განათლების ხარისხის განვითარების ეროვნული ცენტრი

HTML5 & CSS 3
JavaScript
WordPress

ე. ჩიკაშუა, ნ. ელიზბარაშვილი, ზ. დოლიძე, გ. კევლიშვილი

ვებ ინტერფეისის დეველოპერი

თბილისი 2015

წინამდებარე სტუდენტის სახელმძღვანელო **„ვებ ინტერფეისის დეველოპერი“** განკუთვნილია პროფესიული სტუდენტებისათვის, თუმცა სახელმძღვანელოს შინაარსიდან და მასში განხილული პრაქტიკული მასალიდან გამომდინარე მისი გამოყენება შესაძლებელია ბაკალავრიატის დონის შესაბამის საგნობრივ დისციპლინებში და ასევე სასარგებლო რესურსია ვებ დეველოპერის პროფესიით დაინტერესებული ნებისმიერი პირისათვის.

რეცენზენტები:

თორნიკე რაზმაძე
სს „აი თი დი სი“ უფროსი დეველოპერი

ია მოსაშვილი
საქართველოს ტექნიკური უნივერსიტეტის პროფესორი

შინაარსი

თავი 1.ვებსაიტის მარკირება - html

1.1 ვებგვერდის სტრუქტურის აგება

მიმდინარე პარაგრაფის თემატიკა

- თანამედროვე ვებტექნოლოგიები, ვებმარკირების დანიშნულება და გამოყენების სფერო
- კოდის წერის სინტაქსი და ეთიკა
- ვებგვერდის შენახვის ფორმატი
- ვებგვერდის მაკეტის შექმნის ტეგები
- სათაურების ტეგები
- აბზაცის ფორმატირების ტეგები
- სიმბოლოების ფორმატირების ტეგები
- სიების შექმნის ტეგები
- ხაზოვანი და ბლოკური ტეგები

თანამედროვევებტექნოლოგიები, ვებმარკირებისდანიშნულებადაგამოყენებისსფერო

ინტერნეტში არსებული ნებისმიერი ინფორმაცია, რომლთა დათვალიერება შესაძლებელია ინტერნეტ-ბრაუზერების საშუალებით, ინახება რომელიმე ვებ სერვერზე საიტის სახით. ვებ საიტი შედგება ვებგვერდებისაგან, რომელთა მარკირება და ერთმანეთთან დაკავშირება ხდება HTML (Hyper Text Markup Language)ჰიპერტექსტების მარკირების ენის საშუალებით.

HTML ინტერნეტის ფუნდამენტურ საბაზო ტექნოლოგიას წარმოადგენს. HTML ენა ბრიტანელმა ფიზიკოსმა ტიმ ბერნს ლიმ (Tim Berners-Lee) შეიმუშავა 1989 წელს ჟენევაში. თავდაპირველად ის მეცნიერულ-ტექნიკური ინფორმაციის გასაცვლელად შეიქმნა, მას შემდეგ მუდმივად ვითარდება.

ვერსიები	წელი
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

ვებგვერდის ბრაუზერში ასახვისათვის გამოიყენება HTML ენა, რომლის საშუალებით ხორციელდება ვებ-გვერდის სტრუქტურის (Layout) შექმნა, ხოლო გაფორმება ანუ ვიზუალური სახის მიცემა ხდება CSS (Cascading Style Sheets - კასკადური სტილების ცხრილები) სტილების საშუალებით.

html კოდის დასაწერად შესაძლებელია გამოვიყენოთ ნებისმიერ ტექსტური რედაქტორი, თუმცა რეკომენდირებულია ვებ ედიტორების გამოყენება, ვინაიდან ვებ ედიტორში განსხვავებული ფერებით აისახება html კოდი და ვებგვერდის ტექსტური ნაწილი.

ვებ ედიტორებია: notepad++, edit+, sublime Text ჩვენ სახლემძღვანელოში მაგალითების განსახილველად გამოვიყენებთ ღია კოდის პროგრამას - **notepad++,** რომლის გადმოწერაც შესაძლებელია შემდეგი მისამართიდან: https://notepad-plus-plus.org/download/v6.8.6.html (სურ. 1.1)

Download-ით გადმოწერეთ საინსტალაციო პაკეტი და დააყენეთ თქვენს კომპიუტერზე.

ვებგვერდების დასათვალიერებლად გამოიყენება სპეციალური პროგრამა, რომელსაც ბრაუზერს უწოდებენ (browse - დათვალიერება). გავრცელებული ბრაუზერებია: Google Chrome, Mozilla Firefox, Internet Explorer, Apple Safari, Opera.

სურ. 1.1 . საინსტალაციო პაკეტის გადმოწერა

კოდის წერის სინტაქსი და ეთიკა

ნებისმიერი html დოკუმენტიშედგებაბრაუზერშიგამოსატანი ინფორმაციისაგან (ტექსტი, გრაფიკულიობიექტი)დამმართველიელემენტებისაგან, რომელის სახელითავსდებაკუთხურ (< და >) ფრჩხილებსშორის და ეწოდება**ტეგი**(Teg).

როგორცწესიგამოიყენება:

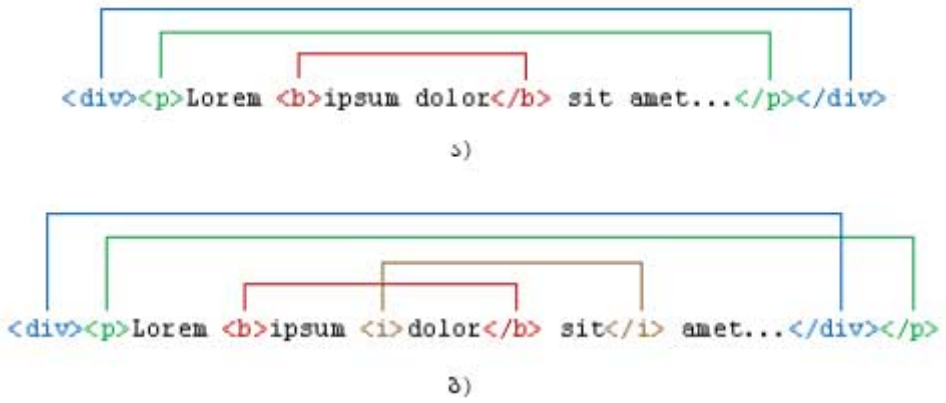
გამხსნელიტეგი - **<ტეგის_სახელი>** მაგ. <html> , <head> , <title> , <body>

დამხურავიტეგი - **</ტეგის_სახელი>** მაგ. </html> , </head> , </title> , </body>

ტეგებისგახსნისადადახურვისმიმდევრობაშემდეგნაირადგამოიყურება:

<ტეგი 1><ტეგი 2><ტეგი 3> ..... </ტეგი 3></ტეგი 2></ტეგი 1>

პირველად იხურება ბოლოს გახსნილი ტეგი და ა.შ. გახსნის უკუთანმიმდევრობით, სხვამიმდევრობითტეგებისგანლაგებამშეიძლებაშეცდომამოგცეთ (სურ 1.2).



სურ. 1.2 ტეგების დახურვის სტრუქტურა ა - მართებული, ბ - არასწორი

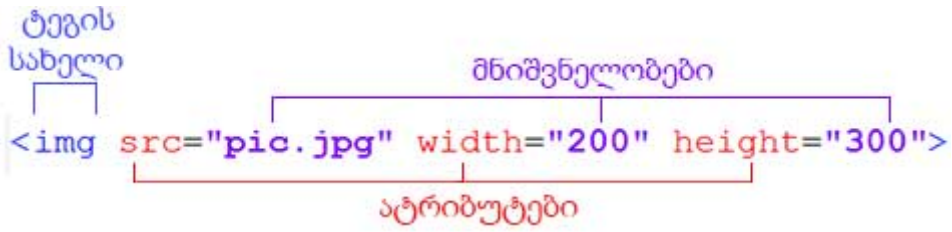
არსებობსგამონაკლისიტეგები, რომელთათვისაცარგამოიყენებადამხურავიტეგი და იხურება თავის თავშივე მაგ: <img/>,
.

ცალკეული ტეგის დანიშნულება დაწვრილებით აღწერილია შემდგომ თვებში

თითოეულტეგსშეიძლებაჰქონდესატრიბუტი. შეიძლებაითქვას, რომატრიბუტებიტეგისთვისებებისსახელებია, რომლებსაცგარკვეულიმნიშვნელობისმიღებაშეუძლია. ტეგისსახელიგანსაზღვრავსობიექტს, ატრიბუტისსაშუალებითკიხდებაამობიექტისსხვადასხვათვისებაზემნიშვნელობისმინიჭება. ატრიბუტისკონსტრუქცია -**თვისება="მნიშვნელობა"**.

ტეგები და ატრიბუტებიაუცილებლადიწერებაპატარასიმგოლოებით, ატრიბუტის მნიშვნელობაიწერებაბრჭყალებში.

ერთტეგშიშესაძლებელიარამოდენიმეატრიბუტისგამოყენება, რომელიცრომელიცტეგისაგანდაერთმანეთისაგანინტერვალითაადაშორიშორებული. (სურ. 1.3).



სურ. 1.3 ტეგის კონსტრუქცია

ატრიბუტებიმხოლოდგამხსნელტეგებსაქვთ, ხოლოდამხურავტეგებსარა. საბოლოოდტეგისკონსტრუქციასაქვსშემდეგისახე (სურ. 1.4)

სურ. 1.4 ტეგის სრული კონსტრუქცია

html კოდის წერის ეთიკის ნორმების უმნიშვნელოვანესი თემაა კოდის კომენტირება.

კომენტარებიშესაძლებელიაგანვათავსოთ HTML დოკუმენტშინებისმიერადგილას, რადგანაციგი არაისახება ბრაუზერის მიერ ვებგვერდზე.

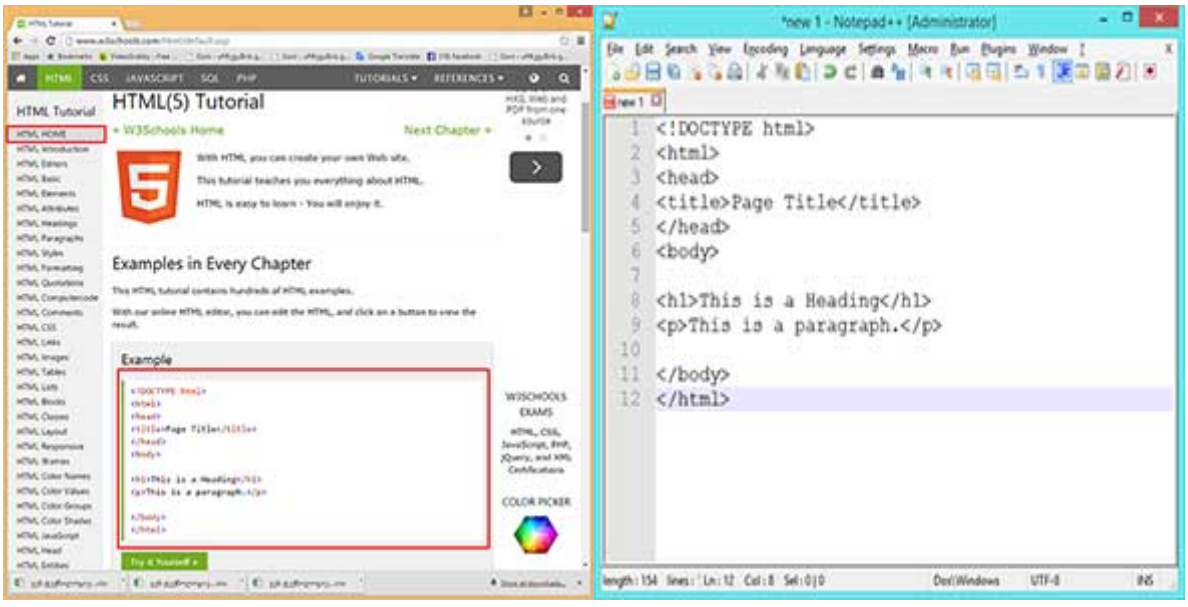
კომენტარშიიწერებაშენიშვნის/კოდის განმარტებითიტექსტები, რომელიც განკუთვნილია ვებ დეველოპერისათვის.

<!-- ერთსტრიქონიანი კომენტარი -->

ვებგვერდისშენახვისფორმატი

პირველი ვებგვერდის შესაქმნელად გავხსნათ ვებ ედიტორი (notepad++) http://www.w3schools.com– დანგადმოაკოპირეთშემდეგიკოდი (სურ. 1.5)

გადაამოწმეთ html დოკუმნტის კოდირება. იმისათვის რომ მას ჰქონდეს უნიკოდის მხარდაჭერა **Encoding** მენიუში აქტიური უნდა იყოს - **Encoding in UTF-8**



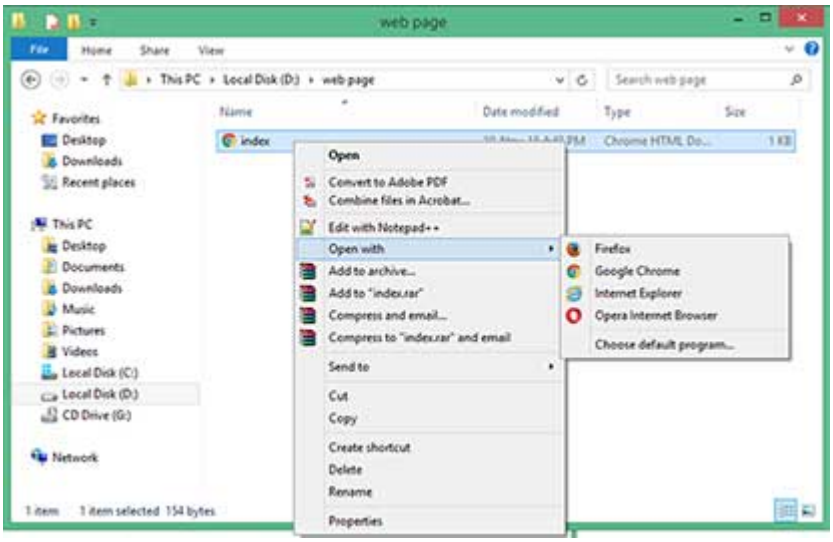
სურ. 1.5 პირველი html კოდი

html კოდიტექსტურიფაილისსახითრომარშეინახოს,დოკუმენტისდამახსოვრებისას (File >Save) Save As დიალოგურიფანჯარის**file name** - ველშივუთითებთ**ფაილისსახელი.html** (სურ.1.6)

თუ კოდის წერისათვის იყენებთ უმარტივეს ტექსტურ რედაქტორს notepad ფაილის შენახვისას Save As დიალოგურ ფანჯარაში **Encoding** - ველში აირჩიეთ - **UTF-8.**

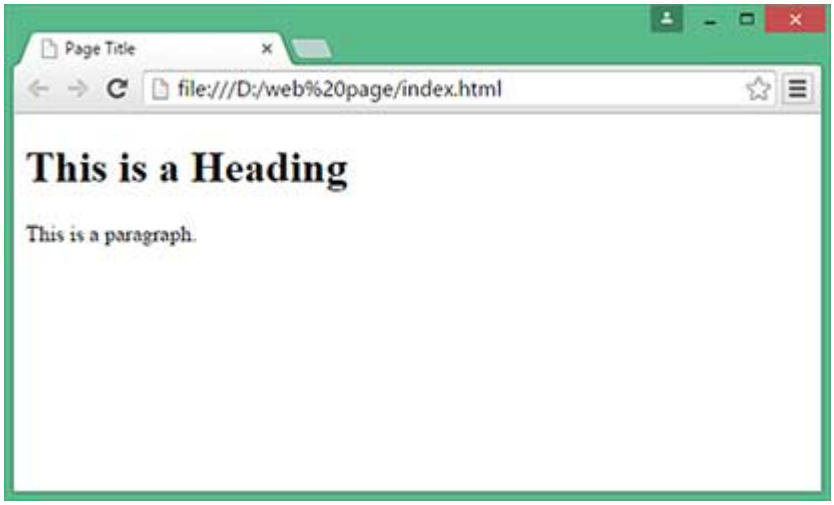
სურ. 1.6 ვებგვერდის შენახვა

ვებგვერდის ნახვა შესაძლებელია ნებისმიერ ბრაუზერში. გახსენით საქაღალდე სადაც შეინახეთ თქვენი ფაილი და index.html ფაილის კონტექსტურ მენიუში (მაუსის მარჯვენა კლავიშზე დაჭერით) აირჩიეთ ბრძანება Open With და თქვენთვის სასურველი ბრაუზერი (სურ. 1.7).



სურ. 1.7 ვებგვერდის გახსნა ბრაუზერში

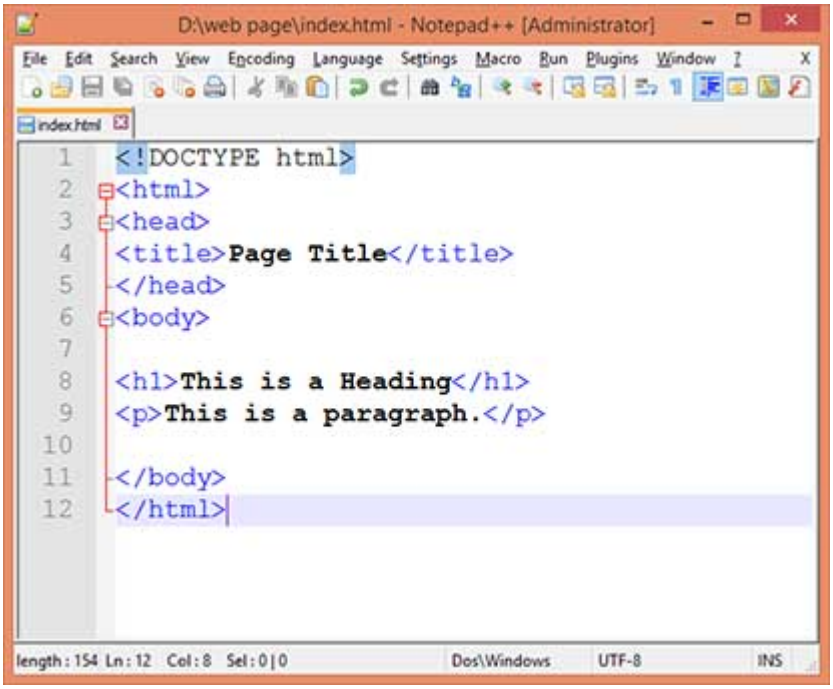
ჩვენს მიერ შექმნილი პირველი ვებგვერდი შემდეგნაირად გამოიყურება (სურ. 1.8.)



სურ. 1.8 პირველი ვებგვერდი

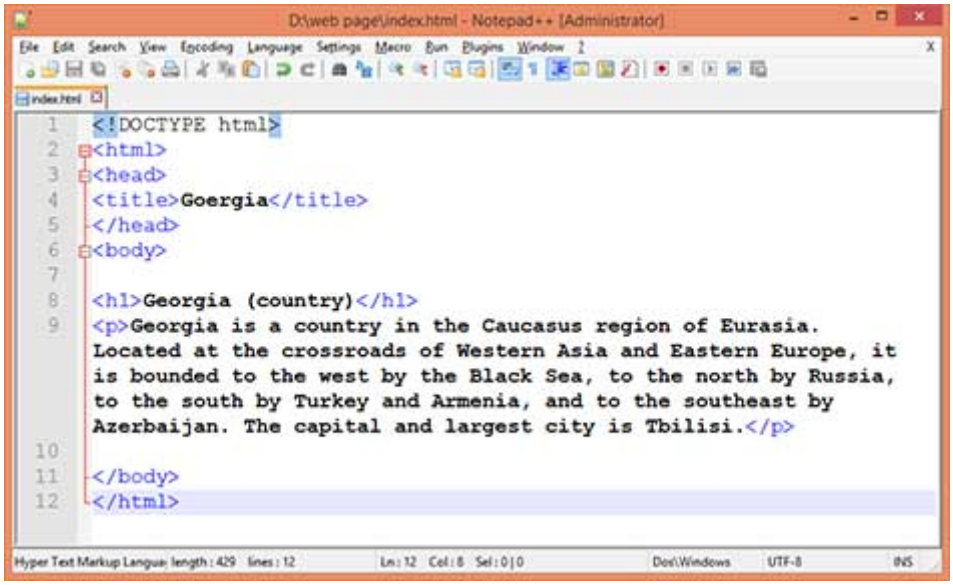
ვებგვერდის რედაქტირება შესაძლებელია ვებ ედიტორში. გავხსნათ ჩვენს მიერ შექმნილი index.html ფაილი ვებ ედიტორში - notepad++.

გახსენით საქაღალდე, სადაც შეინახეთ index.html ფაილი და კონტექსტურ მენიუდან (მაუსის მარჯვენა კლავიშზე დაჭერით) აირჩიეთ ბრძანება Edit with Notepad++ (სურ. 1.7). ფაილის გახსნა ასევე შესაძლებელია ვებ ედიტორიდან file > Open ბრძანებით. ვებ ედიტორში html კოდის ტეგები აისახება ლურჯი ფერით, ტექსტი - შავით (სურ 1.8).



სურ 1.8 პირველი index.html ფაილი

Html დოკუმენტში შეტანილი ცვლილებები (სურ.1.9) ბრაუზერში ვებგვერდზე რომ აისახოს, უნდა შევინახოთ რედაქტირებული ფაილი (Ctrl+S) და ბრაუზერში განვაახლოთ ვებგვერდი ინსტრუმენტების ველზე არსებული Reload ღილაკით ან კლავიატურიდან F5 ფუნქციონალური კლავიშით (სურ. 1.10).



სურ 1.9 index.html ფაილის რედაქტირებული ვერსია

სურ. 1.10 ვებგვერდის რედაქტირების შემდეგ

ვებგვერდისმაკეტისშექმნისტეგები

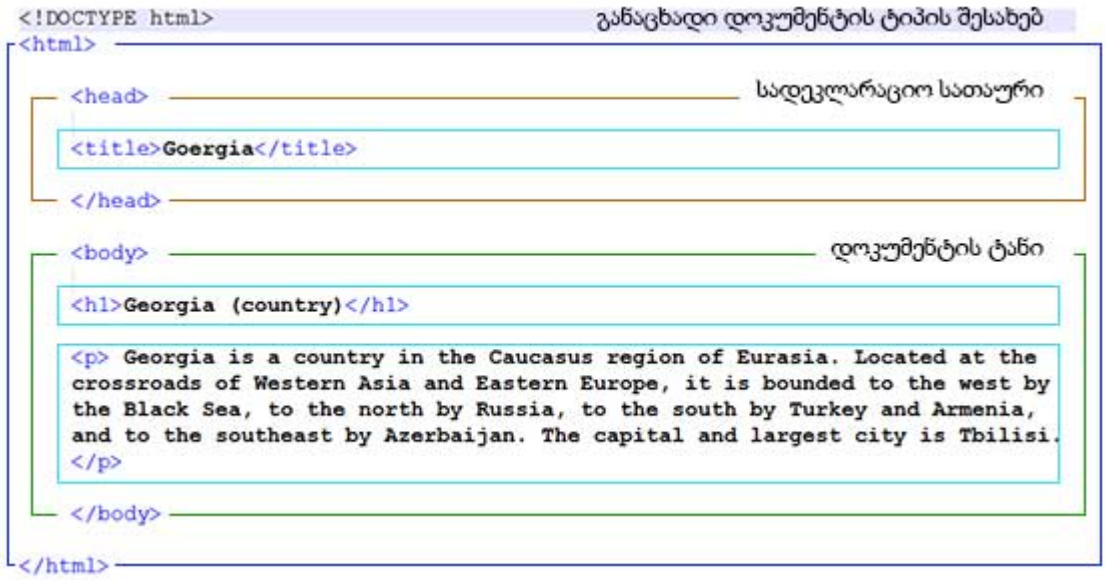
დოკუმენტისსტრუქტურისშექმნისადამასშისაჭიროინფორმაციისშენახვისათვისგამოიყენებასხვადას ხვაელემენტები, რომლებიცმოიცავენდოკუმენტისშექმნისყველასაჭიროპუნქტს. ვებგვერდის შექმნისასგამოიყენებასაბაზოკომპონენტები:

- განაცხადიდოკუმენტისტიპისშესახებ;
- სადეკლარაციოსათაური;
- დოკუმენტისტანი

<!DOCTYPE> - განაცხადიდოკუმენტისტიპისშესახებ, ვერსიებისმიხედვით სხვადასხვა სახის ჩანაწერი არსებობა. მაგალითისათვის იხ. ცხრილი 1.1

ჩანაწერები	ვერსია
<!DOCTYPE html>	HTML5
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	HTML 4.01
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	XHTML 1.0

ცხრილი 1.1 განაცხადი დოკუმენტის ტიპის შესახებ



სურ. 1.11 html დოკუმენტის ზოგადი სტრუქტურა

<html> ... </html>- არისვებდოკუმენტისპირველიდაბოლოტეგი, რომელიცმიუთითებსრომაღნიშნულიფაილიარისვებგვერდიდაარაუბარლოტექსტი

<head> ... </head> - სათაურისბლოკი, სადაციწერებაზოგადიინფორმაციავებგვერდისშესახებ.

<title> ... </title> - დოკუმენტისსახელი, რომელიცგამოისახებაბრაუზერისსათაურისველში (სურ. 1.12).

<body> ... </**body>**–დოკუმენტისტანი, სადაცაღიწერებავებგვერდისშემცველობა. (სურ. 1.12).

სურ. 1.12 ვებგვერდის სტრქტურის ასახვა ბრაუზერში

ჩვენსმიერშექმნილივებგვერდისსათაურიდაშემცველობაწარმოდგენილიიყოლათინურიშრიფტით. ახლა ჩვენს მიერ შექმნილ index.html შევიტანოთ ცვლილებები და ავკრიფოთ სურ. 1.13 ა-ზე მოცემული ქართული ტექსტი (აუცილებელიაგამოვიყენოთქართული უნიკოდ სიმბოლოები - KA). ბრაუზერი ავტომატურად ვერ აღიქვამს ქართულ დამწერლობას (სურ. 1.13 ა).

იმისათვის რომ ბრაუზერმა სწორად აღიქვას ქართული ტექსტი აუცილებელია სათაურისბლოში (<head> - </head>) დავამატოთ <meta charset="UTF-8"> ტეგი.

html დოკუმენტი

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>საქართველოს შესახებ</title>
5  </head>
6  <body>
7  <h1>საქართველო</h1>
8  <p> საქართველო — სახელმწიფო ევრაზიაში, კავკასიაში, შავი ზღვის
   აღმოსავლეთ სანაპიროზე. ესაზღვრება ჩრდილოეთიდან რუსეთი, სამხრეთიდან
   თურქეთი და სომხეთი, და სამხრეთ-აღმოსავლეთიდან აზერბაიჯანი.
   ტრანსკონტინენტური ქვეყანა სამხრეთ-აღმოსავლეთ ევროპისა და დასავლეთ
   აზიის გასაყარზე მდებარეობს, თუმცა სოციოპოლიტიკურად და კულტურულად
   ევროპის ნაწილია.</p>
9  </body>
10 </html>
```

ვებ გვერდი

áƒ¡áƒáƒ₩áƒáƒ áƒ—áƒ•áƒ”áƒšáƒ

áƒ¡áƒáƒ₩áƒáƒ áƒ—áƒ•áƒ”áƒšáƒ â€" áƒ¡áƒáƒ₩áƒáƒ áƒ—áƒ•áƒ”áƒšáƒ›áƒ˜ áƒ›áƒ“áƒ”áƒ‘áƒáƒ áƒ”áƒáƒ‘áƒ¡ áƒ¡áƒáƒ›áƒ®áƒ áƒ”áƒ—-áƒáƒ¦áƒ›áƒáƒ¡áƒáƒ•áƒšáƒ”áƒ—áƒ˜áƒ¡ áƒ“áƒ áƒ“áƒáƒ¡áƒáƒ•áƒšáƒ”áƒ—áƒ˜áƒ¡ áƒ¡áƒáƒ–áƒ¦áƒ•áƒáƒ áƒ–áƒ”. áƒ¡áƒáƒ¥áƒáƒ áƒ—áƒ•áƒ”áƒšáƒ áƒ¡áƒáƒ®áƒšáƒ—áƒáƒœ áƒ”áƒ áƒ—áƒáƒ“ áƒ¡áƒáƒ–áƒ¦áƒ•áƒáƒ áƒ—áƒáƒœ áƒáƒ®áƒšáƒáƒ¡ áƒ¡áƒáƒ›áƒ®áƒ áƒ”áƒ—-áƒáƒ¦áƒ›áƒáƒ¡áƒáƒ•áƒšáƒ”áƒ—áƒ˜áƒ¡ áƒáƒ–áƒ˜áƒ˜áƒ¡ áƒ’áƒáƒ¡áƒáƒ§áƒáƒ áƒ¨áƒ˜ áƒ›áƒ“áƒ”áƒ‘áƒáƒ áƒ”áƒáƒ‘áƒ¡, áƒ—áƒ£áƒ›áƒªáƒ áƒ¡áƒáƒªáƒ˜áƒáƒáƒáƒšáƒ˜áƒ¢áƒ˜áƒ™áƒ£áƒ áƒáƒ“ áƒ“áƒ áƒ™áƒ£áƒšáƒ¢áƒ£áƒ áƒ£áƒšáƒáƒ“ áƒ”áƒ•áƒ áƒáƒáƒ˜áƒ¡ áƒœáƒáƒ¬áƒ˜áƒšáƒ˜áƒ.

ა) არასწორი html კოდი - meta ტეგის გარეშე

html დოკუმენტი

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>საქართველოს შესახებ</title>
5
6  <meta charset="UTF-8">
7
8  </head>
9  <body>
10 <h1>საქართველო</h1>
11 <p> საქართველო — სახელმწიფო ევრაზიაში, კავკასიაში, შავი ზღვის
   აღმოსავლეთ სანაპიროზე. ესაზღვრება ჩრდილოეთიდან რუსეთი, სამხრეთიდან
   თურქეთი და სომხეთი, და სამხრეთ-აღმოსავლეთიდან აზერბაიჯანი.
   ტრანსკონტინენტური ქვეყანა სამხრეთ-აღმოსავლეთ ევროპისა და დასავლეთ
   აზიის გასაყარზე მდებარეობს, თუმცა სოციოპოლიტიკურად და კულტურულად
   ევროპის ნაწილია.</p>
12 </body>
13 </html>
```

ვებ გვერდი

საქართველო

საქართველო — სახელმწიფო ევრაზიაში, კავკასიაში, შავი ზღვის აღმოსავლეთ სანაპიროზე. ესაზღვრება ჩრდილოეთიდან რუსეთი, სამხრეთიდან თურქეთი და სომხეთი, და სამხრეთ-აღმოსავლეთიდან აზერბაიჯანი. ტრანსკონტინენტური ქვეყანა სამხრეთ-აღმოსავლეთ ევროპისა და დასავლეთ აზიის გასაყარზე მდებარეობს, თუმცა სოციოპოლიტიკურად და კულტურულად ევროპის ნაწილია.

ბ) მართებული html კოდი

სურ. 1.13 ვებგვერდზე ქართული ტექსტის ასახვა

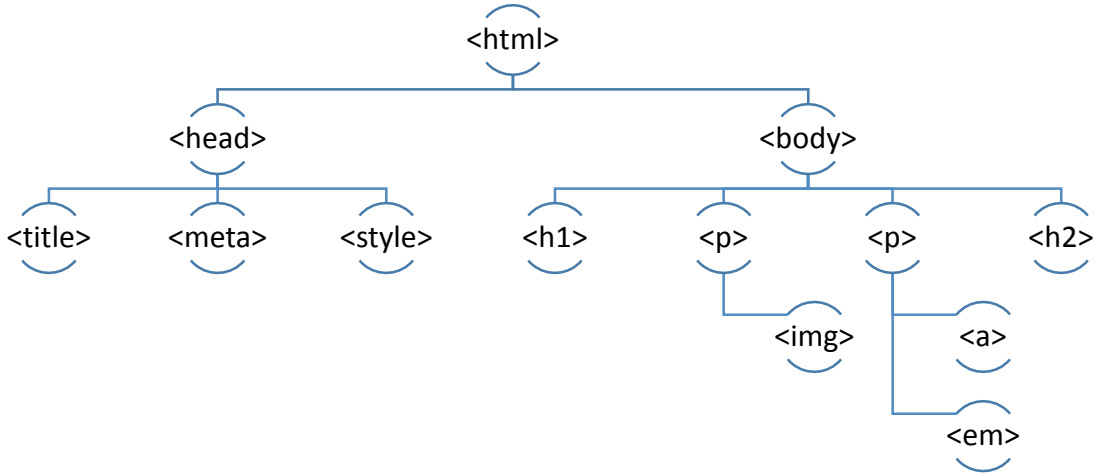
meta ტეგებშიინახებადამატებითიინფორმაციავებ-გვერდისშესახებ. ამელემენტშიარსებულინფორმაციასბრაუზერებიიყენებენვებ-გვერდისდამუშავებისას, ხოლოსაძიებოსისტემებიინდექსირებისას.

<meta name="author"content="ავტორი">ტეგი გამოიყენებავებსაიტისავტორისმისათითებლად, content ატრიბუტშიუთითებთავტორს.

<meta name="description" content="ვებგვერდის ანოტაცია">ტეგი გამოიყენება ვებ საიტის ანოტაციისათვის, content ატრიბუტშიუთითებთსაიტის მოკლე ანოტაციას.

<meta name="keywords" content="საკვანძო სიტყვა1, საკვანძო სიტყვა2">ტეგი გამოიყენება ვებ საიტის საკვანძოსიტყვებისმისათითებლად.

ვებგვერდზეტეგებისგანლაგებასაქვსხისებრისტრუქტურადატეგებილოგიკურადაერთმანეთშიჩალაგებულისურ. 1.14



სურ. 1.14 html დოკუმენტში ტეგების განლაგების სტრუქტურა

პრაქტიკული დავალება

შექმენით ვებგვერდის მაკეტი ვებსტანდარტების მოთხოვნათა დაცვით. რომელზეც ასახავთ ტექსტს „ეს ჩემი შექმნილი პირველი ვებგვერდია“.

სათაურებივებ-გვერდისმნიშვნელოვანელემენტებსწარმოადგენენ. ვებგვერდზეშესაძლებელიამნიშვნელობისმიხედვითსხვადასხვადონისსათაურებისშექმნა. ვებ-გვერდისსათაურებშიარსებულიტექსტიმნიშვნელოვანწილადმოქმედებსსაძიებოსისტემებისმიერსაიტის ინდექსირებაზე,
რადგანრობოტებისუმეტესობაძებნისპროცესშიმნიშვნელოვანყურადღებასაქცევენსათაურებისშემცველობას.

html დოკუმენტშიშესაძლებელიაექვსიდონისსათაურისშექმნაშემდეგიტეგებით: <**h1**>, <**h2**>, <**h3**>, <**h4**>, <**h5**>და<**h6**>. ყველაზემნიშვნელოვანიაპირველიდონისსათაური<h1>დამნიშვნელობისმიხედვითკლებულობს<h6>-მდე. დოკუმენტისსტრუქტურისშექმნისასსაჭიროადავიცვათსათაურებისსუბორდინაციულიიერარქია.

HTML დოკუმენტში პირველი დონის სათაური <h1> გამოიყენება მხოლოდ ერთხელ.

ბრაუზერშიპირველიდონისსათაური<h1>აისახებაუფროდიდიზომისშრიფტით, ვიდრემომდევნოდონეები (სურ. 1.15).

სურ. 1.15 html დოკუმენტში სათაურის დონეები

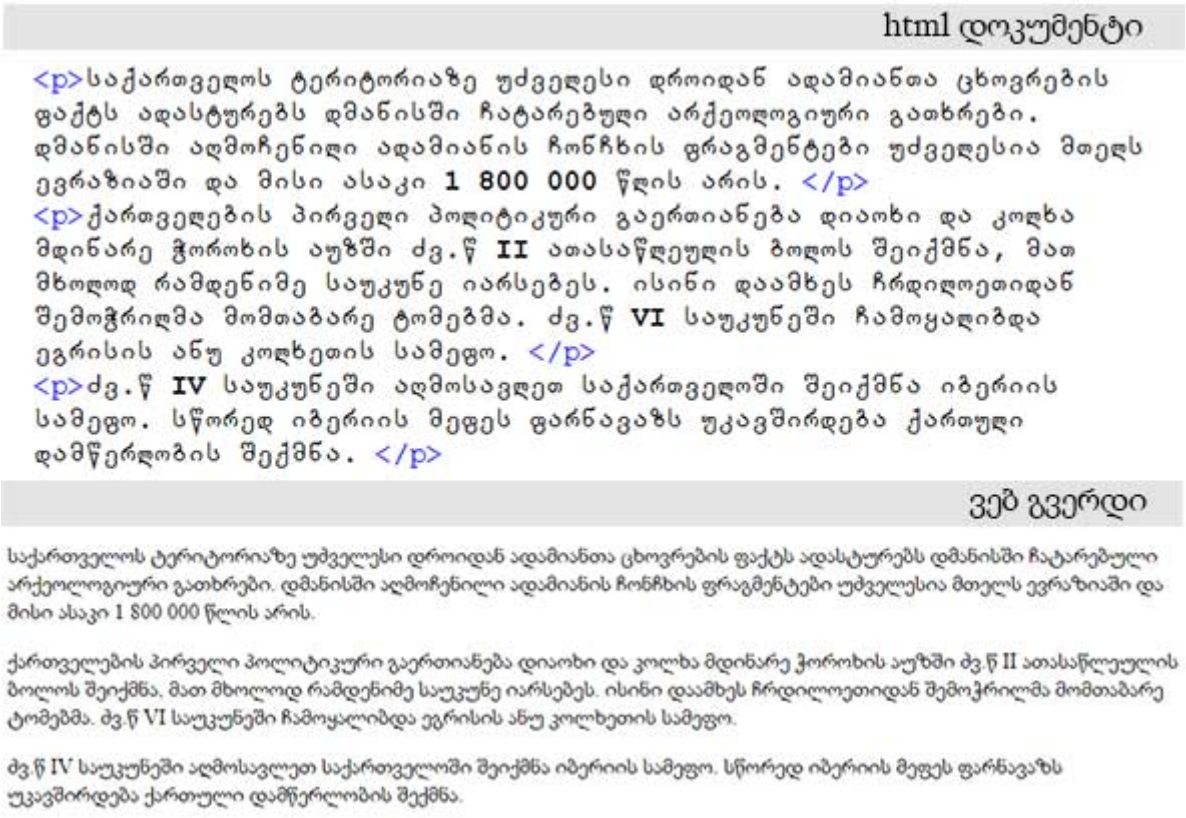
ბრაუზერში სათაურების (h1-h6) თვისებებისათვის (შრიფტის ზომა, სტილი) ავტომატურად მინიჭებული მნიშვნელობების ცვლილება შესაძლებელია სტილების კასკადური ენის (css) გამოყენებით. დაწვრილებით ეს საკითხი განხილულია შემდგომ თავში (იხ. თავი 2).

აბზაცები html დოკუმენტსყოფსლოგიკურნაწილებადდაერთმანეთისგანგამოიყოფაერთისტრიქონისდაშორებით. აბზაცის ორგანიზებისათვის html დოკუმენტში გამოიყენება <**p**> ტეგი (დამხურავი ტეგი - ** **).

აბზაცის და მასში მოთავსებული ტექსტის ბრაუზერში ვიზუალური თვისებების განსაზღვრა შესაძლებელია სტილების კასკადური ენის (css) გამოყენებით. დაწვრილებით ეს საკითხი განხილულია შემდგომ თავში (იხ. თავი 2).

ნუ გამოიყენებთ:

- აბზაცის ტეგების ცარიელკონსტრუქციას -
- აბზაცშისხვააბზაცისტეგს - ააა ააა ბბ ბბ აააა აააა

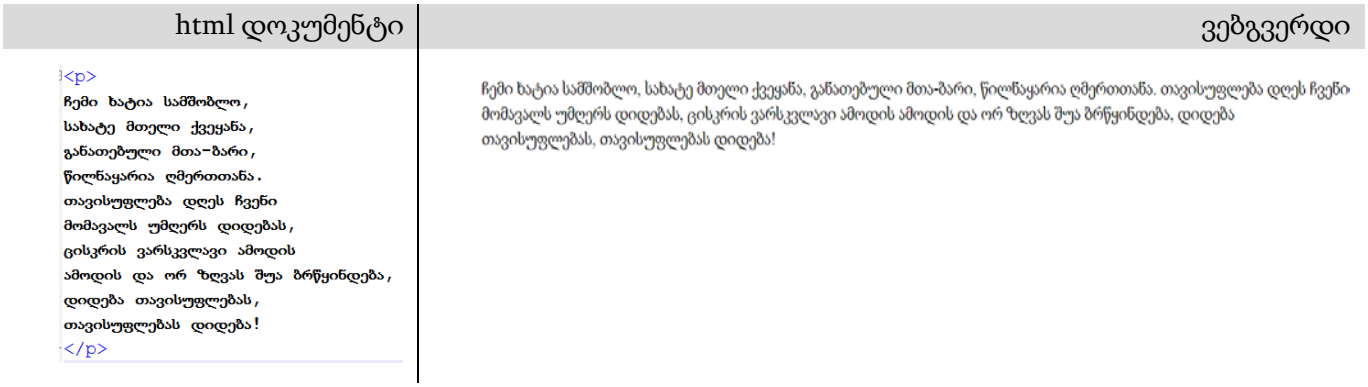


სურ. 1.16 აბზაცის ფორმატირება

ასეთიკონსტრუქციებმა შესაძლება ბრაუზერების სხვადასხვა ვერსიებში არიმუშავებს და კოდის წერის სტანდარტიც დარღვეულია.

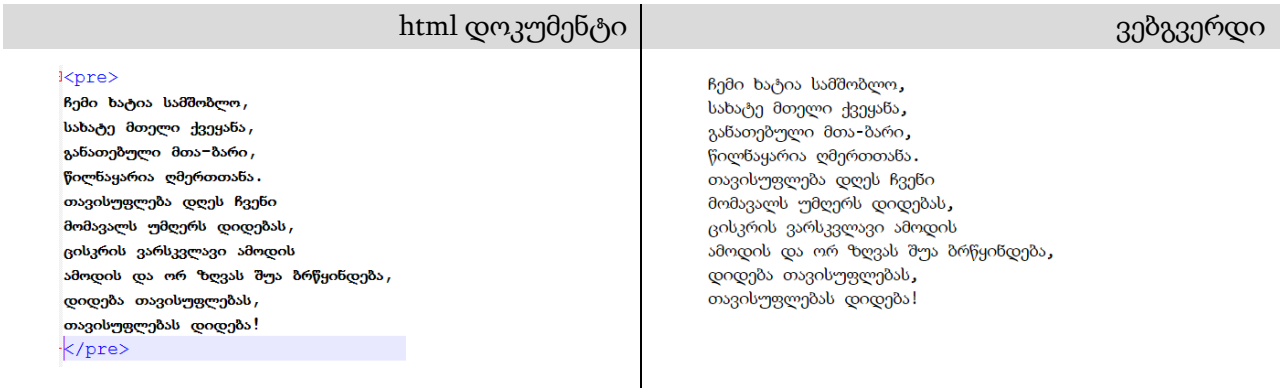
ახალ სტრიქონზე გადასვლისათვის ისე, რომ არ დახუროს მიმდინარე აბზაცი გამოიყენება **
** ტეგი. ეს ელემენტი არ მოითხოვს დამხურავ ტეგს, მაგრამ რეკომენდირებულია ის დაიხუროს გამხსნელ ტეგშივე **
,** რათა ყველა ბრაუზერმა ასახოს კორექტულად.

ვებ ვგერდზე ტექსტის განლაგება განისაზღვრება ტეგების საშუალებით და არ აქვს მნიშვნელობა ვებ ედიტორში მათ განლაგებას (სურ. 1.17). ედიტორში ახალ აბზაცზე enterკლავიშით გადატანილი ტექსტი ბრაუზერში ერთ სტრიქონზე აისახება. ვინაიდან ბრაუზერი ინფორმაციის განლაგებისათვის იყენებს მხოლოდ ტეგებს.



სურ. 1.17 არასწორად განაწილებული აბზაცი

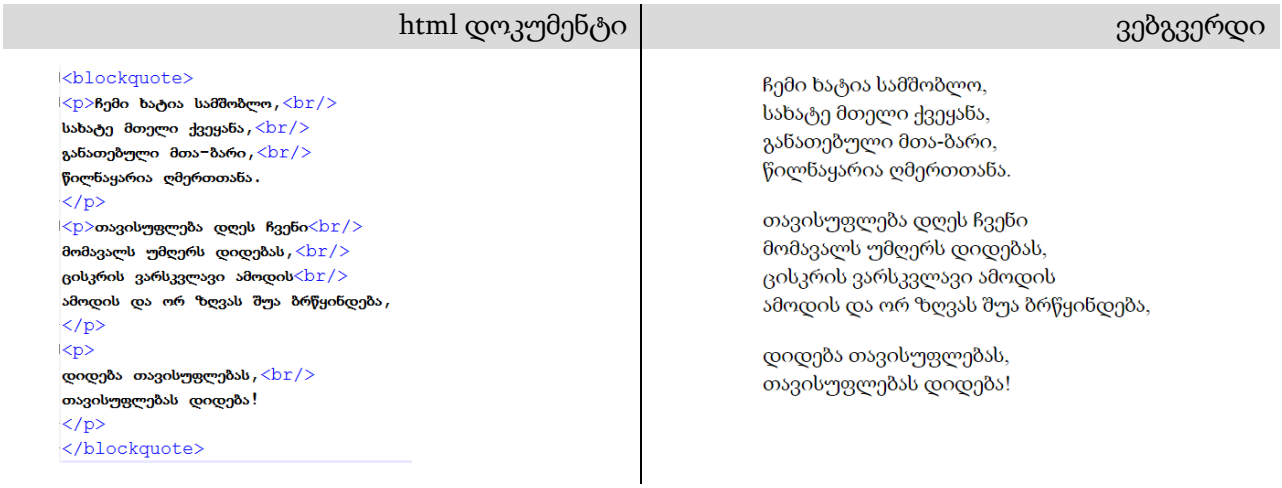
იმისათვის რომ, ვებ ედიტორში გამოყენებული განლაგება ბრაუზერში უცვლელად რომ აისახოს გამოიყენება **<pre>** ტეგი (სურ. 1.18).



სურ. 1.18 <pre> ტეგის გამოყენება

ციტატების, ვრცელი ფრაზებისა და გამონათქვამებისათვის გამოიყენება **<blockquote>**ტეგი. მასში არსებულ ტექსტი ბრაუზერში მარცხნიდან ტაბულაციის შეწეული აისახება. <blockquote> ... </blockquote>ელემენტში დაშაშვებია ფორმატირების სხვა ელემენტების გამოყენება (სურ. 1.19).

ვებგვერდის ლოგიკურ ნაწილებად დასაყოფად გამოიყენება <hr> ტეგი და ბრაუზერში ჰორიზონტალური ხაზის სახით აისახება. აღნიშნულ ტეგს არ აქვს დამხურავი ტეგი და იხურება გამხსნელ ტეგშივე <hr/>



სურ. 1.19 <blockquote> ტეგის გამოყენება

****ტეგიგანკუთვნილიატექსტზეაქცენტირებისათვისდაბრაუზერშიაისახებამუქად.

****გამოყოფსტექსტისფრაგმენტსდაბრაუზერშიაისახება, როგორც*დახრილად*

<abbr>განსაზღვრავსტექსტს, როგორცაბრევიატურას, რომლის**title**ატრიბუტშიიწერებააბრივიატურისგაშიფვრა.ბრაუზერში აბრივიატურის სრული ტექსტიაისახება,შესაბამის აბრივიატურასთან მაუსის მიტანისას.საძიებო სისტემები ყურადღებას აქცევენ აბრევიატურის სრულ ვარიანტს.

<q> ტეგი გამოიყენება ვებ-გვერდზე მოკლე ფრაზის გამოსატანად და მასში არსებულ ტექსტს ბარაუზერი გამოყოფს როგორც ციტატას. ზოგადად ეს ტექსტი ბრაუზერში აისახება დახრილი შრიფტით ან ბრჭყალებით.

<sup> ტეგი გამოიყენება ტექსტის ზედა ინდექსად გამოსახვისათვის.

<sub> ტეგი გამოიყენება ტექსტის ქვედა ინდექსად გამოსახვისათვის.

html დოკუმენტი

```
<p>
<strong>პროფესიული განათლების კვირეული </strong>
</p>
<p>საქართველოს <strong>WorldSkills Georgia</strong>-ს ფარგლებში, გერმანიის საერთაშორისო თანამშრომლობის
საზოგადოების  <abbr title="Gesellschaft für Internationale Zusammenarbeit">(GIZ) </abbr>
მხარდაჭერით, ქვეყნის მასშტაბით პროფესიული განათლების კვირეული ჩატარდა. </p>
<p>კვირეულის დასკვნითი ღონისძიება <em> მოსწავლე-ახალგაზრდობის ეროვნული სასახლეში </em> გაიმართა, რომელსაც
საქართველოს განათლებისა და მეცნიერების მინისტრი თამარ სანიკიძე დაესწრო.</p>
<p>პროფესიული განათლების დევიზია<q>გახდი პროფესიონალი და დასაქმდი</q> </p>

<p>აღნიშნული კვირეულის ფარგლებში სტუდენტთა წამუშევრების გამოფენა გაიხსნება 1 ნოემბერს 15<sup>00</sup>სთ–ზე.
</p>
```

ვებგვერდი

პროფესიული განათლების კვირეული

საქართველოს **WorldSkills Georgia**-ს ფარგლებში, გერმანიის საერთაშორისო თანამშრომლობის საზოგადოების (GIZ) მხარდაჭერით, ქვეყნის მასშტაბით პროფესიული განათლების კვირეული ჩატარდა.

Gesellschaft für Internationale Zusammenarbeit

კვირეულის დასკვნითი ღონისძიება *მოსწავლე-ახალგაზრდობის ეროვნული სასახლეში* გაიმართა, რომელსაც საქართველოს განათლებისა და მეცნიერების მინისტრი თამარ სანიკიძე დაესწრო.

პროფესიული განათლების დევიზია"გახდი პროფესიონალი და დასაქმდი"

აღნიშნული კვირეულის ფარგლებში სტუდენტთა წამუშევრების გამოფენა გაიხსნება 1 ნოემბერს 15^{00}სთ–ზე.

სურ. 1.20 ტექსტის ფორმატირების ტეგები

პრაქტიკული დავალება

შექმენით (განათლებისა და მეცნიერების სამინისტროს ვებ საიტზე ბოლოს გამოქვეყნებული სიახლის იდენტური) ვებგვერდის სტრუქტურა აბზაციებისა და ტექსტის ფორმატირების ტეგების გამოყენებით.

სიებივებგვერდზეგამოიყენებამონაცემებისორგანიზებისადადაჯგუფებისათვის. იგიშეიძლებაგამოყენებულიქნესმენიუს, სტრუქტურებისდასხვამსგავსიობიექტებისშესაქმნელად.

Html დოკუმენტშიში გამოყოფენ სამი ტიპის სიებს

- მარკირებული/დაუნომრავი – Unordered List
- დანომრილი – Ordered List
- განმარტებითი - Description list

მარკირებული სიის შემთხვევაში მისი შემცველი ყოველი პუნქტის წინ გამოისახება სხვადასხვა სახის სიმბოლო. ასეთ სიებს არათანმიმდევრულ/დაუნომრავ სიებსაც უწოდებენ. რადგან ამ ჩამონათვლის ელემენტებში რიგითობას მნიშვნელობა არა აქვს.

მარკირებული სიის (Unordered List) ორგანიზებისათვის გამოიყენება **
**ტეგი, როლის დახურვა **
** ხდება სიის ბოლოს.

მარკირებული სიის პუნქტების შესაქმნელად გამოიყენება**
**ტეგი, რომლის დახურვა **
** აუცილებელია შესაბამისი პუნქტის ბოლოს (სურ.1.21).

html დოკუმენტი	ვებგვერდი

```
<!DOCTYPE html>
<html>
<head>
<title>საქართველოს კუთხეები</title>
<meta charset="UTF-8">
</head>
<body>

<p>საქართველოს კუთხეები:</p>
<ul>
    <li>ქართლი</li>
    <li>კახეთი</li>
    <li>იმერეთი</li>
    <li>გურია</li>
    <li>სამეგრელო</li>
</ul>

</body>
</html>
```

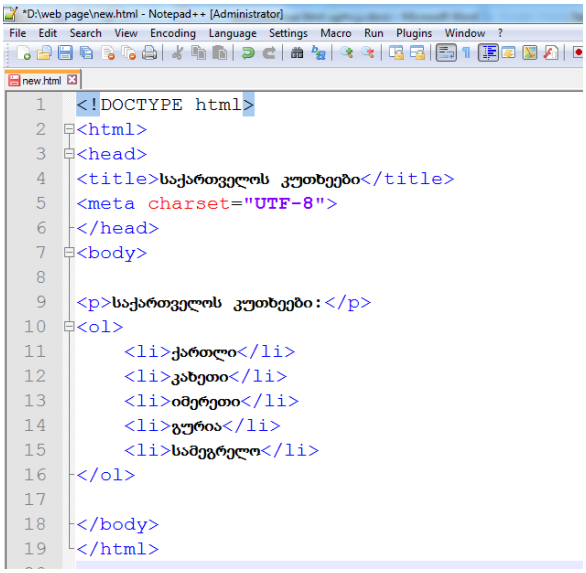
საქართველოს კუთხეები:

- ქართლი
- კახეთი
- იმერეთი
- გურია
- სამეგრელო

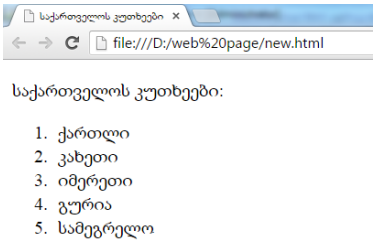
სურ. 1.21 მარკირებული სია

დანომრილი სია გამოიყენება როცა საჭიროა რიგითობის დაცვა, მას თანმიმდევრულ სიასაც უწოდებენ. დანომრილი სიის (Ordered List) შესაქმნელად გამოიყენება **
**ტეგი, რომელიც იხურება **
**სიის ბოლოს. ისევე როგორც მარკირებულ სიაში ჩამონათვლის ელემენტების შესაქმნელად გამოიყენება **
**ტეგი, რომლის დახურვა **
** აუცილებელია შესაბამისი ელემენტის ბოლოს (სურ.1.22).

html დოკუმენტი	ვებგვერდი



```
<!DOCTYPE html>
<html>
<head>
<title>საქართველოს კუთხეები</title>
<meta charset="UTF-8">
</head>
<body>

<p>საქართველოს კუთხეები:</p>
<ol>
    <li>ქართლი</li>
    <li>კახეთი</li>
    <li>იმერეთი</li>
    <li>გურია</li>
    <li>სამეგრელო</li>
</ol>

</body>
</html>
```



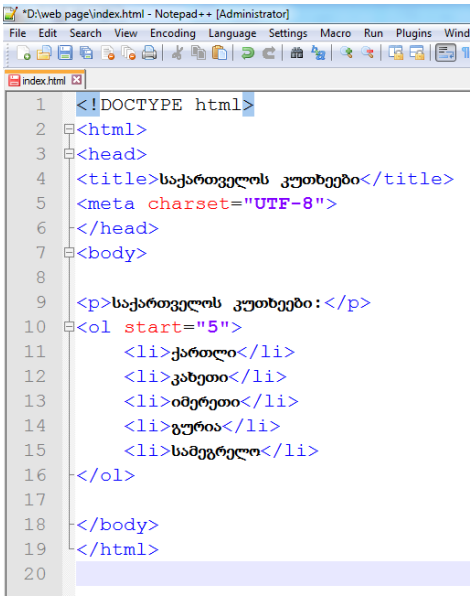
საქართველოს კუთხეები:

1. ქართლი
2. კახეთი
3. იმერეთი
4. გურია
5. სამეგრელო

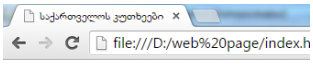
სურ. 1.21 დანომრილისია

დანომრილი სიის ორგანიზებისას შესაძლებელია start ატრიბულის გამოყენება, რომლიც მნიშვნელობაც განსაზღვრავს სიის პირველი ელემენტის ნომერს (სურ.1.23).

html დოკუმენტი	ვებგვერდი



```
<!DOCTYPE html>
<html>
<head>
<title>საქართველოს კუთხეები</title>
<meta charset="UTF-8">
</head>
<body>

<p>საქართველოს კუთხეები:</p>
<ol start="5">
    <li>ქართლი</li>
    <li>კახეთი</li>
    <li>იმერეთი</li>
    <li>გურია</li>
    <li>სამეგრელო</li>
</ol>

</body>
</html>
```



საქართველოს კუთხეები:

5. ქართლი
6. კახეთი
7. იმერეთი
8. გურია
9. სამეგრელო

სურ. 1.23 დანომრილი სიის საწყისი რიცხვის განსაზღვრა

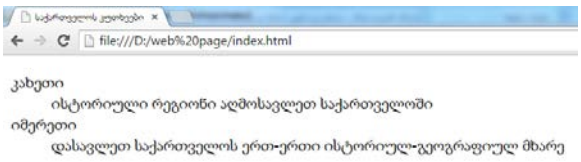
განმარტებითი სია გამოიყენება სიტყვების, ფრაზების ან ტერმინების განმარტებისათვის. განმარტებითი სია შედგება ორი ნაწილისაგან, პირველში აღიწერება განსამარტებელი სიტყვას, ხოლო მეორში ამ სიტყვის განმარტება.

განმარტებითი სიის Description list ორგანიზებისათვის გამოიყენება **<dl>**ტეგი, რომელიც იხურება **</dl>**სიის ბოლოს. განსამარტი სიტყვების აღწერა ხდება **<dt>** ტეგით, განმარტების კი – **<dd>** ტეგით. ორივე ტეგი იხურება ცალკეული სიტყვისა **</dt>** და განმარტების **</dd>** ბოლოს (სურ.1.24).

html დოკუმენტი

```
<!DOCTYPE html>
<html>
<head>
<title>საქართველოს კუთხეები</title>
<meta charset="UTF-8">
</head>
<body>

<dl>
    <dt>კახეთი</dt>
        <dd> ისტორიული რეგიონი აღმოსავლეთ საქართველოში</dd>
    <dt>იმერეთი</dt>
        <dd>დასავლეთ საქართველოს ერთ-ერთი ისტორიულ-გეოგრაფიულ მხარე</dd>
</dl>

</body>
</html>
```

ვებგვერდი



სურ. 1.24 განმარტებითი სია

ვებგვერდზე ხშირად საჭირო ხდება **მრავალდონიანი სიის** ორგანიზება მაგ. ჩამოშლადი მენიუს ან/და საიტის რუქის შესაქმნელად (სურ.1.25).

მრავალდონიანი სიის კოდის წერისას, ყურადღება მიაქციეთ კოდის წერის ეთიკას და ყველა მომდევნო დონის სიის ამსახველი ტეგი გახსენით ერთი ტაბულაციით მარჯვნივ შეწული. ქვედონის სიის დახურვის შემდეგ კი ერთი ტაბულაციით მარცხნივ გამოწევით გააგრძელეთ ზედა დონის სიის პუნქტების ასახვა (სურ.1.25 – html დოკუმენტი).

ბრაუზერში სხვდასხვა დონის სიის ელემენტები განსხვავებული სიმბოლოებით აისახება (სურ.1.25 – ვებგვერდი). პირველი დონე – შავი წრე (disc), მეორე დონე – თეთრი წრე (circle), მესამე დონე – შავი კვადრატი (square).

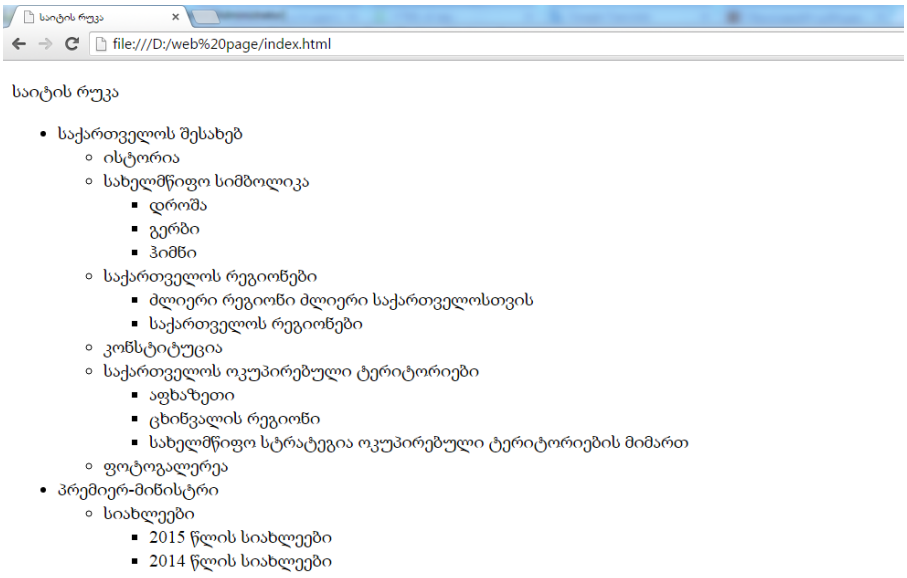
მრავალდონიანი სიის ორგანიზებისათვის შესაძლებელია გამოვიყენოთ დანომრილი სია ან/და მარკირებისა და ნუმერაციის კომბინაცია.

html დოკუმენტი

```
<!DOCTYPE html>
<html>
<head>
<title>საიტის რუკა</title>
<meta charset="UTF-8">
</head>
<body>
<p>საიტის რუკა</p>
<ul>
<li>საქართველოს შესახებ</li>
    <ul>
    <li>ისტორია</li>
    <li>სახელმწიფო სიმბოლიკა</li>
        <ul>
        <li>დროშა</li>
        <li>გერბი</li>
        <li>ჰიმნი</li>
        </ul>
    <li>საქართველოს რეგიონები</li>
        <ul>
        <li>ძლიერი რეგიონი ძლიერი საქართველოსთვის</li>
        <li>საქართველოს რეგიონები</li>
```

```
        </ul>
    <li>კონსტიტუცია</li>
    <li>საქართველოს ოკუპირებული ტერიტორიები</li>
        <ul>
        <li>აფხაზეთი</li>
        <li>ცხინვალის რეგიონი</li>
        <li>სახელმწიფო სტრატეგია ოკუპირებული ტერიტორიების მიმართ</li>
        </ul>
    <li>ფოტოგალერეა</li>
    </ul>
<li>პრემიერ-მინისტრი</li>
    <ul>
    <li>სიახლეები</li>
        <ul>
        <li>2015 წლის სიახლეები</li>
        <li>2014 წლის სიახლეები</li>
        </ul>
    </ul>
</ul>
</body>
</html>
```

ვებგვერდი



სურ. 1.25 მრავალდონიანი სია

დანომრილი სიის ელემენტები ბრაუზერში შეიძლება ასისახოს არაბული ციფრებით (1,2,3 და ა.შ.); ლათინური ასომთავრული ან ხელნაწერი ალფავიტით (A,B,C… ან a,b,c..); რომაული ციფრებით (I, II, III …).

ბრაუზერშიმარკირებისა და ნუმერაციისვიზუალურითვისებებისგანსაზღვრაშესაძლებელიასტილებისკასკადურიენის (css) გამოყენებით. დაწვრილებითესსაკითხიგანხილულიაშემდგომ თავში (იხ. თავი 2).

პრაქტიკული დავალება

შექმენით მრავალდონიანი სიის სტრუქტურა. მაგ. ეროვნული სასწავლო გეგმების პროტალის საიტის რუქის იდენტური http://ncp.ge/ge/sitemap

ხაზოვანიდაბლოკურიტეგები

ტეგები იყოფა **ბლოკურ**(block-level) და**ხაზოვანი** (inline-level) ტიპისტეგებად.

ბლოკური ტიპისტეგებში განთვსებული ინფორმაცია ბრაუზერში აისახებაახალისტრიქონიდან და შესაბამისი

ტეგიმთელჰორიზონტალურსტრიქონსიკავებსიმშემთხვევაშიცკითუამტეგშიმხოლოდერთისიტყვაამო თავსებული. დიდი ზომისტექსტის შემთხევვაშიერთისტრიქონის შევსების შემდეგ ტექსტი ავტომატურადგადადისმეორესტრიქონზე.

ჩვენს მიერ განხილული ტეგებიდან ბლოკური ტიპის ტეგებია: <p>, <blockquote>, <pre>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ol>, <ul>, <dl>, <li>, <dd>, <dt>

ვებგვერდზე დიდი ზომის ინფორმაციის გამოსაყოფად ხშირად გამოიყენება** **ტეგი.

ხაზოვანიტეგებიბრაუზერშიიკავებენმხოლოდმათში მთავსებული ტექსტის ზომის ადგილს. შეიძლება ერთ სტრიქონზე რამოდენიმე ხაზოვანი ტეგი იყოს ასახული.

ჩვენს მიერ განხილული ტეგებიდან ხაზოვანიტეგებია: <abbr>,
, , <q>, , <sub>, <sup>.

ტექსტის მცირე ფრაგმენტის გამოსაყოფად ასევე ხშირად გამოიყენება **** ხაზოვანი ტეგი.

ბლოკურ ტეგებში შეიძლება გამოვიყენოთ სხვა ბლოკური ან/და ხაზოვანი ტეგები. ხაზოვან ტეგებში კი დაუშვებელია ბლოკური ტეგის გამოყენება.

ცალკეული ტეგების ვიზუალურითვისებებისგანსაზღვრაშესაძლებელიასტილებისკასკადურიენის (css) გამოყენებით. დაწვრილებითესსაკითხიგანხილულიაშემდგომ თავში (იხ. თავი 2).

1.2 ვებგვერდზე ობიექტებისა და ბმულების ასახვა

მიმდინარე პარაგრაფის თემატიკა

- გრაფიკული და მულტიმედია ობიექტის ფორმატები და მათი ვებგვერდზე ასახვის საშუალებები
- ბმულის ტიპები და პარამეტრები
- ცხრილის ტეგები და მისი პარამეტრები

გრაფიკული და მულტიმედია ობიექტის ფორმატები და მათი ვებგვერდზე ასახვის საშუალებები

ვებგვერდზე ინფორმაციის წარმოდგენის მრავალფეროვნების მიზნით შესაძლებალია გრაფიკული და მულტიმედია ობიექტის გამოყენება. წამოუდგენელია ვებსაიტი გრაფიკული გამოსახულებებისგარეშე. ვინაიდან გრაფიკული ობიექტების ვებგვერდზე ასახვა ზრდის საიტის მეხსიერების ზომას/მოცულობას, აუცილებელია მათიგამოყენებისას მიღებული ნორმების დაცვა.

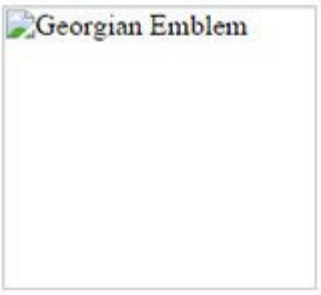
ვებგვერდზე გამოსაყენებელი გამოსახულების პარამეტრები ოპტიმალურად უნდა შეირჩეს, ისე რომ არც ხარისხი გაურესდეს და არც ვებგვერდისბრაუზერშიწარმოდგენა გართულდეს.

html დოკუმენტში გრაფიკული ობიექტის/სურათის ასახვა ხდება შემდეგიტეგით (სურ. 1.25):

html დოკუმენტი

```
<img src="geo_emblem.png" alt="Georgian Emblem" title="საქართველოს
                                    გერბი"  width="174" height="157"border="4"/>
```

ვებგვერდი



ა) ბ)

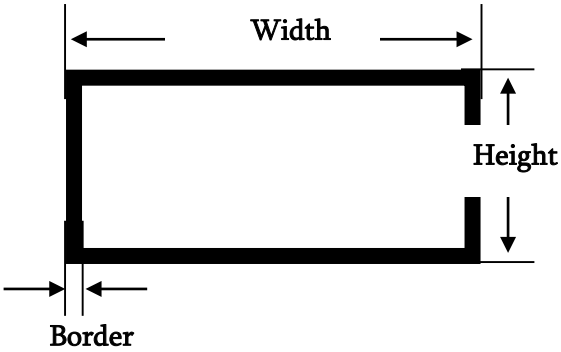
სურ. 1.25 ა) ვებგვერდზე გრაფიკული ობიექტის ასახვა

ბ) ვებგვერდზე ალტერნატიული ტექსტის ასახვა

src – ატრიბუტის მინიშვნელობაა ვებგვერდზე ასასახი სურათის მისამართი.

alt–ალტერნატიულიტექსტი, მისიმნიშვნელობავებგვერდზე აისახება როცა სხვადასხვა მიზეზის (ტექნიკური გაუმართაობა, მომხმარებელს გათიშული აქვს ბრაუზერში სურათის ასახვის რეჟიმი და ა.შ.) გამო არ ხდება სურათის ბრაუზერში ჩატვირთვა.

title– სათაური, რომლისმნიშვნელობაგამოჩნდება ბრაუზერში სურათზემაუსისმაჩვენებლისმიტანისასამოტივტივებადიშეტყობინებისსახით.



სურ. 1.26 სურათის ზომისა და ჩარჩოს ატრიბუტები

width– სურათის სიგრძე. **height**– სურათის სიმაღლე. **border**– სურათის ჩარჩოს სისქე.

თუ არ მიუთითებთ სურათის ზომებს (სიგრძე და სიმაღლე) ბრაუზერში აისახება რეალური ზომითანუიმსიგრძითდასიგანით, რაცსურათსგააჩნია. სურათის ზომის მცირეცვლილებებიდასაშვებია**width** და **height**ატრიბუტებისსაშუალებით, მაგრამთუსაჭიროასურათისზომებისმნიშვნელოვნადშეცვლაუმჯობესიაგამოვიყენოთგრაფიკულირედაქ

ტორი. ვინაიდან html დოკუმენტში სურათის ზომის გაზრდით მისი ხარისხი იკლება, ხოლო შემცირებით სურათის მეხსიერება რჩება უცვლელი და ვიზუალურად ბრაუზერში აისახება მცირე ზომის სურათი და საიტის ასახვა გართულდება.

border ატრიბუტის მნიშვნელობა ავტომატურად ნულის ტოლია, თუმცა შეიძლება მიეთითოს სასურველი ზომა ან მოეხსნას ჩარჩო (border=0).

img ტეგიარმოითხოვსდამხურავტეგს(</img> - არასწორია) დაეს ტეგი იხურებაგამხსნელტეგშივეანუთავისთავში (სურ. 1.25).

html დოკუმენტისმიმართსურათისმდებარეობიდანგამომდინარე src ატრიბუტში სხვადასხვანაირადხდებამისიმისამართისმითითება.

სურათის მისამართებისმითითებისვარიანტები (სურ. 1.27):

1. html დოკუმენტიმიმართავსინტერნეტში, რომელიმესერვერზეგანთავსებულფაილს,მაშინ src ატრიბუტისმნიშვნელობადმიეთითებაამსურათისუნივერსალურიმისამართი (URL).

```
<img src="http://mes.gov.ge/geo_emblem.png" alt="Georgian Emblem"/>
```

2. html დოკუმენტი და სურათი ერთ საქაღალდეშია (სურ. 1.27), მაშინ სურათის მისამართი ნაცვლად მიეთითება მხოლოდ ფაილის სახელი.

```
<img src="picture1.gif" alt="picture1"/>
```

3. html დოკუმენტი მიმართავს ქვესაქაღალდეImage-ში არსებულ picture2.gifფაილს (სურ.1.27), მაშინ მიეთითება **ქვესაქაღალდის სახელი/ფაილის სახელი.**

```
<img src="Image/picture2.gif" alt="picture2"/>
```

4. html დოკუმენტი მიმართავს ქვესაქაღალდე Image-სი ქვესაქაღალდე New-ში არსებულ picture3.gif ფაილს (სურ.1.27), მაშინ მიეთითება **ქვესაქაღალდის სახელი/ქვესაქაღალდის სახელი/ფაილის სახელი.**

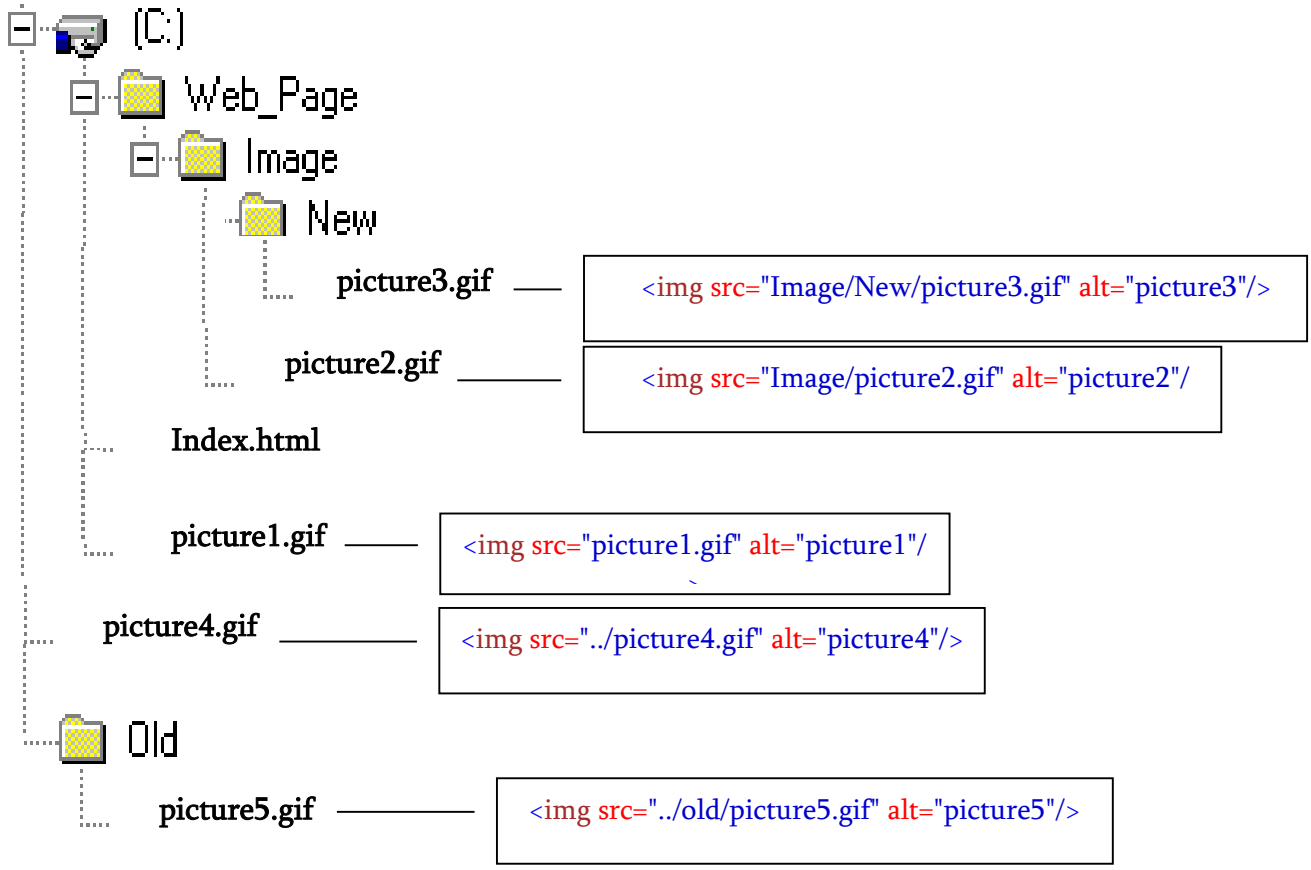
```
<img src="Image/New/picture3.gif" alt="picture3"/>
```

5. html დოკუმენტი მიმართავს საქაღალდის გარეთ არსებულ picture4.gif ფაილს (სურ.1.27), მაშინ მიეთითება **../ფაილის სახელი.**

```
<img src="../picture4.gif" alt="picture4"/>
```

6. html დოკუმენტი მიმართავს საქაღალდის გარეთ არსებულ სხვა Old საქაღალდის picture5.gif ფაილს (სურ.1.27), მაშინ მიეთითება **../საქაღალდის სახელი/ფაილის სახელი.**

```
<img src="../old/picture5.gif" alt="picture5"/>
```



სურ. 1.27 სურათის ასახვა მდებარეობის გათვალისწინებით

ვებ გვერდის ნაწილებად გასაყოფად გამოიყენება ჰორიზონტალური ხაზი <hr />, რომელსაც არ აქვს დამხურავი ტეგი (</hr>არასწორია) და იხურება გამხსნელ ტეგშივე.

ბმულის ტიპები და პარამეტრები

ბმულები ერთერთი უმთავრესი საკითხის ვებ საიტზე. იგი უამრავ ვებგვერდს აერთიანებს ერთ ვებ საიტიად. ვებგვერდები შესაძლებელია სხვადასხვა სერვერზეც კი იყოს განლაგებული, მაგრამ სწორად ორგანიზებული ბმულების საშუალებით მომხმარებელი აღიქვამს ერთ მთლიან საიტად.

არსებობს ორი ტიპის ბმული: **გარე ბმული,** რომელიც ერთმანეთთან აკავშირებს სხვადასხვა ობიექტებს (ვებგვერდები, სურათები, სხვადასხვა ტიპის ფაილები),გვეხმარება მათ შორის ნავიგაციაში და ქმნის

ერთიან სივრცეს. **შიდა ბმული,** რომელიც გამოიყენება ერთი ვებგვერდის ფარგლებში ნავიგაციის ოპტიმიზაციისათვის.

html დოკუმენტში ბმულის ასახვა ხდება შემდეგიტეგით:

```
<a href="მისამართი">ბმულის ტექსტი</a>
```

მაგ.

```
<a href="http://mes.gov.ge/">საქართველოს განათლებისა და მეცნიერების სამინისტრო</a>
<a href="about.html">ჩვენს შესახებ</a>
```

href ატრიბუტს ენიჭება იმ ობიექტისმისამართი, რომელზეცვაპირებთგადამისამართებას. თუ გადამისამართება ხდება რეალურ საიტზე მაშინ მიეთითება საიტის url-ს. სხვადასხვა ფაილზე გადამისამართების შემთხვევაში ფაილის მისამართს (html დოკუმენტიდან შესაბამის ფაილამდე სრულ გზა). მისამართის მითითების ხერხები დაწვრილებით იხილეთ წინა ქვეთავში - სურათის მისამართებისმითითებისვარიანტები (სურ. 1.27).

ბმულზე გადასვლისას შესაბამისი ვებგვერდი ავრომატურად იხსნება იმავე ფანჯარაში. ახალ ფანჯარაში გახსნის შესაძლებლობას იძლევა **target** ატრიბუტი. თუ target ატრიბუტს მივანიჭებთ _blank მნიშვნელობას, მაშიან შესაბამისი ბმულის გახსნა მოხდება ახალ ფანჯარაში.

მაგ. `<a href="http://mes.gov.ge/" target="_blank">განათლებისა და მეცნიერების სამინისტრო</a>`

როცა ვებგვერდზე დიდი ზომის ინფორმაციაა განთავსებული და არ ეტევა მონიტორის ერთ არეში,ნავიგაციისგასაადვილებლადგამოიყენება შიდაბმულები. ბმულისსაშუალებითშეგვიძლია გადავაადგილდეთ იმავე დოკუმენტის სხვადასხვა ადგილას.

შიდაბმულებისორგანიზების პირველ ეტაპზეუნდამოვნიშნოთ ადგილი,სადაცხდება ბმულისგადამისამართება.

მაგ.

```
<h2 id="Chapter1">პარაგრაფი1. .....</h2>
.
<h2 id="Chapter2">პარაგრაფი2. ....</h2>
.
<h2 id="Chapter3">პარაგრაფი3. ....</h2>
```

აღნიშნულ ადგილებზე გადამისამართება ხდება შემდეგი ტეგებით:

```
<ul>
  <li><a href="#Chapter1">პარაგრაფი1.</a></li>
  <li><a href="#Chapter2">პარაგრაფი2.</a></li>
  <li><a href="#Chapter3">პარაგრაფი3.</a></li>
</ul>
```

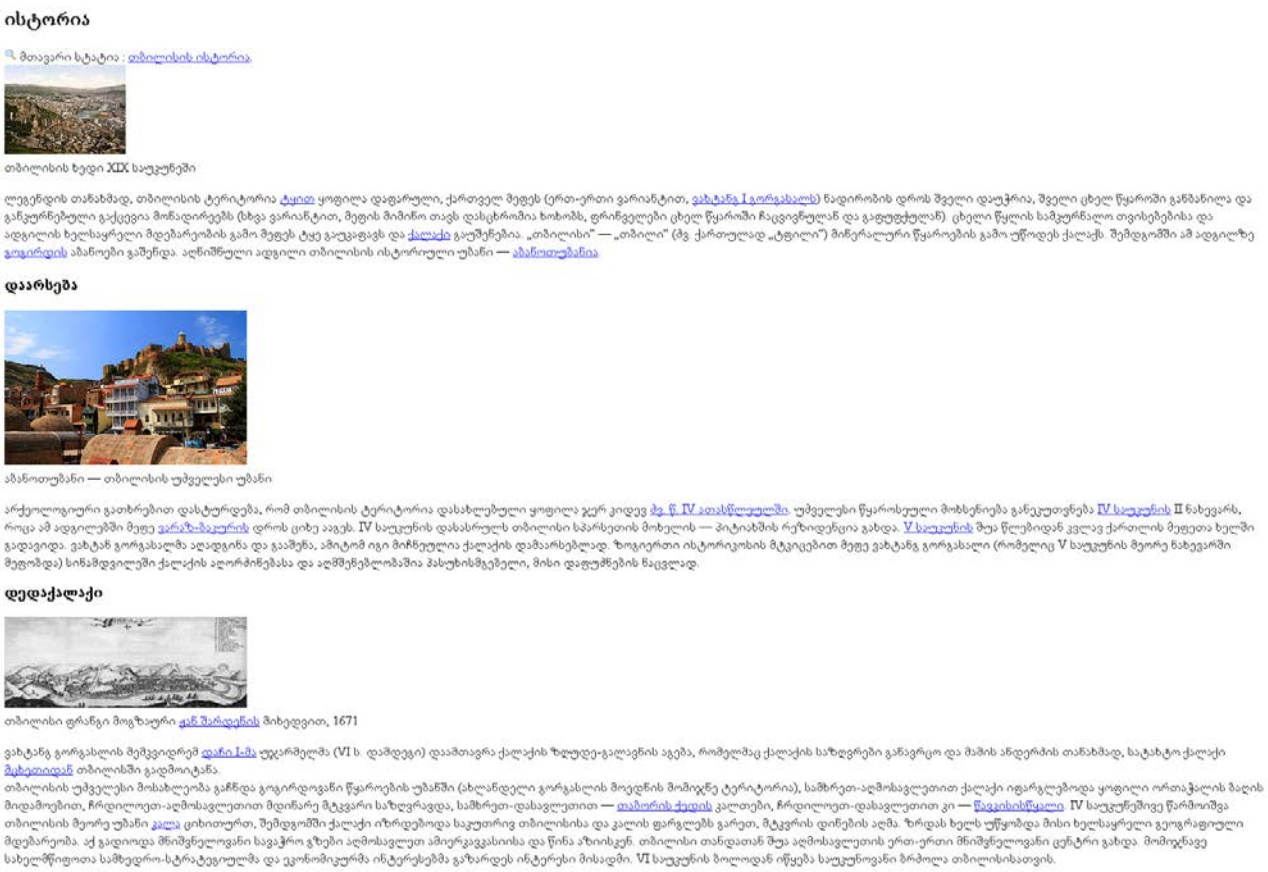
პრაქტიკული დავალებები/კითხვები თვითშეფასებისათვის

შექმენით ვიკიპედიიდან თბილისის ისტორიის იდენტური ვებგვერდის სტრუქტურა (სურ. 1.29). შესაბამისი სურათებისა და ბმულების ასახვით.

https://ka.wikipedia.org/wiki/%E1%83%97%E1%83%91%E1%83%98%E1%83%9A%E1%83%98%E1%83%A1%E1%83%98

სურ. 1.28 თბილისის ისტორიის ვებგვერდი

აღნიშნული გვერდი სტილების გამოყენების გარეშე გამოიყურება შემდეგნაირად, რომლის იდენტური ვებგვერდი უნდა შექმნათ html–ის საშუალებით.



სურ. 1.29 თბილისის ისტორიის ვებგვერდის html სტრუქტურა

ცხრილებიგამოიყენებაპროდუქციისფასის, ვალუტისკურსის, ამინდისპროგნოზის, სტუდენტთაშეფასებებისდასხვასტატისტიკურიმონაცემებისვებგვერდზეასახვისათვის.

ა)

სახელი	გვარი	ასაკი
დავით	ბარბაქაძე	6
თამარ	ბარბაქაძე	4

ბ) <table>

<tr>	<th>სახელი</th>	<th> გვარი </th>	<th> ასაკი </th>	</tr>
<tr>	<td> დავით </td>	<td> ბარბაქაძე </td>	<td> 6 </td>	</tr>
<tr>	<td> თამარ </td>	<td> ბარბაქაძე </td>	<td> 4 </td>	</tr>

</table>

სურ. 1.30 ა) ცხრილის სახე ვებგვერდზე

ბ) ცხრილის ორგანიზება ტეგების საშუალებით

სურ. 1.30 ა–ზე მოცემული ცხრილის ამსახველი html კოდის გამოიყურება შემდეგნაირად.

```
<table width="100%" border="1">
  <tr>
    <th>სახელი</td>
    <th>გვარი</td>
    <th>საკი</td>
  </tr>
  <tr>
    <td>დავით</td>
    <td>ბარბაქაძე</td>
    <td>6</td>
  </tr>
  <tr>
    <td>თამარ</td>
    <td>ბარბაქაძე</td>
    <td>4</td>
  </tr>
</table>
```

ცხრილის ორგანიზებისათვის გამოიყენება შემდეგი ტეგები:

<table>... </table>– ცხრილი

<tr>... </tr>– სტრიქონი

<td>... </td>– უჯრა

<th>... </th>– სათაურის სტრიქონის უჯრები

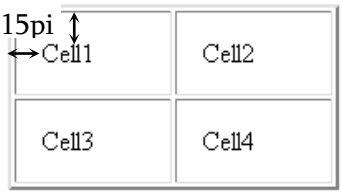
ინფორმაციის (ტექსტური, გრაფიკული და.ა.შ)შეტანახორციელდებაუჯრებში.

ცხრილის ცარიელი უჯრის ასახვა შესაძლებელია <td> </td>ელემენტით.

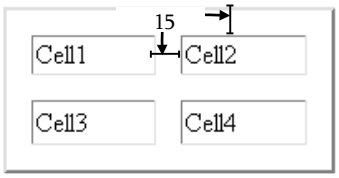
ცხრილის პარამეტრების განსაზღვრა შესაძლებელია შემდეგი ატრიბუტებით:

- **border** – რომლისმნიშვნელონობა განსაზღვრავსცხრილისჩარჩოებისსისქესპიქსელებში. თუატრიბუტი border არ არის განსაზღვრული ცხრილისჩარჩოებიუხილავია.
- **width** – რომლისმნიშვნელონობა განსაზღვრავსცხრილის სიგრძის ზომას. თუ არ არის განსაზღვრული ცხრილის სიგრძე, მაშინ ისდამოკიდებული ხდებაურებშიჩაწერილიინფორმაციისზომაზე. width ატრიბუტის მნიშვნელობაშეიძლებამიეთითოსორი ერთეულით:აბსოლიტური – პიქსელებში (width="540") ან ფარდობითი – პროცენტებში (width="70%"). პროცენტულად ცხრილის ზომის განსაზღვრა ხდება ეკრანის ზომის პროპორცულად და სხვადასხვა ეკრანზე მოხდება სხვადასხვა სიგრძის ცხრილის ასახვა.
  width ატრიბუტი შესაძლებელია გამოვიყენოთ უჯრის ამღწერ td ტეგშიდა ის განსაზღვრავს შესაბამისი უჯრის სიგრძეს.
- **height** ატრბუტი გამოიყენება სიმაღლის განსასაზღვრად, როცორც ცხრილის (table) ასევე სტრიქონის (tr) ან უჯრის (td) ტეგში. ერთ სტრიქონზე სხვადასხვა სიმაღლის უჯრის შექმნა შეუძლებელია, ამიტომ სიმაღლის განსაზღვრა საკმარისია სტრიქონის ერთერთ უჯრაში.
- **cellspacing** – ცხრილისუჯრებსშორის დაშირება (სურ.1.31. ბ).
- **cellpadding** –ცხრილისუჯრაშიარსებულინფორმაციასადამისჩარჩოებსშორის დაშორება (სურ.1.31. ა).

ა)



ბ)

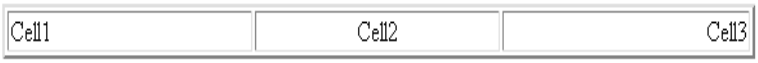


სურ. 1.31 ა) უჯრაში არსებულ ინფორმაციასა და მის ჩარჩოებს შორის დაშორება

ბ) ცხრილის უჯრებს შორის დაშირება

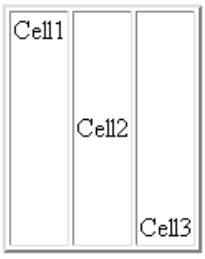
- **align –** უჯრაში არსებული ინფორმაციის ჰორიზონტალური მდებარეობის განსაზღვრა (სურ.1.32. ა). შესაძლო მნიშვნელობებია: left – მარცხნივ გასწორება, center – ცენტრირება, right – მარჯვნივ გასწორება
- **valign –** უჯრაში არსებული ინფორმაციის ვერტიკალური მდებარეობის განსაზღვრა (სურ.1.32. ბ). შესაძლო მნიშვნელობებია: top – ინფორმაციის უჯრის ზედა ნაწილში განთავსება, middle – შუაში და bottom – ქვედა ნაწილში განთავსება.
  align/valign ატრიბუტების სტრიქონის განმსაზღვრელ ტეგში გამოყენებიშ შემთხევვაში მდებარეობები გავრცელდება შესაბამისი სტრიქონის ყველა უჯრაზე.

ა)




```
<table width="60%" border="1">
  <tr>
    <td align="left">Cell1</td>
    <td align="center">Cell2</td>
    <td align="right">Cell3</td>
  </tr>
</table>
```

ბ)




```
<table width="60%" border="1">
  <tr>
    <td align="top">Cell1</td>
    <td align="middle">Cell2</td>
    <td align="bottom">Cell3</td>
  </tr>
</table>
```

სურ. 1.32 უჯრაში ინფორმაციის მდებარეობა

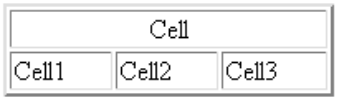
ა) ჰორიზონტალური ბ) ვერტიკალური

– **bgcolor** – უჯრაში ფერის ჩასხმა. მიეთითება ფერის შესაბამისი ექვსნიშნა კოდი. მაგ. #ffffff - თეთრი ფერი, #000000 - შავი ფერი, #ff0000 – წითელი და.ა.შ.

<u>შენიშვნა:</u> ცხრილის ვიზუალური სახისა და მასში განთავსებული ინფორმაციის მდებარეობის განსაზღვრა მართებულია მოხდეს არა ზემოთაღწერილი ატრიბუტების მეშვეობით არამედ css კასკადური სტილების ენის გამოყენებით. css თვისებები და მათი გამოყენების შესაძლებლობები დაწვრილებით აღწერილია შემდეგ თავში.

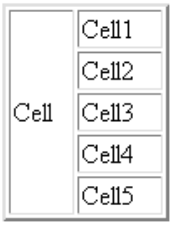
ცხრილების ორგანიზებისას ხშირად დგება უჯრების გაერთიანების საჭიროება, რომლის შესაძლებლობასაც იძლევა ატრიბუტები: colspan – ჰორიზონტალურად უჯრების გაერთინება და rowspan – ვერტიკალურად უჯრების გაერთიანება. მათზე მინიჭებული მნიშვნელობა განსაზღვრავს გასაერთიანებელი უჯრების რაოდენობას.

ა)




```
<table width="60%" border="1">
  <tr>
    <td colspan="3">Cell</td>
  </tr>
    <tr>
    <td>Cell1</td>
    <td>Cell2</td>
    <td>Cell3</td>
  </tr>
</table>
```

ბ)




```
<table width="30%" border="1">
  <tr>
    <td rowspan="5">Cell</td>
        <td>Cell1</td>
  </tr>
    <tr><td>Cell2</td> </tr>
    <tr><td>Cell3</td> </tr>
<tr><td>Cell4</td> </tr>
<tr><td>Cell5</td> </tr>
</table>
```

სურ. 1.33 უჯრების გაერთიანებაა) ჰორიზონტალურად ბ) ვერტიკალურად

სურ. 1.34–ზე მოცემულია ვალუტის კურსის ცხრილის ნიმუში და მისი ბრაუზერში გამოსახვისათვის საჭირო html დოკუმენტი.

ვებგვერდი

საქართველოს ბანკი BANK OF GEORGIA	ვალუტის კურსი		1 ლარი =	
	ყიდვა	გაყიდვა	ყიდვა	გაყიდვა
აშშ დოლარი	2.375	2.455	0.421	0.407
ევრო	2.575	2.695	0.388	0.371
გირვანქა სტერლინგი	3.43	3.59	0.292	0.279

html დოკუმენტი

```
<!DOCTYPE html>
<html>
<head>
<title>ვალუტის კურსი</title>
<meta charset="UTF-8">
</head>
<body>
<table width="80%" border="1">
<tr>
    <td rowspan="2" width="220"><img src="img/bog.jpg" alt="საქართველოს ბანკი"></td>
    <td colspan="2" bgcolor="#dddddd" align="center">ვალუტის კურსი</td>
    <td colspan="2" bgcolor="#dddddd" align="center">1 ლარი =</td>
</tr>
<tr bgcolor="#dddddd" align="center">
    <td>ყიდვა</td>
    <td>გაყიდვა</td>
    <td>ყიდვა</td>
    <td>გაყიდვა</td>
</tr>
<tr>
    <td><img src="img/usa.png" alt="">აშშ დოლარი</td>
    <td>2.375</td>
    <td>2.455</td>
    <td>0.421</td>
    <td>0.407</td>
</tr>
<tr>
    <td><img src="img/eur.png" alt="">ევრო</td>
    <td>2.575</td>
    <td>2.695</td>
    <td>0.388</td>
    <td>0.371</td>
</tr>
<tr>
    <td><img src="img/gbp.png" alt="">გირვანქა სტერლინგი</td>
    <td>3.43</td>
    <td>3.59</td>
    <td>0.292</td>
    <td>0.279</td>
</tr>
</table>

</body>
</html>
```

სურ. 1.34 ვალუტის კურსის ცხრილი

პრაქტიკული დავალება

შექმენით სურ. 1.35–ზე მოცემული ცხრილის ბრაუზერში წარმოსახვისათვის საჭირო html დოკუმენტი.

თბილისის საერთაშორისო აეროპორტი

საერთაშორისო გამგზავრება			
თარიღი	დრო	ავიაკომპანია	მიმართულება
09.01.2016	4:25 PM	Airzena Georgian Airlines	როსტოვი
09.01.2016	5:10 PM	Ukraine International	კიევი
09.01.2016	5:25 PM	S7 Airlines	სტამბული
09.01.2016	6:15 PM	Turkish Airlines	
ადგილობრივი გამგზავრება			
თარიღი	დრო	ავიაკომპანია	მიმართულება
10.01.2016	4:25 PM	Airzena Georgian Airlines	ზათუმი

სურ. 1.35 ცხრილის პრაქტიკული ნიმუში

1.3 ვებგვერდზე ფორმების ასახვა

სწავლის შედეგის შესაბამისი თემატიკა

- ფორმის ელემენტები და მათი ტიპები
- ვებგვერდზე ფორმის ველების ასახვის ტეგები და მათი შესაბამისი პარამეტრები

ვებსაიტზე მომხმარებელთა ინტერაქტიულობის ორგანიზებისათვის გამოიყენება ფორმები. იგი იძლევა საშუალებას ვიზიტორისგან მივიღოთ სხვადასხვა სახის ინფორმაცია და დავუბრუნოთ პასუხი.

ფორმების ვებგვერდზე ასახვა ხდება html-ის საშუალებით, მაგრამ ინფორმაციის მიღება, დამუშავება, დაბრუნებისათვის გამოიყენება ვებ პროგრამირების ენები (PHP, Perl).

ამ თავში განვიხილათ html-ის საშუალებით ვებგვერდზე სხვადასხვა ტიპის ფორმების ველების ასახვისა და მათი პარამეტრების გაწერის შესაძლებლობებს.

ფორმის ელემენტები და მათი ტიპები

ნებისმიერი ფორმის საბაზო ტეგია <form>, რომელიც მოითხოვს დამხურავ ტეგს </form>. აღნიშნული ტეგის ატრიბუტები განსაზღვრავენ ფორმის დამუშავებისათვის აუცილებელ ინფორმაციას (სურ. 1.36).

```
<form action="action_page.php" method="GET" accept-charset="UTF-8" enctype="application/x-www-form
    urlencoded"name="test"target="_blank">
    .
    ფორმის ელემენტები
    .
</form>
```

სურ. 1.36. ფორმის საბაზო ტეგი

- **action** ატრიბუტისმნიშვნელობადმიეთითებაფაილისმისამართი/სახელი (მაგ. *.php ან *.asp), რომელშიცპროგრამულადააღწერილითურამოქმედებებიუნდაშესრულდესფორმიდანმიღებულმონაცემებზე. ამფაილსქმნისვებპროგრამისტი.
- **method**ატრიბუტიუთითებს, თუროგორუნდამოხდესმონაცემებისგადაცემა და მისი მნიშვნელობადამოკიდებულიაგადასაცემიმონაცემისზომასადამნიშვნელობაზე.
  GET ღია მეთოდის გამოყენებითმონაცემებისგადაგზავნახდებასამისამართოველიდან URL-ისსაშუალებით. ამმეთოდითშესაძლებელიამხოლოდ 255 სიმბოლოსგადაგზავნა. ესმეთოდიგამოიყენებაძებნისდროსდაამდროსთუდაკვირდებითსამისამართოველსშენიშნავთთქვენსმიერშეყვანილინფორმაციას. იგიგამოიყენებავებგვერდის გამოძახების/ ძებნის შემთხვევაში.
  POST დახურული მეთოდი გამოიყენება დიდი ზომის ინფორმაციის სერვერზე გასაგზავნად. ასე გადაგზავნილი მონაცემები არ ჩანს სამისამართო ველში, რადგანაც ისინი მოთავსებულია შეტყობინების ტანში - body.
- **accept-charset** ატრიბუტით განისაზღვრება ფორმის ველების კოდირება. თუ ეს ატრიბუტი არ არის განსაზღვრული ავტომატურად გამოიყენება ვებგვერდისათვის განსაზღვრული კოდირება.
- **enctype**ატრიბუტის მნიშვნელობა განსაზღვრავსსერვერზეგადაგზავნილიინფორმაციისკოდირებისმეთოდს. სერვერისათვის მონაცემების უსაფრთხოდ გადაცემის მიზნით, ბროუზერი ახდენს მათ კოდირებას სპეციალური მეთოდით. სტანდარტულ კოდირებას წარმოადგენს - application/x-www-form-urlencoded. თუ ფორმა შეიცავს ფაილის არჩევა/მიბმის ტიპის ველს, მაშინ გამოიყენება კოდირების მეთოდი - multipart/form-data; თუ ფორმის მონაცემების გადაეცემა ხდება

ფოსტის საშუალებით, მაშინ გამოიყენება - text/plain (ამ შემთხვევაშიaction ატრიბუტში მიეთითება mailto:საფოსტო მისამართი).

– **name** ატრიბუტით ფორმასენიჭებაუნიკალურისახელი, რომელისსაშუალებითაც ხდება ფორმისელემენტებთანწვდომა. ამავე დანიშნულებით შესაძლებელია გამოვიყენოთ ატრიბუტი **id.**

Form ტეგისზემოთ განხილული ყველა ატრიბუტი ემსახურებოდა ფორმის მონაცემების დამუშავებას და განკუთვნილია ვებპროგრამისტისთვის.

– **tarjet** ატრიბუტით განისაზღვრება რომელ ფანჯარაში გაიხსნას სერვერიდან დაბრუნებული ინფორმაცია. target="_blank" ინფორმაცია გაიხსნება ახალ ფანჯარაში, target="_self" - იმავე ფანჯარაში. თუ ატრიბუტი tarjet არ არის მითითებული, მაშინ მიღებული ინფორმაცია ავტომატურად გაიხსნება იმავე ფანჯარაში.

ვებგვერდზე ფორმის ველების ასახვის ტეგები და მათი შესაბამისი პარამეტრები

ფორმის ყველაზე მნიშვნელოვანი ელემენტია <input type="ველის ტიპი" name="ველის სახელი">

name ატრიბუტის მნიშვნელობა ველს ანიჭებს უნიკალურ სახელს. მომხმარებლის მიერ შესაბამის ველში შეყვანილი ინფორმაცია ველის სახელით, ცვლადის სახით გადაეცემა შესაბამის ფუნქციას/ფაილს სადაც ხდება მისი დამუშავება. ამ ატრიბუტის მნიშვნელობას იყენებს ვებპროგრამისტი და აუცილებელია გათვალისწინებული იქნას კოდის წერის ეთიკა და ლოგიკურად იქნას შერჩეული.

type ატრიბუტის მინშვნელობა განსაზღვრავს ფორმის ველის ტიპს.

ცხრილი 1. 2 ველის ტიპები

type ატრიბუტის მინშვნელობა	ველის ტიპები
text	ტექსტის შესაყვანი ველი
password	პაროლის შესაყვანი ველი
submit	ღილაკი, ფორმის მონაცემების გასაგზავნად
button	ღილაკი
reset	ღილაკი ფორმის გასუფთავებისათვის
image	სურათის ველი
radio	გადამრთველი
checkbox	ალმით მოსანიშნი
File	ფაილის ასატვირთი ველი

ტექტსის/პაროლის შესაყვანი ველი და ღილაკის ტიპები

სურ. 1.36-ზე მაგალითისათვის მოცემულია ავტორიზაციის ფორმის ნიმუში, რომელიც შედგება სამი ტიპის ველისაგან:

1. <input type="text" name="username"> - ტექსტის შესაყვანი ველი
2. <input type="password" name="psw"> - პაროლის შესაყვანი ველი. ველის თავისებურებიდან გამომდინარე მასში შეყვანილი სიმბოლო დაფარულია (სისტემიდან გამომდინარე გამოისახება მუქი წერტილებით ან ფიფქებით) და პაროლის შეყვანისას შეუძლებელია მისი დანახვა.
3. <input type="submit" value="Submit"> - ღილაკი, რომელიც ფორმაში არსებულ მონაცემებს გადაუგზავნის სერვერზე არსებულ დამმუშავებელ ფაილს (action.php).

html დოკუმენტი	ვებგვერდი

```
<form action="action.php" method="GET" enctype="application/x-www-form urlencoded"name="login">
 მომხმარებელი:<br />
 <input type="text" name="username"><br />
 პაროლი:<br />
 <input type="password" name="psw"><br />
 <input type="submit" value="Submit">
</form>
```

მომხმარებელი:
eka.chikashua
პაროლი:
••••••••
Submit

სურ. 1.36 ავტორიზაციის ფორმის ნიმუში1

ტექსტური/პაროლის შესაყვანი ველის ტეგში შეგვიძლია გამოვიყენოთ ორი ატრიბუტი:

- **Size** რომლის მნიშვნელობაც განსაზღვრავს ჩარჩოს სიგრძეს (ავტომატურად size="20"). ტექსტური ველის სიგრძე განისაზღვრება ერთმანეთის გვერდზე მდგომი მონოსიგანის (monospace font- ერთნაირი სიგანის მქონე სიმბოლოებიანი შრიფტი) შრიფტის მქონე სიმბოლოების რაოდენობით. თუ შრიფტის ზომა იცვლება სტილების საშუალებით თავისთავად შესაბამისად იცვლება ველის სიგრძეც. size ატრიბუტი განსაზღვრავს ველის სიგრძეს (სურ. 1.37) და არა მასში შესატანი სიმბოლოების რაოდენობას. თუ მონაცემთა შესატან ველში შეტანილი სიმბოლოების რაოდენობა ველის სიგრძეს აღემატება, მაშინ პირველად აკრეფილი ტექსტის გადაფარვა ხდება.
- **Maxlenght** - განსაზღვრავს ველში შესაყვანი სიმბოლოების დასაშვებ მაქსიმალურ რაოდენობას. მითიტებულ მნიშვნელობაზე მეტი სიმბოლოს შეტანის მცდელობის დროს ბრაუზერი ხმოვან სიგნალს გამოსცემს და ზედმეტი სიმბოლოს შეტანის საშუალებას არ იძლევა. თუ მოცემული ატრიბუტი არ იქნება მითითებული, მაშინ მისი მნიშვნელობა არ არის განსაზღვრული.

ღილაკის შექმნისას უნდა გავითვალისწინოთ მისი დანიშნულება/ფუნქცია:

- type="button" - ჩვეულებრივი ღილაკი
- type="submit" - ღილაკი, რომელიც ფორმაში არსებულ მონაცემებს გადაუგზავნის სერვერზე არსებულ დამმუშავებელ ფაილს
- type="reset" - ღილაკი, რომლის საშუალებითაც ხდება ფორმის მონაცემების გასუფთავება და სტანდარტულ მნიშვნელობებზე დაბრუნება

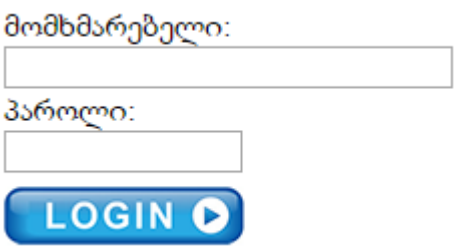
ღილაკის შექმნისას **value** ატრიბუტზე მინიჭებული მნიშვნელობა აისახება ტექსტის სახით ღილაკზე (სურ. 1.36). შესაძლებელია ღილაკზე სურათის გამოსახვა. ამ შემთხვევაში type="image" და აუცილებელია ატრიბუტში src მიეთითოს სურათის შესაბამისი ფაილის მისამართი (სურ. 1.37). სასურველია alt

ატრიბუტია გამოყენებაც, რათა სურათის არჩატვირთვის შემთხვევაში მომხმარებელი მიხვდეს ღილაკის დანიშნულებას.

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="login">
  მომხმარებელი:<br />
  <input type="text" name="username" size="30"><br />
  პაროლი:<br />
  <input type="password" name="psw" size="15"><br />
  <input type="image" src="login.png" alt="LogIn">
</form>
```

ვებგვერდი



სურ. 1.37 ავტორიზაციის ფორმის ნიმუში2

გადამრთველები/რადიოღილაკები

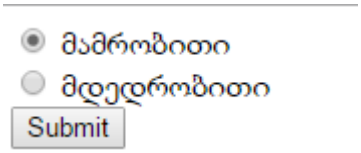
გადამრთველი/რადიოღილაკების ტიპის ველის (type="radio") თავისებურება იმაში მდგომარეობს, რომ ის მომხმარებელს შემოთავაზებული რამოდენიმე ვარიანტებიდან მხოლოდ ერთი მონაცემის არჩევის საშუალებას აძლევს (სურ. 1.38). გადამრთველების შემთხვევაში ველის სახელი name="gender" ერთი და იგივე უნდა იყოს, რადგან ისინი ფაქტიურად ერთ ელემენტს წარმოადგენენ, ხოლო მნიშვნელობა ენიჭებათ სხვადასხვა value="male" და value="gender" და სერვერზე გადაიგზავნება მომხამრებლის მიერ არჩეული მნიშვნელობა.

Checked ატრიბუტი განსაზღვრავს რომელი მნიშვნელობა იყოს ავტომატურად მონიშნული. იგი მოსახერხებელია, ისეთი პასუხების მოსანიშნად, რომელსაც ხშირად ირჩევენ მომხმარებლები. ამ ატრიბუტის მითითების გარეშე არცერთი მნიშვნელობა არ იქნება ავტომატურად მონიშნული.

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="genderForms">
  <input type="radio" name="gender" value="male" checked>
  მამრობითი<br />
  <input type="radio" name="gender" value="female">
მდედრობითი <br />
  <input type="submit" value="Submit">
</form>
```

ვებგვერდი



სურ. 1.38 გადამრთველი ველის ნიმუში

გადამრთველი/რადიოღილაკის სურ. 1.38-ზე განხილული ნიმუშის მიხედვით შექმნისას, მომხმარებელმა სასურველი ვერსიის მოსანიშნად მაუსის მაჩვენებელი აუცილებლად უნდა დააჭიროს ფორმის აღმნიშვნელ წრეს/ღილაკს (შესაბამის ტექსტზე დაჭერით ველი არ ინიშნება).

ფორმასთან მომხამარებლის მუშაობის გასაადვილებლად გამოიყენება label ელემენტი, რომელიც ფორმის ვიზუალურ მხარეს არ ცვლის, მაგრამ ამ შემთხვევაში სასურველი ვერსიის მონიშვნა შესაძლებელია შესაბამის ტექსტზე მაუსის მაჩვენებლის დაჭერითაც.
input ელემენტში, რომელთანაც უნდა დაკავშირდეს label ელემენტში აღწერილი ტექსტი ვამატებთ id ატრიბუტს, რომლის მნიშვნელობა label ელემენტში მითითებული for ატრიბუტის მნიშვნელობის იდენტური უნდა იყოს.

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-www-form
urlencoded"name="genderForms">
  <input type="radio" name="gender" id="male" value="male">
<label for="male">მამრობითი</label><br/>
  <input type="radio" name="gender" id="female" value="female">
<label for="female">მდედრობითი </label><br/>
  <input type="submit" value="Submit">
</form>
```

სურ. 1.39 გადამრთველი ველის ნიმუში label ელემენტის გამოყენებით

მომხმარებლის ფორმის აღქმის გასამარტივებლად, შესაძლებელია ფორმის ველების დაჯგუფება fieldset ელემენტის საშუალებით. fieldset ელემენტი აჯგუფებს ფორმაში განთავსებულ ველებს და შემოსაზღვრავს მათ ჩარჩოთი. ჯგუფის სათაურის განსაზღვრა კი შესაძლებელია legend ელემენტით (სურ. 1.40).

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-www-form
urlencoded"name="genderForms">

  <fieldset>
  <legend>სქესი:</legend>

<input type="radio" name="gender" id="male" value="male">
<label for="male">მამრობითი</label><br/>
  <input type="radio" name="gender" id="female" value="female">
<label for="female">მდედრობითი </label><br/>
  <input type="submit" value="Submit">

</fieldset>
</form>
```

ვებგვერდი

სურ. 1.40 გადამრთველის ველების დაჯგუფება

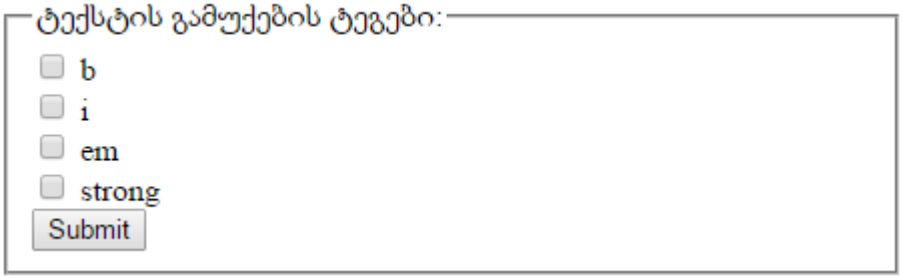
ალმით მოსანიშნი

ალმით მოსანიშნი ველი (type="checkbox") მომხმარებელს ასევე სთავაზობს არჩევანის გაკეთების საშუალებას, მაგრამ ზემოთგანხილული გადამრთველებისაგან განსხვავებით, ამ შემთხვევაში მომხმარებელს შეუძლია რამდენიმე ვარიანტის ერთდროულად მონიშვნა. გადამრთველების მსგავსად, ალმით მოსანიშნი ფორმის შემთხვევაშიც, ველის სახელი name=" fontWeight" ერთი და იგივე უნდა იყოს, რადგან ისინი ფაქტიურად ერთ ელემენტს წარმოადგენენ, ხოლო მნიშვნელობა ენიჭებათ სხვადასხვა value="b", value="i", value="em" და name="strong" და სერვერზე გადაიგზავნება მომხამრებლის მიერ არჩეული მნიშვნელობა.

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-www-form
urlencoded"name="fontWeight">
  <fieldset>
  <legend>ტექსტის გამუქების ტეგები:</legend>
<input type="checkbox" name="fontWeight" id="b" value="b">
<label for="b">b</label><br/>
  <input type="checkbox" name="fontWeight" id="i" value="i">
<label for="i">i</label><br/>
  <input type="checkbox" name="fontWeight" id="em" value="em">
<label for="em">em</label><br/>
  <input type="checkbox" name="fontWeight" id="strong" value="strong">
<label for="strong">strong</label><br/>
  <input type="submit" value="Submit">
</fieldset>
</form>
```

ვებგვერდი



სურ. 1.41 გადამრთველის ველების დაჯგუფება

!შენიშვნა: ფორმის ველებში, რომლიდანაც შეიძლება რამოდენიმე მნიშვნელობის სერვერზე ერთდროული გადაგზავნა ატრიბუტის name მნიშვნელობად გამოიყენება მასივები.

ფაილის ასატვირთი ველი

ხშირად იქმნება ფაილის სერვერზე ატვირთვის საჭიროება. მომხმარებლებმა შეიძლება ატვირთონ სხვადასხვა ტიპის ფაილები, მაგ. გრაფიკული, ვიდეო ან აუდიო ფაილები.

ფაილის ასატვირთი ველის შესაქმნელად ატრიბუტს type ენიჭება მნიშვნელობა file, ხოლო accept ატრიბუტის მნიშვნელობით განისაზღვრება ასატვირთი ფაილის ტიპი.

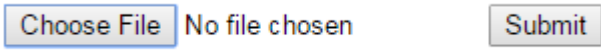
```
<input type="file" name="picture" accept="file_extension">
<input type="file" name="picture" accept="image/*">
<input type="file" name="picture" accept="audio/*">
<input type="file" name="picture" accept="video/*">
<input type="file" name="picture" accept="media_type">
```

ფაილის ასატვირთი ველის უჩნდება ღილაკი Choose File, რომელზე დაჭერაც იძლევა ასატვირთი ფაილის შერჩევის საშუალებას.

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="applicati
on/x-www-form urlencoded"name="fileUpload">
<input type="file" name="picture" accept="image/*">
 <input type="submit" value="Submit">
</form>
```

ვებგვერდი



სურ. 1.42 ფაილის ასატვირთი ველის ნიმუში

ჩამოშლადი ჩამონათვლის ველები

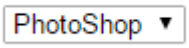
მომხმარებლისათვის შეთავაზებული მნიშვნელობიდან სასურველის არჩევის საშუალებას იძლევა ჩამოშლადი ველები. გადამრთველებისა და ალმით მოსანიშნი ველისაგან განსხვავებით ჩამოშლადი ველი საშუალებას იძლევა კომპაქტურად განვალაგოთ მასში შემავალი პუნქტები და წინასწარ განსაზღვროთ ველის რამდენი პუნქტი იყოს მომხმარებლისათვის ხილული.

ჩამოშლადი ველის შესაქმნელად გამოიყენება select ელემენტი, რომელშიც ჩამონათვალის პუნქტები აღიწერება option ტეგის საშუალებით. option ელემენტის value ატრიბუტს ენიჭება მნიშვნელობა, რომელიც გადაეცემა სერვერს შესაბამისი პუნქტის არჩევის შემთხვევაში. selected ატრიბუტი განსაზღვრავს ავტომატურად მონიშნულ პუნქტს (სურ. 1.43).

html დოკუმენტი

```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="Programs">
<select>
 <option value="Word">Word</option>
 <option value="Excel">Ecxel</option>
 <option value="Photoshop" selected>Photoshop</option>
 <option value="Illustrator">Illustrator</option>
</select>
</form>
```

ვებგვერდი



სურ. 1.43 ჩამოშლადი ველის ნიმუში

როცა არცერთ option ელემენტში არ არის მითითებული selected ატრიბუტი, ავტომატურად პირველი პუნქტი ჩანს ფორმაში, თუ გვსურს თავდაპირველად ცარიელი იყოს ფორმა შესაძლებელია პირველ პუნქტად დაემატოს ცარიელი Option ელემენტი <option value=" "> </option>
ჩამოშლადი ველის პუნქტების სისტემატიზირება, რათა მომხმარებელს გაუადვილდეს მუშაობა, შესაძლებელია optgroup ელემენტის საშუალებით, რომელიც label ატრიბუტის საშუალებით ჩამონათვალის პუნქტებს უქმნის სათაურებს (სურ. 1.44).

html დოკუმენტი	ვებგვერდი

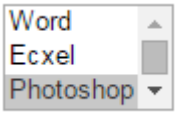
```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="Programs">
<select>
<optgroup label="Microsoft office">
  <option value="Word">Word</option>
  <option value="Excel">Ecxel</option>
</optgroup>
<optgroup label="Adobe">
  <option value="Photoshop" selected>Photoshop</option>
  <option value="Illustrator">Illustrator</option>
</optgroup>
</select>
</form>
```

სურ. 1.44 ჩამოშლადი ველის პუნქტების სისტემატიზირების ნიმუში

ზემოთგანხილული ჩამოშლადი ველის მაგალითებში მომხმარებლისათვის ველის ჩამოშლამდე ხილული არის მხოლოდ ერთი პუნქტი. თუმცა size ატრიბუტის საშუალებით შესაძლებელია განისაზღვრო ხილვადი ელემენტების რაოდენობა. თუ size ატრიბუტის მნიშვნელობა ჩამოშლადი ველის პუნქტებზე ნაკლებია ავტომატურად ჩნდება ვერტიკალური გადაფურცვლის ბილიკი (სურ.1.45).

html დოკუმენტი	ვებგვერდი

```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="Programs">
<select value="3">
  <option value="Word">Word</option>
  <option value="Excel">Ecxel</option>
  <option value="Photoshop" selected>Photoshop</option>
  <option value="Illustrator">Illustrator</option>
</select>
</form>
```



სურ. 1.45 ჩამოშლადი ველის ნიმუში, როცა სამი პუნქტია ხილული

შენიშვნა: select ელემენტში multiple ატრიბუტის დამატება <select multiple>იძლევა რამოდენიმე პუნქტის ერთდროულად მონიშვნის საშუალებას (ctrl კლავიშის დახმარებით).

ტექსტური არე

დიდი ზომის ტექსტების შესაქმნელად გამოიყენება ელემენტი textarea. ველის ზომა დამოკიდებულია ატრიბიტებზე cols - სიგრძე და rows - სიმაღლე.

html დოკუმენტი	ვებგვერდი

```
<form action="action.php" method="GET" enctype="application/x-
www-form urlencoded"name="mail">
 წერილი:<br />
 <textarea cols="30" rows="10">&amp;nbsp; </textarea>  <br />
 <input type="submit" value="Submit">
</form>
```

წერილი:

Submit

სურ. 1.46 ტექსტური არის ფორმის ნიმუში

კითხვები თვითშეფასებისათვის

1. **როგორ განიმარტება HTML?**
   a. Hyper Text Markup Language
   b. Hyperlinks and Text Markup Language
   c. Home Tool Markup Language
2. **რომელი ფორმატით უნდა შევინახოთ ვებგვერდი?**
   a. exe
   b. doc
   c. html
   d. ნებისმიერიზემოჩამოთვლილი
3. **რომელი ტეგებია სავალდებულო ვებგვერდის სტრუქტურის შესაქმნელად**
   a. doctype, html, head, title და p
   b. html და body
   c. html, title, body და p
   d. doctype, html, head, title და body
4. **სტანდარტულად რა ჰქვია ვებ საიტის საწყის ფაილს**
   a. first.html
   b. page.html
   c. index.html
   d. index.exe
5. **Html დოკუმენტში აუცილებელია ყველა ტეგი იყოს დახურული**
   a. მცდარია
   b. ჭეშმარიტია
6. **რომელი ტეგი განსაზღვრავს ვებგვერდის პირველი დონის სათაურს**
   a. <heading>
   b. <h6>
   c. <head>
   d. <h1>
7. **რომელი ტეგი უზრუნველყოფს ვებ ბრაუზერის სათაურის ველში ვებგვერდის სათაურის ასახვას?**
   a. <body>
   b. <head>
   c. <title>
8. **რომელი პროგრამა უზრუნველყოფს ვებგვერდის დათვალიერებას?**
   a. ნებისმიერიტექსტურირედაქტორი
   b. ნებისმიერიგრაფიკულირეადქტორი
   c. ნებისმიეროპერაციულსისტემაში
   d. ნებისმიერიბრაუზერი
9. **რომელი პროგრამაშია შესაძლებელი html კოდის აკრეფა**
   a. გრაფიკულრედაქტორებში
   b. ინტერნეტბრაუზერებში
   c. ვებედიტორებში ან/დატექსტურრედაქტორში
10. **html მარკირებისენაუზრუნველყოფავებგვერდისვიზუალურგაფორმებას**
    a. მცდარია
    b. ჭეშმარიტია
11. **html დოკუმენტშიტეგებიაუცილებელიააიკრიფოსპატარაასოებით**
    a. მცდარია

b. ჭეშმარიტია

12. რომელიტეგიგამოიყენება ვებგვერდზე მნიშვნელოვანი ტექსტისგამოსაყოფად?
    a. <p>
    b.
    c.
    d.
13. რომელი ტეგი გამოიყენება მარკირებული სიის ორგანიზებისათვის
    a.
    b.
    c.
    d. <dl>
14. რომელი ტეგი გამოიყენება დანომრილი სიის ორგანიზებისათვის
    a.
    b.
    c.
    d. <dl>
15. რომელი ტეგი უზრუნველყოფს სიის ცალკეული ელემენტის აღწერას
    a.
    b.
    c.
    d.
16. რომელი ატრიბუტი განსაზღვრავს ნუმერილებული სიის საწყის რიცხვს?
    a. type
    b. number
    c. start number
    d. start
17. რომელი ტეგი გამოიყენება აბზაცის შესაქმნელად?
    a. <p>
    b.
    c.
    d. <bloqoute>
18. რომელი ტეგი უზრუნველყოფს ახალ სტრიქონზე გადასვლას?
    a. <line>
    b. <enter>
    c.
19. რომელი ტეგი უზრუნველყოფს ზედა ინდექსში გადაყვანას
    b. <sub>
    c. <down>
    d. <sup>
    e. <up>
20. რომელი ტეგი უზრუნველყოფს ქვედა ინდექსში გადაყვანას
    a. <sub>
    b. <down>
    c. <sup>
    d. <up>

21. ვებ გვერდზე სურათის ასახვისათვის რომელი HTML ტეგია მართებული?

a.
b. <img="picture.jpg" alt=" " />
c.

22. რომელი ტეგი უზრუნველყოფს ბრაუზერში ქართული შრიფტის მხარდაჭერას

a. <meta charset="georgia">
b. <meta charset="UTF-8">
c. <meta fonts="UTF-8">
d. <meta fonts="georgia">

23. რომელი ატრიბუტია აუცილებელი img ტეგში?

a. img
b. name
c. width
d. src

24. რომელი ფორმატის გრაფიკული ფაილები გამოიყენება ვებ გვერდზე?

a. max, fla
b. cdr, pdf
c. png, jpg
d. tiff, bmp

25. რომელი ტეგის საშუალებით ხდება სურათის ჩასმა?

a. image
b. background
c. img
d. picture

26. img ტეგში რა განისაზღვრება alt ატრიბუტის საშუალებით

a. სურათის დასახელება
b. ბრაუზერში სურათის არ ჩატვირთვის შემთხვევაში ალტერნეტიული ტექსტი
c. სურათზე მაუსის მიტანისას ალტერნატიული ტექსტი

27. რომელ ტეგში უნდა ჩავსვათ bgcolor ატრიბუტი მთლიან სტრიქონზე ფონის ჩასხმისათვის

a. <tr>
b. <td>

28. რომელი ატრიბუტებითაა შესაძლებელი ცხრლის უჯრების გაერთიანება

a. colspan / rowspan
b. cellspan / tdspan

29. რომელი ტეგები გამოიყენება ცხრილის ასაგებად

a. <table><tr><td>
b. <table><p>
c. <html><table>
d. <html><body><table>

თავი 2.საიტის სტილებით გაფორმება - css

2.1 ვებსაიტის საბაზო ელემენტების გაფორმება სტილებით

მიმდინარეპარაგრაფისთემატიკა

- სტილებისმნიშვნელობადაგამოყენებისსფერო
- დოკუმენტისსტილებითგაფორმებისმეთოდები
- css–ისკოდისწერისსინტაქსი
- css კოდისწერისეთიკა
- ძირითადისელექტორები
- სელექტორებისთვისებისადამნიშვნლობისარსი
- Class და ID სელექტორებისმნიშვნელობადაგამოყენებისასპექტები

სტილების მნიშვნელობა და გამოყენების სფერო

ვინაიდან HTML -ის ენა მოკლებულია იმ შესაძლებლობებს, რომ გავაფორმოთ დოკუმენტის ვიზუალური მხარე ისე როგორც ეს თანამედროვეობას შეესაბამება, ამისათვის WEB-გვერდების ვიზუალიზაციაში აუცილებლობა წარმოიშვა შეექმნათ დამატებითი ენა რომელსაც ეწოდა CSS(Cascading Style Sheet) ანუ კასკადური სტილების ენა. CSS კასკადური სტილების ენა საშუალებას გვაძლევს დოკუმენტის ვიზუალიზაცია მომხმარებლისთვის იყოს კომფორტული, ადვილად აღსაქმელი და რაც ყველაზე მთავარია დოკუმენტი ყველა მოწყობილობაზე აისახებოდეს ჩვენთვის სასურველი ფორმით. CSS ენის გამოჩენამ WEB ტექნოლოგიების სივრცეში, მისცა დეველოპერებს საშუალება შეექმნათ უფრო მიმზიდველი და ეფექტური საიტები.

აღნიშნული ენის დანერგვამ მკვეთრად გამიჯნა ერთმანეთისგან HTML დოკუმენტის სტრუქტურული ნაწილი მისი ვიზუალური ნაწილისგან. როდესაც ამ ენის შექმნამდე ვიყენებდით ათასგვარ ხრიკს, რომ დოკუმენტი ვიზუალურად დაგვეფორმატებინა ჩვენი სურვილის და მიხედვით, ახლანდელ დროში ეს პრობლემა აღმოიფხვრა CSS ენის წყალობით. ანუ კიდევ ერთხელ ავღნიშნოთ, რომ HTML კოდს ვიყენებთ რეალურად დოკუმენტის სტრუქტურის შექმნაში და CSS კოდს ვიყენებთ ცალსახად მხოლოდ მის ვიზუალურ ფორმირებაში და მიმზიდველი ეფექტების შექმნისთვის. აქვე ერთ გარემოებას გავუსვათ ხაზი, CSS ენა არავითარ შემთხვევაში არ წარმოადგენს HTML ენის შემცვლელ ალტერნატიულ ტექნოლოგიას. როგორც ზემოთ ავღნიშნეთ ეს ორი კოდირების ენა ერთმანეთს მკვეთრად ემიჯნება თავისი დანიშნულების მიხედვით.

ახლა მოდით ავღწეროთ თუ რის გამო არის CSS-ი HTML კოდირების ენისგან განსხვავებული დოკუმენტის ვიზუალიზაციის ნაწილში. როგორც წესი მოგეხსენებათ HTML ენაში დოკუმენტის სტრუქტურა იქმნება თაგების საშუალებით და თითოეული დოკუმენტის ელემენტს, იქნება ეს გრაფიკული გამოსახულება თუ ტექტსური ელემენტი, გარკვეული ატრიბუტის (პარამეტრის) მეშვეობით ენიჭება დაშვებული მნიშვნელობები, როგორიცაა სიგანე, სიმაღლე, ტექსტის ფერი და ასე შემდეგ. რაც შეეხება CSS-ის კოდირების ენას მთელი მისი შექმნის შინაარსი გამოიხატება იმაში, რომ ერთი მითითებების ნაკრები შეგვიძლია გამოვიყენოთ კასკადურად ერთი ჯგუფის ქვეშ გაერთიანებული თაგებისთვის. თუ რა აერთიანებს ამ თაგებს ერთი ჯგუფის ქვეშ ამაზე ქვემოთ დაწვრილებით ვისაუბრებთ.

ქვემოთ მოყვანილ 3.1. ცხრილში გთავაზობთ გაეცნოთ იმ განსხვავებებს რაც ანსხვავებს ერთმანეთისგან CSS ენის ვერსიებს, თუმცა აქვე დავაზუსტოთ, რომ ვერსიებს შორის მხოლოდ ატრიბუტიკის ნაწილშია სხვაობა და არავითარი პრინციპული ან ლოგიკური სხვაობა არ არსებობს.

მინიშნება: თუ თქვენ დაეუფლებით CSS1 -ის გამოყენების პრინციპებს, არ იფიქროთ, რომ ამ ენის მომდევნო ვერსიების შესასწავლად გარკვეული მნიშვნელოვანი ძალისხმევა დაგჭირდებათ

ცხრილი 3.1 CSS ენის ვერსიების შესაძლებლობები

ვერსია	მიღებისთარიღი	შესაძლებლობები
CSS1	01.1996	• ფურცელზეელემენტისასახვისწესისმართვა • ელემენტისდატექსტისურთიერთმიმართება • ელემენტისზომებისმართვა • შიდადაგარედაძვრებისმართვა • ცხრილებშივერტიკალზეგანლაგებისმართვა • ელემენტისსაზღვრებისსტილისმართვა • სიებისდაფორმატებისმართვა • ტექსტისფერისდაფონისმართვა • შრიფტისპარამეტრებისმართვა • ტექსტისთვისებებისმართვა • სტრიქონებსშორისინტერვალებისმართვა
CSS2	05.1998	CSS1-ისშესაძლებლობანიდადამატებითისიახლეები: • ტექსტისმიმართულებისმართვა • ელემენტისპოზიციონირებისმართვა • უბნებისხილვადობისმართვა • საზღვრებიდანგასულიელემენტებისმართვა • კურსორისგარეგნულისახისმართვა • ფენებისმართვა • ელემენტისზღვრულიზომებისმართვა • ცხრილისუჯრედებსშორისდაცილებები • ელემენტისცალკეულისაზღვრებისმართვა • ცხრილისელემენტებისზომებისმართვა • ბრჭყალებისსტილისმართვა • ბეჭდვისასკონტენტისმართვა • ბგერისხმამაღლობის, პაუზებისმართვა
SS2.1	09.2009	• მოხდაCSS2-შიდაშვებულიშეცდომებისგასწორებადაზოგიერთიმომენტისდაზუსტება, მათშორისპერსპექტივის
CSS3		• მომრგვალებულიკუთხეებისმხარდაჭერა • გრადიენტულისაზღვრებისმხარდაჭერა • ელემენტისჩრდილებისმართვა • არასტანდარტულიშრიფტებისგამოყენებისშესაძლებლობა • მომხმარებლისბლოკებისზომებისმართვა • ტექსტისსვეტებადდაყოფა • დაზოგისხვაგრაფიკულიეფექტი

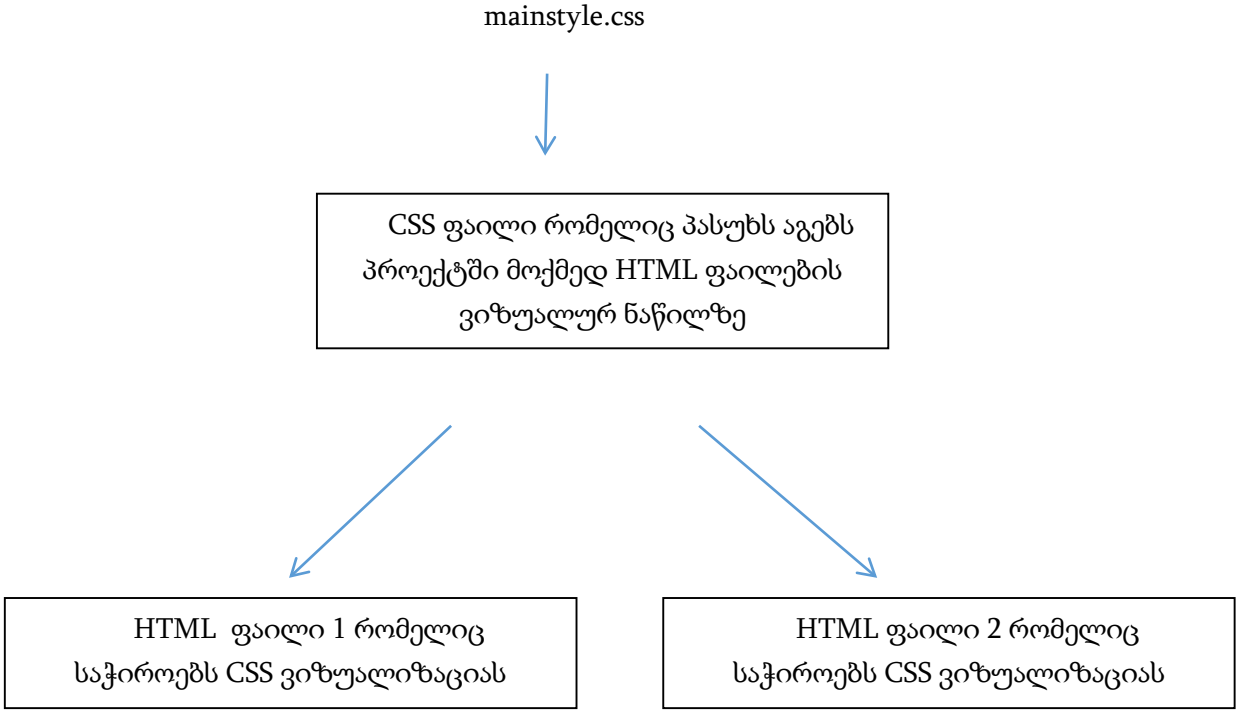
HTML დოკუმენტში არსებობს სტილის შემოტანის 3 მეთოდი, ესენი გახლავთ

1. სტილისდოკუმენტშიიმპორტირება - External style sheet
2. სტილისდოკუმენტისთავის (head)განყოფილებაშიჩაშენება - Internal style sheet
3. ხაზოვანისტილი - Inline style

სტილისდოკუმენტშიიმპორტირება - External style sheet

HTML დოკუმენტში CSS სტილის ჩანეგვის ერთი მეთოდი გახლავთ ცალკე CSS ფაილის შექმნა მაგალითად (mainstyle.css) მისი შემდგომი იმპორტირებისთვის. გთხოვთ ყურადღება მიაქციოთ ფაილის გაფართოებას .css ანუ მსგავსი ტიპის ფაილები, როგორც წესი იქმნება გლობალურად მთელი პროექტის ძირითადი ვიზუალური სტილის შესაქმნელად. მასში ვანთავსებთ ხოლმე, პროექტის ძირითადი შრიფტის ზომას, ფერს, სათაურების სტილებს და ასე შემდეგ. მოკლედ რომ ვთქვათ ასეთი იმპორტირებისთვის გამიზნული CSS ფაილი ემსახურება სრული პროექტის ვიზუალიზაციას და მის საბაზო ეფექტებს. პრინციპი შემდეგია: იქმნება css გაფართოების ფაილი, ხოლო შემდგომ ამ ფაილს იმპორტირებით ვაშენებთ ჩვენთვის სასურველ HTML დოკუმენტში. ქვემოთ წარმოგიდგენთ პრინციპს და სინტაქსს თუ როგორ გამოვიყენოთ ზემოთხსენებული მიდგომა CSS ფაილთან მიმართებაში.

3.1 სურათხე მოცემული სქემიდან ნათლად გამოიხატება, რომ ერთი CSS ფაილი შეგვიძლია გამოვიყენოთ რამოდენიმე ფაილში, რაც იმას ნიშნავს, რომ მთელი პროექტისთვის წინასწარ მოფიქრებული და გათვლილი ფონტის ზომა, დოკუმენტის ფონის ფერი ან მისი გრაფიკული ფონის მართვა და ასე შემდეგ შეგვიძლია ერთი ფაილიდან.



სურ. 3.1. სტილის დოკუმენტში იმპორტირების სტრუქტურა

რაც შეეხება კოდის ნაწილს, მოცემულიკოდისფრაგმენტიაღწერსთუროგორშეგვიძლიაგარე CSS ფაილისიმპორტირებარომელიმე HTML დოკუმენტში.

```
<head>
<link rel="stylesheet" type="text/css" href="mainstyle.css">
</head>
```

აქვეგავაკეთოთმინიშნებაიმისშესახებ, რომიმპორტირებულ mainstyle.css ფაილშიგაწერილიაბსოლიტურადყველაატრიბუტი/თვისება იქონიებსზემოქმედებას HTML ფაილისვიზუალურნაწილზე,რომელშიცგამოვიძახეთკონკრეტული mainstyle.css ფაილიიმპორტირებისმეთოდით. ანუდაიმპროტირებულფაილშითუგაგვაჩნიამითითება, რომფონტისფერიიყოსწითელიმთელიდოკუმენტისთვისანრომელიმეკონკრეტულიობიექტისთვისჩათვალეთ, რომესასეციქნებათურათქმაუნდაყველაფერისწორადგვაქვსორგანიზებული.

სტილის ფაილის იმპორტირების შემთხვევაში საჭიროა ყურადღებით გავითვალისწინოთ ჩვენი პროექტის ფაილური წყობა, ანუ რეალურად თაგს <LINK> ატრიბუტში HREF ენიჭება არა ფაილის სახელი მხოლოდ, არამედ ფაილის URL (Universal Reosource Location) მისამართი.

მაგ. თუ სტილის ფაილებისათვის გამოყოფილის ცალკე საქაღალდე styles, რომელშიც არის მოთავსებული mainstyle.css ფაილი, მაშინ ამ ფაილის html დოკუმენტში ექსპორტირებისას უნდა მიეთითოს საქაღალდის სახელი styles.

```
<head>
<link rel="stylesheet" type="text/css" href="styles/mainstyle.css">
</head>
```

სტილისდოკუმენტისთავისგანყოფილებაშიჩაშენება - Internal style sheet

რეალურად რომ მივუდგეთ ამ საკითხს, ანუ არჩევანის წინაშე რომ დავდგეთ რომელი მეთოდი ჯობია, სტილის იმპორტირება თუ მისი დოკუმენტის სტრუქტურაში ჩაშენება? ცალსახა პასუხი ამაზე არ არსებობს, ყოველივე გამომდინარეობს პროექტის სპეციფიკიდან, ანუ უკეთესი არჩევანის გაკეთება ამ ორ ვარიანტს შორის დამოკიდებულია პროექტის სტრუქტურაზე და ზოგადად არ შეიძლება განისაზღვროს ამ ორიდან რომელი უფრო ოპტიმალურია.

სტილის დოკუმენტის თავში ჩაშენება ხორციელდება მოცემული კოდის ფრაგმენტის საშუალებით.

```
<head>
<style type="text/css">

... ამ ნაწილში ხდება სტილის თვისებების აღწერა ...

</style>
</head>
```

გთხოვთ ყურადღება მიაქციოთ, რომ ეს თაგი კონტეინერია ანუ მოითხოვს შესაბამის დახურვას. ასეთი გზით ერთსადაიმავე დოკუმენტში სტილის ჩაშენება შეგვიძლია უთვალავი რაოდენობით, საჭიროების და მიხედვით.

აქვე დავაზუსტოდ, რატომ მაინცდამაინც დოკუმენტის თავში ვაშენებთ სტილს? როგორც წესი დოკუმენტის ბრაუზერში ვიზუალური ასახვა იწყება ბრაუზერის მიერ თაგი <BODY>-ის წაკითხვის შემდეგ, ანუ რადგან სტილი პასუხს აგებს დოკომენტის ვიზუალურ ნაწილზე უმჯობესია ბრაუზერმა ჯერ წაიკითხოს ის CSS ატრიბუტები და მნიშვლებობები რომელიც აფორმირებს ჩვენს ვიზუალურ ნაწილს და მხოლოდ ამის შემდგომ დაიწყოს დოკუმენტის ბრაუზერში ასახვა.

ხაზოვანისტილი- Inline style

ხაზოვანი სტილი გამოყენების სახელდახელო მეთოდს წარმოადგენს CSS ატრიბუტების თაგში ჩაშენება. სტილის მსგავსი მიდგომით გამოყენება საკმაოდ მოუქნელია და მხოლოდ იშვიათ შემთხვევებში გამოიყენება. მისი გამოყენების სინტაქსი შემდეგნაირია:

```
<img src="image.jpg" alt=" " style="width:300px;height:200px;" />
```

აღნიშნული მაგალითი გამოიყენება კონკრეტულად ერთი თაგისთვის, ამ შემთხვევაში გრაფიკული გამოსახულებისთვის რომელსაც CSS ატრიბუტებით ვანიჭებთ სიმაღლეს (200px) და სიგანეს (300px) პიქსელებში.

CSS ის წერის სინტაქსი

როგორც მარკირების და პროგრამირების ყველა ენას გააჩნია თავისებური წერის სინტაქსი, ასევე გახლავთ CSS ენაც. საკმარისია უმნიშვნელოს სინტაქსი დაირღვეს და ჩათვალეთ, რომ ვეღარ მივიღებთ იმ ეფექტს რაც ჩანაფიქრში გვქონდა. ანუ სინტაქსის ძირითადი პრინციპი შემდეგნაირია: ვირჩევთ სელექტორს (ამაზე მოგვიანებით გვექნება საუბარი), შემდეგ ვირჩევთ კონკრეტულ მის ატრიბუტს (პარამეტრს)და ვანიჭებთ ნებისმიერ დასაშვებ მნიშვლობას.

```
სელექტორის სახელი { თვისება1: მნიშვნელობა; თვისება2: მნიშვნელობა; }
```

მაგალითად: გრაფიკული გამოსახულება გვინდა რომ იყოს 200px სიგანის და 300px სიმაღლის

```
img { height: 300px;  width: 200px; }
```

ზემოხსენებულ მაგალითში img გახლავთ სელექტორი, ანუ კონკრეტულად თაგი რომელსაც გამოსახულება შემოაქვს დოკუმენტზე, ფიგურულ ფრჩხილებში კი ვირჩევთ მის კონრეტულ პარამეტრებს

და ვანიჭებთ მნიშვნელობებს, ანუ მაგალითად პარამეტრი height-ის არჩევის შემდგომ ვწერთ ორწერტილს, რაც მინიჭებას ნიშნავს და ამის შემდგომ ამ პარამეტრს ვანიჭებთ კონკრეტულ მნიშვნელბას ამ შემთხვევაში 300px-ს და ვუკონკრეტებთ რომ საზომი ერთეული არის px ანუ (Pictre Single Element) ყველაზე უმცირესი წერტილი მონიტორის რეზოლუციაზე (pixel). პირველ პარამეტრზე მნიშვნელობის მინიჭების შემდეგ იწერება წერტილმძიმე და შესაძლებელია სხვა პარამეტრის გამოყენება.

აქვე გავაკეთოთ პატარა მაგრამ მნიშვნელოვანი მინიშნება, როგორც ყველა კოდირების სინტაქსი ითვალისწინებს უშუალოდ კოდში კომენტარის ჩანაწერის წარმოებას, ასევე CSS იც არ გახლავთ გამონაკლისი. CSS ის კოდში კომენტარი იწერება შემდეგნაირად

/* ეს გახლავთ CSS კომენტარი */

CSS კოდის წერის ეთიკა

ზემოთმოყვანილი მაგალითიდან ჩანს თუ როგორ შეგვიძლია CSS სინტაქსის გამოყენება HTML დოკუმენტში, თუმცა აქვე უნდა აღინიშნოს, ვრცელი პროექტების ფორმირების დროს ძალიან მნიშვნელოვანია დავიცვათ გარკვეული უკვე კარგად აპრობირებული წერის მანერა და ეთიკა, ვინაიდან ვრცელი პროგრამული კოდების გარჩევის დროს მარტივი იყოს მათი თვალისთვის და გონებისთვის აღქმა. ზემოთმოყვანილი მაგალითი დამეთანხმებით თვალისთვის უფრო გარჩევადი იქნებოდა ქვევით მოცემული ფორმით რომ ჩაგვეწერა. ანუ სელექტორი და ყოველი მომდევნო პარამეტრი ახალი სტრიქონიდან ტაბულაციით მარჯვნივ შეწეული რომ დავწეროთ.

```
img                                                    {
        height: 300px;
        width: 200px;
        }
```

სელექტორები და მათი ზოგადი გამოყენების ცნებები

ტერმინი „სელექტორი“ CSS-თან მიმართებაში გამოიყენება დოკუმენტში კონკრეტული ელემენტის პარამეტრების (ატრიბუტების) მნიშვნელობის სამართავად. დოკუმენტში კონკრეტულ ელემენტს შეგიძლია მივმართოთ სხვადასხვა მიმართვის წესის გამოყენებით. ესენია:

1. მივმართოთელემენტსთაგისსახელით
2. მივმართოთელემენტსID-ისმეშვეობით
3. მივმართოთელემენტსწინასწარგამოცხადებულიკლასისმეშვობით

განვიხილოთზემოთჩამოთვლილისელექტორებისსპეციპიკაციადამათიგამოყენებისასპეკტები.

ელემენტზეთაგისსახელითმიმართვამსგავსიმიმართვაგამოიყენებაკასკადურადდოკუმენტშიკონკრეტულითაგისმიმართინსტრუქციისგადასაცემად.

მაგ:

```
p {
    font-size: 15px;
}
```

აღნიშნული კოდის ნაწილი მიუთითებს, რომ დოკუმენტში ყველა თაგი სახელად უნდა იყოს 15 პიქსელის ზომის ფონტით.

კონკრეტულ ელემენტს მისი ID -ის მეშვეობით მიმართვა. ID უნიკალური უნდა იყოს დოკუმენტში (ანუ არ უნდა მეორდებოდეს).

```
#paragraph {
    font-size: 15px;
}
```

ზემოთმოცემული კოდიდან ჩანს რომ id სელექტრზე მიმართვისათვის ვიყენებთ # დიეზის სიმბოლოს. მოცემული მითითება აღნიშნავს, რომ მხოლოდ ის ელემენტი დოკუმენტში რომელსაც გააჩნია id="paragraph" აისახება 15 პირქსელის ზომის ფონტით.

მაგ. Hello World!

დოკუმენტის ელემენტზე კლასის მეშვეობით მიმართვა. წინასწარ გამოცხადებული გარკვეული პარამეტრები შეგვიძლია გამოვიძახოთ ნებისმიერ დოკუმენტის ელემენტთან

```
.myStyle {
    font-size: 15px;
}
```

ზემოთმოცემული კოდიდან ჩანს რომ კლასსზე მიმართვისათვის ვიყენებთ . წერტილს. აღნიშნული მითითება უზრუნველყოფს HTML ატრიბუტით CLASS გამოძახებული კლასის გავრცელებას ამ ელემენტზე.

მაგ. Hello World!

2.2 ვებსაიტისტექსტურიელემენტებისგაფორმება

მიმდინარეპარაგრაფისთემატიკა

- ზომისერთეულები და მათი გამოყენების ასპექტები
- ვებგვერდისფონისპარამეტრებიდამნიშვნელობები
- შრიფტისთვისებებიდამათიმნიშვნელობები
- ვებფონტებისშესაძლებლობები
- ვებფონტებისგენერირებისსაშუალებები
- ვებფონტებისასახვისშესაძლებლობები
- ტექსტისთვისებებიდამნიშვნელობები
- ბმულებისთვისებებიდაგამოყენებისშესაძლებლობები

ზომის ერთეულები და მათი გამოყენების ასპექტები

WEB გვერდის შესაქმნელად, მასზე ობიექტების განთავსება და ზომების მითითება შესაძლებელია სხვადასხვა საზომი ერთეულებით:

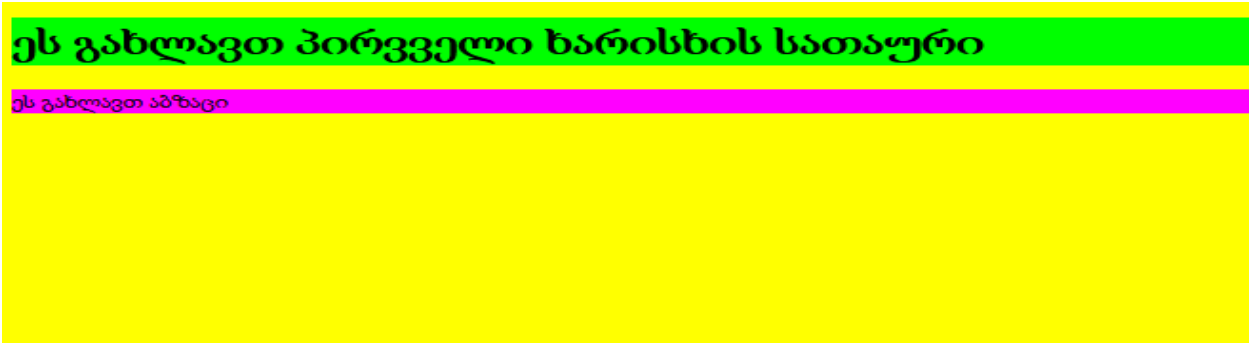
cm	სანტიმეტრი
mm	მილიმეტრი
In	ინჩი (1ინჩი = 96პიქსელი (px) = 2.54 სანტიმეტრი
px	პიქსელი (1 პიქსელი = 1/96 ედი 1ინჩი)
pt	პოინტი (1პოინტი =1/72-ი 1 ინჩი)

თუმცაპრაქტიკამგვაჩვენა, რომყველაზემოსახერხებელი, ზემოთჩამოთვლილისაზომიერთეულებიდანპიქსელებია (PX). ვინაიდანყველამოწყობილობისმონიტორისრეზოლუციაზომებსპიქსელებში.

ვებგვერდის ფონის პარამეტრები და მნიშვნელობები

background-color: ფერი – განსაზღვრავსფონისფერსელემენტისათის

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-color: yellow;
}

h1
{
    background-color: #00ff00;
}

p
{
    background-color: rgb(255,0,255);
}
</style>
</head>
<body>

<h1>ეს გახლავთ პირველი ხარისხის სათაური</h1>
<p>ეს გახლავთ აბზაცი</p>

</body>
</html>
```



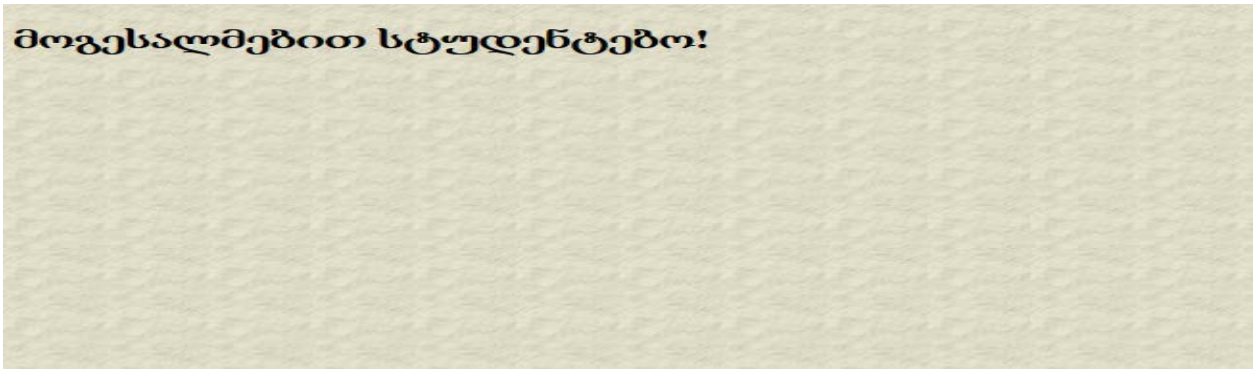
სურ. 3.2. სხვადასხვა ფონის ელემენტები

ზემოთმოყვანილ მაგალითში გამოყენებულია სწორედ ის სელექტორები რომელზეც წინა ქვეთავში გვქონდა საუბარი. დოკუმენტს აშკარა მითითებით გადავცემთ დოკუმენტის ტანის ფერს (yellow - ყვითელი) ანუ გლობალურად ფონის ფერს, პირველი დონის სათაურის ფონის ფერს (მწვანე – 00ff00) და აბზაცის ფონის ფერს (ვარდისფერს - rgb(255,0,255)). ზემოთ აღწერილი მაგალითიდან გამომდინარე, რამდნი აბზაციც გვექნება დოკუმენტში და რამდენი პირველი დონის სათაურიც <h1> ყველა მათგანი გაიზიარებს ზემოთაღწერილ სტილს.

background-image: url("ფაილის სახელი") - უზრუნველყოფს გრაფიკული ფონის შექმნას კონკრეტული ელემენტისტისთვის.

```
<!DOCTYPE html>
<html>
<head>
<style>
body
 {
    background-image: url("paper.gif");
 }
</style>
</head>
<body>

<h1>მოგესალმებით სტუდენტებო!</h1>

</body>
</html>
```

სურ. 3.3. ფონად გამოყენებული გრაფიკული გამოსახულება



სურ. 3.4. ვებგვერდის გრაფიკული ფონი

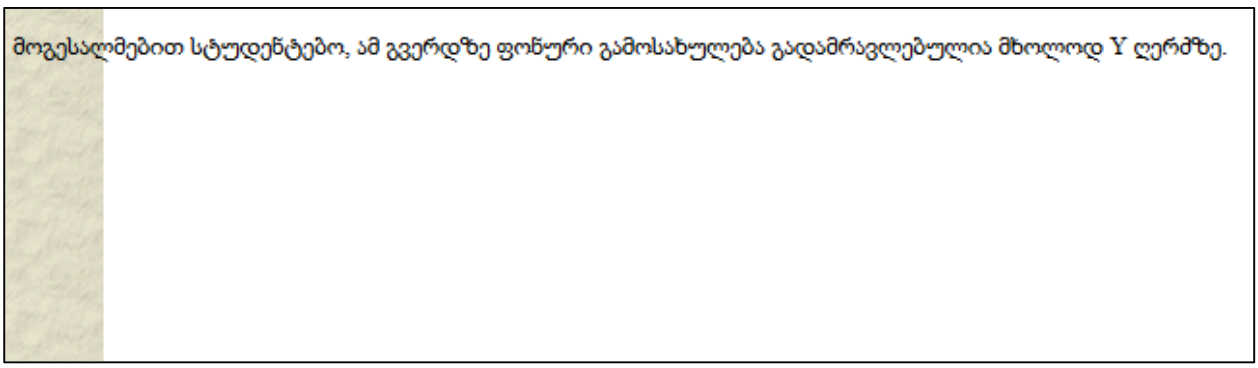
ზემოთ მოყვანილ მაგალითს თუ დააკვირდებით, ნახავთ რომ დოკუმენტის ფონზე შემოტანილი არის ერთი გრაფიკული გამოსახულება (სურ. 3.3), რომელიც მოზაიკის წესით არის გადამრავლებული (სურ. 3.4). ე.ი. როდესაც ფონად ვუთითებთ დოკუმენტის რომელიმე ელემენტს გრაფიკულ გამოსახულებას ის ავტომატურად მრავლდება X და Y ღერძზე უსასრულოდ, თუმცა ქვემოთ მოყვანილ მაგალითებში ვიხილავთ, რომ CSS ის მეშვეობით ამ ქმედების კონტროლიც შეგვიძლია

შენიშვნა: თუ ფონური გამოსახულების ფაილი და html დოკუმენტი არ მდებარეობს ერთ საქაღალდეში, ფაილზე მიმართვისათვის საჭიროა მიეთითოს სრული მისამართი.

background-repeat - განსაზღვრავსფონური გამოსახულებისგანმეორებადობას

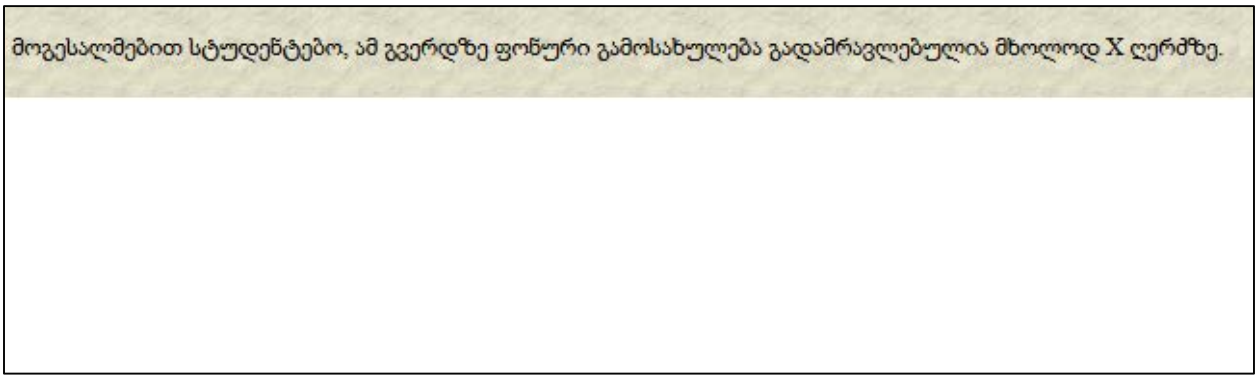
- background-repeat: repeat-y - ფონის გამოსახულების y ღერძზე განმეორადობა;
- background-repeat: repeat-x - ფონის გამოსახულების x ღერძზე განმეორადობა;
- background-repeat: no-repeat - ფონის გამოსახულება განმეორდება.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("paper.gif");
    background-repeat: repeat-y;
}
</style>
</head>
<body>

<p>მოგესალმებით სტუდენტებო, ამ გვერდზე ფონური გამოსახულება გადამრავლებულია მხოლოდ Y
ღერძზე.</p>

</body>
</html>
```



სურ. 3.5. ფონის გამოსახულების y ღერძზე განმეორადობა

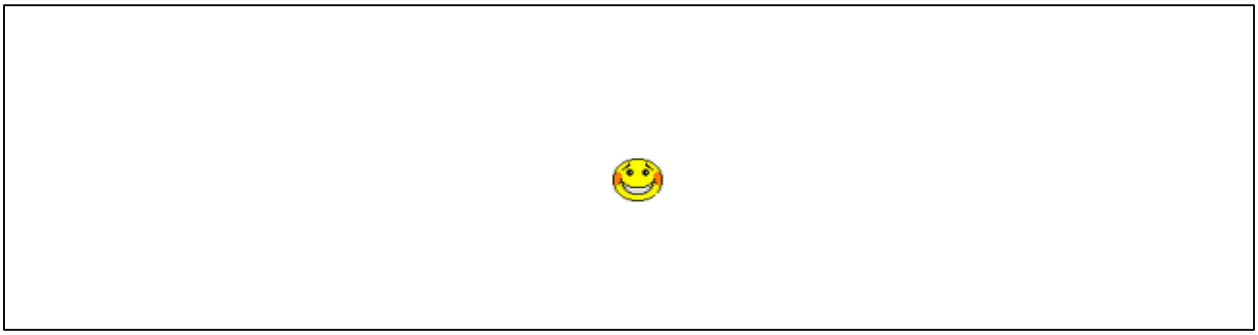
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("paper.gif");
    background-repeat: repeat-x;
}
</style>
</head>
<body>

<p>მოგესალმებით სტუდენტებო, ამ გვერდზე ფონური გამოსახულება გადამრავლებულია მხოლოდ X ღერძზე.</p>

</body>
</html>
```



სურ. 3.6. ფონის გამოსახულების x ღერძზე განმეორადობა

background-position - განსაზღვრავს ფონის სურათის პოზიციის დაწყებას

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url('smiley.gif');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
}
</style>
</head>

<body>
</body>

</html>
```



სურ. 3.7. ფონის გამოსახულების პოზიციის განსაზღვრა

შესაძლებელია გამოვიყენოთ ფონის განსაზღვრის გაერთიანებული ვერსია, სადაც მნიშვნელობები მიეთითება შემდეგი თანმიმდევრობით:

1. ფონის ფერი
2. გრაფიკული გამოსახულება
3. გამოსახულების განმეორებადობა
4. გამოსახულების ასახვის პოზიცია

```
body {
      background: #ffffff url("img_tree.png") no-repeat right top;
  }
```

შენიშვნა: თუ რომელიმე მნიშვნელობა არ არის მითითებული, ბრაუზერი გამოიყენებს ამ თვისების ავტომატურ მნიშვნელობას.

მაგალითისათვის განვიხილოთ სურ. სურ. 3.8-ზე მოცემული ვებგვერდის გაფორმება ფონური ელემენტებით.

ვებგვერდი დაყოვილია ოთხ ძირითად ნაწილად: header - თავი, menu- მენიუ, main - ძირიტადი ნაწილი, footer - ბოლო. შესაბამისი ბლოკური ელემენტები უნდა განისაზღვროს html დოკუმენტში და აღიწეროს ცალკეულის ფონური პარამეტრები. სურ. 3.9- ზე მოცემულია შესაბამისი გრაფიკული გამოსახულებები.

სურ. 3.8. ვებგვერდის გაფორმება ფონური ელემენტებით

menubg.png

logo.png

mainbg.png

სურ. 3.9. მოცემული გრაფიკული გამოსახულებები

```
<!DOCTYPE html>
<html>
<head>
<title>ფონები</title>
<meta charset="UTF-8">
<style>
#header{
    background:#ffffff;
    }
#menu{
    background:url(menubg.png); repeat-x;
    height:35px;
    }
#main{
    background:#e9eff1 url(mainbg.png) no-repeat right bottom;
    height:400px;
    }
#footer{
    background:#707070;
    height:30px;
    }
</style>
</head>
<body>

<div id="header">
    <img src="logo.png" alt=""/>
</div>

<div id="menu"> </div>

<div id="main"> </div>

<div id="footer"> </div>

</body>
</html>
```

სურ. 3.10. ფონის თვისებების გამოყენების ნიმუში

შრიფტთან მუშაობა ამთავითვე დავიხსომოთ: მომხმარებლის ეკრანზე აისახება მხოლოდ ის შრიფტები, რომლებიც დაყენებულია მის კომპიუტერზე, შესაბამისად, ჯობია, "ეგზოტიკური" შრიფტების გამოყენებისაგან თავი შევიკავოთ ან უკიდურეს შემთხვევაში, თუ მიგვაჩნია, რომ ეს აუცილებელია, ასეთი შრიფტებით გაკეთებული ჩანაწერები გრაფიკულ ფორმატში გადავიყვანოთ (შევნიშნავთ, რომ ნათქვამი არ ეხება უნიკოდს).

ქვემოთ ჩამოთვლილია სტილების შემნახველ კასკადურ ცხრილებში გამოყენებული ის ატრიბუტები, რომლებიც ემსახურება შრიფტის მართვას:

- **font-family** – შრიფტის სახეს განსაზღვრავს.
- **Font-style** – სტილის აღმნიშვნელი
- **font-size** – განსაზღვრავს შრიფტის ზომას (აბსოლუტური ან ფარდობითი სახით ზემოთ ჩამოთვლილი საზომი ერთეულების გამოყენებით)

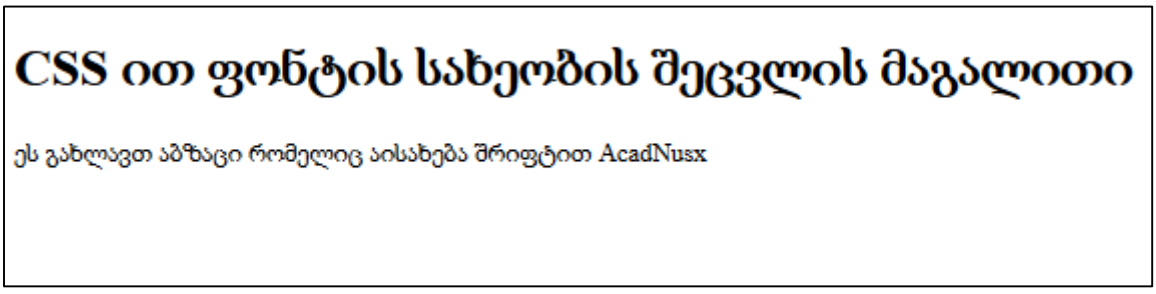
მაგალითად: font-size: 12px, font-size: 100%, font-size: 2.5em;

განვიხილოთ თითოეული ზემოთ ჩამოთვლილი შრიფტთან სამუშაოდ ჩამოთვლილი ატრიბუტი.

font-family- როგორც უკვე ავღნიშნეთ მოცემული ატრიბუტით შეგვიძლია განვსაზღვროთ თუ რომელი შრიფტი გამოიყენოს დოკუმენტმა ტექსტის ასახვის დროს. ვინაიდან ზემოთ უკვე გავუსვით ხაზი იმ გარემოებას, რომ მომხმარებელი შრიფტს დაინახავს მხოლოდ იმ შემთხვევაში თუ ეს შრიფტი მის კომპიუტერში უკვე არსებოს, ვცდილობთ დოკუმენტის ტექსტის გადმოსაცემად შევარჩიოთ სტანდარტული ტიპის შრიფტები, ანუ საუბარი არის იმ ტიპის შრიფტებზე, რომლებიც ავტომატურად მოყვება ჩვენთვის ცნობილ და პოპულარულ ოპერაციულ სისტემებს.

მაგალითად: შემდეგი კოდი გვაჩვენებს თუ როგორ უნდა ავსახოთ ტექსტი AcadNusx-ურის ტიპის შრიფტით ანუ ყველასათვის ნაცნობი ქართული შრიფტით.

```
<!DOCTYPE html>
<html>
<head>
<style>
P
{
   font-family: "AcadNusx";
}
</style>
</head>
<body>
<h1>CSS ით ფონტის სახეობის შეცვლის მაგალითი</h1>
<p >ეს გახლავთ აბზაცი რომელიც აისახე შრიფთით AcadNusx</p>
</body>
</html>
```



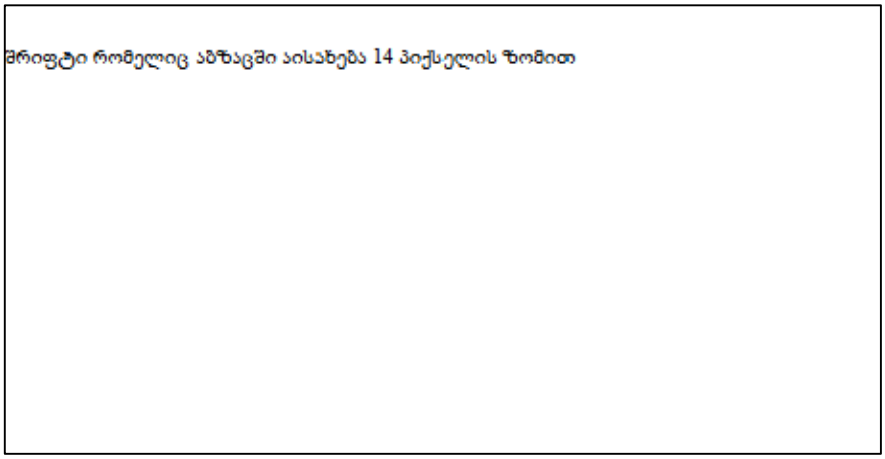
აქვე გვინდა პატარა მინიშნება გავუკეთოთ იმ გარემოებას, რომ შრიფტის სახელი ზუსტად ისე უნდა იყოს მითითებული ამ ატრიბუტის მნიშვნელობაში რეგისტრის მიხედვით როგორც შრიფტს სინამდველში ჰქვია.

font style – სტილის აღმნიშვნელი, რაცშეეხებამოცემულატრიბუტსმანშეიძლებამიიღოსშემდეგიმნიშვნელობები:

- normal;
- italic;
- oblique;
- initial;
- inherit;

font-size – განსაზღვრავს შრიფტის ზომას ჩვენთვის სასურველი საზომი ერთეულით

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    font-size: 14px;
}
</style>

</head>
<body>

<h1>CSS font-family</h1>
<p>შრიფტი რომელიც აბზაცში აისახება 14 პიქსელის ზომით</p>


</body>
</html>
```



ვებ ფონტების შესაძლებლობები

ვებ ფონტები ჩვეულებრივი სტანდარტული ფონტებისაგან (შრიფტებისაგან) განსხვავებით ვებ გვგერდებზე გამოიყენება არასტანდარტული შრიფტებით ტექსტის ასახვის მიზნით. ანუ თუ ჩვენ გვინდა ჩვენს მიერ შექმნილ ვებ პროექტში გამოვიყენოთ რაიმე განსხვავებული შრიფტი, რომელიც რეალურად შესაძლებელია, რომ მომხმარებელს არ ჰქონდეს კომპიუტერზე დაინსტალირებული მაშინ დასახმარებლად მივმართავთ სწორედ ვებ ფონტებს. ვებ ფონტების მთელი ხიბლი და მისი გამოყენების მოხერხებულობა იმაში მდგომარებოს, რომ იგი იტვირთება ბრაუზერში ვებ გვერდთან ერთად (დინამიურად) ასეთი მიდგომით ჩვენ და მომხმარებელიც გარანტირებულია, რომ შრიფტი ნებისმიერ შემთხვევაში ისეთი სახით აისახება ბრაუზერში როგორც ეს ჩვენ (ავტორებს) გვქონდა გათვლილი. თუმცა არსებობს ერთი ნიუანსი რომელიც ძალიან საყურადღებოა, უნდა გავითვალისწინოთ ის მნიშვნელოვანი ფაქტიც, რომ ვებ ფონტებს ყველა ბრაუზერი ერთნაირად არ აღიქვამს და ამ დეტალმა შეიძლება გაუთვალისწინებელი პრობლემა შეგვიქმნას ბრაუზერების გარკვეულ ვერსიებთან. მოდით აქვე დავაზუსტოდ, ვებ ფონტებს გააჩნია 4 ფორმატი, სწორედ ეს იწვევს ბრაუზერებთან მიმართებაში ვებ ფონტების მოქნილად გამოყენების საშუალებას ვინაიდან პრობლემის აღმოფხვრაში სწორედ ოთხივე ფორმატის ფონტის შემოტანა გვიწყობს ხელს.

ეს ფორმატებია:

- TTF
- WOFF
- EOT
- SVG

ანუ იმისათვის რომ ვუზრუნველყოთ ჩვენი ვებ პროექტი ვებ ფონტების სწორად ასახვისთვის, ამისათვის საჭირო გახლავთ ჯერ დავაგენერიროთ ეს შრიფტები სპეციალური ინსტრუქციის მიხედვით და შემდგომ განვათავსოთ იგი სერვერზე ვინაიდან მოგვეცეს საშუალება საჭიროების და მიხედვით მივაკითხოთ მის URL მისამართს და ვუზრუნველყოთ ვებ გვერდთან სინქრონულად მათი ჩატვირთვა მომხმარებლის კომპიუტერში.

ვებ ფონტების გენერირების საშუალებები

რაც შეეხება ვებ ფონტის (შრიფტის) გენერირების პროცესს, ეს საკმაოდ მარტივია, ძალიან ბევრი ონლაინ რესურსის მოპოვება შეიძლება მსგავსი ოპერაციის შესასრულებლად. პროცედურა დაახლოებით ყველგან ერთმანეთის ანალოგიურია, თქვენ, თქვენს კომპიუტერში უკვე არსებული შრიფტი უნდა ატვირთოთ რომელიმე შრიფტ-გენერატორ საიტზე და ის თქვენს შრიფტს დაგიბრუნებთ უკვე ვებფონტის ფორმატში. საცდელად შეგიძლიათ გამოიყენოთ ეს რესურსი: https://www.web-font-generator.com მოცემულ ვებ რესურსზე შესვლისთანავე მიყევით ძალიან მარტივ ინსტრუქციას.

ვებ ფონტების ასახვის შესაძლებლობები

რაც შეეხება ვებ ფონტების ბრაუზერში დოკუმენტზე ასახვს, ამისათვის არსებობს CSS ის ენაში სპეციალურად განკუთვნილი დეკლარაცია- ეს გახლავთ შემდეგი სახის ჩანაწერი @font-face, რომელიც უზრუნველყოფს დინამიურად ვებ ფონტების ჩატვირთვას ვებ გვერდთან ერთად. მისი სრული გამოყენების სინტაქსი და წესი შემდეგნაირია:

```
<!DOCTYPE html>
<html>
<head>
<style>
@font-face {
    font-family: myFirstFont;
    src: url(web_font_file.woff);
}

div {
    font-family: myFirstFont;
}
</style>
</head>
<body>

<div>დინამიური ვებ ფონტი.</div>

<p><b>შენიშვნა:</b> ინტერნეტ ექსპლორერი 8 და უფრო ადრეული ვერსიები არ აღიქვამს WOFF ფორმატის
ვებ ფონტებს. ის აღიქვამს მხოლოდ EOT ფორმატის ვებ ფონტებს.</p>

</body>
</html>
```

დინამიური ვებ ფონტი.

შენიშვნა: ინტერნეტ ექსპლორერი 8 და უფრო ადრეული ვერსიები არ აღიქვამს WOFF ფორმატის ვებ ფონტებს. ის აღიქვამს მხოლოდ EOT ფორმატის ვებ ფონტებს.

ვებ ფონტების შესაძლებლობები

მოგეხსენებათ ნებისმიერ ვებ გვერდზე ყველაზე მეტად ინფორმაციის გადმოცემა აქტუალურია ტექსტური სახით. ამიტომ CSS ში ტექსტის დამუშავების და მათი ფორმატირების მდიდარი საშუალებეი გახლავთ ჩადებული. ჩვენ ახლა თითოეულ მათგანს ჩამვთვლით და ავღწერთ.

თვისება	მნიშვნელობა	აღწერა
color	თექვსმეტობითი ათვლის სისტემით მინიჭებული მნიშვნელობა, RGB(Red	აღნიშნული თვისების მნიშვნელობით შეგვიძლია

	Green Blue) მნიშვნელობა და ფერის სიტყვიერი წარმოდგენა	შევცვალოთ ტექსტის ფერი კონკრეტული ელემენტისთვის
text-align	Left, right, center, justify	მოახდენს ტექსტის ჰორიზონტალურ ხაზზე გასწორების უზრუნველყოფას, მარცხნივ, მარჯვნივ, ცენტრში ან ჩაასწორებს ტექსტს ორივე კიდეზე მარცხნიდან და მაჯვნიდან
Text-decoration	Underline, overline, line-through, none	ტექსტს გამოიტანს ქვევიდან ხაზგასმულს, ან ზემოდან ხაზგასმულს , ან გადახაზახულს და ან საერთოდ მოხსნის დეკორაციას. ბოლო მნიშვნელობა ძალიან ხშირად გამოიყენება ბმულებთან რათა ბმული არ აისახოს ქვემოდან ხაზგასმული
Text-transform	Capitalize, uppaercase, lowercase	ახდენს ტექსტის ტრანსფორმაციას, პირველი მნიშვნელობა ანიჭებს წინადადებაში ტექსტის ყველა სიტყვის პირველ სიმბოლოს მაღალ რეგისტრს, მეორეს აბსოლიტურად ტექსტის ყველა სიმბოლო გადაყავს მაღალ რეგისტრში, უკანასკნელი კი ყველა სიმბოლოს გადაიყვანს დაბალ რეგისტრში
Text-indent	მნიშვნელობა საზომ ერთეულებში	გამოიყენება ბლოკის დონის ტაგებთან და განსაზღვრავს ტაგის ტექსტის პირველი ხაზის აბზაცად გამოყოფას იმ ინტერვალით რასაც მივუთითებთ
Letter-spacing	მნიშვნელობა საზომ ერთეულებში	განსაზღვრავს სიმბოლოებს შორის დაშორებას მითითებული ინტერვალით

Line-height	მნიშვნელობა საზომ ერთეულებში	განსაზღვრავს ტექსტის ხაზებს შორის დაშორებას მითითებული ინტერვალით
Direction	Ltr, rtl	მიუთითებს ტექსტის მიმართულებას(მარცხნიდან მარჯვნივ ან მარჯვნიდან მარცხნივ)
Word-spacing	მნიშვნელობა საზომ ერთეულებში	განსაზღვრავს სიტყვებს შორის დაშორიებას მითითებული ინტერვალით

ბმულების თვისებები და გამოყენების შესაძლებლობები

როგორც HTML-ის ენის სპეციფიკაციიდან მოგეხსენებათ, ბმული ბრაუზერისთვის იყოფა კატეგორიებად, ასე ვთქვათ:

- ბმული რომელიც ჯერ მომხმარებელს არ უნახავს -LINK
- ბმული რომელიც გააქტიურებულია -ACTIVE LINK
- ბმული რომელიც უკვე მომხმარებებლმა დაათვალიერა ანუ მოახდინა ამ ბმულით გარკვეულ URL მისამართზე მიმართვა VISITED LINK
- ბმული რომელზეც მაუსის კურსორს მივიტანთ -HOVER (იმართება მხოლოდ CSS-ის საშუალებით)

სინამდვილეში ამ ბმულების ისტორიას ინახავენ ბრაუზერები, რომლებიც შესაბამისად ბმულის სტატუსიდან გამომდინარე ასახავენ მათ სხვადასხვა ფერით. აგრეთვე ბმულის გამოყოფა დოკუმენტზე ბრაზუერების მიერ ხდება მათი ქვემოდან ხაზგასმული სტილით(UNDERLINE).

უმეტეს შემთხვევაში გამომდინარე მოსაზრებიდან, რომ ჩვენი პროექტის დიზაინი მოითხოვს მეტ დახვეწილობას და ვიზუალურად ჩვენებურად წარმოჩენას, CSS -ი გვაძლევს საშუალებას უფრო მოქნილად ვმართოდ ამ ბმულების ვიზუალური თვისებები.

ბმულის თვისებების შეცვლა ხდება გლობალურად ყველა კატეგორიაზე ერთად

```
a
{
  color: #ff0000;
}
```

ვინაიდან უკვე ავღნიშნეთ, რომ ბმული ბრაუზერისთვის არსებობს კატეგორიზირებული სახით, აქედან გამომდინარე თითოეულ მათგანს რომ მივმართოთ საჭიროა გამოვიყენოთ ქვესელექტორები.

მაგალითად, თუ ჩვენ ვაპირებთ გარკვეული თვისებები შევუცვალოთ მხოლოდ მომხმარებლის მიერ უკვე ნანახ ბმულს(VISITED LINK), მაშინ მასთან მიმართვის სინტაქსი შემდეგნაირი იქნება:

```
a:visited
{
  color: #ff0000;
}
```

მოცემული კოდი უთითებს ბრაუზერს, რომ ლინქი რომელიც ვიზიტორს უკვე რეალიზებული აქვს ანუ ნანახი მისთვის ბრაუზერი ამ ბმულის ფერს ასახავს წითელი შრიფტის სახით.

HTML ენისაგან განსხვავებით CSS ის საშუალებით შეგვიძლია აგრეთვე ვმართოთ ბმული მაუსის კურსორის მიტანის მომენტში, ანუ ვცვალოთ მისი თვისებები. მაგალითად

```
a:hover
{
  color: #ff0000;
 background-color:#00ff00;
 font-size:20px;
}
```

მოყვანილი მაგალითის თანახმად, ბმულზე, რომელზეც მომხმარებელი მიიტანს მაუსის კურსორს, აღნიშნული ბმულის ტექსტის ფერი გახდება წითელი, ფონის ფერი გაუხდება მწვანე და შიფტის ზომა აისახება 20 პიქსელის ზომით.

ახლა კი ვნახოთ სრული მაგალითი თუ როგორ მივაკითხოთ შესაბამისი ქვესელექტორების მეშვეობით ყველა კატეგორიის ბმულს

```
a:link /* ბმული რომელზეც ჯერ არ დაუწკაპუნებია მომხმარებელს  */
{
  color: #ff0000;
}
a:active /* ბმული რომელიც აქტიურია */
{
  color: #ff0000;
}
a:visited /* ბმული რომელიც უკვე დაათვალიერა მომხმარებელმა და ამ ისტორიას ინახავს ბრაუზერი  */
{
  color: #ff0000;
}



a:hover /* ბმული რომელზეც მაუსის მიტანისას უნდა შეიცვალოს მისი თვისება  */
{
  color: #ff0000;
}
```

მიმდინარეპარაგრაფისთემატიკა

- ხაზოვანიდაბლოკურიელემენტებისგამოყენებისთავისებურებები
- ხაზოვანიდაბლოკურიელემენტებისთვისებები
- სიისთვისებებიდამნიშვნელობები

ხაზოვანი და ბლოკური ელემენტების გამოყენების თავისებურებები

მოგეხსენებათ HTML დოკუმენტის სტრუქტურაში ერთმანეთისგან მკაცრად არიან გამიჯნულები ხაზის დონის ელემენტები(ხაზოვანი INLINE ELEMENTS) და ბლოკის დონის ელემენტები (ბლოკური BLOCK LEVEL ELEMNTS). ამ ორი ჯგუფის ტიპის ტაგებს გააჩნიათ საკუთარი თვისებები. მოდით აქვე ავღნიშნოთ ამ თუ რა პრინციპული განსხვავება არის ბლოკურ და ხაზოვან ელემენტებს შორის.

ხაზოვანი ელემენტი (INLINE ELEMENT) დოკუმენტზე აისახება როგორც მიმდინარე ხაზის ელემენტი, ანუ რამოდენიმე ხაზოვანი ელემენტის გამოყენების შემთხვევაში ჩვენ მივიღებთ მიმდინარე ხაზში ჩამწკრივებული ელემენტების კასკადს. მაგალითად:

```
<!DOCTYPE html>
<html>
<head>
  <title>დოკუმენტის ტიტული</title>
</head>

<body>
 <b>ხაზოვანი ელემენტი მსხვილი შრიფტით</b>
 <i>ხაზოვანი ელემენტი დახრილი შრიფტით</i>
 <u>ხაზოვანი ელემენტი ხაზგასმული შრიფტით</u>
</body>
</html>
```

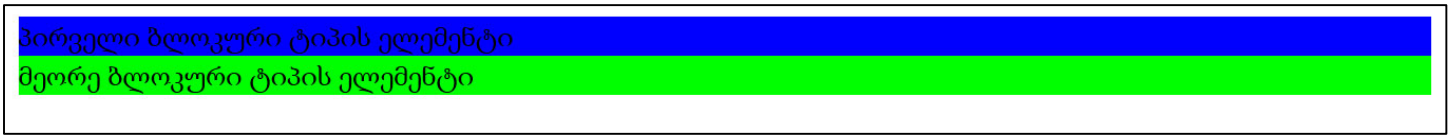
მოცემული სამივე ელემენტი აისახება ერთმანეთის გვერდიგვერდ ერთ ჰორიზონტალურ სივრცეში (ხაზში). სწორედ ამიტომაც მსგავს ელემენტებს უწოდეს ხაზოვანი ტიპის ელემენტები.

ხაზოვანი ელემენტი მსხვილი შრიფტით *ხაზოვანი ელემენტი დახრილი შრიფტით* ხაზოვანი ელემენტი ხაზგასმული შრიფტით

რაც შეეხება ბლოკური ტიპის ელემენტებს (BLOCK LEVEL ELEMENTS) მათ პრინციპულად სხვა თვისება გააჩნიათ, ანუ თუ ჩვენი სურვილის მიხედვით არ მივუთითეთ მათ გარკვეული CSS თვისებები და მათი მნიშვნელობები ავტომატურად ისინი განლაგდებიან დოკუმენტზე რათქმაუნდა თანმიმდევრობით, მხოლოდ ერთმანეთის ქვეშ ახალ ახალი ხაზიდან. ანუ თითოეული ბლოკური დონის ელემენტი დოკუმენტზე მოიცავს დოკუმენტის სიგანის მთელ ჰორიზონტალურ სივრცეს, როგორც ბლოკს. ამიტომაც ეწოდებათ მათ ბლოკური დონის ელემენტები. როგორც წესი ბლოკური დონის

ელემენტებს ხაზოვანი ელემენტებისგან განსხვავებით ვიყენებთ ხოლმე დოკუმენტის სტრუქტურულად აგებისთვის ვინაიდან ჩვენი სურვილის მიხედვით სწორად გავანაწილოთ დოკუმენტზე ელემენტები და მოქნილად ვმართოთ მათი განლაგება ურთიერთ მიმართ.

```
<!DOCTYPE html>
<html>
<head>
  <title>დოკუმენტის ტიტული</title>
</head>
<style type="text/css">
 #div1
  {
    background-color:#0000ff;
  }
#div2
  {
    background-color:#00ff00;
  }
</style>
<body>
 <div id="div1">პირველი ბლოკური ტიპის ელემენტი</div>
 <div id="div2">მეორე ბლოკური ტიპის ელემენტი</div>

</body>
</html>
```



ხაზოვანი ტიპის ელემენტები თავის თვისებებით ვერ შეიძლება მოვაქციოთ ერთ ჯგუფში, ვინაიდან მათ ყველას თავისი სპეციფიკის მიხედვით ერთმანეთისგან განსხვავებული თვისებები გააჩნიათ. მაგალითად ხაზოვანი ტიპის ელემენტია ტაგი <img /> ანუ გრაფიკული გამოსახულება, რომლის თვისებებიც რადიკალურად განსხვავდება ასეთივე ხაზოვანი ტიპის ტაგისაგან <b>.

რაც შეეხება ბლოკური ტიპის ელემენტებს მათ გააჩნიათ საერთო თვისებები, როგორიც გახლავთ, ბლოკის სიგანე, სიმაღლე, ფონის ფერი და ასე შემდეგ. რათქმაუნდა ქვემოთ ყველა ამ თვისებას გავნიხილავთ დეტალურად.

თვისება	მნიშვნელობა	აღწერა
Background-color	თექვსმეტობითი ათვლის სისტემით მინიჭებული მნიშვნელობა, RGB(Red Green Blue) მნიშვნელობა და ფერის სიტყვიერი წარმოდგენა	აღნიშნული თვისების მნიშვნელობით შეგვიძლია შევცვალოთ ბლოკური ტიპის ელემენტისთვის ფონის ფერი
Border-color	თექვსმეტობითი ათვლის სისტემით მინიჭებული მნიშვნელობა, RGB(Red Green Blue) მნიშვნელობა და ფერის სიტყვიერი წარმოდგენა	აღნიშნული თვისების მნიშვნელობით შეგვიძლია შევცვალოთ ბლოკური ტიპის ელემენტისთვის ჩარჩოს ფერი
Border-style	None, dotted, dashed, solid, double, groove, ridge, inset, outset	აღნიშნული თვისების მნიშვნელობით შეგვიძლია შევცვალოთ ბლოკური ტიპის ელემენტისთვის ჩარჩოს სტილი, ანუ მისი სხვადასხვა სტილით წარმოდგენა
Border-size	მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ	აღნიშნული თვისების მნიშვნელობით შეგვიძლია შევცვალოთ ბლოკური ტიპის ელემენტისთვის ჩარჩოს სისქე
Width	მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ	გამოიყენება ბლოკის დონის ტაგებთან და განსაზღვრავს მის სიგანეს

Height	მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ	გამოიყენება ბლოკის დონის ტაგებთან და განსაზღვრავს მის სიმაღლეს
Position	Absolute, relative მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ (top, left, right, bottom)	გადავცემთ ბლოკის დონის ელემენტს თუ როგორ უნდა განთავსდეს დოკუმენტზე, როგორც აბსოლიტური პოზიციის მქონე ელემენტი თუ როგორც ფარდობი.
z-index	მთელი რიცხვითი მნიშვნელობა	მივუთითებთ მრავალშრიანი დოკუმენტის არსებობის შემთხვევაში თუ რომელ შრეზე განთავსდეს არსებული ბლოკის დონის ელემენტი.
Float	Left, right	გადასცემს ბლოკის დონის ელემენტს თუ რომელი მხრიდან განთავსდეს მომდევნო ელემენტის მიმართ
Overflow	Visible, hidden, scroll, auto	განსაზღვრავს ბლოკური ტიპის ელემენტის მიდამოში, როგორ აისახოს მისი შიგთავსი.
Margin	მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ და აგრეთვე შეგვიძლია მივანიჭოთ მნიშვნელობა auto რომელიც უზრუნველყოფს ბლოკის დონის ელემენტის ცენტრში გასწორებას ჰორიზონტალურ სიბრტყეზე.	განსაზღვრავ ბლოკური ტიიპის ელემენტის მინდვრებს მის გარშემო, ანუ რამდენად უნდა იყოს ის დაშორებული სხვა ელემენტებისგან
Padding	მნიშვნელობა საზომ ერთეულებში(px, cm) და ასე შემდეგ	განსაზღვრავს ბლოკური ტიპის ელემენტის შიგთავსის დაშორებას მისი კიდეებიდან
Display	None, block, inline	განსაზღვრავს თუ როგორ უნდა აისახოს ბლოკის დონის

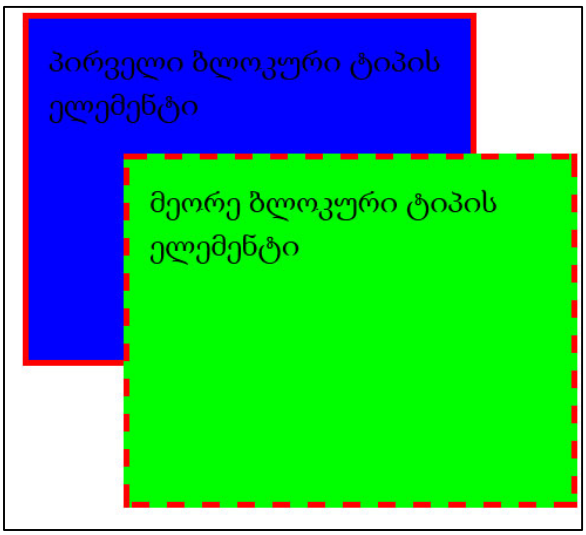
		ელემენტი, იყოს დამალული, ჩანდეს დოკუმენტზე თუ აისახოს როგორც ხაზოვანი ელემენტი.

ზემოთ მოყვანილი ცხრილიდან ჩანს თუ რა თვისებები გააჩნიათ ბლოკის დონის ელემენტებს, მიუხედავად ამისა მაინც გვინდა ყურადღება გავამახვილოთ მათ რამოდენიმე თვისებაზე რომელთა გამოყენებასაც გარკვეული ნიუანსების გათვალისწინება სჭირდება.

თვისება position: რომელსაც ენიჭება მნიშვნელობები, absolute და relative, ანუ დოკუმენტზე ყველა ელემენტი გაჩუმების პრინციპით (Default) პოზიცია იკავებს მის წინა ელემენთან ფარდობითად. თუმცა ჩვენ შეგვიძლია ბლოკის ტიპის ელემენტს მივუთითოთ, რომ ის მიუხედავად ყველაფრისა განთავსდეს კონკრეტულ აბსოლიტურ პოზიციაზე. ასეთ შემთხვევაში მას უნდა გადავცეთ კიდევ დამატებითი ინსტრუქციები, თუ რომელი პოზიცია დაიკავოს მან. ანუ ხსენებულ შემთხვევაში ჩვენ ვუკონკრეტებთ ბლოკის ტიპის ელემენტს თუ ბრაუზერის ფანჯრის რომელი კიდის მიმართ განთავსდეს გარკვეულ პოზიციაზე :top(ზედა კიდე), left(მარცხენა კიდე), right(მარჯვენა კიდე), bottom(ქვედა კიდე)

თვისება z-index: ხშირად საჭიროება მოითხოვს ხოლმე დოკუმენტზე რამოდენიმე ელემენტი განთავსებული იყოს შროვანი განლაგებით, ანუ რომელიღაც ელემენტი ან ელმენტის გარკვეული ნაწილი ამოფარებული იყოს მასზე მაღალი პრიორიტეტის მქონე ელემენტს, ან პირიქით. ასეთ შემთხვევაში თამამად შეგვიძლია მოვიშველიოთ თვისება z-index , სწორედ ეს თვისება განსაზღვრავს ელემენტების განლაგებას შრეებად.

```
<!DOCTYPE html>
<html>
<head>
  <title>დოკუმენტის ტიტული</title>
</head>
<style type="text/css">
 #div1
  {
   position:absolute;
   top:50px;
   left:70px;
   height:150px;
   width:200px;
   border-style:solid;
   border-color:#ff0000;
   border-width:3px;
   background-color:#0000ff;
   margin:20px;
   padding:10px;
   z-index:1;
  }
#div2
  {
   position:absolute;
   top:120px;
   left:120px;
   height:150px;
   width:200px;
   border-style:dashed;
   border-color:#ff0000;
   background-color:#00ff00;
   margin:20px;
   padding:10px;
   z-index:2;
  }
</style>
<body>
 <div id="div1">პირველი ბლოკური ტიპის ელემენტი</div>
 <div id="div2">მეორე ბლოკური ტიპის ელემენტი</div>

</body>
</html>
```



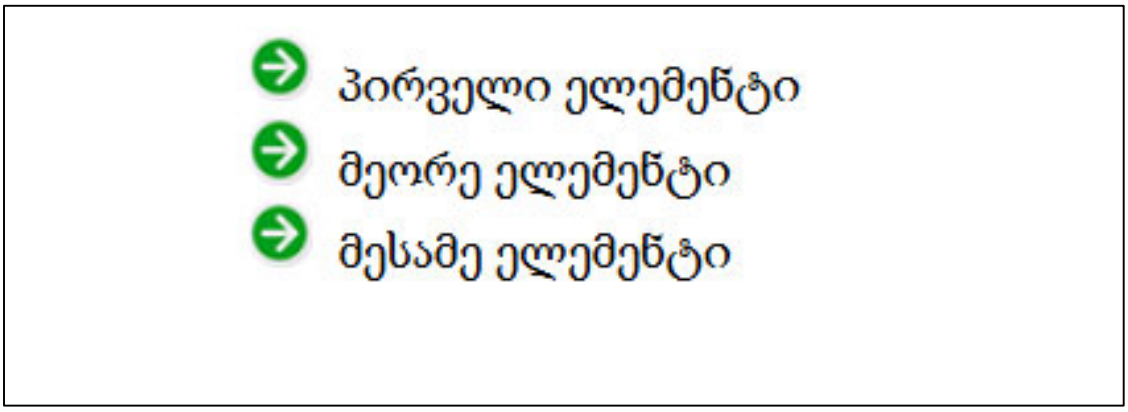
HTML-დანჩვენუკვევიცითრომარსებობსსიისსხვადასხვატიპი, მათიგაფორმებაჩვენშეგვიძლია CSS-ისსაშუალებით. რეალურად სიის გაფორმების კარგი საშუალება გვაქვს, რომ იგი გავაფორმოთ გრაფიკული მარკირებით ნაცვლად სტანდარტული მარკირების ტიპისა. გარდა ამისა CSS -ი შესაძლებლობას გვაძლევს ასევე ვმართოთ სიის სტანდარტული მარკირების ტიპები. სიის მარკირების მართვის სინტაქსი CSS-ის მეშვეობით შემდეგნაირია:

```
<!DOCTYPE html>
<html>
<head>
  <title>დოკუმენტის ტიტული</title>
</head>
<style type="text/css">

ul
{
 list-style-type: circle;
}

</style>

<body>

 <ul>
  <li>პირველი ელემენტი</li>
  <li>მეორე ელემენტი</li>
  <li>მესამე ელემენტი</li>
</ul>

</body>
</html>
```

- პირველი ელემენტი
- მეორე ელემენტი
- მესამე ელემენტი

როგორც მოგახსენეთ სიის მარკირების ტიპი ასევე შეძლება იყოს გრაფიკული გამოსახულება. ქვემეოთ მოცემულ მაგალითში CSS-ის მეშვეობით ჩვეულებრივი HTML სია გაფორმებული გვაქვს გრაფიკული მარკერებით. მთავარი არის, რომ CSS-ში სწორად მივუთითოთ გზა (URL მისამართი) იმ გრაფიკული გამოსახულებისა რომელმაც უნდა გააფორმოს ჩვენს მიერ შექმნილი სია.

```
<!DOCTYPE html>
<html>
<head>
  <title>დოკუმენტის ტიტული</title>
</head>
<style type="text/css">

ul
{
 list-style-image: url("listicon.gif");
}

</style>

<body>

 <ul>
  <li>პირველი ელემენტი</li>
  <li>მეორე ელემენტი</li>
  <li>მესამე ელემენტი</li>
</ul>

</body>
</html>
```



2.4 შექმნილი ნამუშევრის ხარისხის უზრუნველყოფა W3C სტანდარტებისშესაბამისად

მიმდინარეპარაგრაფისთემატიკა

- W3C სტანდარტიდავალიდაციისინსტრუმენტი

ახლა შევეხოთ ისეთ თემას როგორიც არის WEB გვერდის (რესურსის) სტანდარტიზაცია. რას ნიშნავს სტანდარტიზაცია და ვალიდური WEB პროექტი? მოგეხსენებათ ინტერნეტ სივრცეში ნებისმიემიერი WEB გვერდის დათვალიერება ხორციელდება ბრაუზერების მეშვეობით. ვინაიდან წამყვანი ბრაუზერები, როგორებიც არიან Firefox, Google Chrome, Internet Explorer, Opera და ასე შემდეგ ერთიდაიგივე დოკუმენტს, ანუ მასში გაწერილ კოდს, იქნება ეს HTML თუ CSS კოდი აღიქვავენ ერთმანეთისგან უმნიშვნელოდ მაგრამ მაინც განსხვავებულად. სწორედ ამ მიზეზის გამო ორგანიზაციამ W3C (WORLD WIDE WEB CONSORTIUM) -მა ანუ ორგანიზაციამ რომელიც ქმნის ინტერნეტ სტანდარტებს გადაწყვიტა WEB კოდირების ენები მოექცია გარკვეულ ერთ სტანდარტში. ანუ თუ ჩვენ მივყვებით W3C კონსორციუმის სტანდარტებს და ჩვენს პროექტებს ვქნით მათ მიერ დადგენილი სტანდარტის მიხედვით გამოვა, რომ ჩვენი საიტი აბსოლიტურად გამართულად იმუშავებს ყველა ზემოთ შამოთვლილ ბრაუზერში. აქვე ავღნიშოთ, რომ მსგავსი პროექტის შექმნა და ყველა ბრაუზერზე მორგება საკმაოდ შრომატევადი საქმეა, ვინაიდან პროექტის ფორმირების პროცესში უნდა გავითვალისწინოთ ის გარემოება, როგორიცაა ელემენტარულად ჩვენი ყველა WEB დოკუმენტის სხვადასხვა ბრაუზერში ტესტირება. სწორედ ამ პრობლემის გადასაწყვეტად შეგვიძლია მივმართოთ W3C კონსორციუმის ასე ვთქვათ WEB გვერდის ვალიდატორს. რა არის ვალიდატორი? ვალიდატორი გახლავთ ონლაინ პროგრამული უზრუნველყოფა, რომელსაც შეუძლია ჩვენს მიერ შექმნილი პროექტის კოდი შეამოწმოს უკანასკნელი სტანდარტის მიხედვით ვალიდურობაზე.

იმისათვის რომ დავრწმუნდეთ არის თუ არა ჩვენს მიერ დაწერილი კოდი ვალიდური და შეესაბამება თუ არა ის თანამედროვე სტანდარტებს, პირველ რიგში საჭიროა ვეწვიოთ W3C კონსორციუმის ოფიციალურ ვებ-გვერდს შემდეგ ინტერნეტ მისამართზე : https://validator.w3.org სწორედ აღნიშნულ ვებ-მისამართზე გახლავთ განთავსებული ის ინსტრუმენტი (ვალიდატორი), რომლის შესახებაც ზემოთ გვქონდა საუბარი.

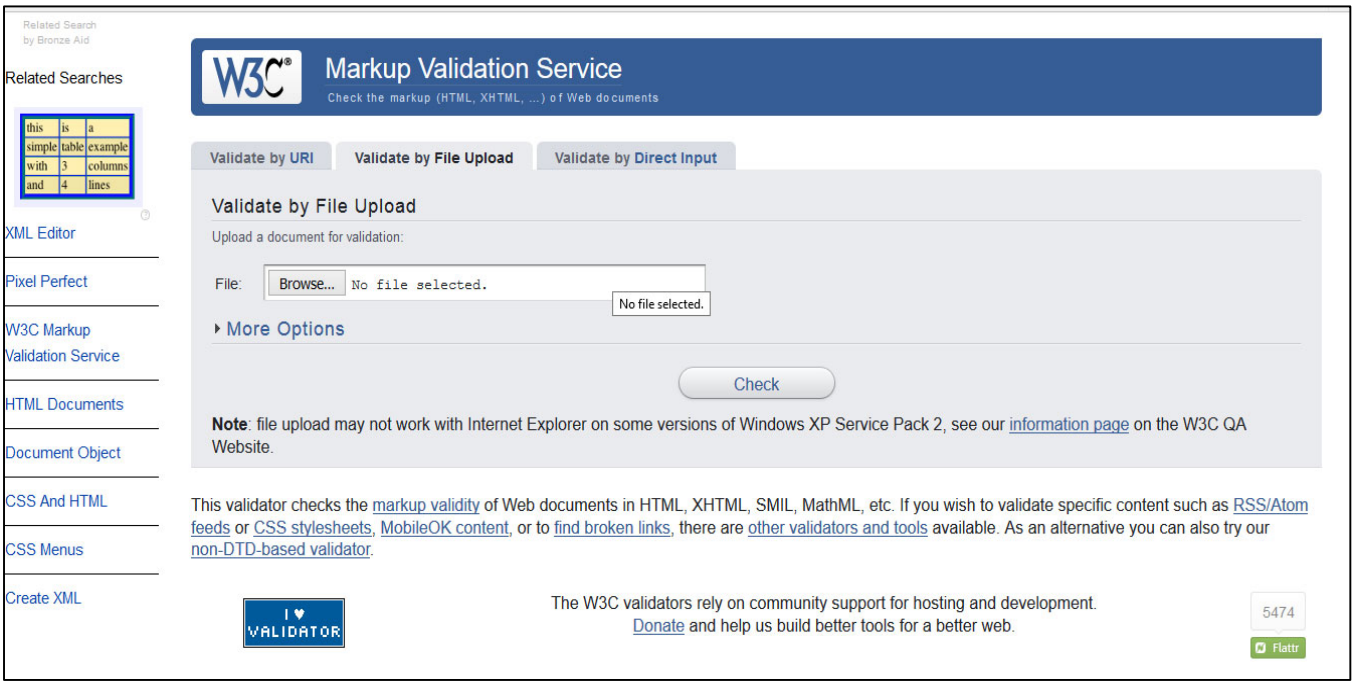
აღნიშნული ვებ-გვერდის ვიზიტისას იხილავთ შემდეგნაირ ინტერფეისს:

აღნიშნული ვალიდატორის ინტერფეისში შენიშნავთ, რომ გაგაჩნიათ თქვენი კოდის ვალიდურობაზე შემოწმების სამი გზა. ეს გახლავთ

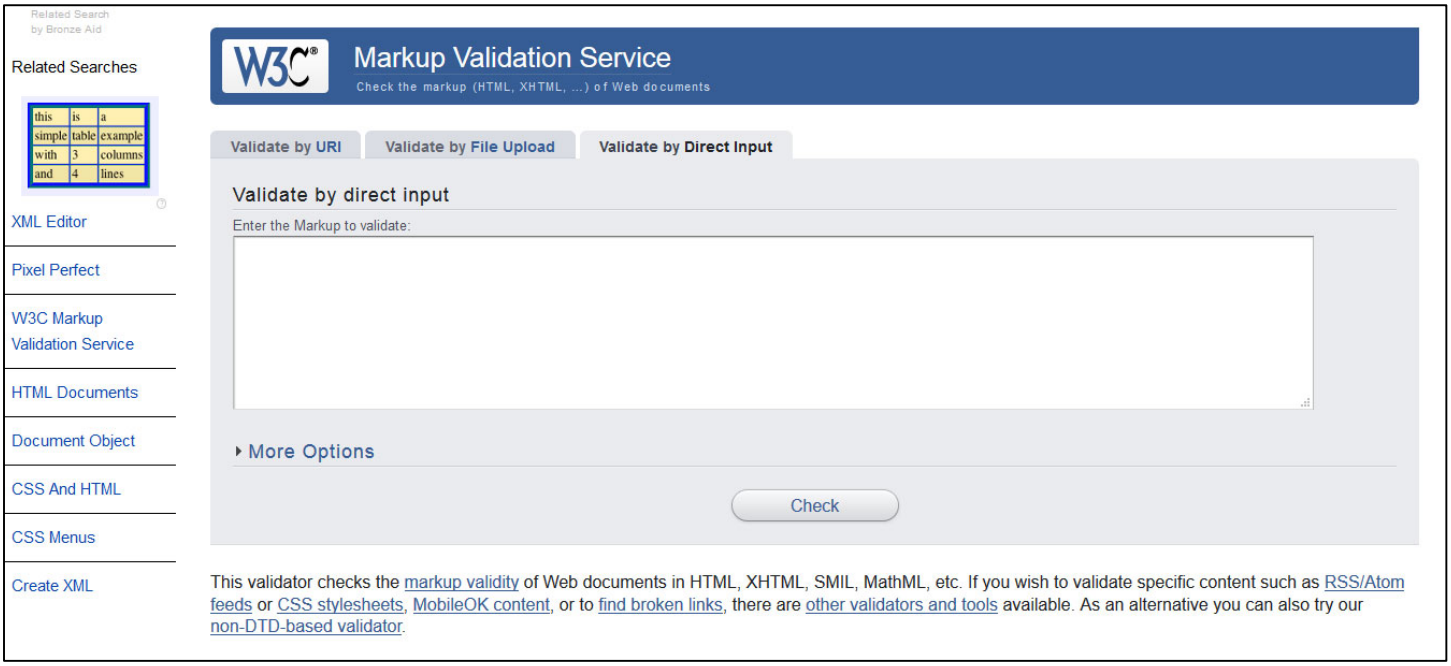
1. Validate by URI
2. Validate by File Upload
3. Validate by Direct Input

პირველივარიანტი, ანუ Validate by URI კოდისვალიდურობისშესამოწმებლადგამოგადგებათმხოლოდმაშინ, როდესაცთქვენიპროექტიანუკოდებიუკვეგარკვეულინტერნეტსერვერზეაგანთავსებულიდაგააჩნიათ ინტერნეტმისმართი. ასეთშემთხვევაშივალიდატორის Address-ისველშიმიუთითებთსწორედამმისამართდავალიდატორიცშეამოწმებსთქვენსფაილსვალიდურობაზე.

Validate by File Upload, ეს მეთოდი მისი სათაურით უკვე თავისთავად მეტყველებს თავის თავზე. ანუ აირჩევთ ნებისმიერ სავალიდაციოთ გამზადებულ ფაილს, ატვირთავთ ვალიდატორის მეშვეობით და ვალიდატორი დაგიბრუნებთ პასუხს, ვალიდურია თუ არა თქვენი შექმნილი კოდი.



და ბოლოს, ყველაზე პრაქტიკულად გამოსაყენებელი ვარიანტი ეს გახლავთ მესამე ვარიანტი. Validate by Direct Input, ანუ ამ შემთხვევაში ყველაზე მარტივად , რედაქტორიდან, სადაც ვქმნით ჩვენს კოდს ვაკოპირებთ კოდის ტექსტს, ვსვამთ ვალიდატორის ფანჯარაში და ვღებულობთ პასუხს თუ რამდენად ვალიდურია ჩვენი კოდი თანამედროვე სტანდარტებთან მიმართებაში.


Related Search
by Bronze Aid
Related Searches
this is a
simple table example
with 3 columns
and 4 lines
XML Editor
Pixel Perfect
W3C Markup Validation Service
HTML Documents
Document Object
CSS And HTML
CSS Menus
Create XML
Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents
Validate by URI
Validate by File Upload
Validate by Direct Input
Validate by direct input
Enter the Markup to validate:
More Options
Check
This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available. As an alternative you can also try our non-DTD-based validator.

კითხვები თვითშეფასებისათვის

1. **HTML კოდის რომელ ნაწილში უნდა მოხდეს სტილების ფაილთან კავშირის აღწერა**
   a. დოკუმენტის დასაწყისში
   b. <body> </body> ნაწილში
   c. <head> </head> ნაწილში
   d. დოკუმენტის ბოლოში
2. **რომელი თვისებითაა შესაძლებელი ბლოკური ელემენტის ცენტრირება**
   a. margin: auto;
   b. margin: center;
   c. align: auto;
3. **კლასზე სტილის მინიჭების რომელი ჩანაწერია მართებული**
   a. .კლასის სახელი{თვისება:მნიშვლელობა}
   b. კლასის სახელი{თვისება:მნიშვლელობა}
   c. კლასის სახელი{თვისება="მნიშვლელობა"}
   d. #კლასის სახელი{თვისება:მნიშვლელობა}
4. **მოცემულ მაგალითში როგორი იქნება „საქართველოს ტექნიკური უნივერსიტეტის“ გაფორმების სტილი**

   .....
   p{text-align:center; color:red}
   .text{font-weight:bold; color:green}
   .....
   ** საქართველოს ტექნიკური უნივერსიტეტი **

   a. წითელი ფერის, მუქი და ცენტრირებული.
   b. მწვანე ფერის, მუქი და ცენტრირებული.
   c. წითელი ფერის და ცენტრირებული.
   d. მწვანე ფერის და ცენტრირებული.
   e.
5. **მოცემულია სტილები:**
   p{text-align:center}
   .text1{font-weight:bold; color:green}
   #text2{font-style:italic;}

   იმისათვის რომ საქართველოს ტექნიკური უნივერსიტეტი იყოს ცენტრირებული, მწვანე, მუქი და დახრილი რომელი ტეგია მართებული.
   a. <p class=”text1” id=” text2”>საქართველოს ტექნიკური უნივერსიტეტი
   b. <p class=”text1 text2”>საქართველოს ტექნიკური უნივერსიტეტი
   c. <p class=”text1” class=” text2”>საქართველოს ტექნიკური უნივერსიტეტი
   d. <p id=”text1 text2”>საქართველოს ტექნიკური უნივერსიტეტი

6. **რომელ ფორმატში უნდა შევინახოთ სტილების აღმწერი ფაილი**
   a. css
   b. html
   c. exe
   d. com
7. **რომელი ჩანაწერია მართებული ერთგვაროვანი თვისებების რამოდენიმე კლასსზე მისანიჭებისათვის**
   a. სელექტორი, სელექტორი {თვისება:მნიშვნელობა}
   b. სელექტორი სელექტორი {თვისება:მნიშვნელობა}
   c. სელექტორი&სელექტორი {თვისება:მნიშვნელობა}
   d. სელექტორი; სელექტორი {თვისება:მნიშვნელობა}

8. რომელი თვისებითაა შესაძლებელი ბლოკური ობიექტების ერთმანეთის გვერდით განლაგება
   a. float:left
   b. align:left
   c. overflow:hidden
   d. float:float
9. რომელია მართებული ჩანაწერი, html დოკუმენტის სტილების აღმწერ ფაილთან დასაკავშირებლად
   a. <link rel=”stylesheet” type=”text/css” href=“layout.css”>
   b. <link rel=”stylesheet” type=”text/css” href="style”>
   c. <style rel=”stylesheet” type=”text/css” href="style”>
   d. <link rel=”stylesheet” type=”text/css” href-"style.html”>
10. ტეგში შესაძლებელია ორი კლასის ერთდროულად გამოძახება შემდეგი ჩანაწერის სახით
    მაგ: <p class="title" class="text">
   a. ჭეშმარიტია
   b. მცდარია
11. ვებ გვერდის ფარგლებში შესაძლებელია კლასსი მხოლოდ ერთჯერადი გამოყენება.
   a. ჭეშმარიტია
   b. მცდარია
12. შეუსაბამეთ ერთმანეთს ფონის თვისებები:
   a. ფონის ფერის განსაზღვრა Answer 1
   b. ფონად სურათის ჩასმა Answer 2
   c. ფონის სურათის მდებარეობის განსაზღვრა Answer 3
   d. ფონის სურათის განმეორებადობის განსაზღვრა Answer 4

13. ვებ გვერდზე ტექსტის ფერის შეცვლისათვის რომელიამართებული ჩანაწერი
   a. body:color=gray
   b. body {color: gray}
   c. body{color=gray}
   d. body{bgcolor:gray}
14. ვებგვერდის ფონის ფერის შესაცვლელად რომელია მართებული ჩანაწერი
   a. body {background:#1a2b3c}
   b. body {bgcolor:#1a2b3c}
   c. body {backcolor:#1a2b3c}
   d. body {color:#1a2b3c}
15. რომელია მართებული ჩანაწერი
   a. body {background:url(back.jpg); background: no-repeat; background: center top}
   b. body {background-image:back.jpg; background-image:no-repeat; background-image:center top}
   c. body {background-image:url(back.jpg); background-repeat: no-repeat; background-position: center top}
   d. body {background:url(back.jpg) no-repeat center top}
16. ფონად სურათის ჩასასმელად რომელია მართებული ჩანაწერი
   a. body {background-image:url(back.jpg)}
   b. body {background-img:url(back.jpg)}
   c. body {background:back.jpg}
   d. body {background-image:back.jpg}
17. ვებ გვერდზე ფონად მცირე ზომის სურათის ჩასმის დროს ავტომატურად ხდება მისი გადამრავლება
   a. ჭეშმარიტია
   b. მცდარია
18. ვებ გვერდზე ფონად შეუძლებელია ერთდროულად იყოს სურათიც და ფერიც ჩასმული
   a. ჭეშმარიტია
   b. მცდარია
19. ვებ გვერდზე შესაძლებელია ცალკეულ აბზაცში ფონად სხვადასხვა ფერის ჩასხმა
   a. ჭეშმარიტია
   b. მცდარია

20. როცა ვებ გვერდზე ფონად ვსვამთ ერთ სურათს (არ მრავლდება) ავტომატურად ის აისახება ეკრანის ცენტრში.
    a. ჭეშმარიტია
    b. მცდარია
21. რომელი თვისება გამოიყენება ტექსტის ფერის შესაცვლელად?
    a. color
    b. font-color
    c. text-color
22. როგორ შეიძლება ყველა ბმულის ფერის შეცვლა წითელი ფერით?
    a. a{color:red}
    b. .a{color:red}
    c. a{all-color:red}
    d. #a{color:red}
23. რომელი თვისებითაა შესაძლებელი ტექსტის გამუქება?
    a. font-weight:bold
    b. font:bold
    c. test-weight:bold
    d. font-style:bold
24. რომელი თვისებითაა შესაძლებელი ტექსტის დახრა?
    a. font-style:italic
    b. text-style:italic
    c. text:italic
    d. font:italic
25. რომელი თვისებითაა შესაძლებელი ტექსტის ზომის შეცვლა?
    a. text-size
    b. text-style
    c. font-size
    d. size
26. რომელი თვისებითაა შესაძლებელი ტექსტის მდებარეობის შეცვლა?
    a. text-align:
    b. align:
    c. font-align:
    d. align-text:
27. რომელი თვისებითაა შესაძლებელი ტექსტის შრიფტის შეცვლა?
    a. font-face:შრიფტის სახელი
    b. text-face:შრიფტის სახელი
    c. font-family:შრიფტის სახელი
    d. font:შრიფტის სახელი
28. რომელი თვისებითაა შესაძლებელი ტექსტის ხაზგასმა?
    a. text-decoration:underline
    b. text-underline:yes
    c. font-decoration:underline
    d. text-style:underline
29. რომელი თვისებითაა შესაძლებელი ტექსტის ხაზგასმის მოხსნა?
    a. text-decoration:none
    b. text-underline:none
    c. underline:none
    d. font-decoration:none
30. რომელი თვისებითაა შესაძლებელი ჩამონათვალის სიმბოლოს განსაზღვრა
    a. list-style-type
    b. list-type
    c. ul-style
    d. ol-style

31. შეუსაბამეთ ერთმანეთს:
    a. border-width:Answer 1
    b. border-color: Answer 2
    c. border-style: Answer 3

32. რომელი ჩანაწერი უნდა გამოვიყენოთ რომ ობიექტი სხვა ობიექტებიდან დაშორებული იყოს შემდეგი პარამეტრებით: ზედა და ქვედა = 20px; მარცხენა და მარჯვენა = 15px;
    a. margin:20px 15px 20px 15px
    b. margin:20px 15px
    c. margin:20px 20px 15px 15px
    d. margin: 15px 15px 20px 20px
33. რომელი თვისებითაა შესაძლებელი ობიექტის მარცხენა დაშორების განსაზღვრა?
    a. margin-left:
    b. left:
    c. align-left:
    d. padding-left:
34. რომელი თვისებითაა შესაძლებელი ობიექტის საზღვრიდან მის შიგთავსამდე დაშორების განსაზღვრა?
    a. margin
    b. padding
    c. align
    d. indent
35. რომელი ჩანაწერია მართებული 500X200 ზომის ობიექტის აღსაწერად
    a. .box {width:500px; height:200px; padding:50px}
    b. .box {width:400px; height:100px; padding:50px}
    c. .box {width:500px; height:200px; margin:50px}
    d. .box {width:400px; height:100px; margin:50px}
36. რომელი ჩანაწერია უნდა გამოვიყენოთ რომ ობიექტი სხვა ობიექტებიდან დაშორებული იყოს შემდეგი პარამეტრებით: ზედა=20px; ქვედა=10px; მარცხენა=30px; მარჯვენა=15px
    a. margin:20px 30px 10px 15px
    b. margin:20px 15px 10px 30px
    c. margin:20px 10px 30px 15px
    d. margin:30px 15px 20px 10px
37. რომელი ჩანაწერის გამოყენებით არის შესაძლებელი ბმულზე მაუსის მიტანისას სტილის განსაზღვრა?
    a. a:hover
    b. a
    c. a:link
    d. a:active
38. რომელი ჩანაწერის საშუალებით შეიქმნება ობიექტის ზედა ჩარჩო 2px-ის ზომის, ერთმაგი და წითელი ფერის.
    a. .box{border-top-width:2px; border-top-color:red; border-top-style:solid}
    b. .box{border-width:2px; border-color:red; border-style:solid}
    c. .box{border-top:2px solid red}
    d. .box{top-width:2px; top-color:red; top-style:solid}
39. თუ მითითებული არ არის ბლოკური (მაგ.div) ელემენტების ურთიერთმდებარეობა, მაშინ ყოველი მომდევნო ბლოკი ავტომატურად იწყება ახალი ხაზიდან ანუ ისმება წინა ბლოკის ქვემოთ.
    a. ჭეშმარიტია
    b. მცდარია
40. ჩარჩოს თვისებების (border-width, border-style; border-color) გამოყენებისას შესაბამი ობიექტი ფორმდება ოთხივე მხრიდან ჩარჩოს არჩეული მნიშვნელობებით.
    a. ჭეშმარიტია
    b. მცდარია

თავი 3.ვებ მარკირების გაფართოებული შესაძლებლობები (HTML5, CSS3)

3.1 ვებსაიტის ძირითადი სტრუქტურის აგება HTML 5-ის ბრძანებების გამოყენებით

მიმდინარე პარაგრაფის თემატიკა

- HTML 5 ის მიმოხილვა
- რა სიახლეებია HTML 5 ში
- რომელი ტაგები აღარ გამოიყენება HTML 5 ში
- HTML 5 ის მეშვეობით დოკუმენტის ძირითადი სტრუქტურის შექმნა

HTML 5-ის მიმოხილვა

პირველ რიგში ავღნიშნოთ ის ფაქტი, რომ HTML 5 გაცილებით მოქნილი არის გამოყენებაში და რაც მთავარია უკვე სტანდარტიზირებულია რაც იმას ნიშნავს, რომ მისი აღქმა ბრაუზერებისთვის პრობლემას არ წარმოადგენს. თუმცა აქვე გვინდა მივანიშნოთ, რომ HTML -ის ნებისმიერი ვერსიის გამოყენებისას ყოველთვის აუცილებელია ჩვენს მიერ შექმნილი კოდის ტესტირება სხვადასხვა ბრაუზერში, ვინაიდან პრაქტიკამ აჩვენა ბრაუზერების მიერ HTML კოდის არაერთგვაროვნად აღქმის შესაძლებლობა. გამონაკლისი ამ შემთხვევაში რათქმაუნდა არც HTML 5 გახლავთ. HTML ის ხსენებულ ვერსიაში საკმაოდ არს ცვლილებები შეტანილი. ჩვენ შევეცდებით ავღწეროთ თუ რა ცვლილებებია მასში, რა დაემატა და რომელი ტაგები იქნა ამოღებული გამოყენებიდან. ყველაზე მნიშვნელოვანი და პრაქტიკაში ადვილად გამოსაყენებელი რაც არის, ეს გახლავთ HTML 5-ის კოდის წერის შემოკლებული სინტაქსი. მაგალითად როგორიცაა დოკუმენტის სტრუქტურის შექმნა, გარკვეული მულტიმედიური და ფორმის ელემენტების გაცილებით მარტივად გამოყენება, გრაფიკასთან მუშაობა და ასე შემდეგ.

რა სიახლეებია HTML 5 -ში?

პირველ რიგში გავამახვილოთ ყურადღება ისეთ მნიშვნელოვან საკითხზე როგორიც გახლავთ HTML დოკუმენტის სტრუქტურის ფორმირება, კერძოდ დოკუმენტის ტიპის დეკლარირება. მაგალითად: თუ HTML 4.01 ვერსიაში დოკუმენტის ტიპის დეკლარირება გვიწევდა შემდეგნაირი სინტაქსით როგორიც არის

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
```

HTML 5 -ში ეს ამოცანა გაცილებით გამარტივებულია და ჩანაცვლებულია შემდეგი დეკლარირების ტიპით

```
<!DOCTYPE html>
```

აქვე ხაზი გავუსვათ იმ ფაქტს, რომ დოკუმენტის ტიპის დეკლარირებისას სიმბოლოების რეგისტრს არავითარი მნიშვნელობა არ გააჩნია ბრაუზერებისთვის. ანუ ზემოთ მოყვანილი ჩანაწერი და მომდევნო ჩანაწერი იდენტურები არიან ბრაუზერებისთვის სტანდარტის მიხედვით

```
<!DOCTYPE HTML>
```

ზუსტად იგივე შეგვიძლია ავღნიშნოთ დოკუმენტის ენობრივ კოდირებასთან დაკავშირებით. HTML 5 ის სტანდარტით გაცილებით მატივად შეგვიძლია დოკუმენტის კოდირების ტიპის მითითება. მაგ:

```
<meta charset="UTF-8>
```

თუმცა გვინდა ყურადღება გავამახვილოთ იმ ფაქტზე, რომ როდესაც HTML 5 ის კოდირების დეკლარირებას არ ვახდენთ მაშინ ავტომატურად ანუ გაჩუმების პრინციპით კოდირების ტიპი ბრაუზერების მიერ აღიქმება როგორც UTF-8.

რაც შეეხებოდა დოკუმენტის სტრუქტურის დეკლარირებას ამაზე უკვე ზემოთ ვისაუბრეთ. ახლა შევეხოთ იმ თემას თუ რა პრინციპულად განსხვავებული ელემენტები არის დამატებული HTML 5ის მარკირების ენაში. ესენი გახლავთ:

- **სემანტიკური ტიპის ელემენტები:**<header>, <footer>, <article>, <section>.
- **ფორმის ელემენტები:** number, date, time, calendar, and range
- **გრაფიკული ტიპის ელემენტები:**<svg>, <canvas>
- **მულტიმედიური ტიპის ელემენტები:**<audio>, <video>.

რათქმაუნდა ყველა ზემოთ ჩამოთვლილ ელემენტს დეტალურად განვიხილავთ და ვნახავთ მათ თვისებებს პრაქტიკულ მაგალითებზე.

რომელი ტაგები აღარ გამოიყენება HTML 5 -ში?

როგორც მოგეხსენებათ HTML მარკირების ენა მუდმივ ფორმირებას განიცდის და სულ დამუშავების პროცეში არის. გარკვეული წლების მანძილზე ცდილობდნენ მის სტანდარტიზაციაზე და უნივერსიალიზაციაზე. სწორედ ასეთი მცდელობების გამო ხდება მისი ფორმირება. მაგალითად HTML 4 ვერსიაში გადაწყვიტეს გამოეყენებინათ ახალი ტიპის ტექსტის ლოგიკური ფორმატირების ტაგები. ასევე იყო დავა და საუბარი დარჩენილიყო თუ არა სტანდარტის მიხედვით FRAME-ები. HTML 5ის მარკირების ენაში ამ თემატიკასთან დაკავშირებით საკმაოდ მკვეთრი ცვლილებებია შეტანილი, კერძოდ: ამოღებულია გამოყენებიდან შემდეგი ტაგები:

<acronym>
<applet>
<basefont>
<big>
<dir>
<font>
<frame>
<frameset>
<noframes>
<strike>
<tt>

ქვმოთ მოყვანილია მაგალითი თუ რა მარტივი არის HTML 5 ის მეშვეობით დოკუმენტის სტრუქტურის შექმნა და ძირითადი სტრუქტურის შემადგენელი ელემენტების დეკლარირება:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8
<title>დოკუმენტის ტიტული</title>
</head>
<body>

</body>
</html>
```

3.2 ვებსაიტზე მულტიმედია ელემენტების გამოყენება

მიმდინარე პარაგრაფის თემატიკა

- რას წარმოადგენ მულტიმედიური ელემენტები, მოკლე მიმოხილვა
- მულტიმედიური ელემენტების ფორმატები და სფეციფიკა
- WEB-ზე გამოსაყენებლად რეკომენდეუბლი ვიდეო ფაილების ფორმატები
- WEB-ზე გამოსაყენებლად რეკომენდეუბლი აუდიო ფაილების ფორმატები
- ვებსაიტზე მულტიმედიური ელემენტის შემოტანა
- ვებსაიტზე მულტიმედიური ელემენტის თვისებები

რას წარმოადგეს მულტიმედიური ელემენტები? მოკლე მიმოხილვა

საიტზე მულტიმედიურ ელემენტებს ვუწოდებთ ყველა იმ ელემენტს, რომელიც არ გამოისახება ტექსტის სახით. ანუ მულტიმედიური ელემენტები არსებობს როგორც

- ვიდეო
- აუდიო
- გრაფიკული
- ანიმაცია

თითოეულ მულტიმედიურ ელემენტს გააჩნია თავისი სპეციფიური ფორმატი. HTML 5 ვერსიამდე მულტიმედიური და განსაკუთრებულად ვიდეო-აუდიო და ანიმაციური ფაილების ჩაშენებას ბრაუზერში და შემდგომ მის ასახვას უზრუნველყოფდა FLASH ფლეიერი. დღესდღეობით კი HTML 5 ის მეშვეობით გაცილებით მარტივად შეგვიძლია მივაწოდოთ მომხმარებელს ჩვენს მიერ ჩაშენებული მულტიმედიური ფაილი იქნება ეს ვიდეო, ანიმაცია თუ აუდიო ფაილი.

ვიდეო ფაილების ფორმატები და მათი აღწერა

ფორმატი	გაფართოება	აღწერა
MPEG	.mpg .mpeg	MPEG. შემუშავებულია „ Moving Pictures Expert Group“ -ის მიერ პირველი ყველაზე პოპულარული ვიდეო ფორმატის მქონე ფაილი WEB-ზე გამოსაყენებლად. მისი მხარდაჭერა გააჩნია ყველა ბრაუზერს მაგრამ არ გამოიყენება HTML 5 -ში. მის ნაცვლად ვიყენებთ MP4-ის ფორმატის ფაილებს.
AVI	.avi	AVI (Audio Video Interleave). შემუშავებულია კომპანია Microsoft-ის მიერ. კარგი ხარისხით აისახება WINDOWS-ის გარემეოში მაგრამ არა ბრაუზერებში
WMV	.wmv	WMV (Windows Media Video). შემუშავებულია კომპანია Microsoft-ის მიერ. კარგი ხარისხით აისახება WINDOWS-ის გარემეოში მაგრამ არა ბრაუზერებში.
QuickTime	.mov	QuickTime. შემუშავებულია კომპანია Apple-ის მიერ. კარგი ხარისხით აისახება Apple-ის გარემეოში მაგრამ არა ბრაუზერებში.
RealVideo	.rm .ram	RealVideo შემუშავებულია Real Media-ს მიერ რათა უზრუნველყო ვიდეოს გადმოცემა ნელი ინტერნეტ კავშირის შემთხვევაშიც. დღემდე გამოიყენება ონლაინ ვიდეოს ფაილების გადმოსაცემად და ინტერნეტ ტელევიზიებისთვის. არ გამოიყენება ბრაუზერებისთვის.
Flash	.swf .flv	Flash- შემუშავებულია კომპანია Macromedia-ს მიერ და როგორც წესი მოითხოვს დამატებით კომპონტენს , ეგრეთწოდებულ plug-in-ის რათა ბრაუზერს ჰქონდეს მისი ასახვის საშუალება.
Ogg	.ogg	Theora Ogg. შემუშავებულიაა „Xiph.Org Foundation“-ის მიერ. ამ ფორმტს გააჩნია სრული HTML5-ის მხარდაჭერა.
WebM	.webm	WebM. შემუშავებულია შემდეგი კომპანიების მიერ: web giants, Mozilla, Opera, Adobe და Google. ამ ფორმტს გააჩნია სრული HTML5-ის მხარდაჭერა.
MPEG-4 ან MP4	.mp4	MP4. შემუშავებულია „Moving Pictures Expert Group“ -ის მიერ. ამ ფორმტს გააჩნია სრული HTML5-ის მხარდაჭერა და რეკომენდებულია YouTube-ის მიერ.

აქვე გვინა მივანიშნოთ, რომ ზემოთ ჩამოთვლილი ვიდეო ფორმატებიდან მხოლოდ MP4, WebM და Ogg ფორმატის ვიდეო ფაილების მხარდაჭერა გააჩნია HTML5-ის მარკირების ენის სტანდარტს.

აუდიო ფაილების ფორმატები და მათი აღწერა

ფორმატი	გაფართოება	აღწერა
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). წარმოადგენს ელექტრონული მუსიკის მთავარ ფორმატს. მას კარგად აღიქვავს ყველა ციფრული მოწყობილობა. მათ შორის კომპიუტერული და მუსიკალური მოწყობილობები. არ გამოიყენება ბრაუზერებში.

RealAudio	.rm .ram	RealAudio. შემუშავებულია „ Real Media“ -ის მიერ რათა უზრუნველყო დაბალი იტერნეტ კავშირის შემთხვევაში მისი გადმოცემა. არ გამოიყენება ბრაუზერებში.
WMA	.wma	შემუშავებულია კომპანია Microsoft-ის მიერ. კარგი ხარისხით ჟღერს WINDOWS-ის გარემეოში მაგრამ არა ბრაუზერებში.
AAC	.aac	AAC (Advanced Audio Coding). შემუშავებულია კომპანია Apple-ის მიერ როგორც „iTunes“ ძირითადი ფორმატი. საუკეთესოდ ჟღერს Apple-ის ფირმის კომპიუტერებზე მაგრამ არა ბრაუზერებში.
WAV	.wav	WAV. შემუშავებულია IBM და Microsoftის მიერ. აუკეთესოდ ჟღერს Windows, Macintosh, და Linux-ის ოპერაციულ სისტემებზე. გააჩნია HTML5-ის სრული მხარდაჭერა.
Ogg	.ogg	Theora Ogg. შემუშავებულიაა „Xiph.Org Foundation“-ის მიერ. ამ ფორმტს გააჩნია სრული HTML5-ის მხარდაჭერა.
MP3	.mp3	MP3 ფაილები როგორც წესი წარმოადგენენ MPEG ფაილების ფორმატის აუდიო ნაწილს. MP3 ყველაზე პოპულარული და გავრცელებული ფორმატია აუდიო ფაილების გადმოსაცემად. ის გახლავთ კარგად კომპრესირებული ტიპის ფაილები საუკეთესო ხარისხით. ამის გამო შედეგად MP3 ფორმატის ფაილები ზომაში საგრძნობლად პატარა გახლავთ. მისი აღქმის მხარდაჭერა გააჩნია ყველა ბრაუზერს.
MP4	.mp4	MP4. შემუშავებულია „Moving Pictures Expert Group“ -ის მიერ. ეს ფორმატი რეალურად შემუშავებულია ვიდეო ფაილებისთვის მაგრამ შეიძლება გამოყენებული იქნეს აგრეთვე აუდიო ფაილებისთვისაც. მას გააჩნია სრული HTML5-ის მხარდაჭერა.

აქვე გვინა მივანიშნოთ, რომ ზემოთ ჩამოთვლილი ვიდეო ფორმატებიდან მხოლოდ MP3, WAV და Ogg ფორმატის აუდუიო ფაილების მხარდაჭერა გააჩნია HTML5-ის მარკირების ენია სტანდარტს.

ვინაიდან ზემოთ აღწერილი ცხრილებიდან შეგვიძლია გავაანალიზოთ თუ რომელი ფორმატის ვიდეო ან აუდიო ფაილი სჯობს რომ გამოვიყენოთ ბრაუზერებში მათი სტანდარტის მიხედვით გამოყენების უზრუნველყოფის მიზნით მივდივართ დასკვნამდე , რომ

სჯობს ვიდეო ფაილების ჩვენს პეოექტში ჩაშენებისთვის შევარჩიოთ შემდეგი ფორმატის ფაილები:
MP4, WebM და Ogg

ხოლო რაც შეეხება აუდიო ფაილებს გამოვიყენოთ შემდეგი ფორმატის ფაილები:
MP3, WAV და Ogg.

მოგეხსენებათ HTML 5 ის ვებმარკირების ენის შემოღებამდე ვებსაიტზე ვიდეოს შემოტანის სტანდარტიზირებული მეთოდი არ არსებობდა. როგორც ზემოთ უკვე ავღნიშნეთ ვიდეო ფაილის აღსაქმელად ბრაუზერს ესაჭირეობადა ეგრეთწოდებული Flash Player- ი, ანუ დამატებითი კომპონენტი (Plug-in)-ი. HTML 5 ის სტანდარტის მიხედვით ეს პრობლემა უკვე აღმოფხვრილია და ვიდეო ფაილის ბრაუზერში ინტეგრაცია გაცილებით მარტივია ტაგი <video> -ს მეშვეობით, რომელსაც რათქმაუნდა გააჩნია საკუთარი თვისებები როგორიც არის ვიდეოს ზომები, ანუ სიმაღლე და სიგანე და ასე შემდეგ.

ქვემოთ წარმოდგენილია პრაქტიკული მაგალითი თუ როგორ უნდა ჩავაშენოთ ჩვენთვის სასურველი ვიდეო ფაილი ბრაუზერში.

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  თქვენს ბრაუზერს არ გააჩნია მულტიმედიური ფორმატის მხარდაჭერა
  </video>
</body>
</html>
```

ახლა განვიხილოთ ტაგი <video>- ს გამოყენების სფეციფიკა. რაც შეეხება თვისებებს HEIGHT-ს და WIDTH -ს ვფიქრობთ ეს თვისეებები ყველასათვის უკვე ნათელია და ყველამ ვიცით მათი დანიშნულება. გადავიდეთ თვისებაზე CONTROLS ეს თვისება ვიდეო კონსოლს (ფლეიერს) განუსაზღვრავს კონტროლის ღილაკების თვისებებს, როგორიცაა PLAY-ს ღილაკი, ღილაკი PAUSE და ღმის საკონტროლებელი ღილაკი VOLUME.

აგრეთვე დააკვირდებოდით ტაგში <video> განთავსებულია ტაგი <source>. სწორედ ამ ტაგის და მისი თვისებების მეშვეობით უკავშირდება დოკუმენტი კონკრეტულ ვიდეო ფაილს. ჩვენს შემთხვევაში ფაილის URL-ს მისამართს და მის სახელს წარმოადგენს „movie.mp4“. გარდა ამისა მოცემულ ტაგს გააჩნია თვისება TYPE , რომელიც განსაზღვრავს მედია ფაილის ტიპს.

ქვემოთ მოყვანილ ცხრილში ჩამთვლილია დღევანდელი სტანდარტის მიხედვით გამოყენებადი ფაილის ფორმატის მედია ტიპები:

ვიდეო ფაილის ფორმატი	მედია ტიპი
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

ზემოთ მოყვანილ მაგალითში შენიშნავდით ტაგები <video></video>- ს შორის განთავსებულ ტექსტს:

„თქვენს ბრაუზერს არ გააჩნია მულტიმედიური ფორმატის მხარდაჭერა “

ეს ტექსტი აისახება ბრაუზერებში მხოლოდ იმ შემთხვევაში თუ ბრაუზერს არ გააჩნია ვიდეო ფორმატის მქონე ფაილების მხარდაჭერა. ვინაიდან მომხარებელი არ შევიყვანოთ შეცდომაში და ინფორმირებული იყოს იმის თაობაზე, რომ თუ ბრაუზერი არ ასახავს ვიდეო ფაილს ეს არ არის საიტის ხარვეზი და იზრუნოს იმაზე, რომ იხილოს ჩვენს მიერ მიწოდებული საიტი შესაბამისი ბრაუზერის მეშვეობით.

ახლა წარმოგიდენთ შემდეგ მაგალითს

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  თქვენს ბრაუზერს არ გააჩნია მულტიმედიური ფორმატის მხარდაჭერა
  </video>
</body>
</html>
```

ზემოთ მოყვანილ მაგალითში გვინდა ხაზი გავუსვათ თვისებას AUTOPLAY-ს, აღნიშნული თვისება უზრუნელყოფს ვიდეო ფორმატის ფაილის ავტომატურად ჩართვას. ასეთ შემთხვევაში ვიდეო კონსოლის ღილაკები რომლებზეც ზემოთ გვქონდა საუბარი PLAY, PAUSE და VOLUME უკვე აღარ არის მომხმარებლისთვის ხელმისაწვდომი. ანუ ვიდეო ჩატვირთვისთანავე გაეშვება ავტომატურ რეჟიმში.

თუ გავაკეთებთ მოკლე რეზიუმეს, აღმოჩნდება, რომ ვებსაიტზე HTML 5 ის სტანდარტის მიხედვით უნდა გამოვიყენოთ სამი ფორმატის ვიდეო ფაილები, ვინაიდან ვუზრუნველყოთ მომხმარებლისთვის ვიდეო მასალის მიწოდება ბრაუზერებისთვის აღსაქმელ ფორმატში. ესფრომატები გახლავთ:

- MP4
- WebM

- Ogg

ქვემოთ მოყვანილ ცხრილში ასახულია რომელ ბრაუზერს გააჩნია კონკრეტული ვიდეო ფაილების ფორმატის მხარდაჭერა.

ბრაუზერერი	MP4	WebM	Ogg
Internet Explorer	დიახ	არა	არა
Chrome	დიახ	დიახ	დიახ
Firefox	დიახ	დიახ	დიახ
Safari	დიახ	არა	არა
Opera	დიახ	დიახ	დიახ

HTML დოკუმენტში აუდიო ფაილის შემოსატანად გამოიყენება ტაგი <audio>, რომელსაც ტაგი <video>-მსგავად გააჩნია თვისება controls . აღნიშნული თვისება უზურნველყოფს აუდიო კონსოლის კონტროლის ღილაკების არსებობას, როგორებიცაა PLAY, PAUSE და VOLUME.

```
<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  თქვენს ბარუზერს არ გააჩნია აუდიო ფაილების აღქმის მხარდაჭერა
</audio>

</body>
</html>
```

აღნიშნულ ტაგს ასევე ვიდეოს მსგავსად გააჩნია თვისება AUTOPLAY. ამ თვისების გამოყენების შემთხვევაში კონსოლი საერთოდ არ აისახება ბრაუზერში და აუდიო ფაილი დოკუმენტის ჩატვირთვისთანავე იწყებს აუდიო ფაილის ფონურ რეჟიმში გაშვებას.

აგრეთვე აუდიო ფაილის ტაგსაც, ვიდეოს მსგავსად გააჩნია ტაგი <source>, რომელიც მისი თვისებების მეშვეობით უკავშირდება კონკრეტულ აუდიო ფაილს. ჩვენს შემთხვევაში ფაილის URL-ს მისამართს და მის სახელს წარმოადგენს „horse.ogg“. გარდა ამისა მოცემულ ტაგს გააჩნია თვისება TYPE , რომელიც განსაზღვრავს მედია ფაილის ტიპს.

ქვემოთ მოყვანილ ცხრილში წარმოდგენილია იმ აუდიო ფაილების მედია ტიპები რომლებიც გამოიყენება HTML5-ის სტანდარტის მიხედვით.

აუდიო ფაილის ფორმატი	მედია ტიპი
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

ტაგებს შორის მოთავსებული ტექსტი:

„თქვენ ბრაუზერს არ გააჩნია აუდიო ფაილების აღქმის მხარდაჭერა“

აისახება მხოლოდ მაშინ როდესაც ბრაუზერი ვერ აღიქვამს აუდიო ფალის ფორმატს და ვერ გადმოსცემს მას მომხმარებლისთვის.

ქვემოთ მოყვანილია მაგალით სადაც აუდიო კონსოლს მითითებული აქვს თვისება AUTOPLAY

```
<!DOCTYPE html>
<html>
<body>

<audio autoplay>
  <source src="horse.ogg" type="audio/ogg">
  თქვენს ბარუზერს არ გააჩნია აუდიო ფაილების აღქმის მხარდაჭერა
</audio>

</body>
</html>
```

HTML 5 ის სტანდარტის მიხედვით უნდა გამოვიყენოთ სამი ფორმატის აუდიო ფაილები, ვინაიდან ვუზრუნველყოთ მომხმარებლისთვის აუდიო მასალის მიწოდება ბრაუზერებისთვის აღსაქმელ ფორმატში. ესფრომატები გახლავთ:

- MP3
- Wav
- Ogg

ქვემოთ მოყვანილ ცხრილში ასახულია რომელ ბრაუზერს გააჩნია კონკრეტული ვიდეო ფაილების ფორმატის მხარდაჭერა.

ბრაუზერერი	MP3	Wav	Ogg
Internet Explorer	დიახ	არა	არა
Chrome	დიახ	დიახ	დიახ
Firefox	დიახ	დიახ	დიახ
Safari	დიახ	დიახ	არა
Opera	დიახ	დიახ	დიახ

მულტიმედიური ელემენტების დოკუმენტში ჩასაშენებლად ასევე აქტუალურად გამოიყენება ტაგი <object>, ამ ტაგის მეშვეობით შეგვიძლია დოკუმენტში ჩავაშენოთ სხვა მულტიმედიურ ელემენტებთან ერთად როგორც FLASH, JAVA APPLET და PDF ფორმატის ფაილები ასევე HTML ფორმატის ფაილები. ქვმეოთ მოყვანილია მაგალითები რომლებიც ასახავენ თუ როგორ გამოიყენება აღნიშნული ტაგი პრაქტიკაში.

FLASH ფორმატის ფაილის შემოტანა დოკუმენტში

```
<!DOCTYPE html>
<html>
<body>

<object width="400" height="50" data="flashfile.swf"></object>

</body>
</html>
```

HTML ფორმატის ფაილის შემოტანა დოკუმენტში

```
<!DOCTYPE html>
<html>
<body>

<object width="100%" height="500px" data="snippet.html"></object>

</body>
</html>
```

ასევე ანალოგიური გახლავთ ტაგი <EMBED> რომელსაც უკვე დიდი ხანია ვიყენებთ HTML-ში და მისი მხარდაჭერა HTML-ის მარკირების ენის ძველ ვერსიებსაც კი ჰქონდათ.

FLASH ფორმატის ფაილის შემოტანა დოკუმენტში ტაგი <embed> ის მეშვეობით

```
<!DOCTYPE html>
<html>
<body>

<embed width="400" height="50" src="flashfile.swf">

</body>
</html>
```

HTML ფორმატის ფაილის შემოტანა დოკუმენტში ტაგი <embed> ის მეშვეობით

```
<!DOCTYPE html>
<html>
<body>

<embed width="100%" height="500px" src="snippet.html">

</body>
</html>
```

გრაფიკული ფორმატის ფაილის შემოტანა დოკუმენტში ტაგი <embed> ის მეშვეობით

```
<!DOCTYPE html>
<html>
<body>

<embed src="audi.jpeg">

</body>
</html>
```

თავი 4.საიტის ინტერაქტივისა და ეფექტების გამოყენება - java script

4.1 ბიბლიოთეკის კოდის სტრუქტურის გაცნობა

მიმდინარე პარაგრაფის თემატიკა

- თანამედროვე ბიბლიოთეკები
- ბიბლიოთეკების გამოყენების მიზანი
- შესატყვისი რედაქტირება და მათი დადებითი მხარეები
- კოდის სტრუქტურა

4.2 მოძიებული ბიბლიოთეკის ინტეგრაცია ვებგვერდთან

მიმდინარე პარაგრაფის თემატიკა

- ბიბლიოთეკის დამაკავშირებელი მისამართები
- ბიბლიოთეკასთან თანდართული ფაილები
- html ელემენტების მზა ჩონჩხი
- სტილური ფაილი

4.3 მზა კოდში ცვლილებების შეტანა

მიმდინარე პარაგრაფის თემატიკა

- ტექსტის ცვლილება
- სურათის პარამეტრები
- ბმულის მისამართები
- ელემენტების ზომის პარამეტრები
- ეფექტების სიჩქარე
- ეფექტის ტრაექტორია

თავი 5.საიტის ინტერაქტივისა და ეფექტების შემუშავება - JavaScript

Javascript მსოფლიოს ყურადღების ცენტრში 1995 წლიდან მოექცა. მისი მეშვეობით შესაძლებელი იყო ვებ გვერდზე პროგრამული ნაწილის განთავსება. საწყის ეტაპზე იგი აღიქმებოდა მხოლოდ ბრაუზერ Netscape Navigator_ში.

Javascript ენაზე მოთხოვნის სწრაფმა ზრდამ გამოიწვია უმოკლეს დროში მისი ინტეგრაცია წამყვან ბრაუზერებში. ამ პროგრამირების ენამ საშუალება მისცა დეველოპერებს შეექმნათ ვებ საიტებისთვის სხვადასხვა პროგრამული დანამატები (მაგ. რუკები), მაგრამ ძირითადად იგი გამოიყენებოდა ვებ გვერდზე ინტერაქციისა და სხვადასხვა ეფექტების შესამუშავებლად.

გასათვალისწინებელია, რომ Javascript პროგრამიურების ენას არავითარი საერთო (წარმოების თვალსაზრისით) არ გააჩნია JAVA პროგრამირებასთან. ხშირ შემთხვევაში იგი აღიქმება ხან Java პროგრამირების შვილობილ ენად, ხანაც უშუალოდ java_დ.დაუშვებელია Javascript_ის java_დ მოხსენიება. ეს ორი ერთმანეთისაგან განსხვავებული პროგრამირების ენებია. უნდა აღინიშოს, რომ მათ შეუძლიათ თანადროული მუშაობა.

მას შემდეგ, რაც არსებული პროგრამირების ენის მოხმარება გასცდა Netscape Navigator და გახდა უფრო ფართოდ გამოსაყენებელი ორგანიზაცია ECMA_ს მიერ შემუშავებულ იქნა სტანდარტი ECMAScript - დოკუმენტი, რომელშიც აღწერილი იყო ენის მუშაობის პრინციპები.

საწყის პერიოდში, როდესაც ECMAScript სტანდარტი იწყებდა განვითარებას მსოფლიოში დომინირებდა პროგრამირების ენის Java. Java იმდენად პოპულარული გახლდათ - მიიჩნეოდა, რომ მომავალში ყველა პროგრამიერბის ენა ჩაიყლაპებოდა ამ გიგანტის მიერ. სწორედ არსებული სიტუაციიდან გამომდინარე ორგანიზაცია ECMA_მ განახორციელა გამართლებული მარკეტინგული სვლა და პროდუქტს უწოდა Javascript_ი (თავდაპირველი სახელწოდება არსებული **LiveScript**). ამ პერიოდიდან დოკუმენტაცია ECMAScript მოიხსენიება Javascript_ად. არსებულ ენას ECMAScript_ის სტანდარტი გააჩნია დღესაც უბრალოდ ფართო მასებისთვის ის რჩება Javascript ენად.

პროგრამების შესასრულებლად არა აქვს მნიშვნელობა, თუ რომელ პროგრამირების ენაზე ისინი დაწერილი, გამოიყენება 2 მეთოდი : **„კომპილაცია“**და **„ინტერპრეტაცია“**.

- **კომპილაცია** - ეს არის შემთხვევა როცა პროგრამის კოდს იღებას სპეციალური პროგრამა „**კომპილატორი**“ , გარდაქმნის სხვა ენად, როგორც წესი მანქანურ ენად (ნულებისა და ერთების ენად 0-1) . ეს კომპილირებული კოდი დამოუკიდებლად Source (საწყისი)კოდისგან ეშვება პროგრამის სახით.

- **ინტერპრეტაცია** - ეს არის შემთხვევა, როცა პროგრამის კოდს იღებას ინსტრუმენტი, რომელიც მოიხსენიება **„ინტერპრეტატორად“** და ასრულებს ისე როგორი სახითად მიუვიდა სკრიპტი. Javascript სწორეად ასეთი ტიპის პროგრამირების ენაა. ეს მიდგომა გამოიყენება ბრაუზერებში javascript_ის კოდი მიღებისას.

თანამედროვე ინტერპრეტატორები გარდაქმნისა და უშუალოდ გაშვების პორეცსშიც ახდენენ javascript კოდის ოპტიმიზაციას, ამიტომ Javascript მუშაობს ძალიან სწრაფად.

დღეს ფაქტობრივად ყველა ბრაუზერში ჩაშენებულია Javascript ინტერპრეტატორი, ამიტომ მისი შესაძლებელია მისი აქტიური გამოყენება ვებ გვერდებზე. **რა თქმა უნდა შესაძლებელია Javascript გამოყენება ბრაუზერის გარეშეც.**

რა შეუძლია Javascript_ს ?

Javascript_ის შესაძლებობლები დამოკიდებულია გარემოზე რომელშიც უწევს ფუნქციონირება. მაგ.: ბრაუზერში მას შეუძლია ყველაფერი რაც შეეხება ვებ გვერდის მანიპულაციას, მომხმარებელს, ურთიერთობას სერვერთან გარკვეული დოზით:

- ახალი HTML ტეგების შექმნა, არსებულის წაშლა, ტეგებისათვის სტილების გაწერა, მათი დამალვა/გამოჩენა;
- მომხმარებლების მოქმედებებზე რეაგირება, მაუსისა და კლავიატურის ხდომილებებზე (event) კონრეტული ფუნქციონალის მინიჭება;
- სერვერზე მოთხოვნების გაგზავნა, ინფორმაციის ჩატვირთვა ვებ გვერდის განახლების გარეშე (ajax ტექნოლოგია);
- Cookie_თან ურთიერთობა;

ეს მხოლოდ ნაწილია იმ შესაძლებლობისა, რომელსაც ფლობს პროგრამირების ენა Javascript.

მიმდინარე მოდულის ფარგლებში ჩვენ შევეხებით javascript პროგრამირებას, რომელიც ორიენტირებულია მომხარებლის მხარეზე, შესაბამისად მისი მოქმედების გარემო იქნება ბრაუზერი.

5.1 მარტივი ამოცანის გადაწყვეტა JavaScript ენის ძირითადი ელემენტების გამოყენებით

მიმდინარე პარაგრაფის თემატიკა

- დეკლარაციების გამოცხადება
- არითმეტიკული, ლოგიკური, ბინარული, სტრიქონული და პირობითი ოპერატორები
- შედეგის გამოტანის ოპერატორები

როგორც შესავალში აღვნიშნეთ javascript ფართო შესაძლებლობების მქონე პროგრამირების ენაა. ჩვენს შემთხვევაში ორიენტირებულნი ვიქნებით javascript_ის გამოყენებაზე ვებ გვერდში, შესაბამისად სამუშაო გარემო html დოკუმენტთან უშუალო კავშირში იქნება.

javascript კოდის არეალის გამოყოფა html დოკუმენტის სხვადასხვა ადგილასაა შესაძლებელი. გამომდინარე დავალების ინტერესებიდან ის შეიძლება გამოძახებულ იქნას:

- <head> JS </head>ტეგში;

- <body> JS </body>ტეგში;
- შესაძლებელია პროგრამული კოდის სხვა დოკუმენტში ფორმირება და შემდგომში მისი იმპორტორება **html**დოკუმენტში;

თითოეულ ამ ხერხს დაწვრილებით გზადაგზა გავარჩევთ. განვიხილავთ თუ რა დატვირთვის მატარებელია ზემოთ ჩამოთვლილი გზებით სკრიპტის ასახვა. მაგრამ საწყის ეტაპზე შევეცადოთ მოვამზადოთ Javascript არეალი <body>ტეგში. ამისათვის საჭიროა გავხსნათ და დავხუროთ <script> … </script>ტეგი. არსებულ ტეგში განთავსებული ინფორმაცია წარმოადგენს პროგრამულ კოდს.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<script>
.....
</script>

</body>
</html>
```

მიმდინარე ეტაპზე დასაშვებია <script>ტეგის უატრიბუტოდ გამოცხადება.

შესაძლებელია <script type="text/javascript">გამოყენებაც, რომელიც არც თუ ისე შორეულ წარსულში აქტიურად გამოიყენებოდა. დეველოპერთა დიდი ნაწილი არსებულ ჩანაწერს დღესაც ინტენსიურად იყენებს.

შესაძლოა რომელიმე საიტთან შეხებისას <script language="JAVASCRIPT">გახსნის შემდეგნაირი ჩანაწერი აღმოაჩინოთ. ის საკმაოდ ძველმოდურია და ფაქტობრივად აღარსად გამოიყენება. თუ მეორე ჩანაწერზე აღვნიშნეთ რომ ჯერ კიდევ აქტიურად მოიხმარება, ამ ტიპის ჩანაწერი არც თუ ისე დადებითად დაგახასიათებთ - ის არავალიდურია.

შესაძლოა Javascript_ით გაფორმებულ დოკუმენტში თვალი მოკრათ მსგავსი ტიპის ჩანაწერს

```
<script type="text/javascript"><!--
   ...
//--></script>
```

სადაც <script>გამხსნელ - დამხურავ ტეგებს შორის მოთავსებულია html კომენტარები. ეს მეთოდი გამოყენებოდა იმისათვის, რომ თუ ბრაუზერს არ ექნებოდა Javascript_ის მხარდაჭერა მასში არსებული კოდი არ გამოისახებოდა ეკრანზე ის ავტომატურად გადაიქცეოდა კომენტარად.

შესაბამისად დღევანდელ გარემოში არსებულ ყველა ბრაუზერს აქვს Javascript_ის მხარდაჭერა და მიმდინარე კოდის გამოყენება აღარაა აქტუალური.

javascript_შიინფორმაციისგამოტანისრამოდენიმემეთოდიარსებობს. მაგ.:

- alert('შეტყობინება 1')- ინფორმაციისგამოტანადიალოგურიფანჯარისმეშვეობით;
- document.write('შეტყობინება 2') - ინფორმაციის გამოტანა ვებ გვერდის შემცველობაში;
- console.log('შეტყობინება 3') - ინფორმაციის გამოტანა კონსოლში;

სანამ უშუალოდ სათითაო მეთოდს გავარჩევთ ზოგადად ავხსნათ მეთოდის არსი. მეთოდი იგივე ფუნქციაა. მისი მოხსენიება ორივენაირად შეიძლება. ჩემს მიერ გამოყენებული იქნება ორივე მიმართება და იმედია ეს არ გამოიწვევს რაიმე გაურკვევლობას.

პროგრამირებაში ამომავალი წერტილი **ობიექტია**. ნებისმიერი მოქმედება დაკავშირებულია კონკრეტულ ობიექტთან. ობიექტებზე ვრცლად შემდეგ ქვეთავებში ვისაუბრებთ ახლა მხოლოდ მცირეოდენი რაკურსით შემოვიფარგლოთ, რათა გაგიადვილდეთ ფუნქციის/მეთოდის არსის გაგება და არსებული ჩანაწერების დაზეპირება გაუაზრებლად არ მოახდინოთ.

მაგ.: გვაქვს ობიეტი ტელეფონი. მას გააჩნია **თვისებები** და **მეთოდები**. თვისებებში შეიძლება ვიგულისმოთ წონა, ფერი, მეხსიერების ზომა და ა.შ. აი მეთოდები წარმოადგენს მის შესაძლებლობებს. ვთქვათ ტელეფონის მეშვეობით შესაძლებელია ფოტოსურათის გადაღება, შეტყობინების გაგზავნა.

რომ შევაჯამოდ ეს არც თუ ისე ვრცელი აბზაცი მივიღებთ შემდეგს:

- ობიექტის **თვისების** გასაგებად თქვენ სთხოვთ ობიექტს გამოგიტანოს/გიჩვენოს უკვე არსებული თვისება;მაგ.: გამოსახოს ეკრანზე ტელეფონის წონა.
- ობიექტის **მეთოდის** გამოყენებით შენ ზემოქმედებ ამ კონკრეტულ ობიეტზე და ახდენ რაიმე ტიპის მანიპულაციას; მაგ.: გაგზავნოს შეტყობინება თქვენს მიერ აკრეფილი ტექსტით.

შესაძლოა ეს ყოველივე ამ ეტაპზე სრულად გასაგები არ იყოს, ასევე არ წარმოადგენდეს იდეალური ახსნა ობიექტის თვისებასა და მეთოდს შორის განსხვავებისა, მაგრამ იქება მცირეოდენი ბიძგი მომავალში ყველაფრის უკეთ აღსაქმელად.

თვისებასა და **მეთოდს** გააჩნია ერთი ფუნდამენტალური პრინციპი ორივე მიეკუთვნება კონკრეტულ ობიექტს. კუთვნილება გამოისახება - წერიტილით **„ . “** ;

ასევე ყველა მეთოდი გამოსახება ფრჩხილებით-**( )**; ამ ეტაპზე არ ვეხებით თვისებებს და მათ გამოსახვას.

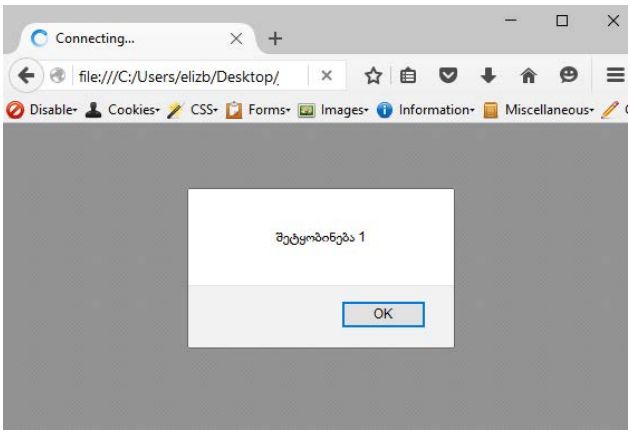
ჩვენს მიერ ჩამოთვლილ გამოტანის ოპერატორებზე ყურადღების გამახვილება კიდევ ერთხელ შეიძლება - **document.write**(), **console.log**() **,** **alert**() **;** როგორც თეორიული მასალიდან გავიგეთ პირობა:

- „ყველა მეთოდი აისახება ფრჩხილებით“ - ჭეშმარიტია.

- „მეთოდი მიეკუთვნება ობიექტს და გამოიყოფა წერტილით“ - პირველ ორ შემთხვევაში პირნათლად სრულდება. ანუ:
    - **document** - წარმოადგენს **ობიექტს** და შესაბამისად**write**() მისი ერითი-ერთი რიგირთი მეთოდია - რომელიც პასუხისმგებელია ინფორმაციის გამოტანაზე ვებ გვერდის ტანში;
    - **console** -ასევე წარმოადგენს **ობიექტს** და მისიერთ-ერთიმეთოდია **log**()- ინფორმაციის გამოტანა კონსოლში;
    - რაც შეეხება **alert()** მეთოდს ჩანაწერში არ ფიგურირებს ობიექტი და მასზე მიმაგრება წერტილის მეშვეობით. alert() მეთოდის სრყლყოფილი ჩანაწერი შემდეგნაირია**window.alert();**ობიექტ window_ს მომავალში დაწვრილებით შევეხებით ამ ეტაპზე არსებული ჩანაწერით წარმოგიდგინეთ, რომ alert() არაფრით განსხვავდება ზემოხსენებული მეთოდებისაგან და ისიც ობიექტის მეთოდია უბრალოდ მასი შემოკლებული ჩანაწერი დასაშვებია (ამ დეტალსაც შემდგომში გავარჩევთ);

Javascript რეგისტრზე დამოკიდებული ენაა შესაბამისად **Alert(), ALERT()** და ა.შარსებული მეთოდების სხვადასხვაგვარი ინტერპრეტაციით დაწერა არ გამოირანს სასუველ შედეგს. არსებული მეთდები გამოიყენეთ მხოლოდ ასოთა იმ რეგისტრით, როგორიც მოცემულია მაგალითად.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

<script>
        alert('შეტყობინება 1');
</script>

</body>
</html>
```



5.1 alert() დიალოგური ფანჯარა

დავუბრუნდეთ ჩვენს მიერ უკვე აღწერილ მეთოდ alert() _ ს და ვნახოთ მისი გამოძახებით მიღებული შედეგი. (იხ.სურ.5.1)

გასათვალისწინებელია, რომ დიალოგური ფანჯარა alert() თანამედროვე ბრაუზერში გასხვავებული დიზაინით და მდებარეობით გამოიხატება.

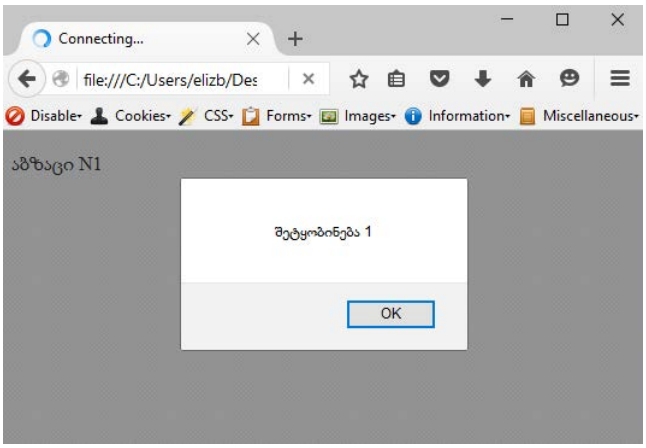
ყურადღება გაამახვილეთ alert() მეთოდში ჩაწერილ სიტყვაზე - მას ფუნქციის **არგუმენტი** ეწოდება. მიმდინარე ეტაპზე მისი დამახსოვრება დიდ მნიშვნელობას არ წარმოადგენს, რეალურ არსს ფუნქციების განხილვისას გაეცნობით.

როგორც ატყობთ ტექსტი ბრჭყალებშია მოთავსებული. დასაშვებია როგორც ერთმაგი ასევე ორმაგი ბრჭყალების გამოყენება. მაგ.: 'შეტყობინება 1' და "შეტყობინება 1" ორივე მართებული ჩანაწერია. თუ მხოლოდ რიცხვითი მნიშვნელობა გამოსატანი დასავებია ბრჭყალების გარეშე მითითება. მაგ: **alert(150);**

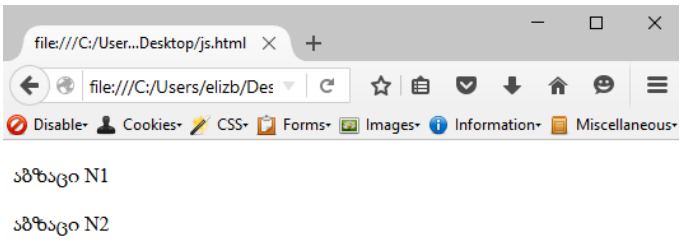
alert() წარმოადგენს მეთოდს, რომელიც იწვევს html კოდის ჩატვირთის შეყოვნებას.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>აბზაცი N1 </p>
<script>
            alert('შეტყობინება 1');
</script>
<p>აბზაცი N2</p>
</body>
</html>
```

კოდის მართლზომიერი მუშაობის შედეგად მონიტორზე გამოისახე alert() დიალოგური ფანჯარის მეშვეობით 'შეტყობინება 1'. თუ უკან ფონს დაუკვირდებით ბრაუზერის <body>ტეგში გამოსახულია წარწერა **„აბზაცი N1“** (იხ.სურ. 1.2.1),შესაბამისად **„აბზაცი N2“** არსად ფიგურირებას. ასევე ბრაუზერის ფანჯრის ზედა title ველში ჩატვირთვის ნიშანი აქტიურია, რაც მიუთთებს რომ ვებ გვერდის ჩატვირთვა არ დასრულებულა. მხოლოდ alert() ფანჯრის **„OK“**ღილაკით დადასტურება გამოიწვევს დანარჩენი კოდის ჩატვირთვას შემდგომ alert() ფუნქციამდე, თუ იგი კიდევ სადმე იარსებებს მიმდინარე ვებ გვერდზე. (გამომდინარე იქიდან, რომ ჩვენს შემთხვევაში გამოყენებულია მხოლოდ ერთი alert() ვებ გვერდი ბოლომდე ჩაიტვირთება). გამოჩნდება ტექსტი **„აბზაცი N2“.** (იხ.სურ. 1.2.2)



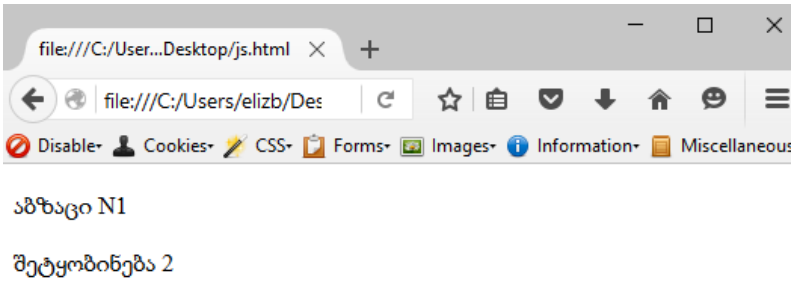
1.2.1 კოდის ჩატვირთვის წყვეტა alert() მეთოდის მიერ



1.2.2 alert() მეთოდის დასტურის შედეგად მიღებული შედეგი

alert() მეთოდისაგან განსხვავებით document.write('შეტყობინება 2') არ წარმოადგენს მცურავ ფანჯარას, მას გამოაქვს ინფორმაცია ბეჭდური სახით, იმ კონკრეტულ სტრიქონზე, რომელზეც ის არის განთავსებული.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>აბზაცი N1 </p>
<script>
        document.write('შეტყობინება 2');
</script>
</body>
</html>
```



დავალების შესაბამისად „აბზაცი N1“ ფორმირდება ტეგის მეშვეობით, ხოლო „შეტყობინება 2“_ის გამოტანა ხორციელდება document.write() მეთოდის დახმარებით (იხ.სურ.1.3)

1.3document.write() ფუნქციის გამოტანილი „შეტყობინება 2“

Javascript სინტაქსი ხაზოვანი ტიპის კითხვადობის რეჟიმის მიმდევარია. თითოეული პროგრამული სტრიქონის ბოლოს რეკომედირებულია წერტილმძიმის - „ ; “ გამოყენება, რაც სტრიქონის დასასრულის აღმნიშვნელია. (სხვადასხვა პროოგრამირების ენაში - მაგ. php, c# „;“ გამოყენება სავალდებულოა). სტრიქონში ედიტორის ერთი კონკრეტული ხაზზე განთავსებული ინფორმაცია არ მოაზრება, პროგრამირებაში სტრიქონად ერთი უწყვეტი კოდი იგულისხმება. იხილეთ მოცემული მაგალითი

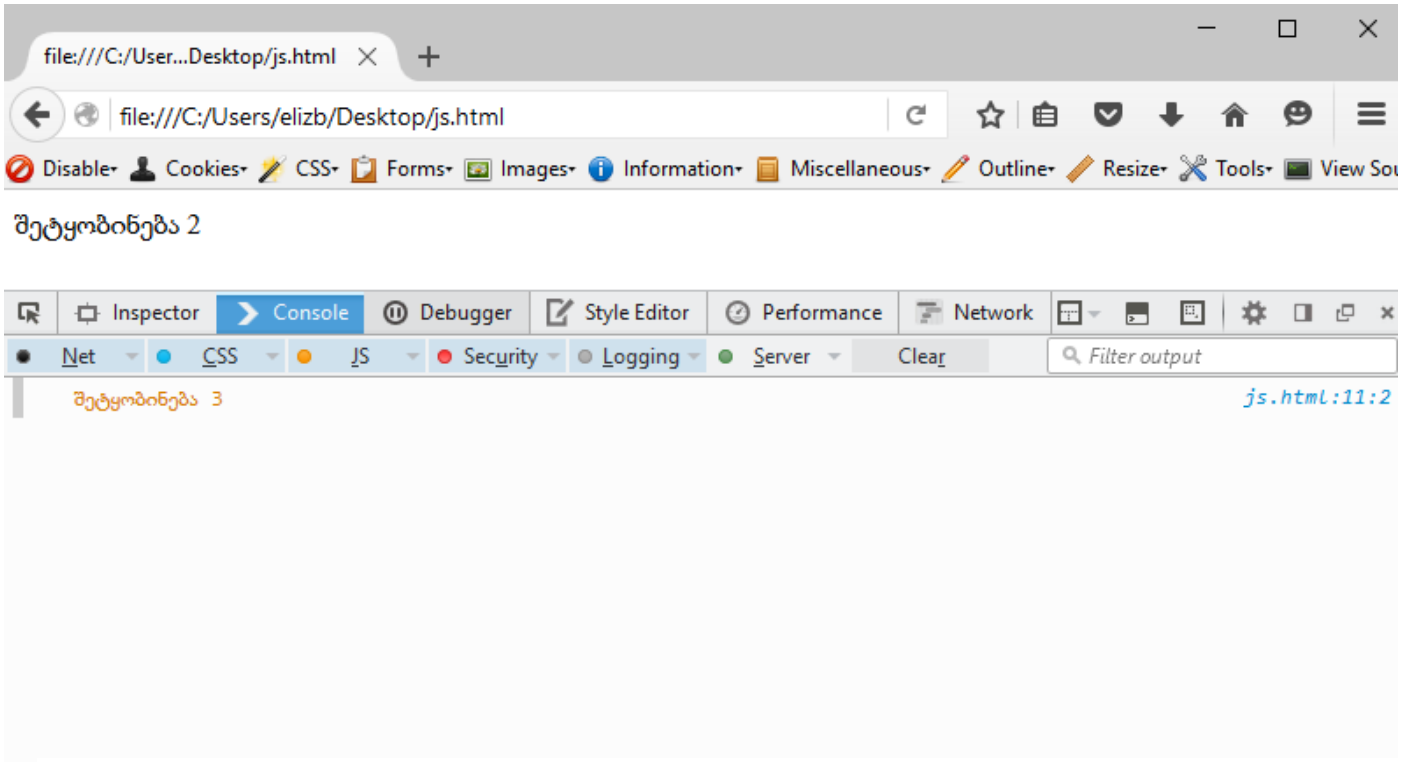
```
1. <script>
2.       alert('შეტყობინება 1');
3.       document.write(' აქ განთავსდება ნებისმიერი სიგრძის ინფორმაციული ტექსტი და ის აღიქმება
         ერთ სტრიონად მიუხედავ იმისა ედიტორში რამოდენ ხაზზეა იგი განლაგებული');
4.</script>
```

რეკომენდირებულია ასევე სხვადასხვა აზრობრივი კოდის სხვადასხვა ხაზზე განთავსება. მიუხედავად იმისა, რომ მაგალითზე ნაჩვენები კოდი უპრობლემოდ იმუშავებს უმჯობესია alert() ფუნქციები განთავსდეს სხვადასხვა სტრიქონზე.

```
<script>
        alert('შეტყობინება 1'); alert('შეტყობინება 2');
</script>
```

```
<script>
        alert('შეტყობინება 1');
        alert('შეტყობინება 2');
</script>
```

console.log()_ ის მეშვეობით გამოტანილი ინფორმაცია გამოისახება ბრაუზერის კონსოლში, რომლის გამოძახებაც შესაძლებელია **ბრაუზერზე მაუსის მარჯვენა კლიკი >Inspect Element**ან **F12**_ის მეშვეობით. (იხ.სურ.1.3.1). რა უპირატესობა გააჩნია კონსოლს თუ ის ხილვადია მხოლოდ ვებში გათვიცნობიერებული ადამიანისათვის. ვებ დეველოპერების უდიდესი ნაწილი მოიხმარს კონსოლს.



1.3.1 კონსოლის ვიზუალური გამოხატულება

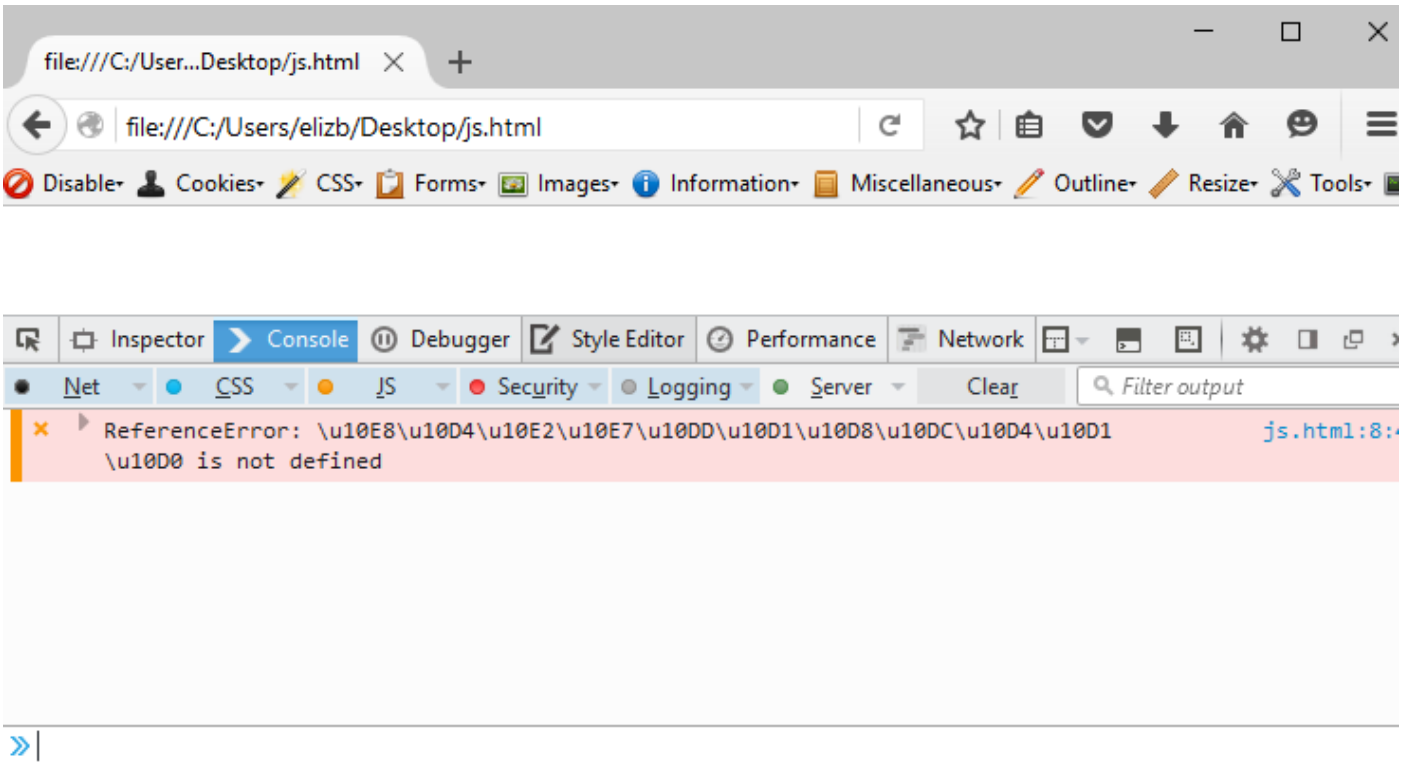
რამეთუ კეთების პროცესში პროგრამისტი არ არის დაზღვეული შეცდომებისაგან. კონსოლი საშუალებას გაძლევთ მარტივად დაადგინოთ შეცდომის ტიპი და მდებარეობა მინიშნებების საშუალებით.

```
<script>
        console.log('შეტყობინება 3');
</script>
```

სურათ 1.3_ზე მოცემულია კონსოლი სადაც წინასწარ მაგალითზე ნაჩვენები console.log() მეთოდის მეშვეობით აისახა შედეგი.

დავუშვათ განზრახ შეცდომა სკრიპტში, რათა მოვახდინოთ კონსოლის რეაგირების დემონსტრება. console.log() მეთოდში გადაცემული არგუმენტი შეტყობინება 3 ბრჭყალების გარეშეა. როგორც აღვნიშნეთ ეს უპატიებელი შეცდომაა. ვნახოთ კონსოლის რეაგირება (იხ.სურ.1.3.2)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<script>
        console.log(შეტყობინება 3);
</script>
</body>
</html>
```



1.3.1. კონსოლის რეაირება შეცდომაზე

ძირითად ნაწილში განსაზღვრულია შეცდომის ტიპი, ხოლო მარჯვენა კიდეში ასახულია სტრიქონის ნომერი, თუ რომელ ხაზზეა შეცდომა დაშვებული js.html:8

კომენტარები

Javascript_ში ისევე როგორც თქვენს მიერ აქამდე შესწავლილ სხვა ენებში (html, css) აქტიურად გამოიყენება კომენატერბის სისტემა.

დიდი კოდის არსებობის შემთხვევაში ბევრი რამ ბუნდოვანი ხდება მისი შემდგომი გარჩევისას. ამიტომ ხშირად გამოიყენებენ კომენტარების სისტემას, რომელიც მიანიშნებს კონკრეტული სტრიქონის ან ლოგიკური ბლოკის დანიშნულებას.

კომენატრები შესაძლოა გამოყენებულ იქნას სკრიპტის ნებისმიერ ადგილას და ის არანაირ უარყოფით ზეგავლენას არ ახდებს კოდის მუშაობაზე, პირიქით დადებით ტონად გასდევს დეველოპერის ნამუშევარს.

ერთხაზიანი კომენტარები აღინიშნება **//** სიმბოლოთი და ზემოქმედებს მხოლოდ 1 სტრიქონზე არსებულ ინფორმაციაზე. დაიმახსოვრეთ: ერთ ხაზში მოაზრება უწყეტი ტექსტი, რომელიც ედიტორში შესაძლოა წარმოდგენილი იყოს რამოდენიმე სტრიქონადაც. ახალი სტრიქონის დასაწყისს უზრუნველყოფს კლავისი **ENTER** აქტივაცია.

```
 // არსებულ მეთოდს გამოაქვს დიალოგური ფანჯარა
alert('შეტყობინება 1');
alert('შეტყობინება 2');// მიღებულია კომენტარის ამ ადგილას განთავსებაც
```

არსებობს ასევე **მრავალზახიანი** კომენტარი, რომელსაც შესაბამისად გააჩნია გამხსენი (**/***) და დამხურავი (***/**)სიმბოლოების ერთობლიობა. მოცემულ მაგალითზე შეტყობინება 2 და 3 არ გამოისახება.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<script>
            /*
            document.write('შეტყობინება 2');
            document.write('შეტყობინება 3');
            */
            document.write('შეტყობინება 4');
</script>
</body>
</html>
```

ცვლადები და მათი ტიპები

ისევე, როგორც პროგრამირების დანარჩენი ენებში Javascript_შიც ინტენსიურად გამოიყენება ტერმინი **ცვლადები.**

რას წარმოადგენს ცვლადი?

ცვლადი არის პირობითად კონტეინერი, რომელსაც შეუძლია შეინახოს გარკვეული მონაცემი. ყოველ ცვლადს გააჩნია **სახელი**, ხოლო შენახულ მონაცემს ცვლადის **მნიშვნელობა** ეწოდება.

ახლი ცვლადის გამოსაცხადებლად javascript_ში გამოიყენება საკვანძო სიტყვა var.ის იშფრება როგორც **variable** - ცვლადი.

როგორც აღვნიშნეთ ცვლადს გააჩნია სახელი. Javascript_ში ცვლადის გამოცხადებისას ყურადღება უნდა გაამახვილოთ ცვლადის სახელებზე, რადგანაც სახელი შეიძლება შედგებოდეს მხოლოდ: **სიმბოლოების $ და _ , ასოების, ციფრებისაგან.** ამასთანავე დაუშვებელია ცვლადის სახელის დაწყება ციფრით.

მაგ. I_ზე მოყვანილია ცვლადის გამოცხადების სწორი გზა, ხოლო მაგ. II მცდარი.

მაგ. I

```
// არსებული მეთოდით ცვლადების გამოცხადება სწორია

   var myName;
  var test123;
var $;
  var _;
```

მაგ. II

```
// არსებული მეთოდით ცვლადების გამოცხადება მცდარია

  var 123test;  // ცვლადის სახელის რიცხვით დაწყება არ შეიძლება;
var my-name;   //  არ შეიძლება სიმბოლო ‘ - ’  ტირეს გამოყენება მხოლოდ ქვედატირე;
```

რაც შეეხება ცვლადების სახელების მიმართ გამოთქმულ რეკომენდაციებს მდგომარეობს შემდომში:

- გამოიყენეთ მხოლოდ ლათინური ანბანის ასოები;
- უმჯობესია თუ მოერიდებით რამოდენიმე სიტვიან ცვლადებ ან მათი გამოყენებისას მაქსიმალურად შეზღუდეთთუნდაც დასაშვები სიმბოლო „_“. მაგ. უმჯობესია გამოიყენოთ myFirstName ვიდრე my_first_name;

- მაქსიმალურად მიაახლოვეთ ცვლადის სახელი მასში შესანახ მნიშვლობის არსთან;

Javascript_ში არსებობს რეზერვირებული სიტყვები, რომელთა სახელებად გამოყენება ასევე დაუშვებელია. ცხრილ I_ში გამოსახულია არსებულ სიტყვათა ნუსხა და პრობლემების თავიდან ასაცილებად შეეცადეთ არ გამოიყენოთ ჩამონათვლიდან არც ერთი. ამ სიტყვების ერთიანად დაზეპირება სავალდებულო არ არის. გზადაგზა უმეტეს მათგანს შეეხებით და დაიმახსოვრებთ. შესაძლოა ასევე მათი ინტერნეტში მოძიება შემდეგი ტექსტის მეშვეობით - **„Javascript reserved words“**

ცხრილი I			
abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

როგორ ზედა ქვეთავში იყო განხილული ცვლადი შედგება 2 ძირითი ნაწილისაგან. ჩვენს მიერ გარჩეულ იქნა ცვლადის სახელი. ახლა გადავიდეთ ცვლადის **მნიშვნელობაზე**.

მაგალითად ისეთ პორგამირების ენებში, როგორებიცაა : C++, C# ცვლადის გამოცხადებისას ხდება მისი ტიპის განსაზღვრა და მასში მხოლოდ მითითებული ტიპის მონაცემების ჩალაგებაა შესაძლებელი. იხ. მაგალითი

```
int n;  // გამოცხადებული ცვლადი n შეიძლება შეიცავდეს მხოლოდ  მთელ რიცხვებს
 string txt;  // ცვლადი txt შეიძლება შეიცავდეს მხოლოდ ტექსტუალურ მნიშვნელობას

// და ა.შ;
```

Javascript_ში არ გვხვდება მკაცრად გამოხატული ტიპიზაცია. რას ნიშნავს ეს ყოველივე ?!

ეს ნიშნავს შემდგომს - Javascript_ში შესაძლებელია ერთი და იგივე ცვლადისათვის სხვადასხვა ეტაპზე სხვადასხვა ტიპის მნიშვნელობის მინიჭება, ისე რომ არავითარი „ტიპზე დაყვანის ოპერაციები“ არ არის საჭირო, ის ავტომატურად გარდაიქმნება იმ ტიპის ცვლადად რა ინფორმაციასაც მიიღებს.

ზოგადადპროგრამირების ენები, რომლებიც არ ხასიათდება მკაცრად გამოხატული ტიპიზაციით მაგ. Javascript, php ამ ამოცანას აკისრებენ ინტერპრეტატორს, ამიტომაც Javascript_ში მათ გამოსაცხადებლად საკმარისია მხოლოდ უკვე ნახსენები საკვანძო სიტყვა var გამოყენება.

```
var myName;        // მისთვის შესაძლებელია ნებისმიერი ტიპის მნიშვნელობის მინიჭება
```

პროგრამირების ენებში არსებობს სხვადასხვა ტიპის ოპერატორები: **არითმეტიკული,ლოგიკური,ბინარული,სტრიქონული,პირობითი.** მათ სრულყოფილად ორიოდ წუთში მივუბრუნდებით ამ ეტაპზე კი მხოლოდ მინიჭების ოპერატორ **„=“** შევეხოთ. დაიმახსოვრეთ ეს ნიშანი ‘=’ პოგრამირებაში არ გამოხატავს „ტოლობას“ ის არის მინიჭება.

ცვლადში დროებითი მნიშვნელობის განსათავსებლაბლად საჭიროა დავწეროთ მაგალითის შესაბამისი გამოსახულება;

```
var myCar = 'Ford Mustang GT 500 Eleanor';

var price = 550000;
```

ყურადღება გაამახვილეთ შემდეგზე: ცვლადი myCarარ უდრის, არ არის ტოლი, მას უბრალოდ ენიჭება'Ford Mustang GT 500 Eleanor' მნიშვნელობა. თანდათან უფრო ცხადი წარმოდგენა შეგექმნებათ მინიჭების ოპერატორზე.

ახლა უშუალოდ რაც შეეხება ამ კონკრეტულ ჩანაწერთა ერთობლიობას. ერთი შეხედვით სხვაობა არ არის არანაირი უბრალოდ myCar_ის მნიშვნელობა მოთავსებულია ბრჭყალებში, ხოლო price_ის არა. ეს მათი ტიპების დამსახურება.

Javascript_ში ტიპებიდაყოფილია 2 ჯგუფად:

1. პრიმიტიული ანუ მარტივი ტიპები:
    a. **String**
    b. **Number**
    c. **Boolean**
    d. **Null**
    e. **Undefined**
2. შედგენილი ტიპი - **Object;**

Nullდა **Undefined** ტიპებს ხშირად **სპეციალურ ტიპებადაც** მოიხსენიებენ.

გავეცნოთ პრიმიტიულ ტიპებს დაწვრილებით, ხოლო **Object**გადავდოთ მომავალისათვის.

String - ტექსტური ტიპი

ტექსტური ტიპი აერთიანებს ყველა სახის ტექსტს. კანონია ყველა ტექსტური ტიპის მნიშვნელობა უნჯა იჯდეს **ერთმაგ** ან **ორმაგ** ბრჭყალებში.Javascript_არ არის სხვაობა არსებულ ბრჭყალებს შორის. იხ. მაგ.

```
var txt ="აქ განთავსდება ტექსტური მნისჰვნელობა";     // ორმაგი ბრჭყალი
txt ='აქ განთავსდება ტექსტური მნისჰვნელობა';         // ერთმაგი  ბრჭყალი
```

Number - რიცხვითი ტიპი

რიცხვითი ტიპი თავის თავში აერთიანებს, როგორც **მთელ,** ასევე **ათწილად** რიცხვებს. არსებობს ასევე სპეციალური რიცხვითი ტიპები -**Infinity** , +**Infinity** , **NaN**.

```
var n =123;
n =12.345;
```

Infinity წარმოადგენს უსასრულობას;

```
// მაგ.:  1_ის  ნულზე გაყოფის შედეგად მიიღება უსასრულობა;

alert(1/0);// +Infinity
alert(-1/0);// -Infinity
```

NaN - იშიფრება როგროც - Not a Number, ანუ რიცხვითი მნიშვნელობა, რომელიც არ უდრის არც ერთ სხვა რიცხვს, ასევე არ უდრის თავის თავსაც. შესაძლოა ცოტა გაუგებარია?! ეს არის ტიპი რომელიც რამენაირ კავშირშია რიცხვთან მაგრამ ამავდროულად არ არის რიცხვი. მაგ.: რიცხვისა და ტექსტის არითმეტიკული ნამრავლი არის **NaN,** ასევე მსგავსი ტიპი მიიღება კვადრატული ფესვი -1 დან და ა.შ.

```
alert("ტექსტი"*15);// NaN
```

Boolean - ბულის ანუ ლოგიკური ტიპი

ის თავის თავში აერთიანებს მხოლოდ 2 მნიშვნელობას **true -** ჭეშმარიტია , **false** - მცდარია;

```
var checked =true;// მიმდინარე სტრიქონზე ცვლადი ჭეშმარიტია
checked =false;// მიმდინარე სტრიქონზე ცვლადი მცდარია
```

Null - მნიშვნელობა, რომელიც არ არსებობს

არსებული ტიპი არ განეკუთვნება არც ერთ ტიპს. ზოგადა მიჩნეულია, რომ null წარმოადგენს ობიექტის ტიპს და მიუთითებს ობიექტის არ არსებობას.. დაიმახსოვრეთ null არ არის 0. ის არის შემთხვევა როცა კონკრეტული ცვლადი უდრის არაფერის;

```
var age =null;
```

Undefined - მნიშვნელობა განუსაზღვრელია

არსებული ტიპი მხოლოდ ერთ მნიშვნელობას შეიცავს და ეს არს „მნიშვნელობა განუსაზღვრელია“.
თუ მოვახდენთ ცვლადის დეკლალირებას მაგრამ მას არ მივანიჭებთ საწყის ეტაპზე არაფერს არ წარმოადგენს არანაირ პრობლემას. უბრალოდ თუ შევეცდებით მის ეკრანზე გამოტანას მნიშვნელობად გვიჩვენებს undefined

```
var x;
alert( x );// გამოიტანს "undefined"
```

არსეობსსპეციალურიოპერატორი და ფუნქცია**typeof,** რომელსაცშეუძლიამონაცემისტიპისდადგენადამისიჩანაწერიშემდეგნაირია:

- ოპერატორის სინტაქსი **typeofx;**
- ფუნქციის სინტაქსი **typeof(x);**

ორივე შემთხვეაში **typeof**მუშაობს იდენტურად. გამოიყენეთ რომელიც გსურთ

```
alert(typeof undefined)// "undefined"
```

```
alert(typeof0)// "number"

alert(typeoftrue)// "boolean"

alert(typeof"foo")// "string"

alert(typeofnull)// "object"
```

შესაძლოა კონკრეტული მნიშვნელობის მაგირად გადავაწოდოთ ცვლადად

```
var x = 2.25;

alert(typeof x )// " number "
```

ცვლადის გამოცხადების გზები

ჩვენს მიერ განხილულ იქნა ცვლადის სახელის შერჩევა, var საკვანძო სიტყვა, ცვლადის მნიშვნელობა შესაბამისად შეგვიძლია ცვლადის გამოცხადების გზებზე საუბარი.

შესაძლებელია ცვლადის გამოცხადება მნიშვნელობის გარეშე და ასევე მასთან ერთად

```
var x;                  // ცვლადის გამოცხადება მნიშვნელობის გარეშე
var y;                  // ცვლადის გამოცხადება მნიშვნელობის გარეშე

var age = 15;                              // ცვლადის გამოცხადება - მნიშვნელობის მინიჭებით
var city =  "თბილისი";          // ცვლადის გამოცხადება - მნიშვნელობის მინიჭებით
```

შესაძლებელია ასევე ცვლადების ჩამოთვლა და მათი ერთდოული დეკლარირება 1 var სიტყვის საშუალებით, გამყოფებად კი გამოიყენება მძიმე“,” .

```
var x,      age = 15, city =  "თბილისი";
```

მრავალჯერ აღნიშნული მიუხედავად კვლავ დავკონკრეტდები, რომ Javascript ხაზოვანი კითხვადობის მიმდევარია.

```
varage = 15;          // ცვლადისთვის პირველადი დეკლარირება;

alert(age);           // შედეგი ამ ეტაპზე 15

age = 103;// ცვლადისთვის ყოველ შემდეგ ჯერზე მნიშვნელობის მინიჭებისას სიტყვა varაღარ არის საჭირო, არსებული სტრიქონის ქვემოთ გამოყენებული ცვლადის მნიშვნელობა ყველგან იქნება 103. ახალი მნიშვნელობის მინიჭებამდე

alert(age);           // შედეგი ამ ეტაპზე  103
```

არითმეტიკული ოპერატორები

არითმეტიკული მოქმედებები გამოიყენება რიცვითი ტიპის ცვლადებზე. ცხრილ II მოცემულია Javas-cript_ში არსებული არითმეტიკული მოქმედებები.

ცხრილი II

ოპერატორი	აღწერა	მაგალითი	y მნიშვნელობა	შედეგი
+	მიმატება	x = y + 2	y = 5	x = 7
-	გამოკლება	x = y - 2	y = 5	x = 3
*	გამრავლება	x = y * 2	y = 5	x = 10
/	გაყოფა	x = y / 2	y = 5	x = 2.5
%	მოდულით გაყოფა	x = y % 2	y = 5	x = 1
++	ინკრიმენტი	x = ++y	y = 5	x = 6
		x = y++	y = 5	x = 5
--	დეკრიმენტი	x = --y	y = 5	x = 4
		x = y--	y = 5	x = 5

პირველი ოღხი ოპერატორი არ საჭიროებს დამატებით განმარტებას, მათთან შეხება სკოლაში საწყისი ასაკიდანვე გიწევდათ.

რაც შეეხება **%** ნიშანს მისი მოქმედება ოდნავ განსხვავებულია და ბევრისთვის შეიძლება გამოცანად რჩება ტერმინი მოდულით გაყოფა. მარტივათ, რომ ვთქვათ მოდულით გაყოფისას შედეგი მიიღება ნაშთი. თუ პასუხად მოგვცა 0 ე.ი გასაყოფი გამყოფზე იყოფა უნაშთოდ.

```
varx = 19 %4;          // 3 - ნაშთი
x = 20 %4;             // 0  და არა 5 რადგანაც გაყოფა მოხდა უნაშთოდ
```

ინკრიმენტი და დეკრიმენტი პროგრამირებაში ფართოდ გამოყენებადი ოპერატორებია. რეალურ რეჟიმში ისინი შემოკლებულ ჩანაწერს წარმოადგენენ. მაგ. ინკრიმენტი: **i++**იგივეა, რაც **i+1**, ასევე **++i** იდენტურია **i+1**ერთი არც თუ ისე უმნიშვნელო განსხვავებისა: **i++**ზრდის ცვლად i მნიშვნელობას 1_ითმხოლოდ **მომავალ სტრიქონზე,** ხოლო**++i**მიმდინარეზე, სადაც მოხდა მისი გამოყენება;

მაგ. I

```
vari = 5;   // საწყის მნიშვნელობად  i  = 5
 alert(i);// 5
 alert(i++);// ამ ეტაპზე i  ისევ 5_ია ის განიცდის მომატებას და გაზრდილ მნიშვნელობად მხოლოდ
ზემდეგი სტრიქონიდან ფიქსირდება
alert(i);// აქ უკვეi გახდა 6 და ყველა შემდეგ ხაზზე მისი მნიშვნელობა 6_ია თუ უცვლელად დავტოვებთ
```

მაგ. II

```
vari = 5;   // საწყის მნიშვნელობად  i  = 5
  alert(i);// 5
  alert(++i);// აქ უკვე i გახდა 6  და ყველა შემდეგ ხაზზე მისი მნიშვნელობა 6_ია თუ უცვლელად დავტოვებთ
alert(i);// 6
```

რაც შეეხება დეკრიმენტს ინკრიმენტის მსგავსად მუშაობს უბრალოდ ახორციელებს 1_ით **კლებას.**

მაგ. I

```
vari = 5;   // საწყის მნიშვნელობად  i  = 5
  alert(i);// 5
  alert(i--);// ამ ეტაზე i  ისევ 5_ია ის განიცდის კლებას  და დაკლებულ  მხოლოდ ზემდეგი სტრიქონიდან ფიქსირდება
alert(i);// აქ უკვეi გახდა 4 და ყველა შემდეგ ხაზზე მისი მნიშვნელობა 4_ია თუ უცვლელად დავტოვებთ
```

მაგ. II

```
vari = 5;   // საწყის მნიშვნელობად  i  = 5
  alert(i);// 5
  alert(--i);//  აქ უკვეi გახდა  4 და ყველა შემდეგ ხაზზე მისი მნიშვნელობა 4_ია თუ უცვლელად დავტოვებთ
alert(i);// 4
```

როგორც ქვეთავის დასაწყისში აღვნიშნეთ არსებული არითმეტიკული მოქმედებები იდეალურად მუშაობენ მხოლოდ ციფრებთან მიმართებაში. რაც შეეხება მათ მუშაობას ტექსტური (string) ტიპიც ცვლადებთან ცოტა განსხვავებულად მოქმედებენ.

მაგალითად თუ მოცმულია შემთხვევა

```
vari = "თბილისი"*5;             // NaN
```

ყველა არითმეტიკული მოქმედება გარდა **„+“** შედეგად დააბრუნებს NaN, რაც ცხადზე ცხადია გამომდინარე იქიდან, რომ ტექსტუალური ტიპის ინფორმაციაზე არითმეტიკული მოქმედებების გამოყენება დაუშვებელია. რაც შეეხება „+“ ორიოდ წუთში დავუბრუნდებით. მაგრამ ვთქვათ ტექსტუალური ტიპით წარმოვადგინეთ რიცხვი რა მოხდება ამ შემთხვევაში? ინტერპრეტატორი ავტომატურად მოახდენს ტიპის დაყვანას თუ ის შესაძლებელია

```
vari = "6"*5;                  //30
i = "11"-3;// 8
i = "10"/2;// 5
i = "10"/4;// 2
```

რაც შეეხება “+” ნიშანს გარდა არითმეტიკული მოქმედებისა javascript_ში ის წარმოადგენს **კონკატინაციის / მიერთების** ნიშანს. თუ ეს ნიშაში მოხვდება ტექსტუალურ მონაცემთან ის ავტომატურად მიერთებას

მოახდენს ხოლო თუ რიცხვით ტიპთან არითმეტიკულ მოქმედებას. ამიტომ საწყის ეტაპზე ხშირად იწვევს დაბნეულობას როცა არითმეტიკული შეკრება სულ სხვა შედეგით სრულდება ვიდრე ამას მოელოდით. მაგ.:

```
vari = "6"+5;              //65
i = "25"+"23"; //2523
```

როგორც ზედა მაგალითიდან გამოჩნდა „+“ ნიშნის გამოყენებით მიღებული შედეგი NaN არ არის, არამედ მოხდა ტექსტების ერთმანეთის გვერდიგვერდ დალაგება. რა თქმა უნდა შესაძლებელია ტექსტური ინფორმაციის რიცხვითში გადაყვანა სპეციალური ფუნქციების გამოყენებით, მაგრამ არსებობს ასევე საკმაოდ მარტივი გზაც, რომელიც მაგალითზეა ნაჩვენები

```
vari = +"6"+5;             //11
i = +"25"++"23";   //48
```

„+“ ნიშანი, როგორც აღვნიშნეთ კონკატინაციას წარმოადგენს და მისი გამოყენება საკმაოდ ინტენსიურად ხორციელდება. მაგ.: მოცემული ცვლადებით მიიღეთ სრულფასოვანი წინადადება და დაბეჭდეთ ეკრანზე.

```
varname= 'თედო';
vartxt = 'საოცნებომანქანა';
varmyCar = 'Ford Mustang GT 500 Eleanor';

//საოცნებო მანქნაFord Mustang GT 500 Eleanor
document.write(txt + myCar   );          //ორი ცვლადის ტექსტი წვეტის გარეშე მიეწებება ერთმანეთს

//საოცნებო მანქნა Ford Mustang GT 500 Eleanor
document.write(txt +' - '+myCar   );

//Ford Mustang GT 500 Eleanor  - საოცნებო მანქნა
document.write(myCar + ' - '+txt  );

//Ford Mustang GT 500 Eleanor ჩემი  საოცნებო მანქნაა
document.write(myCar + ' ჩემი '+txt+ ' ა' );

//თედოს საოცნებო მანქნა არის Ford Mustang GT 500 Eleanor
document.write(name + 'ს '+ txt+ ' არის '+myCar );
```

მინიჭების ოპერატორები

ზედა ნაწილში განხულულ იქნა მინიჭების ოპერატორი „=“. არსებობს სხვა ოპერატორებიც, რომლებით აერთიენებს მინიჭების და არითმეტიკულ ოპერატორებს და საკმაოდ ამარტივებს მიფგომას. ცხრილი III განხილულია შემთხვევა სადაც საწყის მნიშვნელობებად **x = 10** და**y = 5.**

ცხრილი III			
ოპერატორი	**მაგალითი**	**მსგავსი გამოსახულება**	**შედეგი**
=	x = y	x = y	x = 5
+=	x += y	x = x + y	x = 15
-=	x -= y	x = x - y	x = 5
*=	x *= y	x = x * y	x = 50
/=	x /= y	x = x / y	x = 2
%=	x %= y	x = x % y	x = 0

შედარების ოპერატორები

პროგრამირებაში დიდია შედარების ოპერატორების გამოყენების ინტენსივობა. ხშირია პროგრამის შემუშავების პროცესში პირობის წამოჭა, რომლის ჭეშმარიტების დასადგენად შედარების პრინციპის გამოყენებაა საჭირო. ნებისმიერი შედარების შედეგად მიღებული შედეგი წარმოადგენს Boolean ტიპის მნიშვნელობას: ის არის ჭეშმარიტი trueან მცდარიfalse.იხ.ცხრილი IV

ცხრილი IV			
ოპერატორი	**აღწერა**	**პირობა**	**შედეგი**
==	ტოლობა	5 == 8	false
		5 == 5	true
===	ექვივალენტურია, ანუ ტოლია როგორც მნიშვნელობით ასევე ტიპიც საერთო აქვთ	5 === "5"	false
		5 === 5	true
!=	არ უნდრის	5 != 8	true
!==	არაექვივალენტურია	5 !== "5"	true
		5 !== 5	false
>	მეტია	5 > 8	false
<	ნაკლებია	5 < 8	true
>=	მეტია ან უდრის	5 >= 8	false
<=	ნაკლებია ან უდრის	5 <= 8	true

რიცხვითი მნიშვნელობების დროს შედარების ოპერატორებზე ფაქტობრივას სასაუბრო არც არაფერი იმდენად მარტივია ყველაფერი. რაც შეეხება ტექსტური ტიპის მნიშვნელობებს არც მათი შედარებაა წარმოუდგენლად რთული. გამომდინარე იქიდან, რომ თითოეულ სიმბოლოს საკუთარი ASCII კოდი შეესაბამება სირთულეც მოხსნილია. ყურადღება გაამახვილეთ, რომ დადარების პროცესი ხდება ინდივიდუალურად თითოეული სიმბოლოს მიხედვით და ტექსტის დიდი სიგრძე ყოველთვის უპირატესობას არ წარმოადგენს. როგორც მაგალითზეა მოცემული ერთ შემთხვევაში **“a” > “aba”** მცდარია, ხოლო **“b” > “aba”** ჭეშმარიტი. სწორედაც რომ ასეა. გამომდინარე იქიდან რომ დადარება უშალოთ თითოეული სიმბოლოსი ხდება მეორე შემთხვევაში **“b”** შეედარა “**aba**” სიტყვის პირველ ასო “**a**” აღმოჩნდა მასზე მეტი და დაჯაბნა მთლიანი სიტყვა დანარჩენ ასეობზე დადარება საერთოდ აღარ ცათვალა საჭიროდ.

“a” >“aba”	false
“b” >“aba”	true
“abas” >“aba”	true

ლოგიკური ოპერატორები

ლოგიკურ ოპერატორებს საკმაოდ მნიშნველოვანი როლი ენიჭება პირობით ოპერატორებში. მათი დახმარებით დეველოპერებს ეძლევათ საშუალება პირობა უფრო დააკონკრეტონ. ლოგიკური ოპერატორები წარმოდგენილია ცხრილი V სადაც x = 6 და y = 3;

ცხრილი V

ოპერატორი	აღწერა	პირობა	შედეგი
!	უარყოფა	!(x == y)	true
&&	ლოგიკური „და“	(x < 10 && y > 1)	true
||	ლოგიკური „ან“	(x == 5 || y == 5)	false

5.2 ამოცანის გადაჭრა ძირითადი კონსტრუქციების გამოყენებით

მიმდინარე პარაგრაფის თემატიკა

- ამოცანის გადაჭრის ალგორითმის შემუშავება
- ენის ძირითად კონსტრუქციები

პირობითი ოპერატორები

პროგრამირებაში საკმაოდ დიდი როლი უკავიათ პირობით ოპერატორებს. ამოცანების უდიდეს ნაწილში პირობის დასმა და შესაბამისი ლოგიკის მიღება ფაქტობრივად წინა პლანზეა წამოწეული. Javascript_ში გვხვდება**if**და**switch** კონსტრუქციები. ამათგან if უფრო ფართოდ გამოყენებადი და ბევრისთვის საყვარელი ოპერატორია, მაგრამ არც switch_ს უნდა დავუკარგოთ ის დადებითი რაც გააჩნია. გზადაგზა უფრო ვრცლად მაგალითებზე დაყრდნობით გავეცნობით თითეულ მათგანს.

if ოპერტორის რამოდენიმე სახის ჩანაწერი არსებობს. ყველაზე გავრცელებული შემდეგი სკრიპტი

```
if( x  >10) {
alert('გამოიტანოს  შედეგი')
}
```

ნებისმიერი სახით if ოპერატორის გამოყენება, იქნება ეს ფიგურული ფრჩხილებით, მის გარეშე თუ სხვა მეთოდით უცვლელი რჩება არსი: **if ოპერატორს ჭირდება სტანდარტული ფრჩხილები (...) პირობის დასასმელად;**

```
if(პირობა) {
        ბლოკი რომელიც სრულდება პირობის დაკმაყოფილების შემთხვევაში
}
```

ბლოკში არსებული ინფორმაცია იმუშავებს მხოლოდ იმ შემთხვევაში თუ შესრულება if_ის ფრჩხილებში () დასმული პირობა, ანუ პასუხი იქნება ჭეშმარიტი - true.

როგორც დასაწყისში აღვნიშნეთ შესაძლებელია if ჩაწერა ბლოკის - ფიგურული ფრჩხილების გარეშე თუ მხოლოდ ერთი ხაზია მნიშვნელობა

```
if( 20 >10)
alert('შედეგი 1')  // ეკრანზე გამოისახება შედეგი 1
```

თუ შედეგი რამოდენიმეა მაგ.:

```
if( 20  >10)
alert('შედეგი 1')  // ეკრანზე გამოისახება შედეგი 1
alert('შედეგი 2')  // ეკრანზე გამოისახება შედეგი 2
```

მაგრამ იმ შემთხვევაში თუ if პირობა არ დაკმაყოფილდა

```
if(3  >10)
alert('შედეგი 1')  // ეკრანზე არ  გამოისახება შედეგი 1
alert('შედეგი 2')  // ეკრანზე გამოისახება მხოლოდშედეგი 2
```

დასკვნა: თუ რამოდენიმე ხაზიანი შედეგის მიღება გვესაჭიროება if პირობის შესრულების დროს ფიგურული { ... } ფრჩხილების გარეშე არსებული შედეგი არ მიიღება.

ზოდად რეკომენდაციის სახით: უნჯობესია ყოველთვის გამოიყენოთ ფიგურული ფრჩხილები. ის უფრო ნათელს ხდის კოდის ბლოკის დასაწყისსა და დასასრულს.

if პირობის დასაკმაყოფილებლად საჭიროა პირობაში იყოს შედეგი - true. პირობაში განთავსებული მნიშვნელობიდან გარდა 0, null, undefined, '', NaN, false დანარჩენი ყველა მნიშვნელობა ითვლება ჭეშმარიტად.

```
if(0) { // კონსტუქცია არ იმუშავებს  }
if(null) { // კონსტუქცია არ იმუშავებს  }
if(undefined) { // კონსტუქცია არ იმუშავებს  }
if('') { // კონსტუქცია არ იმუშავებს  }
if(NaN) { // კონსტუქცია არ იმუშავებს  }
if(false) { // კონსტუქცია არ იმუშავებს  }

//  გარდა ზემოთ ჩამოთვლილი შემთხვევებისა სხვა ყველა შემთხვევაში პირობა აღიქმება დადებითად

if(-1) { // კონსტუქცია იმუშავებს}
```

დაიმახოსვრეთ Javascript_ში if კონსრუქციის პირობის დაკმაყოფილების შედეგად კოდის კითხვა გრძელდება ბლოკის ფრჩხილებში და მხოლოდ ამის შემდგომ გადადის მის ქვემოთ არსებულ სკრიტების შესრულებაზე.

მაგ.: ამოცანა მდგომარეობს შემდეგში. შემოვიტანოთ ცვლადი რომელშიც განვსაზღვრავთ პიროვნების ასაკს. მოვახდინოთ if ოპერატორის მეშვეობით დადგენა სრულწლოვანია თუ არასრულწლოვანია არსებული პიროვნება. პირობა დაისმის შემდეგნაირად: თუ ადამიანის ასაკი აღემატები 18 წელს ის წარმოადგენს სრულწლოვანს.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<script>
            varage = 25;
            var  age1 =  10;
```

```
        var  user = 'სრულწლოვანი'; // საწყის ეტაპზე განისაზღვრა, რომ ეს მომხმარებელი არის  18_ზე მეტი წლის.

        if(age <18) {
                user = 'არა სრულწლოვანი';
        }

        // თუ if პირობა შესრულდებოდა user ცვლადი მიიღებდა მნიშვნელობად 'არა სრულწლოვანი' და შემდეგ ხაზზე დაიბეჭდებოდა სწორედაც ეს სიტყვა, გამომდინარე იქიდან, რომ   25 ნაკლები არ არის 18_ზე  if  კონსტრუქცია აღარ გააქტიურდა და ცვლად user _ მა შეინარჩუნა პირველადი მნიშვნელობა

        document.write(user); // სრულწლოვანი


        if(age1<18) {
                user = 'არა სრულწლოვანი';
        }

        // არსებულ შემთხვევაში 10 ნაკლებია 18ზე პირობა დაკმაყოფილებულია. user ცვლადმა მიიღო მნიშვნელობა 'არა სრულწლოვანი'

        document.write(user);          // შედეგია:  არასრულწლოვანი

</script>
</body>
</html>
```

გარდა ცალკეული if კონსტრუქციისა არსებობს **if-else** ოპერატორი და მისი ჩანაწერი შემდეგია

```
        if(age1<18) {
                //  შესრულდება ყოველთვის, როცა დაკმაყოფილდება if პირობა
        } else {
                //  შესრულდება ყოველთვის, როცა არ დაკმაყოფილდება if პირობა
        }
```

თუ ყურადღებით დაუკვირდებით else პირობის ფრჩხილები (...) არ გააჩნია, რადგანაც ის if_ის უარყოფას წარმოადგენს და დამატებით პირობებს არ საჭიროებსდ და if ოპერატორში დასმული პირობის „მცდარობის“ შემთხვევაში ყველეთვის შესრულდება.

დაიმახსოვრეთ: else ოპერატორს დამოუკიდებლად არსებობა არ შეუძლია, განსხვავებით if ოპერატორისაგან.

```
    if(age1 <18) {
            document.write('არა სრულწლოვანი');
    } else {
            document.write('სრულწლოვანი');
    }
```

არსებობს if-else კონსტრუქციის შემოკლებული ჩანაწერი, რომელსაც დეველოპერები საკმაოდ აქტიურად გამოიყენებენ მცირე კოდის შემთხვევაში

```
(age1 <18) ?document.write('არა სრულწლოვანი') :document.write('სრულწლოვანი');
```

სადაც სიტყვიერი if არ იწერება, იხსნება პირობის ფრჩხილები, მას მოსდევს „? „ რომელიც აღნიშნავს if_ის ბლოკს, ხოლო ":" else _ს ბლოკია .

არსებობს ასევე შედაერბით რთული კონსტრუქცია, რომელიც **else-if**_ად მოიხსენიება. ეს არის კონსტრუქცია, რომელშიც if პირობის უარყოფასთან ერთად ხდება ახალი პირობის წამოყენება.

```
if(age <18) {
            document.write('მოზარდი');
} elseif (age <60) {
            document.write('ზრდასრული');
    }
```

მოცემულ მაგალითი შემდეგნაირად იშიფრება: თუ ცვლადი age ნაკლებია 18_ზე მაშინ დაბეჭდოს 'მოზარდი', **წინააღმდეგ შემთხვევაში თუ** age ნაკლებია 60_ზე დაბეჭდავს'ზრდასრული'.

ბევრს შეიძლება გაუჩნდეს კითხვა,თუ age = 15 მაშინ შესრულდება if ბკლოკში არსებული კოდი (დაიბეჭდება 'მოზარდი') , მაგრამ 15 ხომ 60_ზეც ნაკლებია? რა თქმა უნდა ნაკლებია მაგრამ else if პირობაზე გადასხვა მხოლოდ იმ შემთხვევაში ხდება, როდესაც არ სრულდება if პირობა.

თუ if პირობა შესრულდა არა აქვს მნიშვნელობა რამდენი else-if იარსებებეს მის ქვემოთ ისინი არასდროს დაკმაყოფილდებიან. იხ.მოცემული მაგალითი

```
var age =15;

    if(age <18) {
            // 15<18 შესაბამისად შესრულდება მხოლოდ ეს კოდი
            document.write('მოზარდი');
} elseif (age <30) {
```

```
            document.write('ზრდასრული');
}
  elseif (age <60) {
            document.write('ასაკოვანი');
}
  elseif (age >60) {
            document.write('ხნიერი');
    }
```

შესაძლებელია else-if კონსტრუქციის გამოყენებისას ის დასრულდეს else ოპერატორით, რომელიც ყველა შესაძლო else if პირობის უარყოფას წარმოადგენდეს

```
var car ='acura';
  if(car =='bmw') {
            document.write('bmw');
}elseif (car =='volvo') {
            document.write('volvo');
}elseif (car =='opel') {
            document.write('opel');
}else {
            document.write('მარკა სრულიად განსხვავებულია ის არის '+car);
}
```

switch – case კონსტრუქცია

ბოლო მოგალითზე დაკვირვებით სია, რომელთანაც არსებულ car _ს ვადარებთ შესაძლოა საკმაოდ დიდი იყოს. შემთხვევა, როცა ერთი კონკრეტული მნიშვნელობის რამოდენიმესთან ერთდროულ შედარებაა საჭირო გამოიყენება **switch – case** კონსტრუქცია რომელიც ბევრად მოხერხებულია, გამომდინარე მისი სინტაქსიდან.

```
 switch(შესადარებელიმნიშვნელობა){

case'მნიშვნელობა 1':
            შესასრულებელი არეალი .....
break;                  // თუ case არსებული მნიშვნელობის თანხვედრა მოხდა break აჩერებს დანარჩენი დადარების ოპერაციებს.

case'მნიშვნელობა 2':
            შესასრულებელი არეალი .....
break;

default:
            შესასრულებელი არეალი .....
```

```
break;
}
```

```
 var car ='acura';

 switch(car){
case'bmw':// if (x === ' bmw')
            document.write('bmw');
break;

case'volvo':// if (x === 'volvo')
            document.write('volvo');
break;

case'opel':// if (x === 'opel')
            document.write('opel');
break;

default:
            document.write('მარკა სრულიად განსხვავებულია ის არის '+car);
break;
}
```

ოპერატორი breakწყვეტს კოდის კითხვადობას თუ მოხდა გადმოცემული მნიშვნელობის და case_ის გატოლება (ექვივალენტურად). არსებული ოპერატორის გამოტოვების შემთხვევაში გამოვა switch_ში არსებული უკანასკნელი მნიშვნელობა, რაც არასასურველ შედეგამდე მიგიყვანთ.

მინიშნება არ დაგავიწყდეთ პირობის შემდეგbreakოპერატორის გამოყენება.

რაც შეეხება ოპერატორ default მის „ტანში“ გამოტანილი მოქმედება სრულდება მაშინ, როცა არ მოხდება ცვლადის არცერთ case მნიშვნელობასთან გატოლება (ექვივალენტურად).

ფაქტობრივად switch-case კონსტრუქცია, რაღაც კუთხით else-if კონსტრუქციის გამარტივებულ ვარიანტს წარმოადგენს (ყველა შემთხვევაში არა), შესაბამისად თუ მას else-if _თან გავაიგივებთ ოპერატორი default წარმოადგენს ყველა პირობის უარმყოფ else ოპერატორს, რომელიც სრულდება დანარჩენი პირობის დაუკმაყოფილებლობის შემთხვევაში.

default გამოყენება არ არის სავალდებულო.

switch-case კონსტრუქციაში შესაძლებელია რამოდენიმე case მნიშვნელობის დაჯგუფება შემდეგი სახით

```
var a =2+2;
```

```
switch(a){
case4:
alert('სწორია!');
break;

case3:// თუ a === 3 ან a ===5 გამოვიდოდა მიმდინარე შეტყობინებები
case5:
alert('შეცდომაა!');
alert('დაგაკლდათ მცირეოდენი');
break;

default:
alert('თქვენ საერთოდ არ ხართ არითმეტიკულ მოქმედებებთან ახლოს ☺');
break;
}
```

ციკლები

სკრიპტის შემუშავების პროცესში ხშირად დგება ამოცანა ერთგვაროვანი მოქმედება განმეორდეს რაიმე ინტენსივობით გარკვეულის ლოგიკის შესაბამისად. კოდის ნაწილის გამეორებადობისათვის Javascript პროგრამირების ენა იყენებს ციკლებს. ადვილად აღწმადი, რომ გამოვიდეს კიდევ ერთხელ განვმარტოთ: ციკლი წარმოადგენს პროცესს, რომელიც რაღაც კანონზომიერებით მეორდება.

მაგ. 1: წელიწადში 4 დროა: გაზაფხული, ზაფხული, შემოდგომა, ზამთარი. ისინი სწორედ იმ თანმიმდევრობით მონაცვლეობენ, როგორც ჩამოვთვალეთ ანუ ემორჩილებიან კონკრეტულ კანონზომიერებებს. ეს არის ციკლი, რომელიც არასდროს (ყოველ შემთხვევაში იმედია) დასრულდება.

მაგ. 2: დღე-ღამე შედგება 24 საათისაგან. სწორედ 24 საათიანი ინტერვალის გავლიშ შემდგომ იწყება ახალი დღე ანუ საათის ათვლა უბრუნდება ნიშნულ 0_ს.

ამ ყოველივეს გასამარტივებლად Javascript_ში გაოიყენება ციკლის ოპერატორები **while, do while, for.** განვიხილოთ თანმიმდევრულად.

ციკლი while სინტაქსი მოცემულია მაგალითზე.

```
while(პირობა){
        მოქმედების შესასრულებელი არეალი
}
```

როგორც მონახაზიდან ჩანს ციკლი while ვიზუალი ძალიან ახლოსაა if კონსტრუქციასთან. მასაც გააჩნია პირობის ჩასაწერი ფრჩხილები (...) და იდენტური ბლოკის გამხსენი და დამხურავი ფიგურული ფრჩხილები { ... } . შესაბამისად არსიც იგივეა თუ პირობა სრულდება ანუ პასუხი ჭეშმარიტია true, მაშინ სრულდება ბლოკის ფრჩხილებში მოთავსებული სკრიპტი და ეს პროცესი გრძელდება იქამდე სანამ პირობის **ჭეშმარიტება** ძალაშია. მაგ.:

```
var a = 5;

while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');
}
```

მოცემული ციკლი იმუშავებს უსასრულოდ. ტექსტი 'გამარჯობა მე ვარ ციკლი' alert დიალოგური ფანჯრის მეშვეობით გამოტანება უწყვეტ რეჟიმში და თითოეულის გათიშვა გამოიწვევს საპასუხოდ ახლის გამოჩენს. **რატომ ?** გამომდინარე იქიდან, რომ 5 ყოველთვის ნაკლებია 9_ზე ეს პირობა არასდორს შეიცვლება. ვთქვათ ამ პოცესის გამეორება გვესაჭიროება მხოლოდ 4_ჯერ. როგორ მოვიქცეთ? შევეცადოთ დავხვეწოთ კოდი და მოვარგოთ ჩვენს მოთხოვნილებებს.

```
var a = 5;

while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');
        a++;                                    // გამოვიყენე ინკრიმენტი
}
```

a ცვლადის ზრდადობა გამოიწვევს იმას, რომ როდესღაც a მიაღწევს მნიშვნელობას როცა ის აღარ იქნება ნაკლები 9_ზე და ციკლი შეწყვეთს მუშაობას. არსებულ შემთხვევაში ეს იქნება მეხუთე ნაბიჯზე, შესაბამისად შეტყობინება ეკრანზე გამოვა 4_ჯერ.

```
var a = 5;

// ეტაპი I
// a = 5 პირობა სრულდება
while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');       // გამოდის ეკრანზე
        a++;                                    // a ცვლადმა განიცადა ინკრიმენტაცია - შემდეგი ხაზიდან
a=6
// აქ უკვე a არის 6 – a=6
}


// ციკლის უზრუნველყოფს ინფორმაციის გამეორებადობას ე.ი ის აბრუნებს კოდის მნიშვნელობას
დასაწყისში და ეკითხება არის a < 9_ზე? გამომდინარე იქიდან, რომ a =6 ის აშკარად ნაკლები გამოდის
9_ზე და პროცესი კიდევ ერთხელ მეორებდება
```

```
// ეტაპი II
// a = 6 პირობა სრულდება
while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');      // გამოდის ეკრანზე
        a++;                                   // a ცვლადმა განიცადა ინკრიმენტაცია - შემდეგი ხაზიდან
a=7
// აქ უკვე a არის 7 – a=7
}


// ეტაპი III
// a = 7 პირობა სრულდება
while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');      // გამოდის ეკრანზე
        a++;                                   // a ცვლადმა განიცადა ინკრიმენტაცია - შემდეგი ხაზიდან
a=8
// აქ უკვე a არის 8 – a=8
}


// ეტაპი IV

// a = 8 პირობა სრულდება
while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');      // გამოდის ეკრანზე
        a++;                                   // a ცვლადმა განიცადა ინკრიმენტაცია - შემდეგი ხაზიდან
a=9
// აქ უკვე a არის 9 – a=9
}


// ეტაპი V
//a = 9 პირობა არ სრულდება.9 არ არის ნაკლები 9_ზე
// შესაბამისად სკრიპტი აღარ კითხულობს ბლოკის შიგთავს და ეკრანზე  alert-აღარ გამოვა.
while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');
        a++;
}
```

ციკლის გამეორებადობის პროცესს **იტერაცია** ეწოდება. მაგალითის შემთხვევაში ციკლი ასრულებს 4 - ოთხ იტერაციას.

უსასრულო ციკლის საჭიროების შემთხვევაში არ არის აუცილებელი ვრცელი გზის გამოყენება

```
var a = 5;

while(a < 9){
        alert('გამარჯობა მე ვარ ციკლი');
}
```

შესაძლებელია პირობაში გაიწეროს true

```
while(true){
        alert('გამარჯობა მე ვარ ციკლი');
}
```

დავალება 1:

გამოვსახოთ ეკრანზე 1_დან 10_მდე ყველა ლუწი რიცხვი შემდეგი სახით. ამ ეტაპზე დავუშვათ , რომ ბოლო მნიშვნელობასაც ექნება გამყოფად ტირე „-“

```
2 – 4 – 6 – 8 -
```

დავალების ამოსახსნელად გვჭირდება ციკლი, ინფორმაციის გამოტანის ოპერატორი, და ცვლადი რომელიც ყველა ჯერზე მოიმატებს 2_ით.

```
var a = 2;

while(a < 10){
        document.write( a + ' - ' );  // დაბეჭდავს 2 - ..... და აშ ყველა ნაბიჯზე
        a += 2;     // იგივეა რაც  a = a + 2  (მინიჭების ოპერატორის შემოკლებული  ჩანაწერი )
}
```

დავალება 2:

ციკლის მეშვეობით გამოვიტანოთ ეკრანზე ყველა რიცხვის ჯამი 1_დან 5_მდე.

```
15
```

დავალების ამოსახსნელად გვჭირდება ციკლი, ინფორმაციის გამოტანის ოპერატორი, და ცვლადი, რომელიც ყველა ჯერზე მოხდება არსებული რიცხვის მიმატება წინა ჯამთან.

```
var i = 1;
var sum = 0;


while(i <= 5){
```

```
        sum += i;  //sum = sum+i I ეტაპი 0+1, II ეტაპი 1+2,  II I ეტაპი 3+3,  IV ეტაპი 6+4, V ეტაპი 10+5
        i++;       //უზრუნველყოფს ციკლის მუშაობას
}

document.write(sum);  // 15
```

განსხვავევით დავალება1_ის ამოხსნისაგან სადაც document.write() მეთოდი უშუალოდ while ციკლის ტანში { ... } იყო გამოძახებული, დავალება2 შედეგის გამოტანის მეთოდი გარეთაა გამოძახებული. ეს რა თქმა უნდა დავალებიდან გამომდინარეა. პირველ შემთხვევაში საჭირო იყო ყველა ეტაპზე შედეგის გამოტანა, ხოლო მეორე შემთხვევაში მხოლოდ საბოლოო შედეგის. მეორე დავალების ციკლი უბრალოდ უზრუნველყოფდა დაჯამებას თითოეულ ეტაპზე (ეტაპები კომენტარის ველშია გაშლილი) და გამოტანა ერთჯერადი პროცედურით შემოიფარგლებოდა.

ცილში შესაძლებელია break და continue ოპერატორების გამოყენება. break_ის შესახებ ინფორმაცია მოწოდებული იყო switch-case კონსტრუქციის განხილვისას. ის გამოიყენებოდა პროცესის შესაჩერებლად / გასაწყვეტად. იგივე ფუნქციონალი აკისრია მას ციკლთან მიმართებაშიც.

break ციკლში მოქმედების უკეთ გასაგებად დავალება 2 შევასრულოდ სახეცვლილი პირობით

დავალება 3:

> ეკრანზე დაბეჭდოს რიცვხთა ჯამი 1_დან 5_მდე, სანამ ის იქნება 12_ზე ნაკლები. ანუ ეკრანზე უნდა გამოიტანოს 10 (I ეტაპი 0+1, I I ეტაპი 1+2, I I I ეტაპი 3+3, IV ეტაპი 6+4, V ეტაპი 10+5), რადგანაც ბოლო ეტაპზე მიიღება რიცხვი 15, რომელიც შესაბამისად აღემატება 12_ს.

```
var i = 1;
var sum = 0;

while(i <= 5){

// არსებულ ეტაპზე ვამოწმებთ ჯამს დამატებული მიმდინარე რიხვი მეტი ხომ არ არის 12, თუ პირობა
შესრულდა sum ცვლადში აღარ მოხდეს მიმდინარე ბიჯის დამატება და ციკლმა შეწყვიტოს მუშაობა;

        if( (sum + i)> 12){
                break;
        }
        sum += i;  //sum = sum+i I ეტაპი 0+1, II ეტაპი 1+2,  II I ეტაპი 3+3,  IV ეტაპი 6+4
        i++;       //უზრუნველყოფს ციკლის მუშაობას
}

document.write(sum);  // 10
```

break ოპერატორისაგან განსხვავებით continue არ წყვეტს ციკლის მუშაობას ის უბრალოდ ახტება იმ ნაბიჯს, რომელზეც გვაქვს შესაბამისი პირობა დადებული.

დავალება 4:

გამოვსახოთ ეკრანზე 1_დან 20_მდე ყველა ლუწი რიცხვი,გარდა **6** და **14** შემდეგი სახით

2 – 4 –8 – 10 – 12 – 16 – 18

```
var a = 2;

while(a < 20){

// პირობა: თუ a მნიშვნელობა ტოლია 6 ან 14 მოახდინოს a გაზრდა 2_ით და დაბრუნდეს ციკლის პირობაში, შესაბამისად ტოვებს document.write()_ს რომელიც უზრუნველყოფს დაბეჭდას

        if(a==6 || a==14 ){
                a += 2;
                continue;
        }

        document.write( a + '-' );//დაბეჭდავს 2 - ..... და აშ ყველა ნაბიჯზე
        a += 2;     // იგივეა რაც  a = a + 2  (მინიჭების ოპერატორის შემოკლებული  ჩანაწერი )
}
```

როგორც ქვეთავის დასაწყისში აღვნიშნეთ Javascript_ში გვხვდება do…while ციკლიც. მისი მუშაობის პრინციპი ოდნავ განსხვავებულია სტანდარტული while ციკლისაგან, ასევე მცირეოდენ შეცვლილია წერის სინტაქსიც, რამეთუ ჯერ მოდის შესასრულებელი მოქმედების ბლოკი ხოლო შემდგომ პირობა. ეს ყოველივე კი უზრუნველყოფს იმას, რომ მოქმედება მინიმუმ **ერთჯერადად** მაინც სრულდება.

```
var i = 2;

do {

   document.write( i + ' - ' );
   i++;

} while(i > 5)

//ეკრანზე მიღებული შედეგი 2 -
```

ციკლი ასრულებს შემდეგ მოქმედებას :

- do_ს მეშვეობით შედის ციკლის ტანში {...} ასრულებს ლოგიკურ ოპერაციას (2-);
- ახდენს **i** ცვლადის გაზრდას 1_ით;
- ამის შემდგომ უყურებს whileციკლის პირობას, გამომდინარე იქიდან, რომ პირობა არ კმაყოფილდება (3>5), ხელახალი პროცესის გამეორებადობა არ მოხერხდა;

სტნდარტული while ციკლის გამოყენების შემთხვევაში შედეგი საერთოდ არ გამოსახებოდა ეკრანზე, რადგანაც იქ ჯერ შედარების ოპერაცია ხორციელდება

```
var i = 2;

while(i > 5){

   document.write( i + '-' );
   i++;

}

//ეკრანზე არავიტარი შედეგი არ გამოისახება
```

მაგრამ მოცმეული მაგალითს თუ შემდეგ პირობას წავუყენებთ i < 5 , არავითარი განსხვავება არ იქნება, რომელი ციკლის მეშვეობით შევასრულებთ მას იქნება ეს whileთუdo…while ციკლები

```
//მაგალითი while  - ციკლით

var i = 2;

while(i < 5){

   document.write( i + '-' );     //ეკრანზე მიღებული შედეგი 2 – 3 -  4 –

   i++;

}


//მაგალითი do…while  - ციკლით

var i = 2;

do {

   document.write( i + ' - ' );    // ეკრანზე მიღებული შედეგი 2 – 3 -  4 -
```

```
  i++;

} while(i < 5)
```

პროგრამისტეიბს მიერ ციკლებიდან ყველაზე ხშირად - for კონსტრუქციაა, რომელიც თავისი სინტაქსით მეტად მობილურია და შემდეგნაირად გამოიყურება. ის ფაქტობრივად იყენებს შაბლონს, რომელშიც თავისთავად მოიაზრება ციკლი while.

```
for(დასაწყისი; პირობა; ბიჯები){
// ... ციკლის ტანი ...
}
```

while ციკლისაგან განსხვავევით for_ის ფრჩხილებში (...) კომპაქტურადაა მოთავსებული ცვლადის გამოცხადება, პირობა და იტერაცია რაც თვალნათელს ხდის თითოეულ ბიჯს.

```
for(var i =0; i <3; i++){
alert( i );                          // alert(0),  alert(1),  alert(2)
}
```

როგორც while და do…while შემთხვევაში იყო შესაძლებელი ინდენთურად მათი ამუშავებისა ასევე შესაძლებელია forციკლი ვამუშაოთ while_ის მსგავსად. for ციკლში დასაშვებია ნებისმიერი მნიშვნელობის გამოტოვება.

```
//ვარიანტი I

var i =0
for(; i <3; i++){
// ... ციკლის ტანი ...
}


//ვარიანტი II
var i =0
for(; i <3;){
// ... ციკლის ტანი ...
i++;
}

//ვარიანტი III
for(;;){
```

```
// ... ციკლის ტანი ...
}
```

ვარიანტი I- ფაქტობრივადარგანსხვავდებაfor სტანდარტულიჩანაწერისაგანუბრალოდსაწყისიმნიშვნელობაამოღებულიაფრჩხილებიდან (...) დაგატანილიაცალკე.

ვარიანტი II - ანალოგურია while ციკლისა

ვარიანტი III - ციკლი, რომელიც იმუშავებს უსასრულოდ.

Javascript_ში არსებობს for in კონსტრუქცია ამ ეტაპზე მისი განხილვა აზრს მოკლებულია, დამხმარე თემებზე, რომელთანაც ის კავშირშია ჯერ არ გქონიათ შეხება, შესაბამიად შემდეგ ქვეთავებში მაგალითებზე დაყრდნობით გავარცევთ მასასც დეტალურად.

5.3 მასივებთან მუშაობა

მიმდინარე პარაგრაფის თემატიკა

- მასივებთან მუშაობის ძირითად ფუნქციები
- ამოცანის გადაჭრის გზები მასივზე დაფუძნებული აბსტრაქტული მონაცემთა სტრუქტურების მეშვეობით
- ამოცანის იმპლემენტაცია მასივების მეშვეობით

მასივი ეს არის მონაცემთა ერთგვარი ტიპი, რომელიც შეიცავს ერთ ან რამოდენიმე მნიშვნელობას ერთდროულად, რომლებიც მასივში განთავსებულნი არიან საკუთარ პოზიციებზე. არსებულ პოზიციებს ინდექსები ეწოდებათ და ისინი წარმოადგენენ რიცხვით მნიშვნელობას 0_დან დადებითი მიმართულებით. გამომდინარე Javascript_ის არამკაცრი ტიპიზაციიდან მასივს მკვეთრად გამოხატული ტიპიზება არ სჭირდება, ასევე მისი თითოეული მნიშვნელობა შეიძლება ერთმანეთისაგან რადიკალურად განსხვავებული ტიპის გახლდეთ. არის შესაძლებლობა მასივის მნიშვნელობა ასევე მასივი იყოს.

მასივის შესაქმნელად ერთ ერთი გზა შემდეგი. საჭიროა ახალი ცვლადის გამოცხადება მისთვის სახელის მინიჭება და კვადრატული ფრჩხილები [....]. მაგალითზე მოცემულია ცარიელი მასივი.

```
var arr =[];
```

მასივის ელემენტების შესავსებად საკმარისია კვადრატულ ფრჩხილებში მძიმით მოხდეს ჩამონათვლის გაკეთება

```
var arr =['მარწყვი', 'ატამი', 'ფორთოხალი'];
```

არსებული მასივი შედგება 3 ელემენტისაგან, შესაბამისად ისინი განთავსებულნი არიან პოზიციებზე რომელიც ათვლას ყოველთვის 0 დან იწყებს. თითოეულ მათგანთან წვდომისათვის უნდა მიმართოთ მასივს სახელწოდებით , კვადრატული ფრჩხილები და იმ კონკრეტული მნიშვნელობის ინდექსით რომელის გამოტანაც გსურთ.

```
var arr =['მარწყვი', 'ატამი', 'ფორთოხალი'];

alert(arr[ 0 ]);                // მარწყვი
alert(arr[ 1 ]);                // ატამი
alert(arr[ 2 ]);                // ფორთოხალი
```

იმ ინდექსით მასივისათვის მიმართვა რომელზეც არანაირი მნიშვნელობა არ არის განთავსებული შედეგად დაგიბრუნებთ undefined

```
var arr =['მარწყვი', 'ატამი', 'ფორთოხალი'];

alert(arr[ 3 ]);                // undefined
alert(arr[ 80 ]);               // undefined
```

თუ მასივის ელემენტების ჩამოთვლისას მათ შორის ხელოვნურად ან შემთხვევით ჩავსვამთ დამატებით მძიმეებს ისინი გამოყოფს მასივის ელემენტისათვის ინდექსებს და მათმნიშვნელობად undefined მოიაზრებს. მაგალითზე მოყვანილი მასივის მნიშვნელობა 'ატამი'_თან წვდომისათვის ინექსი 1 აღარ გამოდგება, რადგანაც არსებულ ინდექსე მასივმა განალაგა მნიშვნელობა რომელიც ამ ეტაპზე undefined არის. მაგალითზე ხელოვნურად არის ადგილები გამოტოვებული ინდექსებიის ზუსტად აღსაქმელად.

```
var arr =['მარწყვი',  ,  ,'ატამი',  ,'ფორთოხალი'];
//          0   1  2  3   4   5

alert(arr[ 1 ]);                // undefined
alert(arr[ 3 ]);                // ატამი
```

მასივისათვის ახალი ელენებტის მინიჭება ან არსებულის ჩანაცვლება ყველთვის შესაძლებელია, საჭიროა მხოლოდ მასივის კონკრეტულ ინდექსს მივანიჭოთ მნიშვნელობა. იმ შემთხვევაში თუ ამ კონკრეტულ ინდექსზე უკვე არსებობს მნიშვნელობა ის ჩანაცვლდება წინააღმდეგ შემთხვევაში დაემატება ახალი.

```
var arr =['მარწყვი' , 'ატამი','ფორთოხალი'];
//            0       1        2

arr[ 0 ]='ვაშლი';     // arr =['ვაშლი' , 'ატამი','ფორთოხალი'];
arr[ 3 ]='ყურძენი'; // 'ვაშლი' , 'ატამი','ფორთოხალი', 'ყურძენი'];
```

როგორც ქვეთავის დასაწყისში აღვნიშნეთ აუცილებელი არ არის მასივი შეიცავდეს ერთგვაროვანი ტიპის ელემენტევს. Javascript_ში შესაძლებელია შერეული ტიპის მნიშვნელობების მინიჭება ერთი კონკრეტული მასივისათვის. აქამდე განხილულ მაგალითებში საქმე მხოლოდ ტექსტური (string) ტიპის მნიშვნელობები ფიგურირებდნენ. გასათვალისწინებელია ის რომ ნასივის ინდექზე მნიშვნელობის განთვსებისას ტიპი სტანდარტული ჩანაწერით (იქნება ის ბრჭყალებით - string, მის გარეშე-number, იქნება ობიექტი და ა.შ) გამოისახება.

```
var arr =['მარწყვი' , true , null, 'კივი', 155];
```

მასივის ინდექსაცია ყოველთვის იწყევა 0_დან. ხელოვნურად მისი ცვლილება შეუძლებელია. მაგ.: ინდექსად 5_ის მნიშვნელობის განსაზღვრა არ განაპირობებს მის ამთვლელ ინექსს. საწყისი ინდექსები ავტომატურად შეივსება undefined მნიშვნელობებად.

```
arr[ 5 ]='საქართველო';          // arr =[ , , , , , 'საქართველო'];
```

როგორც აღვნიშნეთ მასივის კონრეტული მნიშვნელობის გამოსატანად სჭიროამასივის სახელიდა და მისი ინდექსის გამოტანის ოპერატის ატრიბუტად განსაზღვრა (მაგ.: document.write(**arr[5]**)), მაგრამ Javascript_ში შესაძლებელია მასივის სრული ვიზუალიზაცია უშუალოდ მისი სახელის გამოტანით. დაიმახსოვრეთ არსებული გზა ემსახურება მხოლოდ მასივის ვიზუალური შედეგის მიღებას და ძირითადად სატესტოდ გამოიყენება.

```
var arr =['მარწყვი' , true , null, 'კივი', 155];

document.write(arr)// 'მარწყვი' , true , null, 'კივი', 155;
```

არსებობს მასივის შექმინის კიდევ ერთი სახის სინტაქსი`new Array();` ის შედარებით იშვიათად გამოიყენება გამომდინრე იქიდან, რომ კვადრატული ფრჩხილებით [...] ჩანაწერი ბევრად შემოკლებულია.

```
var arr =newArray('მარწყვი' , true , null, 'კივი', 155);
```

არსევე არსებობს ერთი თავისებურება და სხვაობა ამ ორ სინტაქსს შორის. თუ მასივის შედგება რიცხვითი მნიშვნელობისაგან და ის მხოლოდ 1 ცალია, **newArray**() მეშვეობთ მისი გამოცხადება მოგცემთ განსხვავებულ შედეგეს. მაგ.:

```
var arr =[155];
alert( arr[ 0 ] );                              // 155;

var arr1 =newArray(155);
alert(arr1[ 0 ]);                       // undefined
```

მაშინ რა მისია აკისრია ამ შემთხვევაში 155 - newArray(155) ? **ის განსაზღვრავს მასივის სიგრძეს.** (არსებულ საკითხს დავუბრუნდებთ) სხვა ყველა დანარჩენ შემთხვევაში ეს 2 ჩანაწერი მუშაობს ინდენტურად იქნება ეს სხვა ტიპის 1 ან 1_ზე მეტი მნიშვნელობა, თუ რიცვითი ცვითი მნიშვნელობა მხოლოდ ერთ ელემენტზე მეტი.

```
// შედეგი იდენტური
var arr =newArray('მარწყვი' ,true,null, 'კივი', 155);
var arr =['მარწყვი' ,true,null, 'კივი', 155];

// შედეგი იდენტური
var arr =['მარწყვი'];
var arr =newArray('მარწყვი');

// შედეგი იდენტური
var arr =[5 , 120];
var arr =newArray(5 , 120);

// შედეგი განსხვავებული
var arr =[10];
var arr =newArray( 10  );
```

რა არის მასივის სიგრძე, რომელმაც მცირეოდენი გაუგებრობა წარმოშვა?

მასივის სიგრძე ეს არის მასისვის თვისება რომელიც განსაზღვრავს თუ რამდენი ელემენტისაგან შედგება ესა თუ ის კონკრეტული მასივი. მისი სინტაქსი შემდეგია **მასივის სახელი.length**

```
var arr =['მარწყვი' , true , null, 'კივი', 155];

alert( arr.length );   // 5
```

მასივის სიგრძე დინამიურ თვისებას წარმოადგენს და იცვლება მასივის ელემენტების რაოდენობიდან გამომდინარე- უფრო ზუსტად, რომ აღვნიშნოთ მასივის ინდექსების რაოდენობიდან გამომდინარე. შემთხვევას როცა კონკრეტულ ინდექსზე (გარდა 0) ვათავსებთ ერთ ელემენტს მასივის სიგრძეს განსაზღვრავს ინდექსი და არა ელემენტი, რადგანაც ინდექზე განთავსება ნიშნავს იქამდე არსებული გამოტოვებული ინდექსების ავტომატურ შევსებას undefined მნიშვნელობებით.

.length თვისების გამოყენებით შესაძლებელია არსებული მასივის დამოკლება, რომელიც არც ისე ხშირად გამოყენებადი ამ დანიშნულებით. მაგ.:

```
var arr =[1,2,3,4,5];

arr.length =2;// შემცირდა 2 ელემენტრამდე
alert( arr );// [1, 2]

arr.length =5;// შევეცადოთ  დავაბრუნოთ მასივის საწყისი სიგრძე
alert( arr[3]);// undefined: სამწუხაროდ მცდელობამ უშედეგოდ ჩაიარა
```

მასივივის ელემენტების უწყვეტ რეშიმში ამოსაღებად შესაძლებელია უკვე ცნობილი რომელიმე ციკლის ოპერატორის გამოყენება, რომელიც შეასრულებს იმდენ იტერაციას რამდენიც იქნება მასივის სიგრძის (arr.length) .ეს ყოველივე უზრუნველყოფს ყველა ელემენტთან წვდომას. მაგ.:

```
var arr =['მარწყვი' , true , null, 'კივი', 155];

for(var i =0; i <arr.length; i++){
alert(arr[i]);                              // alert('მარწყვი'),  alert(true),  ....
}
```

ციკლების განხილვისას აღვნიშნეთ, რომ არსებობს for_ის სახეცვლილი ჩანაწერი რომელსაც იმ კონკრეტულ ეტაპზე არ შევეხეთ. ეს არის for…in კონსტრუქცია, რომელიც გამოიყენება ობიექტის ყველა მნიშვნელობის გადასათვლელად. არსებულ შემთხვევაში ზედა მააგალითის პირობის გასამარტივებლად ეს კონსტრუქცია იდეალური ვარიანტი იქნებოდა, სადაც key მნიშვნელობა ხოლო arr არსებული მასივი.

```
var arr =['მარწყვი' ,true,null, 'კივი', 155];

for(var key in arr){
alert( arr[key]);// 'მარწყვი' , true , null, 'კივი', 155;
```

```
}
```

გარდა ერთგანზომილებიანი მასივებისა Javascript_ში გვხვდება მრავალგანზომილებიანი მასივები. იმ შემთხვევაში თუ მასივის ინდექსებზე მნიშვნელობებად ასევე მასივები მოგვევლინებიან მაშინ საქმე მრავალგანზომილებიან მასივებთან გქოდეთ. ამ მეთოდის გამოყენებით შესაძლოა მაგ. მატრიცის მიღება. (ვიზუალისათვის წარმოდგენილია სხვადასხვა ხაზზე, არსებილი ერთ ხაზზე განლაგეც არ წარმოადგენს შეცდომას)

მის კონკრეტულ ელემენტთან წვდომა შესაძლებელია მთავარი და ქვე მასივის მასივის ინდექსის მეშვეობით

```
var matrix =[
[1,2,3],
[4,5,6],
[7,8,9]
];

alert( matrix [1] [2]);              // 6
```

გარდა length თვისებისა არსებობს რიგი ფუნქციების / მეთოდებისა, რომლებიც აქტიურ რეჟიმში გამოიყენებიან მასივებთან სამუშაოდ. იხ ცხრილი VI

ცხრილი VI

```
var arr =['მარწყვი' , 'ატამი','ხილი'];
```

მეთოდი	აღწერა	მაგალითი	შედეგი
toString()	ტექსტურ მონაცემად გარდაქმნა	arr.toString()	მარწყვი , ატამი , ფორთოხალი
join()	მნიშვნელობების შეერთება	arr.join(*)	მარწყვი * ატამი * ფორთოხალი
pop()	ბოლო ელმენტის ამოჭრა	arr.pop()	'მარწყვი' , 'ატამი'
push()	ელემენტის/ელემენტების დამატება მასივის ბოლოში	arr.push('ვაშლი')	'მარწყვი' , 'ატამი','ხილი', 'ვაშლი'

shift()	პირველი ელმენტის ამოჭრა	arr.shift('კივი')	'კივი','მარწყვი' , 'ატამი','ხილი';
unshift()	ელემენტის/ელემენტების დამატება მასივის ბოლოში	arr.unshift()	'ატამი','ხილი'
splice()	შესაძლოა ელემენტების ჩამატება კონკრეტულ ინდექსზე, არსებულის ამოჭრა ან ჩანაცვლება	arr.splice(0, 2 , 'კივი')	'კივი','ხილი'
		arr.splice(1, 0 , 'კივი')	'მარწყვი' ,'კივი', 'ატამი','ხილი'
		arr.splice(1, 2)	'მარწყვი'
slice()	მასივის ინდექსების კონკრეტული დიზაპაზონიდან მნიშვნელობების ამოღბა	var arr2=arr.slice(0,2)	'მარწყვი' ,'ატამი'
sort()	სორტირება / დალაგება	arr.sort()	'ატამი','მარწყვი','ხილი'
reverce()	შეტრიალება	arr.reverce()	'ხილი','ატამი','მარწყვი'
concat()	რამოდენიმე მასივის შეერთება	arr2= ['კივი' , 'ბუ'];	'მარწყვი' , 'ატამი','ხილი','კივი' , 'ბუ'
		arr.concat(arr2)	

5.4 მზა ფუნქციების გამოყენება

მიმდინარე პარაგრაფის თემატიკა

- დროისა და თარიღის ფუნქციები
- სტრიქონული ფუნქციები
- ბრაუზერთან სამუშაო ფუნქციები
- დოკუმენტთან სამუშაო ფუნქციები
- ტაიმერის ფუნქციები

ობიექტი ამომავალი წერტილია პროგრამირებაში. ყველაფერი მის გარშემო კრავს მოქმედ ჯაჭვს. რალურ რეჟიმში კონკრეტულ ობიექტთან მუშაობის პროცესი საკმაოდ საინტერესო, რამეთუ დიდი აბათობით ობიექტების უმრავლესობა საკმაო ინფორმაციისა და ფუნქციონალის მატარებელია.

ძირითად ობიექტებს, რომლებიც Javascript_ში აქტიურად გამოიყენება საკუთარი მზა თვისებებისა და მეთოდების / ფუნქციების ერთობლიობა გააჩნია, რომლებიც საკმაოდ ამარტივებს ობიექტთან მუშაობას და ხელს უწყობს დეველოპერს მარტივი ფუნქციონალის გამოყენებით ისეთი შედეგის მიღებაში, რომლის თავად შესაქმნელად შესაძლოა არც ისე მცირე დროითი რესურსის დახარჯვა მოუხდეს.

არსებულ ქვეთავში შეძლებისდაგვარად მაქსიმალური რაოდენობის მეთოდების გარჩევა მოხდება. და თუ შემთხვევით მათ გარდა კიდევ სხვა მეთოდებს მიაკვლიეთ არ იფიქროთ მორიგ მსოფლიო დონის აღმოჩენასთან გქონდეთ შეხება. გამომდინარე იქიდან, რომ ფუნქციათა რაოდენობა არც ისე მცირეა მათი სრულყოფილი ახნა ფაქტობრივად შეუძლებელია.

String_თან მუშაობის მეთოდები

იმედია შეხსენება არ გჭირდებათ, რომ Javascript_ში არსებობს ტექსტუალური ტიპის ინფორმაცია. ერთის შეხედით რას წარმოადგენს იგი? **String** ეს არის სიმბოლოთა ერთობლიობა.

```
var txt ='21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული';//ლიტერალი / მნიშვნელობა

var txt =new String('21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული'); //ობიექტი
```

არსებული შექმნის მეთოდებიდან პირველი მეტად პრიმიტიულია. ამასთანავე Javascript_ში ლიტერარს შეუძლია ობიექტის მეთოდების მიღება და მათთან მუშაობა, საჭიროების შემთხვევაში ინტერპრეტატორი ავტომატურად ახდენს მის გარდაქმნას ობიექტად.

თუ კარგად დაუკვირდებით შეატყოფთ, რომ ტექსტი შედგება სიმბოლოებისაგან, ე.ი ის თავის მხრივ წარმოადგენს ერთგვარ მასივს. შესაბამისად მასივის თვისება **length** არსებულო ობიექტის თვისებასაც წარმოადგენს. ის განსაზღვრავს ტექსტის სიგრძეს.

```
var txt ='21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული'; //ლიტერალი / მნიშვნელობა

alert ( txt.length )  //55
```

მასივის მსგავსად ტექსტის კონკრეტულ სიმბოლოსთან წვდომა შესაძლებელია მის ინდექსზე მიმართვით

```
var txt ='21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული';

alert ( txt[ 0 ] )  //2

alert ( txt[ 3 ] )  //ე
```

რა თქმა უნდა გარდა ამ მსგავსებისა მასივთან String_ს გააჩნია მეთოდთა ერთობლიობა რომელიც მოცემულია ცხრილ VII

ცხრილი VII

`var txt ='21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული';`

მეთოდი	აღწერა	მაგალითი	შედეგი
charAt()	კონკრეტულ ინდექსზე არსებული მნიშვნელობის გამოტანა, იდენტურია txt[…] ჩანაწერისა	txt.charAt(3)	ე
charCodeAt()	მისი საშუალებით შესაძლებელია დადგინდეს სიმბოლოს კოდური რიცხვითი მნიშვნელობა ASCII	txt.charCodeAt(3)	4608
String.fromChar-Code()	რიცხვითი მნიშვნელობის შესატყვისი სიმბოლოს მიღება	String.fromCharCode(97)	a
concat()	შეერთება ერთი ან რამოდენიმე ტექსტური ინფორმაციის	var txt2 ='ამბობენ, რომ ' txt.concat(txt2)	ამბობენ, რომ 21_ე საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული

indexOf()	ახორციელებს სიმბოლოს ძებნას ტექსტში და გამოაქვს მისი პირველი დამთხვევის ინდექსი, სიმბოლოს ვერ მოძებნის შემთხვევაში პასუხია (-1)	txt.indexOf('საუკუნე')	5
lastIndexOf()	ახორციელებს სიმბოლოს ძებნას ტექსტში და გამოაქვს მისი უკანასკნელი დამთხვევის ინდექსი.(ძიება ბოლოდან) სიმბოლოს ვერ მოძებნის შემთხვევაში პასუხია (-1)	txt.lastIndexOf('ულ')	52
replace ()	ტექსტის ჩანაცვლება	txt.replace('21_ე','XIX')	XIX საუკუნე ინფორმაციული ტექნოლოგიების ერადაა მიჩნეული
search()	ფაქტობრივად მუშაობს ისე როგორც მეთოდი **in-dexOf()**	txt.**search**('საუკუნე')	5
slice()	იღებს ტექსტიდან კონკრეტულ მონაკვეთს მითითებულს ფუნქციის არგუმენტებად	txt. **slice(5,11)**	საუკუნ
substring()	იდენტურია ფუნქციისა slice()	txt.**substring (5,11)**	საუკუნ
substr()	იღებს ტექსტიდან კონკრეტულ მონაკვეთს პირველი არგუმენტი საწყისია მეორე რაოდენობა	txt.**substr(5,11)**	საუკუნე ინ
toUpperCase()	რეგისტრის ცვლილება ზედა ინდექზე. ქართულ ტექსტზე აზრი არ აქვს გამოყენებას	var txt2 = 'hello' txt2.**toUpperCase()**	HELLO

toLowercase()	რეგისტრის ცვლილება ქვედა ინდექზე. ქართულ ტექსტზე აზრი არ აქვს გამოყენებას	var txt2 = 'HElIo' txt2. **toLowercase()**	hello

ტექსტური მნიშვნელობის გარდაქმნა რიცხვით მნიშვნელობად

არსებობს რიგი ფუნქციების რომლებიც უზრუნველყოფენ ტექსტური ტიპის ინფორმაციის გარდაქმნას რიცხვითტიპად. რა თქმენა უნდა თუ ტექსტუარი ტიპის ცლადიდან შესაძლებელია რიცხვითი მნიშვნელობის მიღება, წინააღმდეგ შემთხვევაში მიიღება NaN .არსებული მეთოდებია:

- parseInt()
- parseFloat()
- Number();

მოკლედ მიმოვიხილოდ ისინი:

მეთოდი parseInt() უზრუნველყოფს ტექსტიდან **მთელი** რიცხვის გამოყოფას, მხოლოდ იმ შემთხვევაში თუ ტექსტი იწყევა რიცხვითი მნიშვნელობით,წინააღმდეგ შემთხვევაში ნებისიმერი სიმბოლო აღიქმება არარიცხვით ტიპად და შედეგი იქნება NaN

```
alert(parseInt('21_ე საუკუნე'))                    //21
alert(parseInt('21.55'))                           //21
alert(parseInt('ტექსტით დაწყებული 21.55'))        //NaN
```

მეთოდი parseFloat() იღებს ათწილად მნიშვნელობებს. მაგ.:

```
alert(parseInt('21_ე საუკუნე'))                    //21
alert(parseInt('21.55'))                           //21.55
alert(parseInt('ტექსტით დაწყებული 21.55'))        //NaN
```

ხოლო მეთოდი Number() ახდენს მხოლოდ იმ ტექსტის დაყვანას როგორც მთელ ასევე აღწილად მნიშვნელობად რომელიც შედგება მხოლოდ რიცხვითი მნიშვნელობისაგან და ასევე წერტილისაგან „.“

```
alert(Number(true))                                //1
alert(Number(false))                               //0
alert(Number('21'))                                // 21
alert(Number('21.55'))                             // 21.55
alert(parseInt('21 55'))                           //NaN
alert(parseInt('ტექსტით დაწყებული 21.55'))        //NaN
```

დროისა და თარიღის ფუნქციები

Javascript დროსთან სამუშაოდ იყენებს ობიექტ Date(). მიმდინარე დროის შესაქმნელად გამოიყენება new Date() ობიექტი.

```
var now =newDate();
```

```
alert( now );
```

თუ კონსტრუქტორს გადაეცემა ერთი რიცხვითი არგუმენტი ის აღიქმება, როგორც მილიწამის (1/1000 წამი) მნიშვნელობა. თარიღის ამთვლელ წერტილად ნაგულისხმევია 1970 წლის 1 იანვარი GMT+0 (დროითი სარტყლის აღმნიშვნელია)

```
// 24 საათი 01.01.1970 GMT+0 წლის შემდეგ
var getD =newDate(1000*60 * 60 *24);
 // მილიწამი * 1000 = 1 წამი
// 60 * 60 = 1 საათი
// .... * 24 = 1 დღე
alert(getD); // შედეგი  02.01.19702 იანვარი 1970 წლის
```

დროის შექმნა შესაძლებელია სხვა არსებული პარამეტრების გადაწოდებით. სადაც წინასწარ არის განსაზღვრული შემდეგი თანმიმდევრობით:

```
var now =newDate(წელი, თვე, რიცხვი,საათი, წუთი, წამი, მილიწამი);

// 1 იანვარი 2016, 00:00:00
var date =newDate(2016,0,1,0,0,0,0);

// შედეგი იგივეა საათი,წუთი,წამი,მილიწამი ავტომატურად 0_ად არის განსაზღვრული
var date =newDate(2016,0,1);


// დროის შექმნა მილიწამების სიზუსტით
var date =newDate(2016,0,1,2,3,4,567);// 1.01.2016, 02:03:04.567
```

როგორც მაგალითიდან ჩანს აუცილებელი პირობაა:

- წელი შედგებოდეს 4 სიმბოლოიანი ჩანაწერისაგან (2016 და არა 16)
- თვის ათვლა იწყება ნულიდან და არა ერთიდან (იანვარი - 0)

გასათვალისწინებელია ისიც, რომ მოცემილი სტილით დროის შექმნისას აუცილებელი პირობა მინიმუმ 2 არგუმენტის (წელი, თვე) გადაცემაა. დანარჩენი არგუმენტების არ მიწოდების შემთხვევაში ისინი ავტომატურად 0_ებად აღიქმებიან და არაფერი დაშავდება.

დაიმახსოვრეთ Javascript_ში დროის გამოსახვა ლოკალური დროის მიხედვით ხდება და არა UTC (Universal Coordinated Time), ანალოგიურია GMT (Greenwich Mean Time)

დროით ობიექტს გააჩნია მთელი რიგი მეთდებისა, რომელიც უზრუნველყოფს არსებული დროის კონკრეტული მნიშნველობის წარმოჩინებას. მაგ: სასურველია ინფორმირებულნი ვიყოთ მხოლოდ მიმდინარე წლის შესახებ და საერთოდ არ არის საჭირო მთილიანი დროის (წელი,თვე,რიცხვი, საათი,წუთი,წამი,მილიწამი,) გამოსახვა.

ძირითადად დროით ობიექტის მეთდები ორი ფაქტობრივად ერთგვაროვანი სახითაა წარმოდგენილი, რაც უზრუნველყოფს როგორც ლოკალურ ასევე UTC დროსთან მუსაობას. ასევე არსებობს მეთოდთა ერთობლიობარომელიც უზრუნველყოფს ობიექტიდან მონაცემების მიღებას და პირიქით - მინიჭებას. მათგან get… თავსართით დაწყებყლი მეთდები შედეგს იღებს ობიექტიდან, ხოლო set… უზრუნველყოფს მინიჭებას.

ცხრილში VIII მოცემულია დროის ობიექტის მეთოდები განმარტებებითურთ.

ცხრილი VIII

მაგ.მიმდინარე დრო არის varnow=newDate();1 იანვარი 2016, 02:23:14.557

მეთოდი	აღწერა	მაგალითი	შედეგი
getFullYear()	მიიღება 4 ციფირანი წელი	now.**getFullYear**()	2016
getMonth()	გამოაქვს თვე (გაითვალისწინეთ საწყისი თვე 0_ით სიმბოლირდება)	now.**getMonth**()	0
getDaTe()	მიიღება თვის დღე 1_დან 31_მდე	now.**getDay**()	1
getHours()	მიიღება საათი	now.**getHours**()	2
getMinutes()	მიიღება წუთი	now.**getMinutes**()	23
getSeconds()	მიიღება წამი	now.**getSeconds**()	14
getMilliseconds()	მიიღება მილიწამი	now.**getMilliseconds**()	557
setFullYear()	წლის მინიჭება, შესაძლოა გადაეცეს თვეც და რიცხვიც	now.**setFullYear**(2022)	1 იანვარი 2022, 02:23:14.557

setMonth()	თვის მინიჭება, შესაძლებელია რიცხვისაც	now.**setMonth**(5)	1 ივნისი2016, 02:23:14.557
setDate()	რიცხვის მინიჭება	now.**setDate**(31)	31 იანვარი 2016,02:23:14.557
setHours()	მიენიჭება საათი, შესაძლოა წუთი,წამი,მილიწამიც	now.**setHours**(5)	31 იანვარი 2016,05:23:14.557
setMinutes()	ენიჭება წუთი, შესაძლოა წამი, მილიწამიც	now.**setMinutes**(35)	31 იანვარი 2016,02:35:14.557
setSeconds()	ენიჭება წამი, შესაძლოა მილიწამიც	now.**setSeconds**(19)	31 იანვარი 2016,02:23:19.557
setMilliseconds()	ენიჭება მილიწამი	now.**setMilliseconds**(2)	31 იანვარი 2016,02:23:14.2

აღსანიშნავია: ცხრილში ჩამოთვლილ ყველა მეთოდს გააჩნია მსგავსი მორგებული UTC_სარტყელზე. მაგ: **getUTCFullYear()**.

ბრაუზერთან სამუშაო ფუნქციები

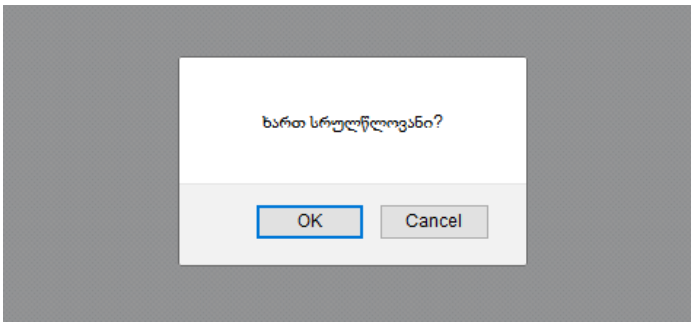
Javascript_ში window წარმოადგენს როგორც გლობალურ ელემენტს ასევე ბრაუზერის ფანჯარას. რეალურ რეჟიმში არსებულმა ობიექტმა განიცადა სრულყოფა და დროთა განმავლობაში მიიღო უამრავი დამატებითი ფუნქცია და მეთოდი. ბრაუზერთა სწრაფმა განვითარებამ შეცვალა დამოკიდებულება window ობიექტის ბევრი თვისებისადმი, გამომდინარე იქიდან რომ სხვადასხვა ბრაუზერი სხვადასხვა მეთოდებთან განსხვავებული თავსებადობით გამოიხატება. მაგრამ ეს სულაც არ ნიშანვს window ობიექტის მეთოდების ცოდნიაგან თავის არიდებას.

Javascript_ის პირველივე შემხებლობისას გავარჩიეთ გამოტანის ოპერატორი alert(). თუ იმ ლოგიკიდან გამოვალთ, რომ window წარმოადგენს სუპერ გლობალურ ობიექტს მაშინ ჩანაწერი window.alert() არ უნდა იყოს ლოგიკას მოკლებული. სინამდვილეში ასეცაა. მაგარამ გამომდინარე იქიდან, რომ window ყველა ობიექტის მშობელია სიმარტივის მიზნით მისი გამოუყენებლობა დასაშვებია. გარდა alert() მეთოდისა window ობიექტს გააჩნია რიგი მეთოდებისა, რომელიც აქტუალურობას არ კარგავს მიმდინარე ეტაპზეც. მათი სრული სია შეგიძლიათ იხილოთ შემდეგ მისამართზე http://www.w3schools.com/jsref/obj_window.asp . ჩვენს მიერ კი განსხილული იქნება რამოდენიმე მათთაგანი.(გამოყენებული იქნება სრული ჩანაწერი)

window.print()

არსებული ფუნქცია უზრუნველყოფს მითითებული ადგილის, არეალის ბეჭდვას.

confirm()

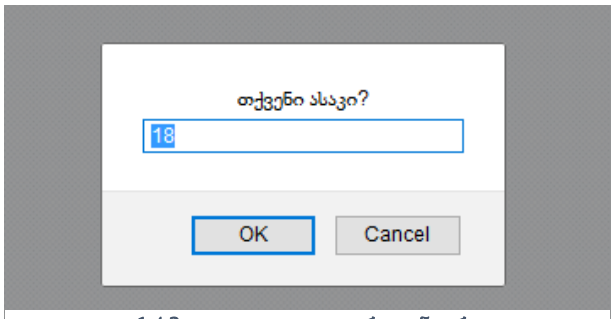


1.4.1 დიალოგური ფანჯარა confirm()

არსებობს დიალოგური ფანჯარა, რომლის მეშვეობითაც შესაძლებელია შეტყობინების გამოტანა ეკრანზე. (იხ.სურ.1.4.1) მისი სინტაქსი შემდგომია **confirm**("შეკითხვა"), დიალოგურ ფანჯარას გააჩნია ორი ღილაკი **OK** და **Cencel.** რომელთა გააქტიურებაც რეზულტატში გვაძლევს OK->**true** , Cencel->**false.**

prompt()

დიალოგური ფანჯარა **prompt()** შედგება 2 ნაწილისაგან ეს არის ტექსტის გამოტანის არეალი ინფორმაციის შესაყვანი ფორმა (იხ.სურ. 1.4.2). **prompt**("თქვენი ასაკი","18"). მეორე მნიშვნელობა default_ად სიცარიელეს წარმოად-გენს და მისი საერთოდ არ მითითებით არაფერი დაშავდება. **prompt**("თქვენი ასაკი")



1.4.2. prompt დიალოგური ფანჯარა

window მეთოდებიდან გამოსაყოფია **window.open(),** რომელიც ახალი ფანჯრის გახსნის ფუნქციონალს წარმოადგენს. მისი სრულყოფილი სინტაქსი შემდეგი საახისაა **window.open**("საიტის URL მისამართი","გახსნილი ფანჯრის სახელი","პარამეტრები"). **window.open()** მეთოდს შესაძლოა გადაეწოდოს მხოლოდ საიტის მისამართი დამატებითი პარამეტრებისა და სახელის გარეშე. ის უპრობლემოდ იმუშავებს.

```
window.open("http://google.ge");
```

არსებული მეთოდის ასეთი სახით გამოყენებისას გაიხსნება ახალი ფანჯარა ბრაუზერის ახალი ჩანართში - ტაბში . თნამედროვე ბრაუზერების უმეტეს ნაწილში მეთოდის ამგვარი ჩანაწერი ფანჯარას სწორედაც, რომ ახალ ჩანართში ხსნის და არა დამოუკიდებელ **popup** ფანჯრად.სრულყოფილი სინტაქსი უზრუნველყოფს ფანჯრის დამატებითი პარამეტრეის განსაზღვრას, რაც თავისთავად შესაძლებელს ხდისგახსნილი ფანჯარა დამოუკიდებელი ზომებით და პარამეტრებით მოგვევლინოს.

```
win = window.open(url, name, params)
```

url - ის წარმოადგენს გასახსნელი ფანჯრის მირამართს.

name- მიეთითება ფანჯრის სახელი. რამდენიმე ფანრის გახსნის მცდელობისას თუ სახელი ერთი იქნა ახალ ფანჯარაში ჩანაცვლდება საიტის ვიზუალი და ყოველ ჯერზე ახალში არ გაიხსნება.

params - პარამეტრები უზრუნველყოფს ფანჯრის სრულყოფას. აღსანიშნავია, რომ დღეს მათი ნაწილი სხვადასხვა ბრაუზერებში სხვადასხვანაირად გამოისახება, შესაძლოა ზოგმა საერთოდ არ იმუშაოს. ასე რომ ყურადღებით გაარჩიეთ დამოუკიდებლად სხვადასხვა პარამეტრის თავსებადობა ბრაზერებთან.იხ. ცხრილი IX

ცხრილი IX.

N	მეთოდი	აღწერა	მნიშვნელობა
1	**left**	ფანჯრის დაშორება ეკრანის / ბრაუზერის **მარცხენა** კიდიდან	რიცხვი
2	**top**	ფანჯრის დაშორება ეკრანის / ბრაუზერის **ზედა** კიდიდან	რიცხვი
3	**width**	ფანჯრის სიგანე	რიცხვი
4	**height**	ფანჯრის სიმაღლე	რიცხვი
5	**menubar**	ბრაუზერის მენიუს ზოლის დამალვა/გამოჩენა	yes/no
6	**toolbar**	ნავიგაციის პანელის დამალვა / გამოჩენა	yes/no
7	**location**	საიტის URL სამისამართე ველის დამალვა / გამოჩენა	yes/no
8	**statusbar**	სტატუსის ველი	yes/no
9	**resizable**	ბრაუზერის ფანჯრის დაპატარავება / გაზრდა	yes/no
10	**scrollbar**	სქროლის დამალვა / გამოჩენა	yes/no

როგორც აღვნიშნეთ მოცემული პარამეტრები დღევანდელ ბრაუზერებში, არც ისე სრულყოფილად მუშაობს, მაგრამ თუ ახალი ფანჯრის გახსნა არა ბრაუზერის ჩანართად (ტაბი) ჩამატება არამედ ფანჯრად ვიზუალიზება გსურთ აუცილებელია width და height პარამეტრების მინიჭება.

```
var newWin = window.open("http://google.ge/","google","width=200,height=200");
```

გახსნილი ფანჯრის დასახურად გამოიყენება window.close() ფუნქცია, რომელიც სხვადასხვა ბრაუზერებში განსხვავებულად მოქმედებს აქტიური ფანჯრის პირობებში, ზოგან ხურავს ავტომატურად, ზოგან გამოაქვს შეკითხვის ფორმა დაახლოებით „დარწმუნებულნი ხართ რომ ნამდვილად გინდათ ფანჯრის დახურვა?“ **newWin -** ფანჯრის დასახურად გამოიყენეთ შემდეგი კონსტრუქცია **newWin.close();**

არსებობს ასევე ფანჯრის გადატანის / გადაადგილების, ზომის ცვლილებისა და სქროლის ფუნქციები მეთოდები

- **win.moveBy(x,y)** - ფანჯრის მოძრაობა - წანაცვლება არსებული პოზიციიდან x , y მნიშვნელობების შესაბამისად. მაგ. ფანჯრის საწყისი მდებარეობაა left-0; top-0; win.moveBy(100,100) მეთოდის გამოყენებთ პირველ ნაბიჯზე ფანჯარა განთავსდება **left-100px**; **top-100;** შემდეგი მოქმედებისას **left-200px; top-200;** და ა.შ

- **win.moveTo(x,y)** - ფანჯრის გადაადგილება ეკრანის / ბრაუზერის x, y კოორდინატებზე. მაგ. ფანჯრის საწყისი მდებარეობაა left-0; top-0; ხოლოwin.moveTo(100,100) მეთოდის გამოყენებთ ფანჯარა განთავსდება ფიქსეირებულ**left-100px;top-100;**

- **win.resizeBy(width,height)** - მითითებული ფუნქციით შესაძლებელია არსებული ფანჯრის ზომების ცვლა მისი საწყისი ზომიდან გამომდინარემრავალჯერადად

- **win.resizeTo(width,height)** მითითებული ფუნქციით შესაძლებელია არსებული ფანჯრის ზომის ცვლა ერთჯერადად.

- **win.resizeTo(width,height)** მითითებული ფუნქციით შესაძლებელია არსებული ფანჯრის ზომის ცვლა ერთჯერადად.

- **win.scrollBy(x,y)** მითითებული ფუნქციით შესაძლებელია სკროლის გადატანა x, y პოზიციებზე მრავალჯერადად

- **win.scrollTo(x,y)** მითითებული ფუნქციით შესაძლებელია სკროლის გადატანა x, y პოზიციებზე ერთჯერადად

object screen

მომხმარებლის მონიტორის რეზოლუციის დასადგენად გამოიყენება ობიექტი screen. მისი ჩაწერა დასაშვებია, როგორც window ობიექტთან, ასევე მის გარეშე :

screen

window.screen

screen ობიექტის თვისებათა ერთობლიობაწარმოდგენილია ცხრილ X:

ცხრილი X.

N	მეთოდი	აღწერა
1	**screen.width**	გამოაქვს მონიტორის სრული სიგანე
2	**screen.height**	გამოაქვს მონიტორის სრული სიმაღლე

3	**screen.availWidth**	გამოაქვს მხოლოდ გამოყენებადი არეალის სიგანე (მაგ.: თუ სტატრის ზოლი ან დამატებითი ინსტრუმენტების პანელი გამოყენებულია მარჯვენა ან მარცხენა მხარეს)
4	**screen.availWidth**	გამოაქვს მხოლოდ გამოყენებადი არეალის სიმაღლე (მაგ.: თუ სტანდარტულ მდგომარეობაშია windows ინსტრუმენტების პანელი განთავსებულია ქვედა პანელზე და მისი სიმაღლეა 40 px რაც აკლდება სრულ სიმაღლეს)
5	**screen.colorDepth**	გამოაქვს ფერთა პალიტრის ბიტური მნიშვნელობა

object Navigator

ობიექტიNavigator შეიცავს ინფორმაციას მომხმარებლის ბრაუზერის შესახებ. არსებული ობიექტის გამოყენებით შესაძლებელია დადგენა თუ რომელ ბრაუზერს იყენებს მომხმარებელი.

navigator ობიექტსგააჩნია შემდეგი თვისებები (იხ. ცხრილ XI):

ცხრილი XI.

თვისებები	აღწერა
navigator.appName	გამოაქვს ბრაუზერის სახელი
navigator.appCodeName	გამოაქვს ბრაუზერის კოდური დასახელება
navigator.appVersion	გამოაქვს ბრაუზერის ვერსია დასახელება
navigator.cookieEnabled	ახდენს დადგენას ჩართულია თუ არა cookie_ების მხარდაჭერის რეჟიმი
navigator.platform	ადგენს ოპერაციულ სისტემას თუ რაზეამორგებული ბრაუზერი
navigator.userAgent	ახდენს მომხმარებლის ბრაუზერის სათაურის გამოტანას, რომელსაც უგზავნის ბრაუზერი სერვერს გვერდის მოთხოვნის მომენტში

object History

ობიექტი**History** ინახავს გვერდების ნავიგაციის მონაცემებს. მას გააჩნია 2 მეთოდი, რომელიც უზრუნველყოფს წინა და მომდევნო მისამართებზე გადასვლას:

- **history.back()** - უკან
- **history.forward()** - წინ

1.5. სანავიგაციო ისრები

ისინი იდენტურ მოქმედებებს ასრულებელ რასაც მოიცავს სურათ 1.5_ზე მონიშნული ისრები.

object Location

Location ობიექტი უზრუნველყოფს სამისამართე პანელი ინფორმაციის მართვას. მისი ფუნქციაა მიიღოს / მიანიჭოს URL მისამათი და მისი კომპონენტები.

Location ობიექტს გააჩნია რიგი მეთოდებისა და თვისებებისა. ცხრილ XII მოცემულია იმ თვისებების ჩამონათვალი, რომესაც იყენებს Location ობიექტი

ცხრილი XII

მაგ. მისამართი შემდეგიაhttp://www.google.com:80/search?q=javascript#test

მეთოდი	აღწერა	მნიშვნელობა
location.hash	URL_ის ნაწილი რომელიც გამოსახულია სიმბოლო '#'_ის მერე	#test
location.host	ჰოსტი და პორტი ერთდროულად	www.google.com:80
location.href	მთლიანი URL მისამართი	http://www.google.com:80/search?q=javascript#test
location.hostname	ჰოსტი პორტის გარეშე	www.google.com
location.pathname	გზა საქაღალდისკენ	/search
location.port	გამოაქვს პორტი	80
location.protocol	პროტოკოლი	http:
location.search	ტექსტი სიმბოლო „?“ შემდგომ	?q=javascript

გარდა თვისებებია Location ობიექტს გააჩნია შემდეგი მეთოდები:

- **location.assign(url)** - გადაეცემა url მისამართი, რომელზეც ახდებს გადამისამართებას. ახდენს მითითებული მისამართის დოკუმენტის ჩატვირთვას
- **location.replace(url)** - ახდენს არსებული დოკუმენტის ჩანაცვლებას იმ დოკუმენტით რომელიც იტვირთება მითითებული url_ის შემთხვევაში. (დოკუმენთში მოიაზრება საიტის გვერდი). **replace()** მეთოდი **assign()**_სგან განსხვავდება იმით, რომ ის არ ინახება ისტორიაში შესაბამისად replace() მეთოდით გახსნილ გვერდს არ გააჩნია **back** უკან დაბრუნების ღილაკი ის დეაქტივირებულია.
- **location.reload()** - არსებული მეთოდის მეშვეობით შესაძლებელია მიმდინარე გვერდის განახლება. შესაძლოა მეთოდს გადავცეთ ატრიბუტი **true** ან **false** . თუ მნიშვნელობა იქნება **true** განიცდის განახლებას ყოველთვის სერვერიდან, ხოლო **false** . ის შემთხვევაში ბრაუზერის ქეშიდან.

ტაიმერის ფუნქციები

ზემოთმულ მასალაზე დაყრდნობით შეგვიძლია დავასკვნათ, რომ ობიექტი window საკმაოდ მნიშვნელოვანი ფუნქციების გამაერთიანებელია. ვებ გვერდზე ხშირია მოცემულობა, როცა საჭიროა გარკვეული ინფორმაციის პერიოდული გამეორება, ან რომელიმე ინფორმაციის ერთჯერადად, მაგრამ კონკრეტული დროის შემდგომ ასახვაა საჭირო. window ობიექტის ძირითად მეთოდებად მოიხსენიება „ტაიმერის“ ფუნქციები: **setInterval( )** და **setTimeout**().

ორივე მეთოდის სინტაქსი ანალოგურია განსხვავებულია მათი მუშაობის პრინციპი. **setInterval( )** უნრუნველყოფს მოქმედების მრავალჯერადად გამეორებადობას, ხოლო **setTimeout**() თვისი არსით ერთჯერადად. რა თქმა უნდა შესაძლებელია ისინი იდენტური ფუნქციონალით ვამუშაოთ, მაგრამ ყოველი აზრს არის მოკლებული, რადგანაც ისინი იმიტომ არსებობენ, რომ გამოიყენოთ დანიშნულებისამებრ.

მეთოდის სინტაქსი შემდეგია: ის შედგება ფუნცქიისა და დროის მნიშვნელობისაგან, შესაძლოა გამოყენებულ იქნას არგუმენთებიც. დრო განისაზღვრება მილიწამებად (1 წთ - 1000 მილიწამი)

```
var timerId =setTimeout(func / code, delay[, arg1, arg2...])
```

განხილვისას შეგვხვდება ტერმინი ფუნქცია და მისი გამოყენება, რომელიც შემდეგ 5.5 თავშია განხილული, უნჯობესია გაურკვევლობის შემთხვევაში ეწვიოთ მითითებულ თავს.

ამოცანა:

> საიტის გახსნიდან 2 წთ_ში გამოგვიტანოს ერთჯერადად alert("მოგესალმებათ საიტი") ფანჯარა შეტყობინებით

ამოცანის გადასაჭრეად საჭიროა setTimeout() ფუნქციის გამოყენება. არსებული დავალების გადასაწყვეტათ გამოვიყენოთ ყველა შესაძლო გზა, რაც გაამარტივებს setTimeout() ფუნქციის არსის გაგებას.

გზა I - შეიქმნას ანონიმური ფუნქცია უშუალოდ setTimeout() მეთოდში

```
setTimeout( function (){ alert("მოგესალმებათსაიტი") } , 2000 )
```

ფუნქცია ვიზუალურად საკმაოდ სასიამოვნოდ გამოიყურება, უბრალოდ დიდი კოდის შემთხვევაში ერთობ მოუხერხებელია. ამისათვის უნჯობესია შეიქმნას ცალკე ფუნქცია.

გზა II ბევრად უფრო მისაღები და კლასიკურია

```
functionshowMessage(){
  alert("მოგესალმებათსაიტი")
}

setTimeout(showMessage, 2000 )
```

არსებული დავალება იმდენად მარტივია შესაძლებელია ფუნქციის გამოყენებისაგან საერთოდ თავის შეკავება და alert() მეთოდის სტრინგად გამოყენება

გზა III

```
setTimeout("alert("მოგესალმებათსაიტი")", 2000 )
```

არსებული ფუნქციაც ანალოგურად იმუშავებს, მაგრამ ამგვარი ჩანაწერი მაინც არარეკომენდირებულია.

თუ ფუნქცია იღებს მნიშვნელობად არგუმენტებს მაშინ დროით მნიშვნელობის შემდეგ მძიმის მეშვეობით შესაძლებელია არგუმენტების გადაცემა. მაგ:

```
functionsum(a,b,c){
  alert(a+b+c)
}

setTimeout(sum, 2000, 5,25,15 )   // 2 წთ _ ში ეკრანზე გამოდის შედეგი 45
```

არსებული ფუნქციის განულება, მწყობრიდან გამოყვანა შესახძლებელია **clearTimeout**()ფუნქციის მეშვეობით. რომელსაც არგუმენტად გადაეცემა ფუნქციის სახელი/ცვლადი რომელსაც ის მიენიჭება გამოცხადება/გამოძახებისას:

```
functionsum(a,b,c){
alert(a+b+c)
}

var t = setTimeout(sum, 2000, 5,25,15 )

clearTimeout(t) // ეკრანზე შედეგი აღარ გამოვა
```

ანალოგური სინტაქტი აქვს **setInterval**() მეთოდს. მისი შესაძლებელია **clearInterval**()მეთოდის საშუალებით.

დავალება:

ეკრანზე გამოვიდეს მთვლელი, რომელიც 10_დან უკუთვლით წამოვა 1_მდე 1 წამის ინტერვალით. და საბოლოოდ გადაამისამართოს გვერდი google.com_ზე.

```
var i = 10;
functiontimer(){
  alert(i)
```

```
  if (i>1){
    i--;
   }else{
    clearInterval(t);
   location.assign("http://google.com");
   }

}

var t = setInterval (timer, 1000 )
```

DOM

კლიენტზე ორიენტირებული Javascript_ის შესაძლებლობებშოი შედის სტატიკური HTML დოკუმენტის დინამიურად გარდაქმნა, ასევე ინტერაქციული ეფექტების შემუშავება. მისი მთავარი ორიენტირი ვებ გვერდების შემცველობასთან მუშაობაა. ცალკე საუბრის თემაა ობიექტები,რომლებიც წარმოადგენს უშუალოდ ვეგ ვერდის შემცველობას.

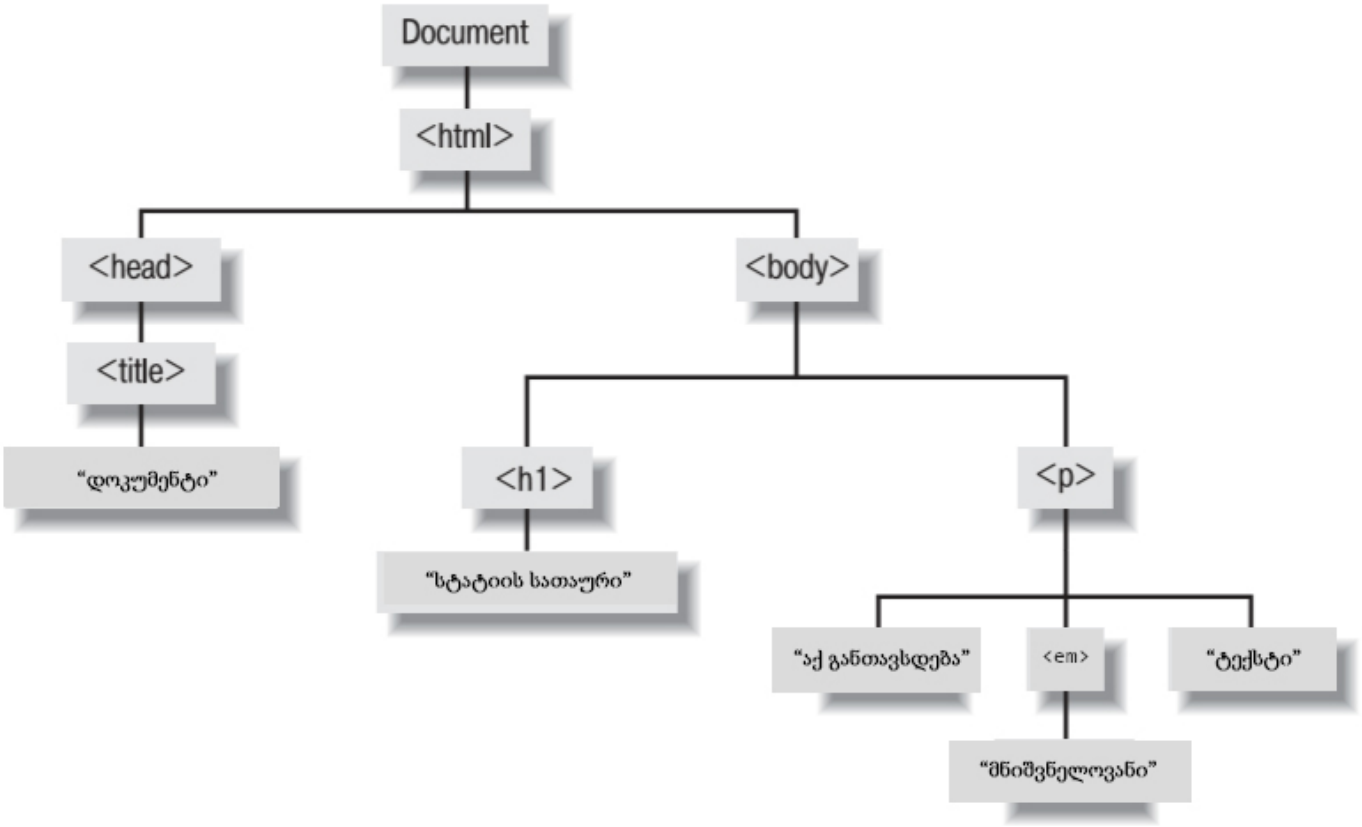
HTML დოკუმენტი შეიცავს ტექსტებს, სურათებს, ჰიპერბმულებს, ფორმის ელემენტებს და ა.შ. Javascript შეუძლია არსებულ ობიექტებთან მუშაობა და მათი მანიპულირება.

DOM (Document Object Model) დოკუმენტის ობიექტური მოდელი - ეს არის დოკუმენტის ობიექტებთან წვდომის განმსაზღვრელი ინტერფეისი.

HTML დოკუმენტი წარმოადგენს ერთმანეთში ჩადგმული ტეგების იერარქიულ სტრუქტურას, რომელიც DOM _ში წარმოდგენილია, როგორც „ობიექტების ხე“. ხის **კვანძები** წარმოადგნენ დოკუმენტის სხვადასხვა ტიპის ინფორმაციას, როგორიცაა HTML ელემენტები/ტეგები, ტექსტური ინფორმაცია, კომენტარები და ა.შ

მაგალითზე მოცემულია მარტივი HTML დოკუმენტი რომლის DOM სტრუქტურა წარმოდგენილია სურათ 1.6_ზე

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<h1>სტატიის სათაური</h1>
<p>აქ განთავსდება <em>მნიშვნელოვანი</em> ტექსტი</p>
</body>
</html>
```



1.6 დოკუმენტის DOM სტრუქტურა

კვანძები ერთმანეთთან მიმართბაში შეიძლება „გენიალოგიურ“ ხის სტრუქტურას მივამსგავსოთ, სადაც კვანძი შესაძლოა წარმოადგენდეს **მშობელ** კვანძს (მაგ.:html). კვანძები რომლებიც განლაგებულნი არიან ერთი დონით ქვემოთ მშობელ კვანძებზე იწოდებიან **შვილეული** კვანძებად (html კვანძსთვის ->head და body). ერთდონეზე მდებარე კვანძებს რომლებსაც გააჩნიათ ერთი მშობელი ერთმანეთისათვის წარმოადგენენ **დედმამიშვილ** კვანძებს (მაგ.: head და body ერთმანეთისათვის, ასევე h1 და p)

DOM სტრუქტურაში კვანძი მოხსენიებულია როგორც - **Node.** სურათ 1.6_ზე წარმოდგენილია სხვადასხვა ტიპის Node_ები (კვანძები). ესენია: დოკუმენტი, ელემენტი, ტექსტი. ცხრილ XIII წარმოდგენილია ძირითადი კვანძები შესაბამისი ტიპის მნიშვნელობით. სრული სია იხილეთ http://www.w3schools.com/jsref/prop_node_nodetype.asp

ცხრილი XIII

ინტრეფეისი	კონსტანტა nodeType	მნიშვნელობა
Element	Node.ELEMENT_NODE	1
Text	Node.Text_NODE	3
Document	Node.DOCUMENT_NODE	9
Comment	Node.COMMENT_NODE	8
DocumentFragment	Node.DOCUMENT_FRAGMENT_NODE	11

Attr	Node.ATTRIBUTE_NODE	2

DOM სტრუქტურის ფუძისეულ კვანძს წარმოადგენს Document ობიექტი.შევეცადოთ გადავიდეთ უშუალოდ სხვა ელემენტების მიმართიანობაზე და ვნახოთ როგორ ხდება მათთან წვდომა.

document.documentElementახდენს წვდომას <html>ტეგთან. არსებობს თვისება რომელიც გამოსახავს ეკრანზე კვანძის სახელს **nodeName.**

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<h1>სტატიის სათაური</h1>
<script>
        alert( document.documentElement.nodeName ) // html
</script>
</body>
</html>
```

ძირითადად მიმართვიანობა არა html არამედ body ტეგზე ხდება ამისათვის გამოიყენება შემდეგი ჩანაწერი**document.body**

უშუალოდინდივიდუალურადელემენტებზემიმართვისათვიშესაძლოაგამოიყენოთ:

- **document.getElementById()-** ელემენტზე მიმართვა კონკრეტული id_ის მიხედვით;
- **document.getElementsByClassName()**-მიმართვა ელემენტზე class სელექტორის მიხედვით
- **document.getElementsByTagName()**-ელემენტებზე ტეგების სახელწოდებებით მიმართვა
- **document.getElementsByName()-** ელემენენტებზე ატრიბუტ name_ის მიხედვით მიმართვა

გასათვალისწინებელია, რომ getElement მხოლობითში იხმარება მხოლოდ ID_სთან მიმართებაში, რადგანაც საიტზე შესაძლებელაი მხოლოდერთი განუმეორებელი id მქონე ელემენტის არსებობა. დანარჩენ შემთხვევებში ერთი და იმავე კლასის, ტეგის სახელისა და ატრიბუტ name_ის ელემენტები მრვალად შეგხვდეს. მათ ოდნავ მოგვინაებით შევეხოთ. იქამდე გავიგოთ არსი უშუალოდ მიმართვიანობისა.

დავალება:

Javascript_ის მეშვეობით შევიტანოთ ტეგში ტექსტი.

მოცემული დავალების შესასრულებლად საჭიროა მოვახდინოთ ტეგზე მიმართვიანობა, და გამოვიყენოთ მეთოდი**innerHTML** მისთვის ინფორმაციის მისანიჭებლად:

```
<html>
<head>
<title>დოკუმენტი</title>
```

```
</head>
<body>
<h1id="caption"></h1>// საწყის ეტაპზე h1  ცარიელია
<script>
        document.getElementById("caption").innerHTML = "საიტის სათაური";
// არსებული ჩანაწერის გამოყენებით ტექსტ - „საიტის სათაური“ ხელოვნურად მოვათავსებთ ტეგ
<h1>_ში და მიიღება შედეგი  <h1id="caption">საიტის სათაური</h1>
</script>
</body>
</html>
```

კონკრეტული თვისების უმრავლესობა ორმაგი ხასიათისაა და როგორც მინიჭების ასევე არსებული ელემენტის თვისების ამოღების უნარი გააჩნია. მაგ.: იმავე innerHTML_ის გამოყენება შემდეგ სახეს მოგვცემს

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<h1id="caption">განთავსებულია სათაური </h1>
<script>
        alert( document.getElementById("caption").innerHTML )
// არსებული ჩანაწერის გამოყენებით ეკრაბზე გამოვა შედეგი  “განთავსებულია სათაური”
</script>
</body>
</html>
```

არსებული მოცემულით შესაძლებელია ასევე საკმაოდ მარტივი და საინტერესო დავალების შესრულება.: მოადინეთ ერთ ტეგში არსებული ტექსტის მეორეში კოპირება და ჩანაწერის **„ეს საინტერესოა**“_ს თნდართვა;

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<h1id="caption">განთავსებულია სათაური </h1>
<em id="copyText"></em>
<script>
        document.getElementById("copyText").innerHTML  = document.getElementById("caption") .in-
nerHTML + "ეს საინტერესოა"
</script>
</body>
```

```
</html>
```

შედეგი ბრაუზერში

<h1>

Javascript_ის მეშვეობით შესაძლებელია კონკრეტული ელემენტების სტილირება. რა თქმა უნდა სტილების სრულფასობანი გაწერა აზრს მოკლებულია და მიის გამოყენება მხოლოდ კონკრეტული ლოგიკის შესაბამისადაა საჭირო. მაგ. საჭიროა ღილაკზე ხელის დაჭერისას ბლოკის გამოცენა / გაქრობა. რა თქმა უნდა ასეთ შემთხვაში დასაშვებბია სტილების მინიჭება.

არსებობს სტილის მინიჭების ორი გზა. ეს არის ობოექტ **style**გამოყენება ან ატრიბუტის მინიჭება. ორივე შემთხვევაში სტილები ელემენტს ენიჭება ხაზოვანი სტილის მინიჭების მეშვეობით (Inline style):

```
<h1id="caption">განთავსებულია სათაური </h1>
<script>
        document.getElementById("caption ").style.color ="red"  // h1 ტექსტი გახდება წითლი
</script>
```

მაგალითზე მოცემული ჩანაწერით შესაძლებელია 1 სტილის მინიჭება და ყოველი ახალი სტილის მიცემისათვის გამოყენებული უნდა იქნას ახალი ჩანაწერი.

```
<h1id="caption">განთავსებულია სათაური </h1>
<script>
  document.getElementById("caption ").style.color ="red"  // h1 ტექსტი გახდება წითლი
  document.getElementById("caption ").style.backgroundColor ="#fff44c"  // h1 ფონი - ყვითელი
</script>
```

Javascript_ით სტილების ბრძანენების სრული ჩამონათვალი შეგიძლიათ იხილოთ შემდეგ მისამართზე http://www.w3schools.com/jsref/dom_obj_style.asp

DOM სტრუქტურის მეშევეობით შესაძლებელია კონკრეტულ ელემენტის ატრიბუტზე წვდომა, მისი მინიჭება ან კითხვადობა. style წარმოადგენს ელემენტის ატრიბუტს როცა ის გაწერილია ხაზოვან რეჟიმში. თუ ერთდორულად რამოდენიმე სტილის მინიჭება გჭირდებათ სასურველია გამოიყენოთ მეორე გზა.

```
<h1 id="caption">განთავსებულია სათაური </h1>
<script>
  document.getElementById("caption ").setAttribute("style","color: red; background-color: #fff44c; font-size: 15px" )
</script>
```

setAttribute() მეთოდის გამოყენებით შესაძლებელაი ნებისმიერი ატრიბუტი მინიჭება html ტეგებისათვის. არსებული მეთოდი მოითხოვს ორ არგუმენტს: ეს არის ატრიბუტის სახელი რომელსაც ანიჭებთ და მნიშვნელობა რომელიც მიეკუთვნება ამ ატრიბუტს.

მაგ.: setAttribute() მეთოდის მეშვეობით მივანიჭოთ ელემენტს **class="logo"**

```
<h1id="caption">განთავსებულია სათაური </h1>
<script>
  document.getElementById("caption ").setAttribute("class","logo" )
</script>
```

მიღებულ შედეგში ტეგ h1 ექნება დამატებული ატრიბუტი class

```
<h1 id="caption" class="logo" >განთავსებულია სათაური </h1>
```

გარდა ელემენტის ატრიბუტის მინიჭებისა შესაძლებელია მოხდეს ელემენტის ატრიბუტის დადგენა **getAttribute()** მეთოდის მეშვეობით, რომელიც თავის მხრივ ერთ არგუმენტს საჭიროებს, თუ რომელი ატტრიბუტის მნიშვნელობის დადენა გვინდა:

მაგ.:

```
<h1 id="caption" class="logo" >განთავსებულია სათაური </h1>

<script>
alert(document.getElementById("caption ").getAttribute("class" )// logo
)
</script>
```

გარდა **document.getElementById()** მიმართვისა შესაძლებელია ელემენტებზე მიმართვა ტეგის სახელისა, კლასებისა და ატრიბუტ name_ის მეშვეობით. ისინი თავის მხრივ საიტზე რამოდენიმე ერთფგვაროვანი შესაძლოა იყოს. ამიტომ ისინი აღიქმებიან როგორც მასივი და უშუალოთ მათგან ერთ ან რამოდენიმეზე მიმართვისას საჭიროა დაკონკრეტება იმ **ინდექსისა** თუ რომელს უკავშირდებით. მაგ.:

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<div class="container">ტექსტი 3</div>
<h1id="caption">განთავსებულია სათაური </h1>
<h2>განთავსებულია  მეორე სათაური </h2>
<p>ვრცელი ტექსტი 1 </p>
<h2>განთავსებულია მესამე სათაური </h2>
<p>ვრცელი ტექსტი 2 </p>
```

```
<div class="container">ტექსტი 5</div>

<div class="about"> ტექსტი კომპანიის შესახებ </div>

<input type="radio" name="gender" />
<input type="radio" name="gender" />
<div class="container">ტექსტი 4</div>

</body>
</html>
```

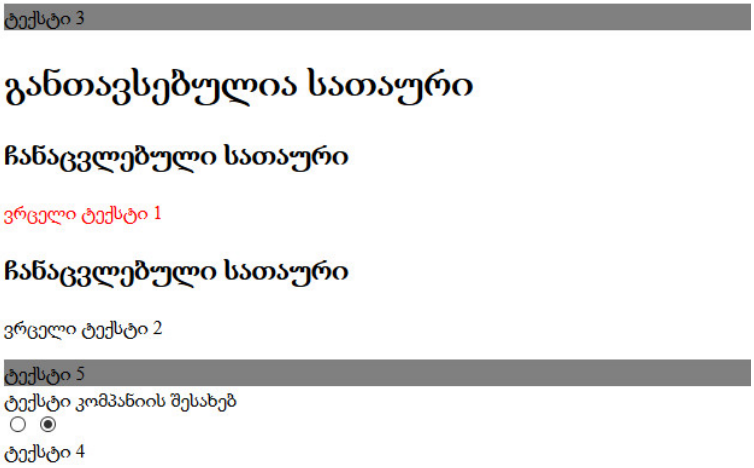
არსებულ კოდზე დაყრდნობით სასურველია შემდეგი სტილის დოკუმნტის მიღება:

- ყველა h2_ში გამოისახოს ტექსტი 'ჩანაცვლებული სათაური';
- მხოლოდ პირველი <p> ტეგის ფერი გახდეს წითელი;
- საიტზე ნებისმიერ ადგილას განთავსებული პირველი და მეორე class="container"ფონად გამოესახოს სერი ფერი;
- მხოლოდ მეორე რადიო name_ით **gender**საწყის მნიშვნელობადიყოს მონიშნული(checked).

```
<script>
// 1 - ყველა h2_შიგამოისახოსტექსტი 'ჩანაცვლებულისათაური';

// გამომდინარე იქიდან რომ არ ვიცით წინასწარ რამდენი ელემენტი h2 იქნება საიტზე გამოყენებული
უმჯობესია ციკლის მეშვეობთ ჩავწვდეთ და ყველა h2 მივაგნოთ ეს მეტად დინამიური იქნება :


for(var i =0; i <document.getElementsByTagName("h2").length; i++){
            document.getElementsByTagName("h2")[i].innerHTML = "ჩანაცვლებული სათაური"
}

// 2- მხოლოდ პირბელი p ტეგის გერი გახდეს წითელი
document.getElementsByTagName ("p")[0].style.color = "red"

// 3-  მხოლოდ პირბელი p ტეგის გერი გახდეს წითელი
document.getElementsByClassName("container")[0].style.backgroundColor = "gray";
document.getElementsByClassName("container")[1].style.backgroundColor = "gray";

შესაძლოა ეს ჩანაწერი შემდეგნაირადაც გამოისახოს
document.getElementsByClassName("container")[0].style.backgroundColor =
document.getElementsByClassName("container")[1].style.backgroundColor = "gray";


// 4 - მხოლოდ მეორე რადიო იყოს cheked
document.getElementsByName("gender")[1].setAttribute("checked","checked")
```

```
// უფრო მოკლე გზაა
document.getElementsByName("gender")[1].checked = true
</script>
```

მოცემილი კოდის მანიპულირებით მიღებული შედეგი გამოსახულია 1.6.2 სურათზე. შეეცადეთ გაარჩიოთ მოცემილი მაგალითი, ააწყოთ თქვენთვის საურველი html დოკუმეტის ჩონჩხი და მოახდინოთ წვდომა ტეგებთან, ცვალოთ მათი სტილები და ინფორმაცია.



1.6.2 სტილშეცვლილი დოკუმენტი

Dom სტრუქტუს მეშვეობით შესაძლებელია ახალი ელემენტების, ტექსტური node_ების შექმნა მათთვის ატრიბუტების გენერირება და მინიჭება, ასევე უკვე არსებული ელმენტების წაშლა. მიდგომა საკმაოდ ხშირად გამოყენებადია. მისი საშუალებით სესაძლებელია html დოკუმენტის ჩონჩხის ცვლილება.

ახალი ელემენტის შესაქმნელად გამოიყენეთ მეთდი **createElement().** ის მოითხოვს ატრიბუტად მხოლოდ ტეგის დასახელებას და თავად ახდენს მის ტეგად გარდაქმნას.

```
<script>
        document.createElement("div");
</script>
```

მაგრამ ტეგის შექმნა მის ავტომატურ დამატებას html ჩონჩხში არ ნიშნავს, ასევე ის ცარიელია, შევეცადოთ შევუქმნათ მას ტექსტური შემცველობა, მივანიჭოთ ატრიბუთი id="newElement". და მხოლოდ ამის შემდეგ განვათავსოთ საჭირო ადგილას ჩვენს მიერ წინასწარ შექმნილ html დოკუმენტში

```
<script>
//<div>ელემენტის შექმნა
var elem = document.createElement("div");

//  ატრიბუტ id შექმნა
var attr = document.createAttribute("id");

//  ატრიბუტ id_ის მნიშვნელობის განსაზღვრა
attr.value = " newElement";

//  ატრიბუტის div ელემენტზე მიბმა / მინიჭება
elem.setAttributeNode(attr); // არსებულ დონეზე მიღებული შედეგია <div id='newElement'></div>
```

```
// ტექსტური კვანძის - ტექსტური ინფორმაციის შექმნა
var txt = document.createTextNode("ტექსტი ახალ ელემენტში");

// ტექსტური ინფორმაციის div _ში განთავსება
elem.appendChild(txt) // არსებულ დონეზე მიღებული შედეგია <div id='newElement'>"ტექსტი ახალ ელემენტში"</div>


// div_ის ბოლო ელემენტად განთავსება body_ტეგში
document.body.appendChild(elem);
</script>
```

მოცემული სკრიპტი შემდეგნაირად იმუსავებს უშუალოდ html დოკუმენტში:

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<div class="container">ტექსტი 3</div>
<h1id="caption">განთავსებულია სათაური </h1>
<div class="about"> ტექსტი კომპანიის შესახებ </div>

// body_ის უკანასკნელ ელემენტად დააამატებს
//-------------------------------------------------------------
<div id="newElement"> ტექსტი ახალ ელემენტში </div>
//-------------------------------------------------------------
<script>
var elem = document.createElement("div");
var attr = document.createAttribute("id");
attr.value = " newElement";
elem.setAttributeNode(attr);
var txt = document.createTextNode("ტექსტი ახალ ელემენტში");
elem.appendChild(txt)
document.body.appendChild(elem);
</script>

</body>
</html>
```

თუ არსებული ბლოკის არა ბოლოში არამედ კონკრეტულ ადგილას განთავსებაა საჭირო საჭიროა მეთოდ **insettBefore()** გამოყენება. ის მოითხოვს ორ ატრიბუტს, პირველი ეს არის რას ვსვამტ, მეორე რომელი ბრძანების წინ.

ზედა მაგალითზე დაყრდნობით შექმნილი ბლოკი არა body ტეგის ბოლოში არამედ ..... <h1 id="caption"></h1>ტეგებს შორი განთვსდეს (წერტილების ადგილას).

ამისათვის შაჭიროა scritp_ის ბოლო ბრძანება document.body.**appendChild**(elem) შეიცვალოს

```
document.body.insertBefore(elem, document.getElementById("caption"))
```

Dom_ის მეშვეობით შესაძლებელია კონკრეტული ელემენტის პირველ(**firstChild**) და უკანასკნელ (**lastChild**) ელემენტებზე წვდომა. ასევე ერთ დონეზე განლაგებულ შემდეგ (**nextSibling**) და წინა (**previousSibling**) ელემენტებზე.

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
<body>
<div>ტექსტი 1 </div>
<ul id="partner">
<li> განათლების სამინისტრო </li>
<li> ჯანდაცვის სამინისტრო </li>
<li> სპორტისა და ახალგაზდობის სამინისტრო </li>
<li> ენერგეტიკის სამინისტრო </li>
</ul>
<div>ტექსტი 2 </div>

<script>
var elem = document.getElementById("partner");
alert(elem.firstChild.innerHTML);// განათლების სამინისტრო
alert(elem.lastChild.innerHTML);// ენერგეტიკისსამინისტრო
alert(elem.nextSibling.innerHTML);// ტექსტი 2
alert(elem.previousSibling.innerHTML);// ტექსტი 1
</script>

</body>
</html>
```

გარდა ელემენტების შექმნისა არსებობს ელემენტების წაშლის **removeChild()** და ჩანაცვლების **replace.Child()** მეთოდები.

```
<html>
<head>
<title>დოკუმენტი</title>
</head>
```

```
<body>
<div>ტექსტი 1 </div>
<ul id="partner">
<li> განათლების სამინისტრო </li>
<li> ჯანდაცვის სამინისტრო </li>
<li> სპორტისა და ახალგაზდობის სამინისტრო </li>
<li> ენერგეტიკის სამინისტრო </li>
</ul>
<div id="txt">ტექსტი 2 </div>
<script>
var elem = document.getElementById("partner");
document.body.removeChild(elem);// წაიშლება მთლიანი <ul>თავისი li_ებით

var elem2 = document.getElementById("txt");
var txt = document.createTextNode("შეცვლილი ტექსტი ");
document.body.replaceChild(txt, elem2);// შედეგი <div id="txt">შეცვლილი ტექსტი </div>
</script>
</body>
</html>
```

5.5 ფუნქციებთან მუშაობა

მიმდინარე პარაგრაფის თემატიკა

- ფუნქციების შექმნა და გამოყენება
- „ლამბდა“ გამოსახულებები

ფუნქციები

ფუნქცია თავისი არსით ავტონომიურად მოქმედ კოდს წარმოადგენს, რომლის ფუნქციონალის გაწერა ერთჯერადად ხდება, ხოლო გამოყენება შესაძლებელია მრავალჯერ.მაგ. საჭიროა ეკრანზე რაიმე რიცხვის კვადრატის გამოტანა. როგორც თქვენთვისაა ცნობილი რიცხვის კვადრატში ასაყვანად საჭიროა არსებული რიცხვის თავის თავზე გამრავლება. ე.ი პროცესი უცვლელია და მისი ყოველ ეტაპზე თავიდან დაწერა კოდის დუბლირებას წარმოადგენს, რაც შესაბამისად არც ისე კარგ ტონად მიიჩნევა.

Javascript_ში ფუნქციები გვევლინებიან ობიექტებად და სახელი რომლის მინიჭებაც ხდება, წარმოადგენს მის იდენთიფიკატორს, შესაბამისად ის რეგისტრზე დამოკიდებულია. მაშასადამე დაუშვებელია ფუნქციის სახელებად გამოყენებულ იქნას Javascript_ის რეზერვირებული სიტყვები (იხ.ცხრილი I), ასევე სხვა გლობალური ობიექტების სახელები. მაგ.: უკვე არსებული ცვლადების დასახელებები და ობიექტის იდენთიფიკატორები. თუ კოდში ერთიდაიგივე სახელით აღწერილია ორი ფუნქცია მაშინ არსებული ფუნქციის გამოძახებისას ხდება წვდომა არსებული სახელით შექმნილ ბოლო ფუნქციასთან. ფუნქცია შედგება ორი ნაწილისაგან ეს არის ფუნქციის სახელი და ტანი, სადაც ხდება სკრიპტის მობილიზება. სტანდარტული (სახელდებული) ფუნქციის დეკლარირება ხორციელდება შემდეგნაირად

```
functionshowMessage(){  //showMessage - სახელი
... ფუნქციის ტანი
}
```

ხოლო ანონიმური ტიპის ფუნქცია

```
var showMessage = function(){  //showMessage - სახელი
... ფუნქციის ტანი
}
```

ფუნქციის დეკლარირება მის ავტომატურ გამოყენებას არ გულისხმობს და არც ითვალისწინებას. როგორც ითქვა ერთგვაროვანი მოქმედების მრავალჯერ გამოყენებაა საჭირო ამისთვის სასურველ ადგილზე საჭიროა ფუნქციის გამოძახება რომელიც მდგომარეობს შემდეგში

```
functionshowMessage(){
alert("შეტყობინება");
}
//ფუნქციის გამოძახება შესაძლებელია მრავალჯერადად. არსებული მაგალითზე დაყრდნობით ეკრანზე გამოვა alert_დიალოგული ფანჯარა ორჯერ ტექსტით 'შეტყობინება'

showMessage()       //ფუნქციის გამოძახება
showMessage()       //ფუნქციის გამოძახება
```

სტანდარტული და ანონიმუტი ტიპის ფუნქციები ერთმანეთისაგან ფუნქციის წვდომით განსხვავდებიან. თუ სტანდარტულ ფუნქციასთან წვდომა ნებისმიერი ადილიდან არის შესაძლებელი, ანონიმურთან მხოლოდ მისი დეკლარირების შემდგომ.

```
//სტანდარტული ფუნქცია
showMessage()       //შედეგი ეკრანზე 'შეტყობინება'

functionshowMessage(){
alert("შეტყობინება");
}

showMessage()       //შედეგი ეკრანზე 'შეტყობინება'

//ანონიმური ფუნქცია
showMessage()       //ფიქსირდება შეცდომა

var showMessage = function(){
alert("შეტყობინება");
}
```

```
showMessage()       //ამ ეტაპზე მუსაობს --- შედეგი ეკრანზე 'შეტყობინება'
```

ფუნქციას, როგორც მნიშვნელობის დაბეჭდვა (რომელიმე გამოტანის ოპერატორის alert(), console.log(), document.write()) , ასევე მისი დაბრუნება შეუძლია. ფუნქციიდან ინფორმაციის დაბრუნება საკმაოდ მნიშვნელოვანი დეტალია, გამომდინარე იქიდან, რომ შესაძლოა მიღებული მნიშვნელობა გამოყენებულ იქნას სხვა დანიშნულებით. მაგ.: დავალება მდგომარეობს შემდეგში ერთი ფუნქციის მეშვეობით უნდა მოხდეს რიცხვის კვადრატის დადგენა, მეორე ფუნქცია ადგენს ამ რიცხვის უდიდეს გამყოფს და შედეგად უნდა მივიღოთ მათი სხვაობა. ამისათვის საჭიროა ფუნქციები რომლების უზრუნველყოფენ: პირველი - კვადრატის დადგენას, მეორე უდუდესი გამყოფის პოვნას. როგორც ატყობთ დავალება არ მდგომარეობს მათ დაბეჭდვაში. შესაბამისად გამოტანის ოპერატორების ფუნქციაში გამოუყენების არავითარი საჭიროება არ არსებობს. პროგრამისტისთვის მნიშვნელოვანია მიიღოს ინფორმაცია, რომელზეც შემდგომ იმოქმედებს. ასეთ შემთვევაში ფუნქციაში გამოიყენება დაბრუნების ოპერატორი **return**

```
functionshowMessage(){
        return"შეტყობინება";
}
```

არსებულ შემთხევაში ფუნქციის გამოძახება არ მოგვცემს ვიზუალურ შედეგს ეკრანზე, მაგრამ იძლევა საშუალებას არსებულ მნიშვნელობაზე მოვახდინოთ მანიპულირება.

```
functionshowMessage(){
        return"შეტყობინება";
}

alert ( "საგანგაშო" + showMessage())    //შედეგი ეკრანზე - საგანგაშო შეტყობინება
```

ფუნქციიდან **return** მეთოდით დაბრუნებული შედეგის მინიჭება შესაძლებელია ასევე რაიმე ცვლადზე. მაგ.:

```
functionshowMessage(){
        return"შეტყობინება";
}
var sms = showMessage( )
alert ( "სასარგებლო" + sms)   //შედეგი ეკრანზე - სასარგებლო შეტყობინება
```

გაითვალისწინეთ ფუნქციაში შეუძლებელია რამოდენიმე მნიშვნელობის რამოდენიმე return მეთოდით მიღება, რადგანაც პირველივე return _ის გამოზახება იწვევს ფუნქციის მოქმედების შეწყვეტას და მის შემდგომ არსებული კოდის შესაბამისად აღარ აღიქმება. უხეშად რომ შევადაროთ, რა გავლენასაც

ახდენდა break ოპერატორი switch კონსტრუქციასა და ციკლებში, ფაქტობრივად იგივე ფუნქციონალის შეთავსება უხდება return_ს გარდა ძირითადი ფუნქციისა, რომელიც აბრუნებს მნიშვნელობას

```
functionshowMessage(){
        return"სასიამოვნო";
        return"სიახლე";
}
var sms = showMessage( )
alert (sms)          //შედეგი ეკრანზე მხოლოდ - სასიამოვნო      და არავითარ შემთხვევაში სასიამოვნო
სიახლე
```

ფუნქციების მნიშნველოვან დანამატს წარმოადგენს **არგუმენტები.** არგუმენტები ეს ის მნიშვნელობებია რომლის გადაცემაც და დამუშავებაც შესაძლებელია ფუნქციის მიერ. შესაძლებელია ფუქნციისათვის, როგორც ერთი ასევე რამოდენიმე არგუმენტის ერთდროულად გადაცემა. მაგ.: დავალების შესაბამისად დავწეროთ რიცხვის კვადრატში აყვანის ფუნქცია.

```
functionsquare(x ){   //xწარმოადგენს არგუმენტეს
        return x * x ;
}

alert ( square( 5) )   //შედეგი ეკრანზე 25
alert ( square( 10) )  //შედეგი ეკრანზე 100
alert ( square( 7) )   //შედეგი ეკრანზე 49
```

შესაძლოა არგუმენტად გადაეცემოდეს ესა თუ ის კონკრეტული ცვლადი

```
functionsquare( x  ){ //xწარმოადგენს არგუმენტეს
        return x * x ;
}

var n = 11;

alert ( square( n) )   //შედეგი ეკრანზე 121
```

დაიმახსოვრეთ არგუმენტის სახელი (მოცემულ შემთხვევაში - **x**) არ წარმოადგენს მნიშვნელობას რომელიც ხილულია ფუნქციის გარეთ. ის მხოლოდ ფუნქციის შგნით აღიქმება. ფუნქციის გარათ x_ის დაბეჭდა არ მოხერხდება , შედეგი ეკრანზე არ გამოისახება.

```
functionsquare( x  ){ //xწარმოადგენს არგუმენტეს
        return x * x ;
```

```
}

alert (x)    //Uncaught ReferenceError: x is not defined - შეგიძლიათ გადაამოწმოთ consol_ში
```

არგუმენტების განსაზღვისას გასათვალისწინებელია, რომ ფუნქციამ უნდა მიიღოს იგივე რაოდენობის არგუმენტების გამოძახებისას.მაგ:

```
functiontest( x,  y , z ){
            return x + y + z ;
}
alert (5, 7, 3)          //15
```

არგუმენტების არასრულად გადაცემის შემთხვევაში არსებული არგუმენტები მიენჭება საწყის ცვლადებს სანამ ისინი იარსებებს, დანარჩენი არგუმენტის მნიშვნელობები გახდება undefined. მაგ.: მხოლოდ 2 პარამეტრის მიწოდების შემთხვევაში ისინი იქნებიან **x,y** არგუმენტები და არა **x, z** ან **z,y** შესაბამისად z არგუმენტი იქნება undefined.

```
functiontest( x,  y , z ){
            return x + y + z ; // 5+3+undefined
}
alert (5, 3)             //NaN

alert (5, 3, 8)          //16
```

დამატებითი არგუმენტების გადაცემის შემთხვევაცი, როცა ისინი არ იყო წინასწარ განსაზღვრული ფუნქცია არ იღებს მათ მხედველობაში, ფაქტობრივად უფულველყოფს მათ და არ წარმოაჩენს. ისინი არ აღიემქებიან არც შეცდომებად. მათი არსებობა საერთოდ არსად ჩანს

```
functiontest( x,  y , z ){
            return x + y + z ; // 5+3+2
}
alert (test(5, 3, 2, 8, 9, 4))       //  შედეგი 10 ---- 8,9,4  უგულველყოფილია, არც შეცდომა ,არც ყურადღების გამახვილება
```

არსებობს გზა თავის დასაცავად, თუ რომელიმე კონკრეტული არგუმენტი არ იქნა გადაცემული მოხდეს ამ არგუმენტის ჩანაცვლება რაიმე მნიშვნელობით რათა ფუნქციამ არ გამოიტანოს შეცდომა. მაგ.:

```
functiontest( x,  y , z ){
```

```
        x = x || 1; // თუ x არგუმენტი არ გამოეცია მნიშვნელობა x გახდება 1_ის ტოლი
        y = y || 10; // თუ y არგუმენტი არ გამოეცია მნიშვნელობა y გახდება 10_ის ტოლი
        z = z || 20; // თუ z არგუმენტი არ გამოეცია მნიშვნელობა z გახდება 20_ის ტოლი

        return x + y + z ;
}

alert( test(5, 3, 2))   //10
alert( test(5, 3))      //  28  - - - - x = 5, y = 3, z = 20რადანაც z არ გადაეცა მნიშვნელობად აიღო 20
alert( test(5))         //  35  - - - - x = 5, y = 10, z = 20
alert( test())          //  31  - - - - x = 5, y = 10, z = 20
```

Javascript_ს გააჩნია **arguments** ობიექტი, რომლის საშუალებით შესაძლებელია ყველა არგუმენტის მიღება წინასწარ განუსაზღვრელი რაოდენობის მიუხედავად. **arguments** ობიექტი იგივე მასივია, შესაბამისად მასივის მეთოდებისა და ფუნქციები მიესადაგება

```
functiontest(){

        return arguments.length;
}

alert( test(5, 3, 2))   //3
```

თითოეულ ელემენტთან წვდომა თქვენთვის ასევე ცნობილია

```
functiontest(){
        return arguments[0]+arguments[2];
}
alert( test(5, 3, 2) )  //7
```

მაგ. საჭიროა დაიწეროს ფუნქცია რომელიც მოახდენს რიცხვთა ჯამის წარმოჩენას, მაგრამ საწყის ეტაპზე არ არის გარკვეული თუ რამდენი რიცხვის დაჯამება იქნება საჭირო. ეს შეიძლება იყოს 4,10, და ა.შ. ამისათვის გამოიყენება ობიექტი arguments რომლის მეშვეობითაც შესაძლებელია არგუმენტების რაოდენობის დადგენა და მათზე მანიპულირება შეცდომების გარეშე (როგორც იყო წინა შემთხვევაში განუსაზღვრელი არგუმენტის მნშვნელობა undefined).

```
functiontest(){
        var sum = 0;
          for(var i =0; i <arr.length; i++){
                    sum+= arguments[i];
          }
```

```
return sum;
}

alert( test(5, 3, 2))   // 10
alert( test(2,11,25, 4, 3))        // 45
alert( test(10, 23))   // 33
```

ფუნქციის თავისებურებას წარმოადგენს „ურთიერთობა ცვლადებთან“. რაც შემდეგში მოიაზრება. ნებისმიერი ცვლადი რომელიც აღწერილია სკრიპტში ფუნქციისათვის წარმოადგენს გლობალური ტიპის ცვლადს. ფუნქციიდან შესაძლებელია მის გარეთ მდებარე ცვლადთან წვდომა და მისი ცვლილება.მაგ:

```
var k = 25;

functiontest(){
        k  += 11     // k = k + 11
}

alert( k ); //  36
```

ფუნქციას გააჩნია ასევე ლოკალური ტიპის ცვლადები. რომელიც არ სცდება არსებული ფუნქციიის ხილვადობის არეალს - მის ტანს.არსებული ცვლადის დეკლარირება ხდება უშუალოდ ფუნქციის ტანში და ის ხილვადია არსებული ფუქნციის შიგთავსში ასევე სხვა ფუნქციებში რომლებიც გამოძახებულია მიმდინარე ფუნქციის ბლოკში {...}, მაგრამ არამც და არამც ძირითადი ფუნქციის გარეთ.

```
functiontest(){
var k = 25;
        k  += 11     // k = k + 11
}

alert( k ); //  k is not defined
```

შემთხვევა როცა ერთი და იგივე სახელით დეკლარირებულია ცვლადი ფუნქციის გარეთ და ფუნქციის შიგნით განიხილება შემდეგნაირად. გლობალური ცვლადი k რჩება გლობალურად და ის ფუნქციიდან არ განიცდის ცვლილებას გამომდინარე იქიდან, რომ არსებულ ფუნქციას თავისი ლოკალური k ვლადი გააჩნია, სწორედ მასი ცვლილება ხდება ფუნქციის შიდა გამოთვლის დროს.

```
var k = 25;

functiontest(){
var k = 11;
        k  += 11     // k = k + 11
```

```
alert( k );
}
test()                  //  22

alert( k ); //  25
```

არსებობს ფუნქციის შექმნის კიდევ ერთი გზა, რომელიც ძალიან იშვიათად გამოიყენება მაგრამ ფუნქციის დეკლალირების სრულყოფილი წარმოდგენისთვის მასაც შევეხოთ.

```
var showMessage =  new Function()
```

არსებული სახით ფუნქცია იღებს და ამუშავებს მონაცემებს string_ის სახით, ძირითადად გამოიყენება მოკლე კოდისთვის

```
var sum =newFunction('a,b',' return a+b; ');

var result =sum(1,2);
alert( result );        // 3
```

Javascript Events

ვებ გვერდების დინამიურობა ძირითადად რაიმე ტიპი ხდომილებებზეა/ქმედებაზეა დამოკიდებული. ხშირია, როდესაც დოკუმენტი ან მისი რომელიმე ელემენტი გარკვეულ ქმედებაზე განიცდის რაღაცა სახის ცვლილებას. მაგალითისათვის ვებ ბრაუზერი ახორციელებს ქმედებას, როდესაც ტვირთავს დოკუმენტს, მომხამრებელი მიერ მაუსის მიტანისას მოქმედებს ჰიპერბმულზე, ასევე ფორმის ღილაკზე ახორციელებს კლიკს. Javascript_ით შესაძლებელია წინასწარ განისაზღვროს კონკრეტული ქმედება, რაიმე ფუნქციის ან სკრიპტის ფრაგმენტის მეშვეობით, რომელიც ამუშავდება მხოლოდ მომხმარებლის მიერ გამოხატული აქტივობის შემდეგ.

ზოგადად სხვადასხვა ტიპის მოქმედება უზრუნველყოფს სხვადასხვა ტიპის ქმედების/ხდომილების წარმოქმნას. მაგ.: ბმულზე ხელის მიტანა და მასზე უშუალო ზემოქმედება (კლიკი) ერთმანეთისაგან განსხვავებული პროცესებია, ისევე როგორც თუნდაც ფორმის ორი ერთნაირი ღილაკი Submit და Reset აზრობრივად ერთმანეთისაგან რადიკალურად განსხვადება.

ბრაუზერების დახვეწამ დროთა განმავლობაში განავრცო Javascript ხდომილებები და დღეს დღეისობით დეველოპერებს საკმაოდ დიდი არჩევანი აქვთ . ცხრილ XIV მოცემულია იმ ძირითადი ხდომილებების ჩამონათვალი, რომელიც აქტიურად გამოყენებადია. ესენია მაუსის, კლავიატურის, ფორმის და ობიექტის ქმედებები. სრული სია შეგიძლიათ იხილოთ მითითებულ ბმულზე -

http://www.w3schools.com/jsref/dom_obj_event.asp

ცხრილი XIV

ხდომილება	აღწერა		ტეგები, რომელსაც მიესადაგება
onclick	მაუსი	ერჯერადი კლიკი (დაჭერა - აშვების პროცესი)	ფაქტობრივად ყველა html ელემენტი
ondblclick		ორმაგი კლიკი	ფაქტობრივად ყველა html ელემენტი
onmousedown		მაუსის ღილაკზე მხოლოდ დაწოლა (აშვება არა)	ფაქტობრივად ყველა html ელემენტი
onmouseup		მაუსის ღილაკზე ხელის აშვება	ფაქტობრივად ყველა html ელემენტი
onmouseout		მაუსის კურსორი ცდება ელემენტის საზღვრებს	ფაქტობრივად ყველა html ელემენტი
onmouseover		მაუსის კურსორი შედის ელემენტის საზღვრებში	ფაქტობრივად ყველა html ელემენტი
onmousemove		მაუსის კურსორი გადაადგილდება ტეგის არეალში	ფაქტობრივად ყველა html ელემენტი
onkeydown	კლავიატურა	კლავიატურის კლავიშზე ხელის დაწოლა	ფაქტობრივად ყველა html ელემენტი
onkeyup		კლავიატურის კლავიშზე ხელის აღება	ფაქტობრივად ყველა html ელემენტი
onkeypress		დაწოლიქნა და აშვებულია კლავიშზე ხელი	ფაქტობრივად ყველა html ელემენტი
onload	ფრეიმები / ობიექტები	დოკუმენტის ჩატვირთვა დასრულებულია	BODY , IMG
onerror		შეფერხება წარმოქმნა სცენარის შესრულებისას	IMG
onabort		სურათის ჩატვირთვის შეფერხება	IMG
onresize		დოკუმენტის ზომის ცვლილება	WINDOW
onscroll		ელემენტის დასკროლვისას	უმრავლესობა ტეგებზე

onunload		ელემენტი ჩატვირთvის პროცესშია	ძირითადად BODY
onfocus	**ფორმები**	ფორმის მონიშვნა / კურსორის ჩაყენება	BUTTON, INPUT, LABEL, SELECT, TEXTAREA
onblur		ფორმის ელემენტიდან ამოსვლა	
onchange		ფორმის ველი არჩეულია	SELECT
onreset		ფორმის გაბულება / ინფორმაციის ჩამოყრა	FORM
onsubmit		ფორმის მონაცემების გაგზავნა	FORM
onselect		ტექსტის არჩევა	INPUT, SELECT

ხდომილების გამოყენება შესაძლებელია უშუალოდ ელემენტში მისი გაწერით ან სკრიპტიდან ელემენტზე მიმართვიანობით.

დავალება

ღილაკზე მაუსის დაკლიკვით ბლოკ id="txt" შეეცვალოს ფონი და გახდეს ის წითელი

```
<divid = "txt">ტექსტი 1 </div>
<buttononclick ="setBg()">ღილაკი </button>
<script>
function setBg(){
        document.getElementById("txt").style.backgroundColor="red";
}
</script>
```

შესაძლებელი იყო არსებული ფუნქცია ცალკე გატანის გარეშე უშუალოდ მეთდ onclick_ში გაგვეწერა. ეს გზა შედარებით არაკომფურტულია დიდ სკრიპტთან მიმართებაში და ნაკლებად გამოიყენება, მაგრამ მისი უგულველყოფა არიქნება სამართლიანი, რადგანაც მცირე კოდებში საკმაო გამოყენება აქვს. კოდის მინიმიზაცია თვალსაჩინოა.

```
<divid = "txt">ტექსტი 1 </div>
<buttononclick ="document.getElementById('txt').style.backgroundColor='red'">ღილაკი </button>
```

არსებობს კიდევ ერთი გზა, როცა event_ის მიბმა უშუალოდ სკრიპტიდან ხორციელდება

```
<div id = "txt">ტექსტი 1 </div>
```

```
<button>ღილაკი </button>
<script>
document.getElementsByTagName("button")[0].onclick = function(){
        document.getElementById("txt").style.backgroundColor="red";
}
</script>
```

ქმედებებთან / ხდომილებებთან ურთიერთბისას მნიშვნელოვანია thisმნიშვნელობის არსის ცოდნა. ის გამოიყენება მიმდინარე ელემენტის გამოსახატად სა კომფორტულია ბევრ შემთხვევაში. რამოდენიმე ელემენტის შემთხვევაში სათითაოდ მოგვიწევს მათი წვდომის (document.getElement…..) განსაზღვრა, თავის მარტივად დასაძვრენად გამოვიყენოთ this

დავალება:

ბლოკებზე ხელის დაწოლისასგამოსახოს ეკრანზე არსებულ ბლოკში არსებული ინფორმაცია

```
<divonclick ="getTxt(this)">ტექსტი 1 </div>
<divonclick ="getTxt(this)">ტექსტი 2</div>
<divonclick ="getTxt(this)">ტექსტი 3</div>

<script>
function getTxt(t){
        alert(t.innerHTML);
}
</script>
```

მოცემულ კოდში this არგუმენტის არცოდნის შემთხვევაში მოგწევდათ კოდის შემდეგნაირად გაშლა

```
<divonclick ="getTxt1()" id ="dv1">ტექსტი 1 </div>
<divonclick ="getTxt2()" id ="dv2">ტექსტი 2</div>
<divonclick ="getTxt3()" id ="dv3">ტექსტი 3</div>

<script>
function getTxt1(){
        alert(document.getElementById("dv1").innerHTML);
}
function getTxt(2){
                alert(document.getElementById("dv2").innerHTML);

}
function getTxt(3){
                alert(document.getElementById("dv3").innerHTML);
}
```

```
</script>
```

ან შესაძლოა სხვა ინტერპრეტაცია ჰქონოდა სკრიპტის ვრცელ ვარიანტს

თავი 6.საიტის მართვის სიტემები

6.1 საიტის მართვის სისტემების (CMS) ინსტალირება/პარამეტრების დაყენება

მიმდინარე პარაგრაფის თემატიკა

- საიტის მართვის სიტემის (CMS) დანიშნულება და გამოყენების ასპექტები
- CMS-ების გამოყენებით გადაჭრადი ამოცანები
- საიტის მართვის სისტემის მოძიება–ინსტალირება
- საიტის მართვის პარამეტრები და მათი გამოყენება

საიტის მართვის სიტემის (CMS) დანიშნულება და გამოყენების ასპექტები

დღეს ჩვენ შეგვიძლია ვთქვათ რომ XXI საუკუნე არის ახალი ერა ადამიანთა ცხოვრებაში. 90_იანი წლებიდან საფუძველი ეყრება ვებ გვერდების კონცეფციის ჩამოყალიბებას.

ამ დროიდან მოყოლებული ვებმა დიდი ევოლუცია განიცადა. შეიცვალა როგორც მისი ვიზუალური ასევე სტრუქტურული მხარე. მუდმივი განახლების და დახვეწის პროცესშია ის ენები და ტექნოლოგიები რომლის შეშვეობითა ხდება საიტების შექმნა. ისტორიას ჩაბარდა სტატიკური ტიპის საიტები და ფართო მოხმარების არეალი გადაეშალა განახლებადი ტიპის ვებ გვერდებს.

საიტის შემცველობის მართვის სისტემის **CMS** (Content management system) შექმნა ხანგრძლივი და შრომატევადი პროცესია. მისი პუსები გამოიხატება უნივერსალურობაში - რისი მიღწევაც ხშირად მხოლოდ გუნდურობის შემთხვევაშია შესაძებელი. დღეს ბაზარზე მოთხოვნადი საიტების ფაქტიურად 100% მორგებული **CMS**_ზე.

ბაზრის მოთხოვნის ადეკვატურად სხვადასხვა ორგანიზაციებმა მომხმარებლებს შესთავაზეს მზა ტიპის სამართავი სისტემები**WordPress** , **Drupal**, **Joomla** , **OpenCart, 1C-Bitrix**და ა.შ.(იხ.სურ.1.1) ეს რა თქმა უნდა უმალ იქნა ათვისებული ვებ დეველოპერებისა და ფართო მასების მიერ რადგანაც უკვე ავღნიშნეთ სამართავი სისტემის შექმინის სირთულეზე.

CMS	Official Sites
Joomla	www.joomla.org
Drupal	drupal.org
Modx	modx.com
WordPress	wordpress.org
Typo3	typo3.org
ImageCMS	www.imagecms.net
CMS Made Simple	www.cmsmadesimple.org

სურ. 6.1 CMS ოფიციალურისაიტები

ძირითადად ორგანიზაციები ორიენტირებულნი არიან კონკრეტული მიმართულების მოთხოვნის დაკმაყოფილებაზე. მაგ.: Wordpress მიმართულია ბლოგური ტიპის საიტების შესაქმნელად, OpenCart ინტერნეტ მაღაზიების ასაწყობად და ა.შ. მაგრამ ეს სულაც არ ნიშნავს რომ ამ CMS_ების შესაძლებლობები მხოლოდ ჩამოთვლილით ამოიწურება.

რა თქმა უნდა თითოეულ სამართავ სისტემას თავისი პლუსები და მინუსები გააჩნია, ამიტომ CMS მისადაგება უნდა მოხდეს კონკრეტული მიმართულებისათვის ინდივიდუალურად. სწორი გათვლა უზრუნველყოფს მაქსიმალურად გამოყენებას იმ დადებითი რესურსისა, რაც კონკრეტულ CMS_ს გააჩნია. ხშირ შემთხვევაში დეველოპერთა უმეტესობა გარკვეულ სამართავ სისტემაზე ისეა მორგებული რომ ნებისმიერი მიმართულებით შეუძლია მისი გამოყენება.

ქვემოთ გარჩეული იქნება ერთ-ერთი ფაროთოდ გამოყენებადი სამართავი სისტემა WordPress. მისი მუშაობის პრინციპები და თავისებურებანი.

WordPress - ერთ ერთი უპოპულარულესი საიტის ინფორმაციის სამართავი სისტემაა დღეს დღეისობით მსოფლიოშო.მის მთავარ უპირატესობად მიიჩნევა გახნისლი, ღია კოდი (open source). ღია კოდის პრინციპი შემდეგია - მომხმარებელს ეძლევა საშუალება ჩაერიოს და თავად მოახდინოს კოდის ცვლილება. დახურული კოდის შემთხვევაში პირს უწევს ლოდინი სანამ პროდუქტის მწარმოებელი კომპანიის დეველოპერები არ გამოასწორებენ შეცდომას. WordPress_ის დადებით მხარეში უნდა მოვიხსენიოთ ის ფუნქციონალური დანამატები (plugins) და ესთეტიკური დიზაინის ელემენტები (Themes) რომლის ფართო და მრავალფეროვანი არჩევანიც გააჩნია ამ CMS_ს.

WordPress_ი პროგრამირების ენად PHP,ხოლო მონაცემთა ბაზად MySQL იყენებს. ეს კიდევ უფრო ზრდის მის პოპულარობას, რადგანაც ამ ენების მოხმარების არეალი საკმაოდ ვრცელია.

WordPress_ის პოპულარობა განპირობებულია ასევე მისი სიმარტივით. მაგ.: მის ინსტალაცია სულ რამოდენიმე წუთს მოითხოვს. WordPress შექმნია ისე, რომ გაფართოვდეს და მოერგოს გასხვავებული მოხმარებლის მოთხოვნილებებს.

CMS-ების გამოყენებით გადაჭრადი ამოცანები

ზედა ჩანართის დაწვრილებით გაცნობის შემთხვევაში შესაძლებელია შემდეგი დასკვნის გამოტანა - მსოფლიო ქსელში საიტები იყოფა ორ ძირითად ჯგუფად **სტატიკურ** და **დინამიურ** საიტებად. რა თქმა უნდა ურივე მიდგომას თავისი უპირატესობები გააჩნია. დღევანდელობიდან გამომდინარე სტატიკური ტიპის საიტები ფაქტობრივად ტოვებენ მსოფლიო ბაზარს და მათ ადგილს ინტენსიურად იკავებენ დინამიური საიტები.

სტატიკური ტიპის საიტები - ეს არის html ენის მეშვეობით ტექსტური და გრაფიკული, მულტიმედია ინფორმაციის წარმოჩენა ბარაუზერის ფანჯარაში. გასათვალისწინებელია ის ფაქტი რომ საიტის ინფორმაცია გამოისახება ყველა ჯერზე ერთნაირად და მასში ცვლილებების შესატანად საჭიროა გამოყენებულ ტექნოლოგიებში კომპეტენტური ადამიანი. სწორედ ის უზრუნველყოფს ინფორმაციის დამატებას ან შეცვლას საიტის კოდში რეალური ჩარევთ, რაც გამოიხატება ფაილების გახსნით და კოდის ცვლილებით.

სტატიკური ტიპის საიტების **უპირატესობაში** შეიძლება ჩაითვალოს:

- საიტის კომპატურობა;
- საიტის ინფორმაციის ბრაუზერში ჩატვირთვის სისწრაფე;
- ვებ სევერის მინიმალური დატვირთვა;
- ახალ სერვერზე საიტის ინფორმაციის გადატანის სიმარივე;

სტატიკური ტიპის საიტების **უარყოფით** ასპექტებად შეიძლება მივიჩნიოთ:

- საიტის ინფორმაციის ყოველი ცვლილება დაკავშირებულია ფინანსურ რესურსთან (კომპეტენტური ადამიანის ხელახალი ჩარევა);
- ახალი ჩანართის ჩამატება უნდა მოხდეს ყველა გვერდზე ინდივიდუალურად;
- საიტის ინფორმაციის სამართავად ყოველ ჯერზე წვდომა უნდა მოხდეს FTP_სთან;
- მომხარებლის არასრული მხარდაჭერა (ბრაუზერების მხარდაჭერა , cookie)
- შეზღუდული ფუნქციონალი;

დინამიური ტიპის საიტები- ეს არის პროგრამული კოდის მეშვეობით შემუშავებული საიტი მრავალფუნქციური შესაძლებლობებით. ვებში გამოყენებადი პროგრამირების ენებიდან საკმაო პოპულარულია PHP. Perl, Asp და ა.შ. ხშირ შემთხვევაში დინამიური საიტების შესაქმნელად

პროგრამული კოდის ნულიდან წერის გზას არ იყენებენ - არამედ მოიხმარენ უკვე მზა CMS (Content Management System). CMS - წარმოადგენსმზა მოდულებისა და კომპონენტების ერთობლიობას, რაც ამარტივებს მუშაობის პროცესს და მათი ყოველ ჯერზე ნულიდან წერა აღარ არის საჭირო. ამასთანავე CMS იძლევა საშუალებას მის პროგრამულ საფუძველზე დამოუკიდებლად შემუშავებული იქნას ნებისმიერი სირთულის პროგრამული ამოცანა. რა თქმა უნდა სრულყოფილი CMS არ არსებობს და ზოგად მათაც გააჩნიათ თავისი უარყოფითი მხარეები, მაგრამ იმ მრავალფუნქციონურობის ფონზე რაც CMS-ს გააჩნია ძნელია მათზე ყურადღების გამახვილება .

მაგ. CMS_ების **უარყოფით** მხარეებში შეიძლება გამოვყოთ შემდეგი თვისებები:

- სტატიკური საიტებისაგან გასხვავებით მოიხმარს მეტ რესურსს;
- შედარებით ნელია;
- მისი დამზადება მეტ ფინანსურ რესურსს მოითხოვს;

როგორც წინამდებარე აბზაცში განვიხილეთ CMS_საც გააცნიათ უარყოფითი მხარეებ, მაგრამ ეს ზღვაში წვეთია იმ დადებით რესურსის ფონზე რისი შემოთავაზებაც შეუძლიათ.

შეძლებისდაგვარად მოკლედ გავეცნოთ CMS _ის გამოყენებით გადაჭრილი ამოცანების ჩამონათვალს:

- საიტის ინფორმაციის მართვა (დამატება, განახლება, წაშლა) საკმაოდ მარტივია და არ მოითხოვს პროგრამირების ენების ცოდნას.
- ფინანსური რესურსის სიმცირე - პირველი პუნქტიდან გამომდინარე საიტის ინფორმაციის მართვა შეიძლება საერთოდ არ იყოს ფინანსურ საკითხთან დაკავშირებული (თუ თქვენ თვითონ უზრუნველყოფთ ყოველივეს. და ეს ნამდვილად შსაძლებელია);
- ინფორმაციის გაფორმება და ვიზუალიზება შესაძლებელია ღილაკებზე დაყვანილი მოქმედებებით. მაგ. გამუქება, დახრა, სიების შექმნა, სწორებებისა და ბმულების გენერირება (იხ.სურ.1.2). დაწვრულებით ამ საკითხს გავარჩევთ ქვედა თავებში;
- ახალი გვერდების, მენიუს პუნქტების, სტატიების დამატება სულ რამოდენიმე წამში - 1 ღილაკის მოქმედებით;
- ანიმაციური და მულტიმედია ობიექტების ზომების და განლაგებების ფორმირება;
- ატვირთული ფაილების დაჭრა და დიზაინზე მორგებულფორმატზე დაყვანა

სურ. 1.2ედიტორისმეშვეობითსაიტისინფორმაციისგაფორმება

რაც ყველაზე მთავარია CMS_ს გააჩნია უდიდესი ფუნცქიონალური შესაძლებლობები.

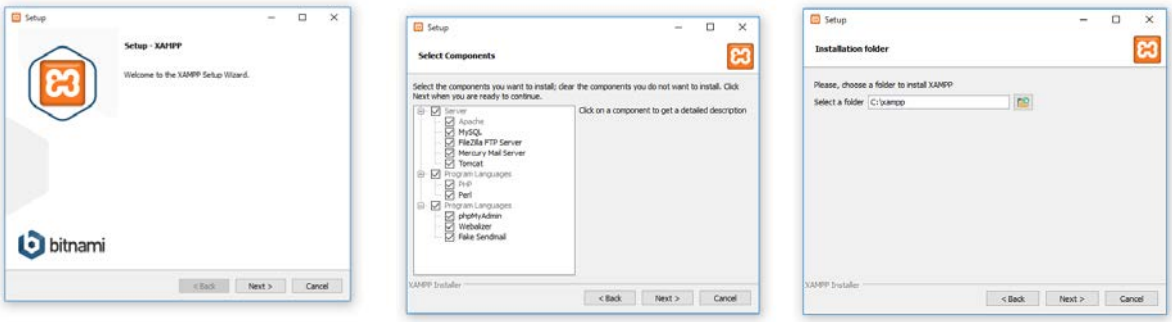
გამომდინარე იქიდან რომ WordPress დაწერილია პროგრამირების ენა PHP_ზე მის გასაშვებად მხოლოდ ბრაუზერის ფანჯარა არ არის საკმარი. საჭიროა ფაილების სერვერზე განთავსება.

სერვერი შიძლება იყოს ლოკალურიც ანუ რომელიმე ვირტუალური პროგრამა-სერვერი. არსებობს რამოდენიმე ფართოდ მოხმარებადი ვირტუალური სერვერი **Denwer, Xampp, Wamp, Vertrigo, UsbWebserver და ა.შ.**ჩვენ განვიხილოთ ერთ ერთი მათთაგანი Xampp სერვერი.

Xampp სევერისჩამოწერა შესაძლებელი ოფიციალური საიტიდან https://www.apachefriends.org/index.html დაის სრულიად უფასოა (იხ. სურ. 2.1).

სურ. 2.1 Xampp სევერისჩამოწერა

შეარჩიეთ თქვენთვის სასურველი ოპერაციული სისტემის ვერსია და ჩამოწერეთ. ფაილი ავტომატურად exe გაფართოებისაა ასე რომ მისი ინსტალაცია არანაირ პრობლემას არ წარმოადგენს. მივყვეთ ნაბიჯებს ფოტოსურათების მიხედვით (იხ. სურ. 2.2).

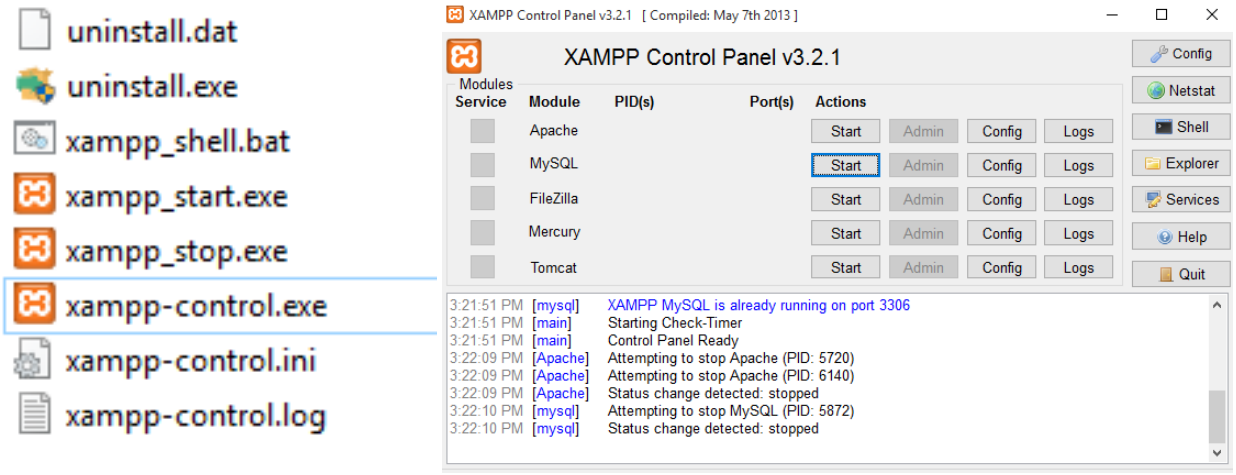


სურ. 2.2 Xampp სევერისჩამოწერისეტაპები

ინსტრუქციის უცვლელად გაყოლის შემთხვევაშიXampp ავტომატურად ყენდება **C დისკზე**(იხ. სურ. 2.2.3).

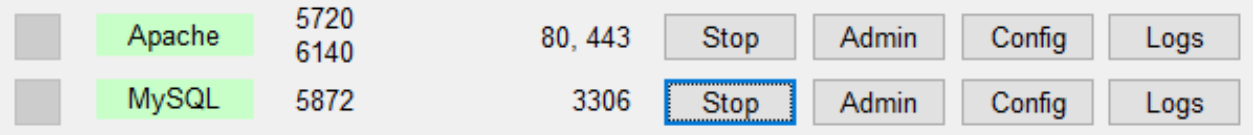
სურ. 2.2.3 Xampp სევერისჩამოწერისეტაპები

პროგრამის სამართავი პანელის (XAMPP Control Panel) გაშვება ხდება ძირითად საქაღალდეში (C:\\xampp) არსებული გამშვები ღილაკის**xampp-control.exe** საშუალებით (იხ. სურ. 2.3).



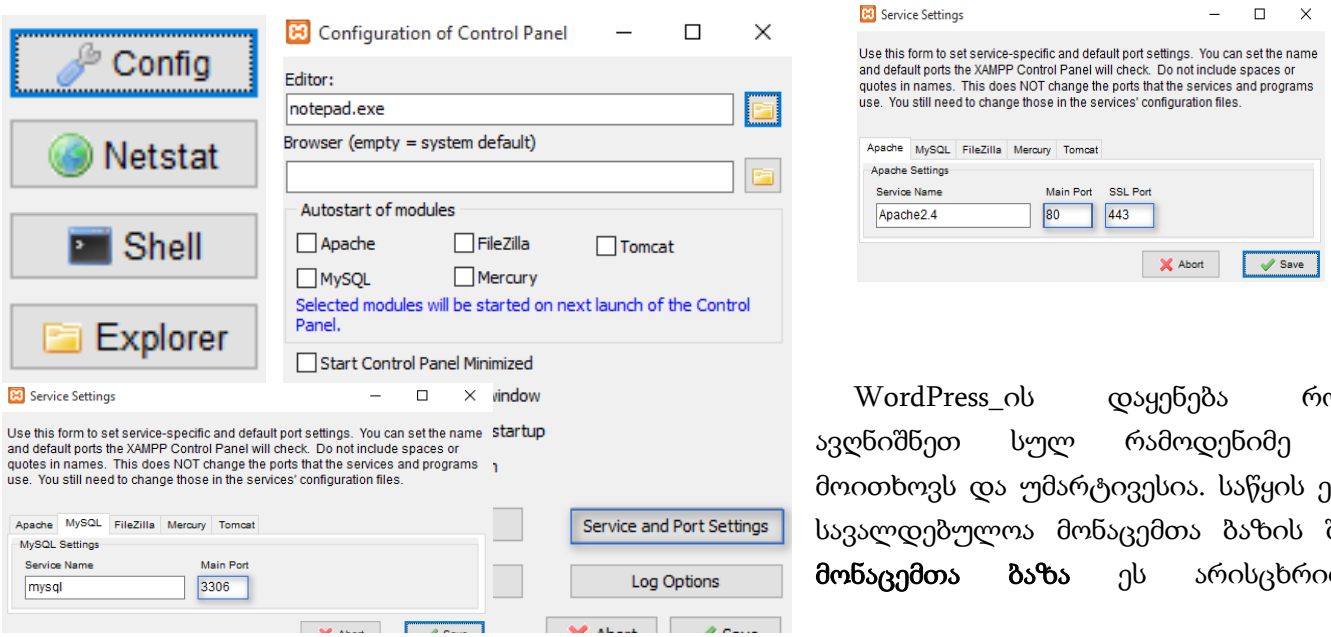
ურ.2.3 Xampp სევერისსამართავიპანელისგამოძახება

აუცილებელია რომ გაეშვას **Apache** (ვებსერვერი) და **MySQL**(მონაცემთა ბაზა) რაც გამოიხატება მათ გამწვანებაში (იხ. სურ. 2.4.).



სურ.2.4Apache სერვერისადა MySQL მონაცემთაბაზისგაშვება

Xampp_ის გაშვებისას ხშირად პრობლემას წარმოშობს სხვა პროგრამები მაგ.: ანტივირუსები,ან პროგრამები რომელთა პორტებიც ემთხვევა xampp_ის პორტებს (skype პორტები - 80 и 443) . ასეთ შემთხვევაში საჭიროა სხვა მოწყობილობების გათიშვაxampp_ის გაშვებამდე. პრობლემა შესაძლოა აღმოფხვრას კონფიგურაციის ფაილიდან პორტების ცვლილებამ **config > service and port settings (იხ.სურ. 2.5).**



სურ.2.5კონფიგურაციისღილაკისმეშვეობითპორტებისარეალისგამოძახება

WordPress_ის დაყენება როგორც ავღნიშნეთ სულ რამოდენიმე წუთს მოითხოვს და უმარტივესია. საწყის ეტაპზე სავალდებულოა მონაცემთა ბაზის შექმნა. **მონაცემთა ბაზა** ეს არისცხრილების ერთობლიობა, მათში გარკვეული ლოგიკით სორტირებული ინფორმაციით. WordPress იყენებს MySQL მონაცემთა ბაზას.

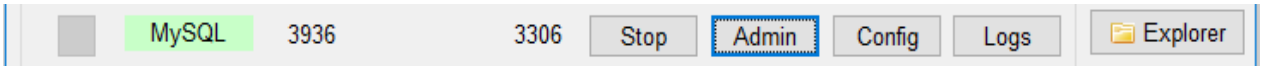
მოცემთა ბაზასთან წვდომა შესაძლებელია სხვადასხვა პროგრამული უზრუნველყოფის (პროგრამები) მეშვეობით . მაგ.:**phpMyAdmin**, **MySQL Workbench**, **SQLyog**და ა.შ.

მონაცემთა ბაზის შესაქმნელად საჭიროა შემდეგი ნაბიჯების გავლა:

1. ვხსნით Xamppcontrol panel;
2. Start - ღილაკების საშუალებით ვუშვებთ Apache და MySQL (იხ.სურ.1,4);
3. დამამთავრებელი ფაზა - ვიძახებთ მონაცემთა ბაზას **MySQL_ის Admin** ღილაკით (იხ. სურ. 3.1).

შედეგი 3 მიღება შესაძებელია ასევე ბრაუზერის სამისამართე ველში შემდეგი ბმულის გაწერითაც **localhost/phpmyadmin**(იხ. სურ. 3.2).

(გაითვალისწინეთ ეს ბმული იმუშავებს xampp_ისპარამეტრების საწყისიმნიშვნელობისას როცა პორტების მისამართები უცვლელია);



სურ3,1Admin ღილაკისმეშვეობით PhpMyAdmin პროგრამითმონაცემთაბაზასთანწვდომა

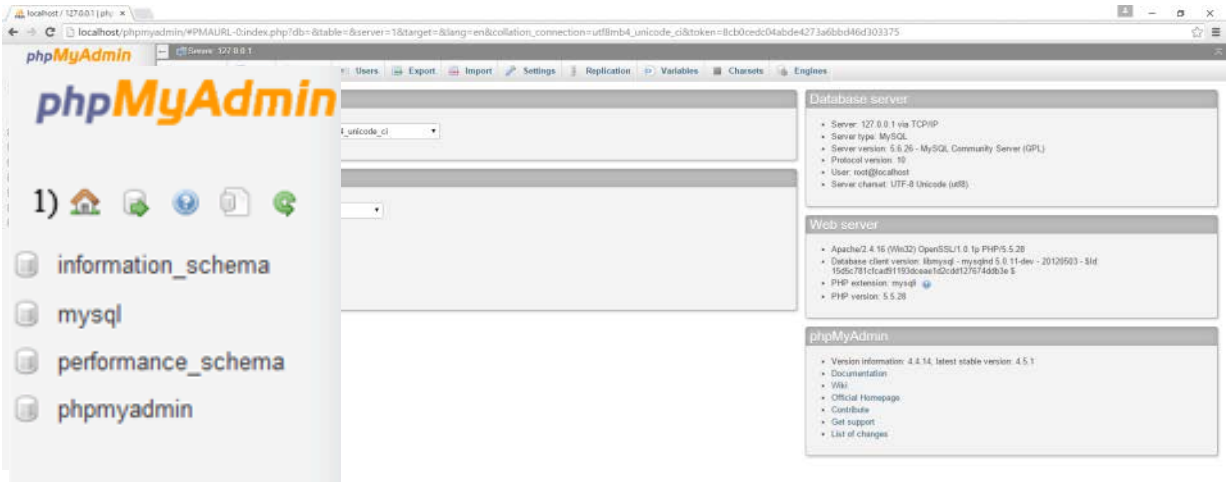
სურ3,2 ბრაუზერისსამისამართეველიდან PhpMyAdmin პროგრამითმონაცემთაბაზასთანწვდომა

მონაცემთა ბაზასთან წვდომა xampp_ის შემთხვევაში საწყისი მოცემულობით (default)შემდეგია:

- მომხმარებელი -**root**
- პაროლი - **არ გააჩნია**

გაითვალისწინეთ ყველა ვირტუალური სერვერის შემთხვევაში ეს მოცემულობა იდენტურად არ მუშაობს !

ინსტრუქციის ზუსთად შესრულების შემთხვევაში ბრაუზერში გამოტანილი ინფორმაცია იდენთურია სურათი 3.3

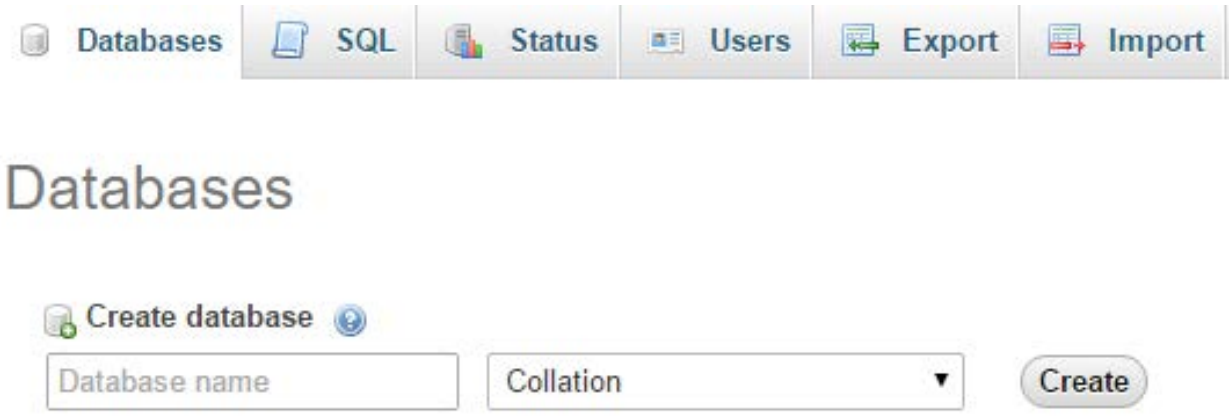


სურ3.3 PhpMyAdmin მეშვეობით MySQL_თანურთიეთობა

სურ3.4 საწყისიბაზები

მარცხენა არეალში გამოტანილია უკვე არებული მონაცემთა ბაზები.(იხ.სურ. 3.4) დაუშვებელია არსებული მონაცემთა ბაზების წაშლა, გამომდინარე იქიდან, რომ ისინი ინახავენ მონაცემთა ბაზებისა და ცხრილების ინფორმაციებს. მაგ.: მონაცემთა ბაზა **information_schema**უზრუნველყოფს სხვადასხვა ბაზების ინფორმაციასთან წვდომას - ბაზების, ცხრილების სახელებთან,სვეტების ტიპებთან.

მომხმარებლის მიერ მონაცემთა ბაზის დასამატებლად საჭიროა ვიხელმძღვანელოდ შემდეგი ინსტრუქტაჟის მიხედვით. (იხ. სურ. 3.5)



სურ3.5 ახალიმონაცემთაბაზისშექმნა

ბმულის მეშვეობით **Database** ვააქტიურებთ სურათზე(იხ სურ. 3.5.1) ნაჩვენებ მოცემულობას

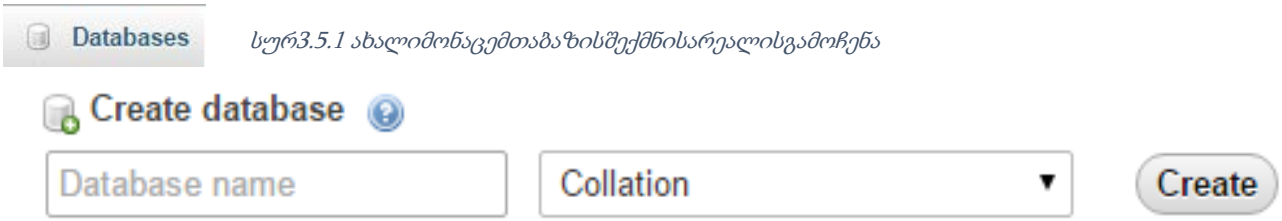
Create databaseარეალში ვუთითებთ ახალი ბაზის სახელწოდებას (იხ სურ. 3.5.2). ბაზის სახელის შერჩევისას უმჯობესია თუ გავითვალისწინებთ შემდეგ დებულებებს:

1. სახელი იყოს ლოგიკურად შერჩეული;

2. ბაზის სახელი შედგებოდეს ლათინური ასოებისაგან.
3. სახელი იყოს მოკლე და ინფორმაციული
4. შეძლებისდაგვარად მოერიდოთ ბაზის სახელში ადგილის გამოტოვებას (my name);
5. თუ დგას აუცილებლობა რამოდენიმე სიტყვიანი სახელის შექმნისა გამოვიყენოთ „_“ ქვედა ტირე (my_name)
6. ტირეების გამოყენება დაყვანილი უნდა იყოს მინიმუმამდე;

სახელის მინიჭების შემდეგ ბაზას ჭირდება კოდირების ენის განსაზღვრა - **Collation**>ჩანართში. (იხ სურ. 3.5.3) უნდა ავირჩიოთ**utf8_general_ci.** კოდირება - ეს არისგარკვეული სიმბოლოების ერთობლიობა. utf-8 ჩვენ შემთვევაში საკმაოდ ოპტიმალური ვარიანტია რადგანაც აერთიანებს უამრავ სიმბოლოს, მართ შორის ლათინურ, ქართულ , რუსულ და ა.შ.

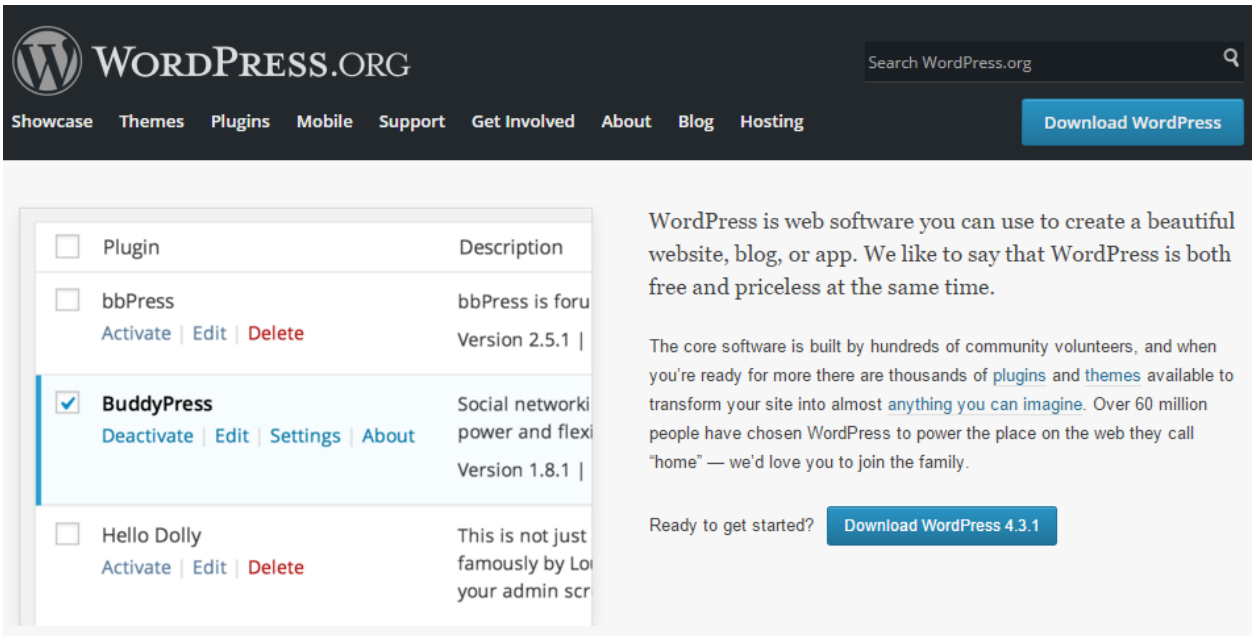
ბაზის სახელის და კოდირების განსაზღვრის შემდეგ **Create** ღილაკის მეშვეობით იქმნება ახალი მონაცემთა ბაზა(იხ სურ. 3.5.4), რომელიც ავტომატურად მარჯვენა არეალში გამოისახება. შევქმნათ მაგ.: **my_base** რომელსაც გამოვიწენებთ WordPress_ის საინსტალაციოდ.



სურ3.5.1 ახალიმონაცემთაბაზისშექმნისარეალისგამოჩენა

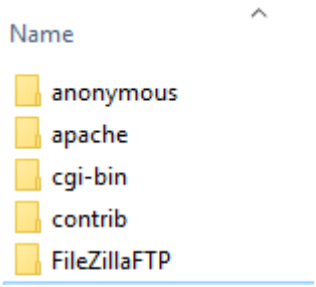
სურ3.5.2 ახალისახელისმინიჭებასურ3.5.3 კოდირებისგანსაზღვრასურ3.5.4 ბაზის შექმნა

ამ პროცედურების დასრულების შემდეგ საჭიროა უშუალოდ CMSWordPress_ის მოძიება. WordPress_ის ჩამოწერა შესაძლებელია მისი ოფიციალური ვებ გვერდიდან და ის სრულიად უფასოაhttps://wordpressorg/ (იხ. სურ. 4.1)



სურ4.1 CMS WordPress_ისსანსტალაციოფაილი

რეკომედირებულია ყოველთვის უახლესი ვერსიის დაყენება.

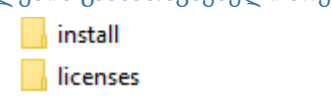


სურ 4.2 WordPress ფაილების განსათავსებელი არეალი

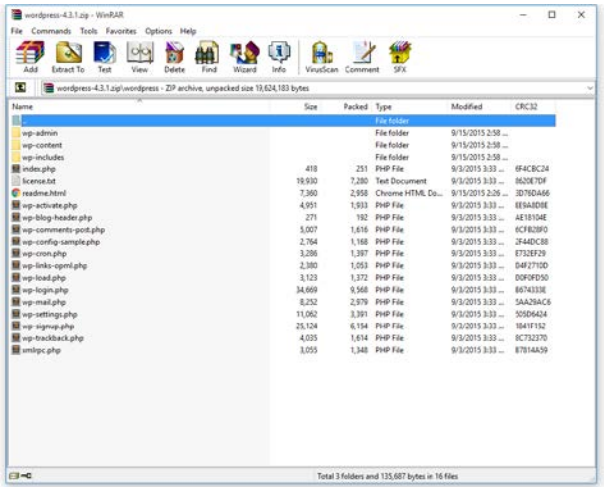
CMS ჩამოიწერება არქივის სახით.

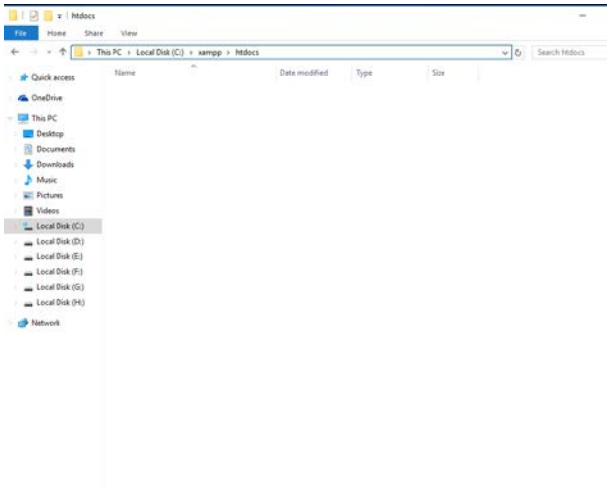
საწყის ეტაპზე xampp სერვერის ძირეულ საქაღალდში(იხ. სურ. 2.2.3) არსებული საქაღალდე **htdocs** ფაილებისაგან უნდა გასუფთავდეს. (იხ. სურ. 4.2)

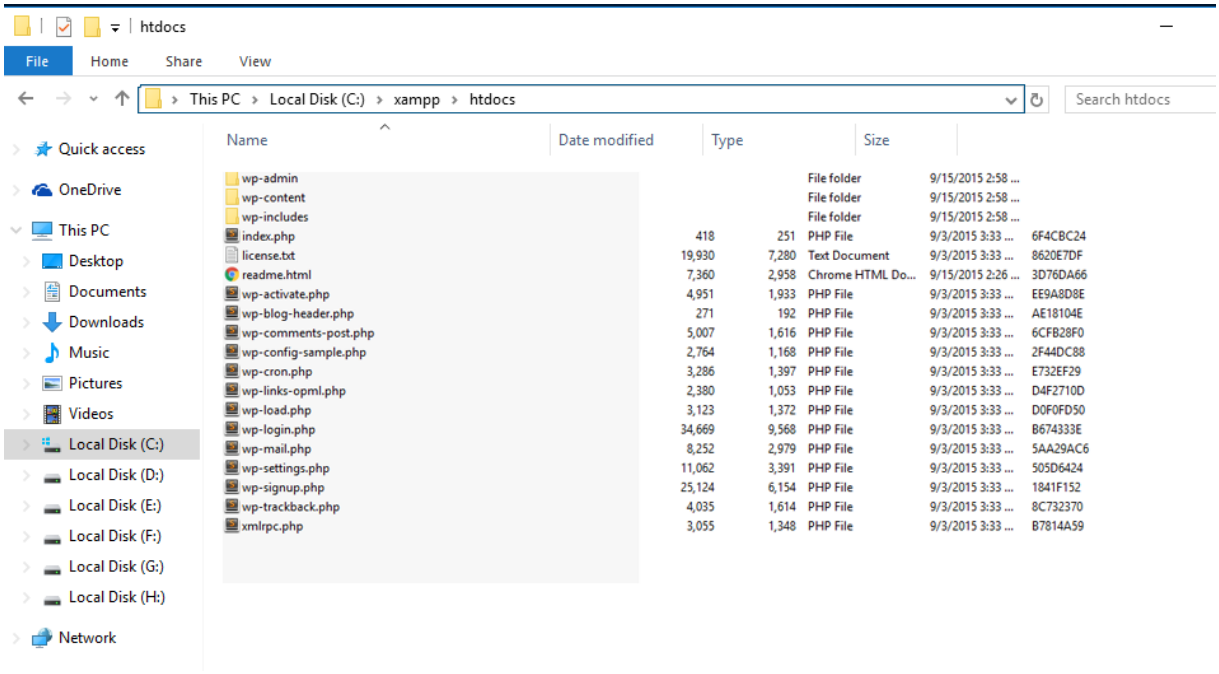
WordPress_ის არქივის განარქვივების შემდეგ ფაილების განთავსება უნდა მოხდეს სწორედ **htdocs**საქაღალდეში(პროცესი იხ.სურ. 4.3)



ახლა კი წინWordPress ინსტალაციისკენ !

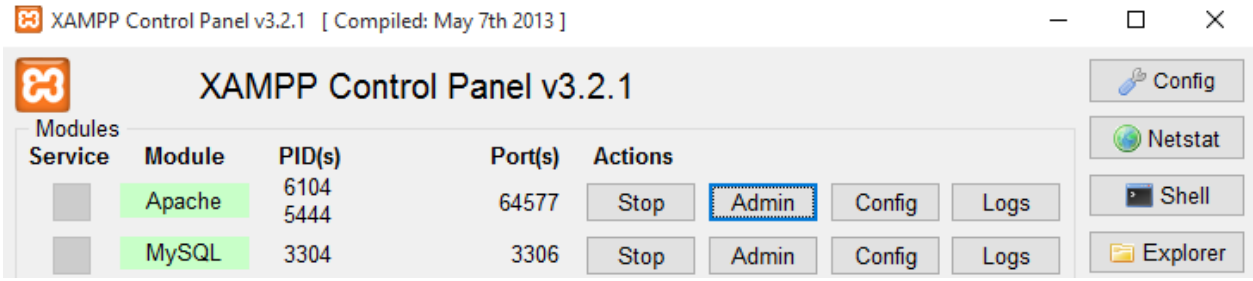






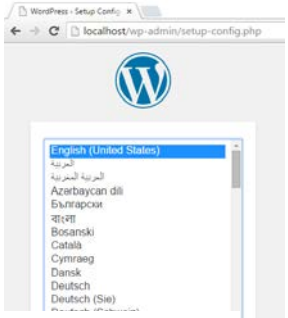
სურ4.3 WordPress ფაილისგანარქივებადამათიგანთავსებასერვერისძირითადსაქაღალდეში

გარემოსა და მონაცემთა ბაზის მომზადების შემდეგ გავხსნათ ბრაუზერი და მისამართში მივუთითოთ **localhost**(ეს იმ შემთხვევაში თუ xampp_ის კონფიგურაციიდან პორტები უცვლელია) ან გამოვიძახოთ სურათზე (იხ. სურ. 4.4) ნაჩვენები **Admin** ღილაკის მეშვეობით



სურ4.4ლოკალურიმისამართისგამოძახება Admin ღილაკისსაშუალებით

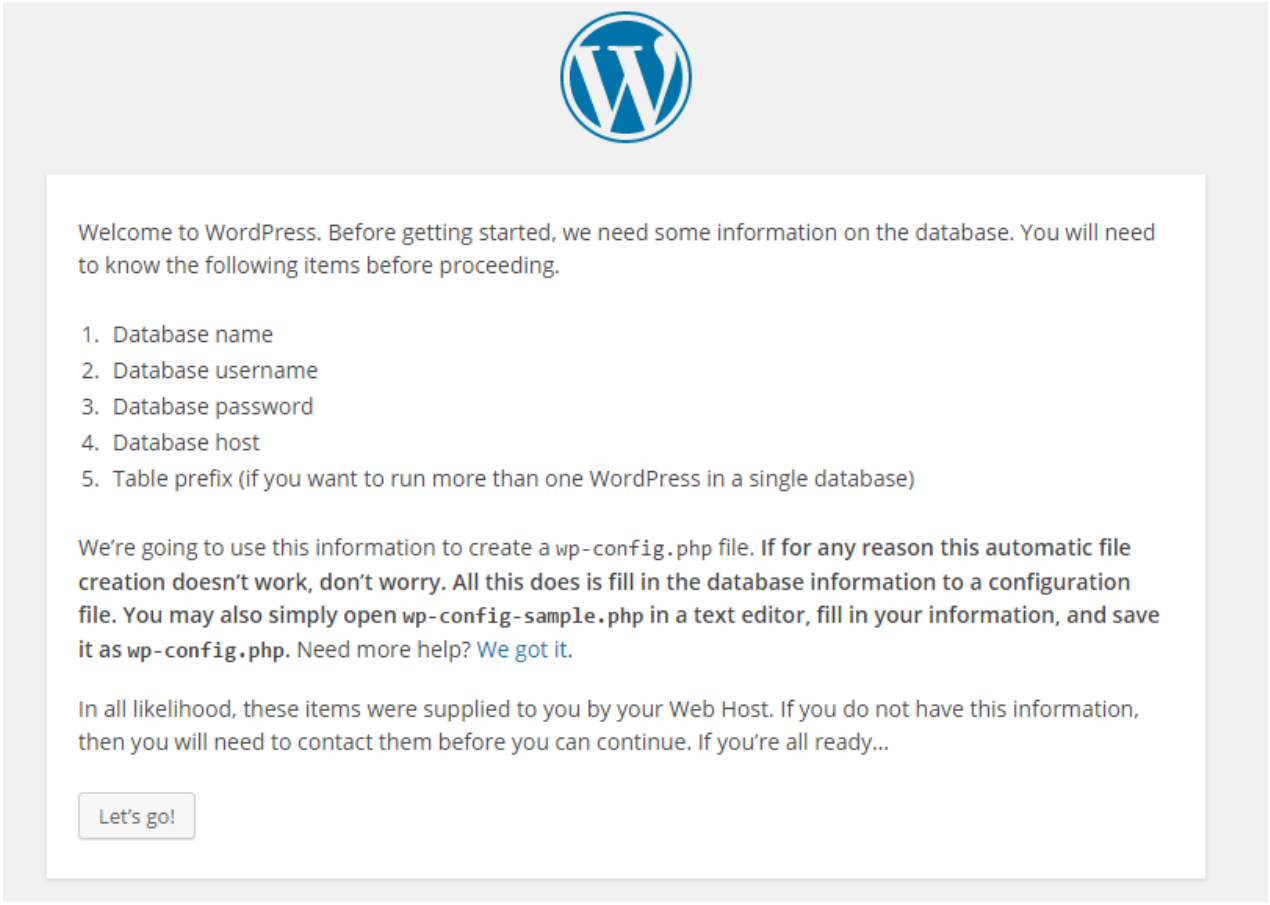
დაიმახსოვრეთ მუშაობის პროცესში დაუშვებელია xampp_ის გამორთვა!!!



სურ. 4.5.1Admin პანელის ენის შერჩევა

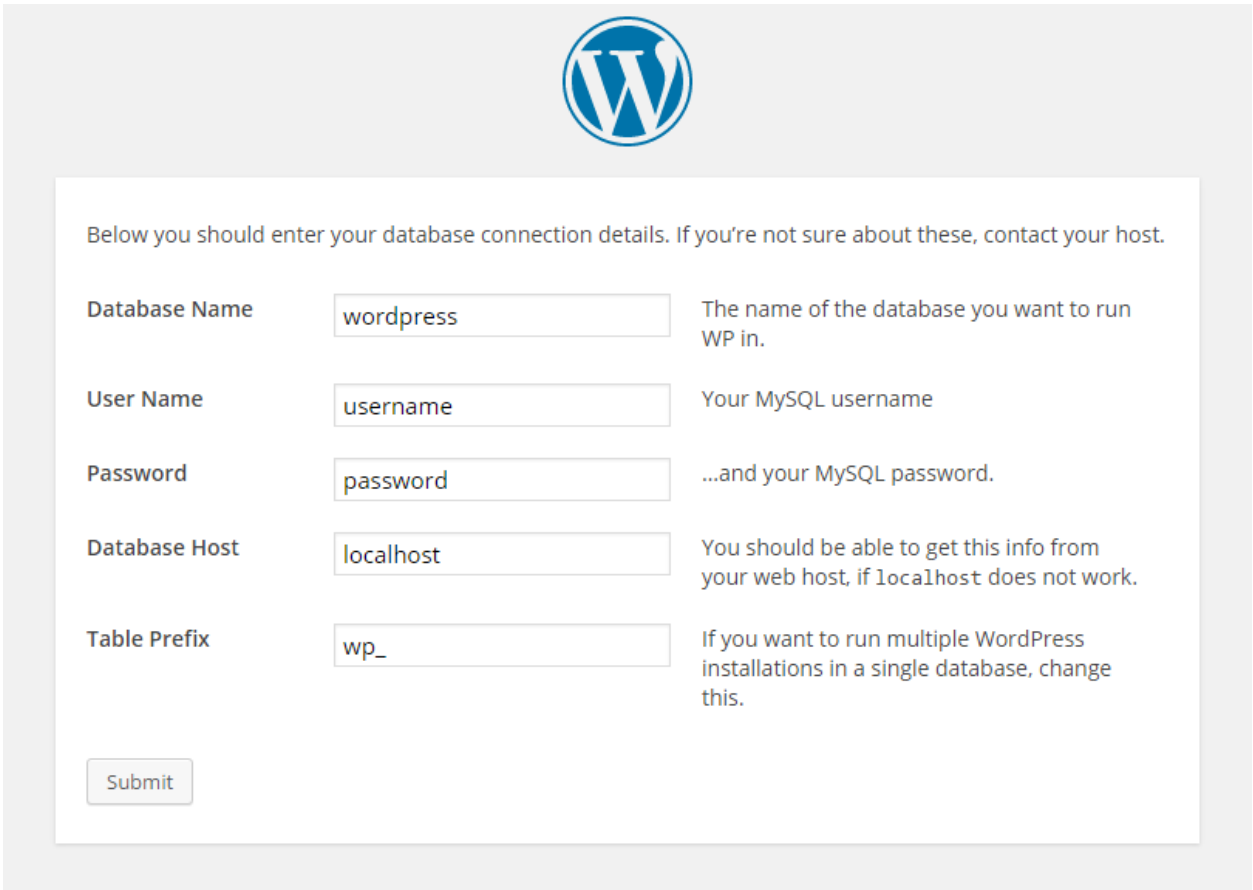
გახსნილ ბრაუზერში გამოდის WordPress_ის დასაყენებელი ნაბიჯები. იხელმძღვანელეთ სურათებისშემდეგი თანმიმდევრობთ (იხ. სურ. 4.5.1–4.5.7):

1. ვირჩევთ ადმინისტრირების (საიტის სამართავი) პანელის ენას (იხ.სურ. 4.5.1)
2. გამოტანილი ფანჯარა გვამცნობს კომფიგურაციის ფაილის ინფორმაციის ცვლილების შესახებ . საჭიროა მოხდეს მონაცამთა ბაზასთან დაკავშირება გადავდივართ**Let's go!**ღილაკის მეშვეობით შემდეგ ეტაპზე; (იხ.სურ. 4.5.2)



სურ 3.5.2საინფორმაციოფანჯარა

მონაცემთა ბაზასთან კავშირის დასამყარებლად და მასში WordPress_ის ცხრილების ინტეგრაციისათვის საჭიროა მოცემული ველების ზედმიწევნით სწორედ შევსება (იხ.სურ. 4.5.3)



სურ4.5.3მონაცემთაბაზასთანკავშირისდამყარება

Database Name >შეგვყავს მონაცემთა ბაზის სახელი. ბაზისა რომელიც შევქმენით PhpMyAdmin_ის მეშვეობით. მაგ.: my_base.

User Name > root . ეს არის მონაცემთა ბაზასთან წვდომის მომხმარებელი. როგორც ავღნიშნეთ xampp_ის შემთხვევაში მომხმარებელი არის - root რომელსაც პაროლი არ გააჩნია.

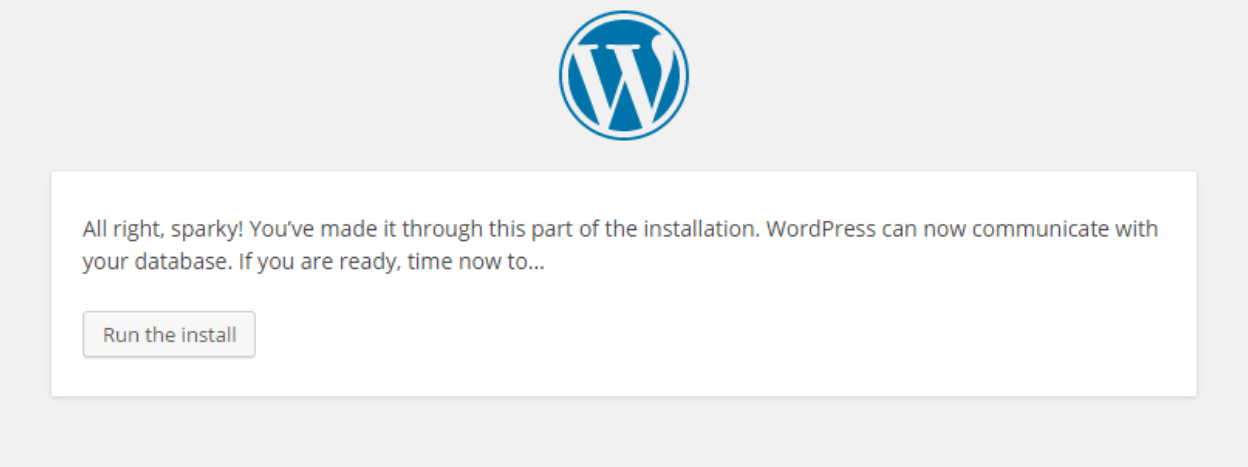
შესაბამისად შემდეგ ველში- **Password** ვირჩევთ სიცარიელეს.

Database Host - აქ ვტოვებთ საწყის მნიშვნელობა localhost_ს.

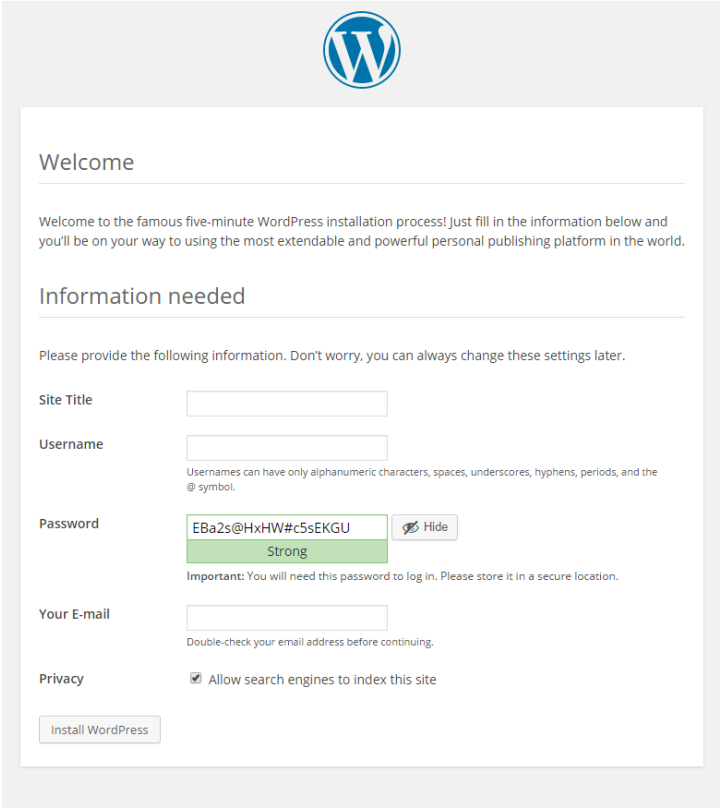
Table Previx - უმჯობესია პრეფიქსის შეცვლა მოხდეს ნებისმიერი სიმბოლოებით . ეს ერთ ერთი რეკომენდაციაა WordPress ცხრილების დასაცავად. მაგ.: თუ საიტი ჩვილების შესახებაა wp_ შევცვალოთ baby_ .

Submit - შეყვანილი ინფორმაციის დასტური.

WordPress_ ის ინსტალაცია - >**Run The install.**(იხ. სურ. 4.5.4)



სურ4.5.4 WordPress_ისინსტალაცია



სურ4.5.5 ინსტალაციისდამამთავრებელინაბიჯები

შემდეგი ნაბიჯი WordPress ინსტალაციის დასასრულის მიგვაახლოვებს (იხ.სურ. 4.5.5).

ამ პარამეტრების ცვლილება საიტის ადმინისტრირების პანელიდანაც შესაძლებელია ასე რომ რაიმე შეცდომის შემთხვევაში არ ღირს დამწუხრება.მნიშვნელოვანია **არ დაგავიწყდეთ მომხამრებლის სახელი და შეყვანილი პაროლი.**

Site Title -საიტის სათაური;

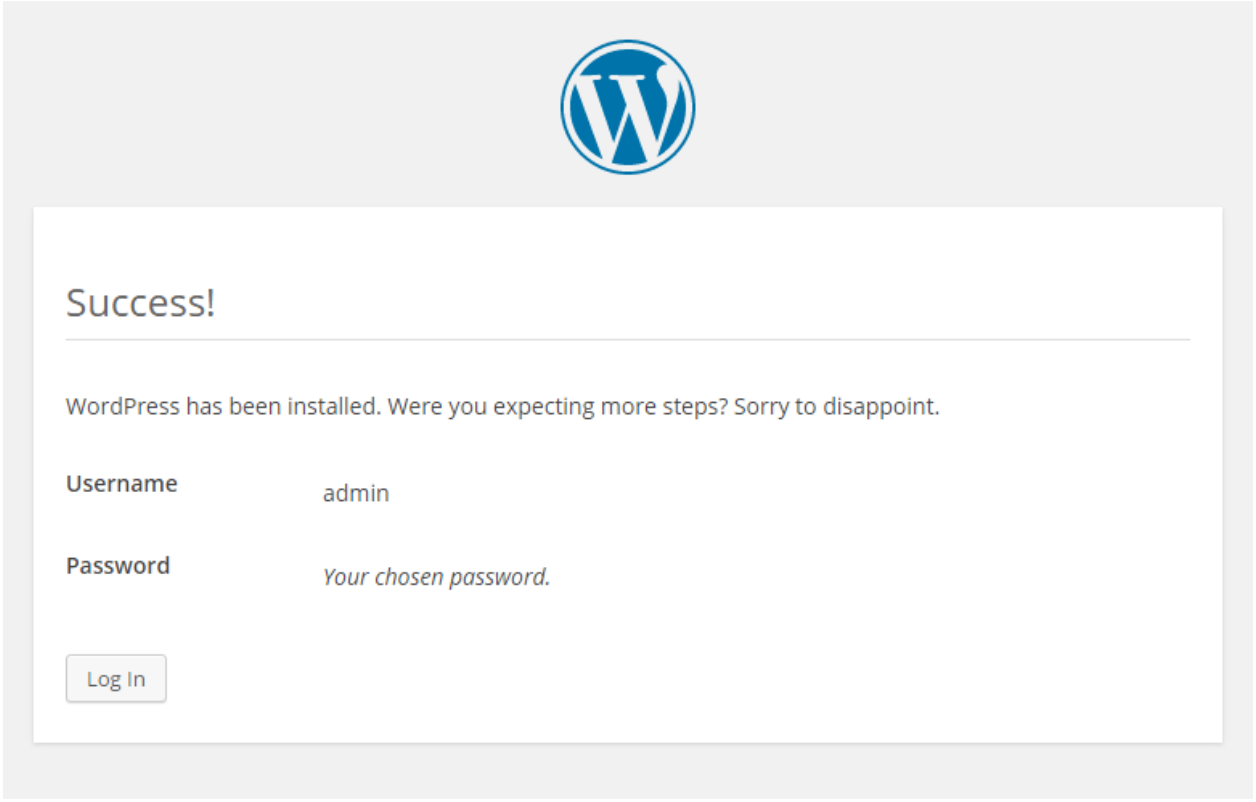
Username - ადმინისტრირების პანელში შემსვლელი მომხმარებლის სახელი. მაგ.(adm);

Password -მომხმარებლის პაროლი. მაგ.(adm123);

Your E-mail - მომხმარებლის საკონტაქტო ფოსტა;

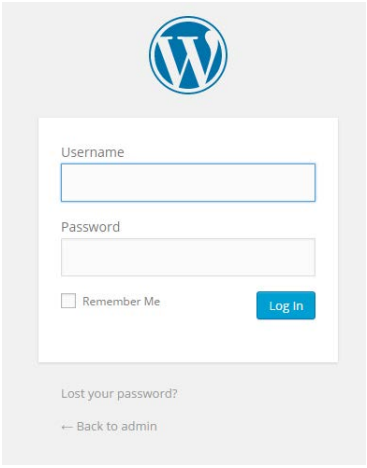
Privacy – ველის მონიშვნის შემთხვევაში საიტი წვდომადი იქნება საძიებო სისტემებისთვის

Install Wordpress - ინსტალაციის დასასრული (იხ. სურ. 4.5.6)



სურ4.5.6 ინსტალაციისდასრულებისასგამოსულიშეტყობინება

Log in - საიტის ადმინისტრირების პანელზე გადასვლა (იხ.სურ. 4.5.7)



4.5.7 ადმინპანელშიშესვლა

ამ პანელზე გადასვლილსას უნდა გამოიყენოთ ის Username და Password რაც ორიოდე წუთის წინ თქვნს მიერვე იქნა რეგისტრირებული.

- username- **adm**
- password – **adm123**

თუ სახელისა და პროლის დამახსოვრება გსურთ მონიშნეთ**Remember Me**ველი.

Log Inღილაკის მეშვეობით შეასრულეთ სამართავ პანელში შესვლის ოპერაცია.

WordPress _ ის საიტის ინსტალაციის დასასრული საშუალებას გვაძლევს ადმინისტრირები პანელში შეუსვლელად ვიხილოთ შედეგი ბრაუზერის სამისამართე ველში localhost_ზე გადასვლით. (იხ.სურ. 4.6)

სურ4. 6 საიტისმთავარიგვერდისგამოძახება

დაიმახსოვრეთ მუშაობის პროცესში დაუშვებელია xampp_ის გამორთვა!!!

აღნიშნული გვერდის წარმატებით ჩატვირთვის შემთხვევაში გაიხსნება WordPress_ის ის კონკრეტული თემა რომელიც აქტუალურია მიმდინარე ვერსიისათვის.

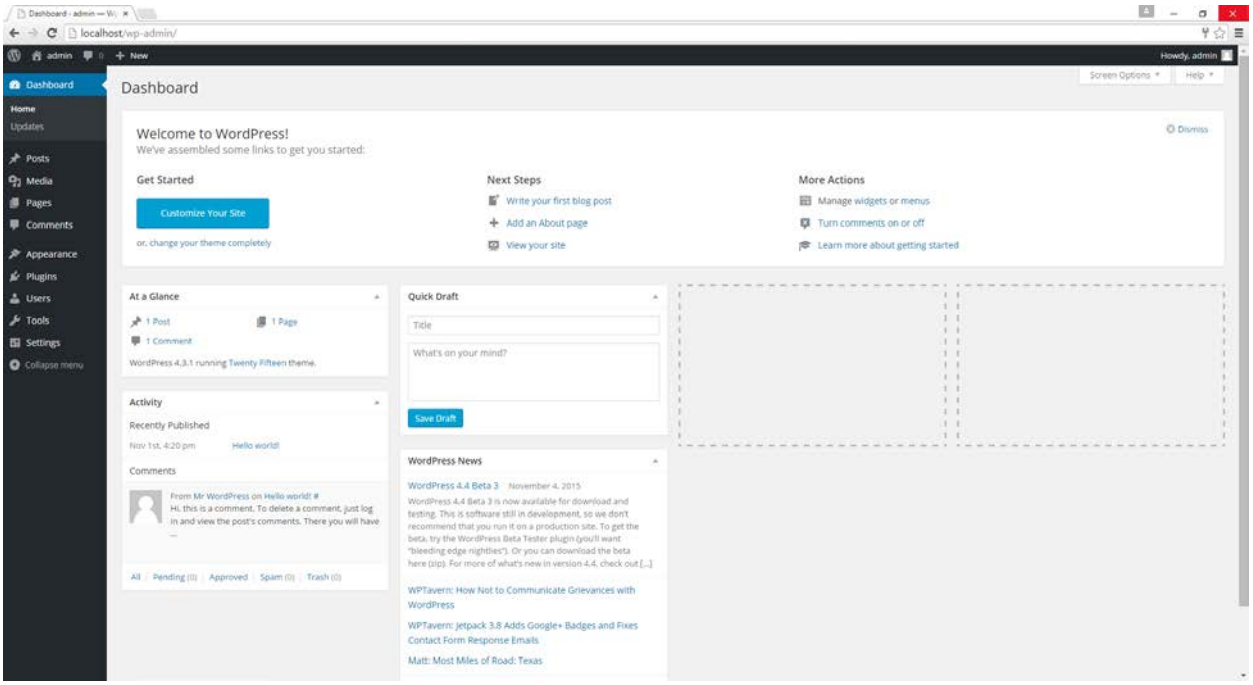
WordPress_ის ინსტალაციის დასრულების შემდეგ შესაძლებელია საიტის სამართავ სისტემაში შეღწევა. რეგისტრაციის საბოლოო ეტაპზე WordPress ავტომატურად სთავაზობს მომხმარებელს ავტორიზაციის ველებს. (იხ.სურ. 4.5.7)

ადმინისტრირებადი პანელის ავტორიზაციის არეალის გამოჩენისათვის მომხმარებელმა ბრაუზერის სამისამართე პანელში უნდა შეიყვანოს **საიტის დომენი /wp-admin**. ამ კონკრეტულ შემთხვევაში, მაშინ როცა საიტი განთავსებულია ვირტუალურ სერვერ xampp_ის htdocs დირექტორიაში დამაკმაყოფილებელი იქნება **localhost/wp-admin**(xampp_ის ავტომატური კონფიგურაციის შემთხვეაში) მისამართის მითითება. (იხ.სურ. 5,1)

სურ5.1 საიტისსამართავიპანელისავტორიზაციისგვერდისგამოძახება

მიმდინარეეტაპის წარმატებით დაძლევის შემთხვევაში ბრაუზერში გამოისახება ადმინ პანელის ავტორიზაციის ფორმა (იხ.სურ. 4.5.7). საწყის ეტაპზე მაშინ, როცა რეგისტრირებულია მხოლოდ ერთი მოხმარებელი, ავტორიზაცია უნდა განხორციელდეს იმ კონრეტული სახელისა და პაროლის მიხედვით რომელიც მომხმარებელმა შეიყვანა WordPress_ის რეგისტრაციის დროს (სურ4.5.5) ჩვენ შემთხვევაში: სახელი - **adm,** პაროლი **- adm123.**

მომხმარებლის სახელისა და პაროლის სწორედ შეყვანის შემთხვევაში დამყარდება წვდომა ადმინისტრირების პანელთან (იხ.სურ.5.2)



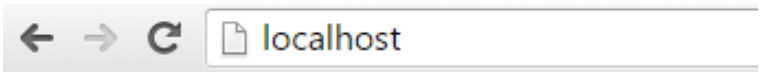
სურ5.2საიტისსამართავიპანელი

საიტის სამართავი სისტემის ვიზუალური იერსახე შეიძება მცირეოდენით განსხვავდებოდეს მოყვანილი მაგალითსგან. გასათვალისწინებელია ის რომ WordPress_ის CMS ხშირ განახლებას და დახვეწას განიცდის. გამომდინარე აქედან შესაძლოა ამ ინფორმაციის გაცნობისას ადმინ პანელის

დიზაინი არ იყოს იდეალურ თანხვედრაში მიმდინარე ვერსიასთან, მაგრამ ფუნდამენტური ცვლილებები დიდი ალბათობით არ იქნება.

ადმინისტრირები პირველი გვერდი საინფორმაციო სახისაა. მის სრულ განხილვას ჩვედა თავებში შევეცდებით ახლა მხოლოდ საიტის მართვის პარამეტრებითა და მათი გამოყენებით შემოვიფარგლოთ.

სანამ უშუალოდ საიტის მართვის პარამეტრებს გაეცნობით გახსენნით ვებ გვერდი მისამართ **localhost** გამოძახების საშუალებით (იხ.სურ.5.3)



სურ. 5.3 საიტისგამოძახება

კონკრეტული მომენტისათვის აქტუალური WordPress ვერსია იყენებს თავისათვის მისაღებს თემას (გავეცნობით ქვედა თავებში), ამიტომ არსებული საიტის დიზაინი (იხ სურ. 5.4) შეიძლება განსხვავდებოდეს იმ მოცემულობისაგან რომელშიც თქვენ იმყოფებით. პარამეტრების კონფიგურირების არსის გაგება დაგეხმარებათ ამ მცირეოდენი დისკომფორტის (რა თქმა უნდა თუ საერთოდ შეგექმნათ) თავიდან არიდებაში.

სურ. 5.4 საიტისპირველიგვერდი (მიმდინარევერსიისაქტუალურითემა)

სურ. 5.4_ ზე ბლოკებად გამოყოფილია აქტუალური საიტის ნაწილები რომელთა კონფიგურაციაც შესაძლებელია ადმინისტრირების პანელიდან ერთ ერთი მენიუს პუნქტის **Setting** - პარამეტრების საშუალებით.

გავყვეთ ეტაპობრივად.

WordPress_ის ინსტალაციის დამამთავრებელ ფაზაში მოცემულობის თანახმად შესავსები იყო შემდეგი ველები:

Site Title - საიტის სათაური;

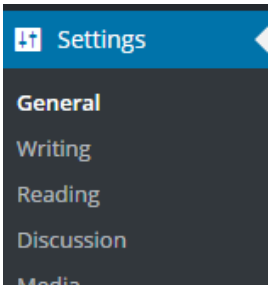
Username - ადმინისტრირების პანელში შემსვლელი მომხმარებლის სახელი.

Password -მომხმარებლის პაროლი;

Your E-mail - მომხმარებლის საკონტაქტო ფოსტა; (იხ.სურ. 4,5,5)

ამ ველების შევსებისას აღინიშნა, რომ მიმდინარე ეტაპზე შეყვანილი პარამეტრების კონფიგურირება შესაძლებელია ადმინისტრირების პანელიდან.

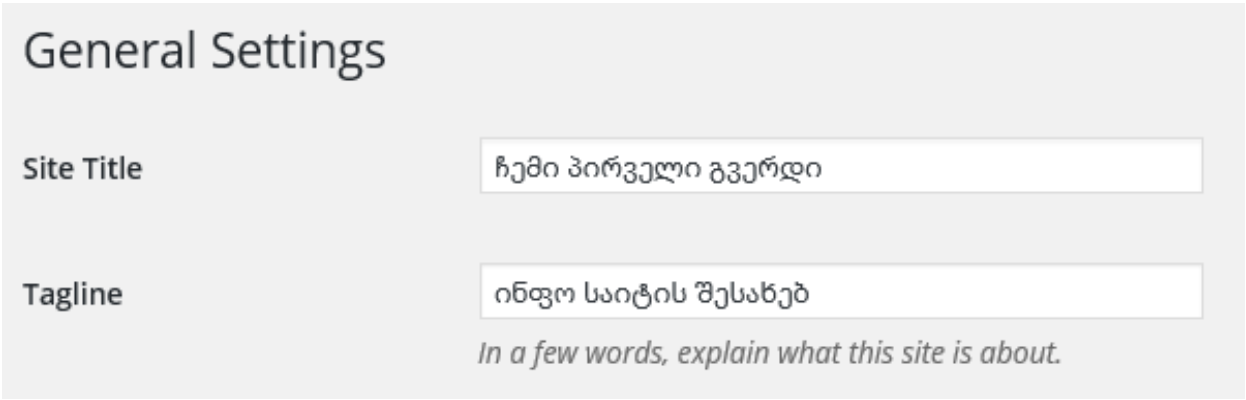
საიტის პარამეტრების ცვლილებისათვის საჭიროა **Settings** მენიუს გააქტიურება. (იხ.სურ.5.5)



სურ. 5.5 საიტის პარამეტრები

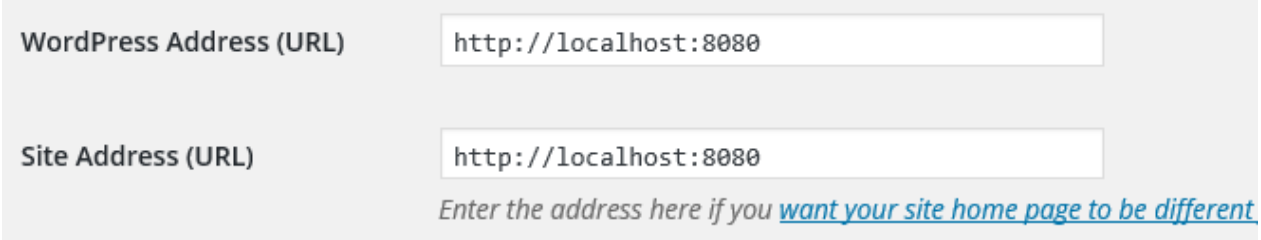
General ჩანართი პასუხისმგებელია სხვადასხვა ტიპის ინფორმაციის მართვაზე.

- **Site Title** - საიტის სათაური. მისი ცვლილება ავტომატურ რეჟიმში გამოიწვევს საიტის სათაურის ცვლილებას რომელიც მიმდინარე ეტაპზეგანთავსებულია სურ. 5.4 - **I -ბლოკში**; (იხ.სურ. 5.5.1.1)
- **Tagline** – მოკლე აღწერა. ტექსტი, რომელიც შეიცავს ინფორმაციას საიტის თემატიკის შესახებ. ის უნდა იყოს რამოდენიმე სიტყვანი მცირე წინადადება. შინაარსობრივად დატვირთული - Tagline მცირეოდენ უპირატესობას ანიჭებს საიტს საძიებო სისტემებთან მიმართებაში, ამიტომ ჯობია არსებობდეს ვიდრე საერთოდ არ მიეთითოს. სურ. 5.4 - **II -ბლოკში**;



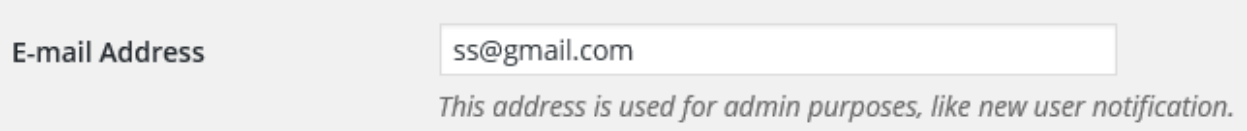
სურ. 5.5.1.1 საიტისსათაურისადამოკლეაღწერისცვლილება

- **WordPress Address (URL)** - ეთითება მისამართი თუ სად არის WordPress ძირითადი ფაილები განთავსებული. მიმდინარე ეტაპზე საიტის ფაილური სტრუქტურა განთავსებულია სერვერის გამშვებ საქაღალდე htdocs_ში. თუ იქნებოდა შემთხვევა მაგ.: ფაილებს განვათავსებით საქაღალდე **site**_ში მაშინ მისამართი დაკორექტირდებოდა http://localhost/site(იხ.სურ. 5.5.1.2)
- **Site Address (URL)** - ეთითება საიტის მისამართი საიდანაც უნდა მოხდეს წვდომა/შესვლა საიტზე. მაგ.: http://anydomain.ge . ამ კონრეტულ შემთხვევაში WordPress Address(URL) იდენტურია Site Address (URL) _ის რადგანაც საიტი განთავსებული ლოკალურ მისამათზე მთავარ საქაღალდეში (htdocs – xampp_ის შემთხვევაში) (იხ.სურ. 5.5.1.2)



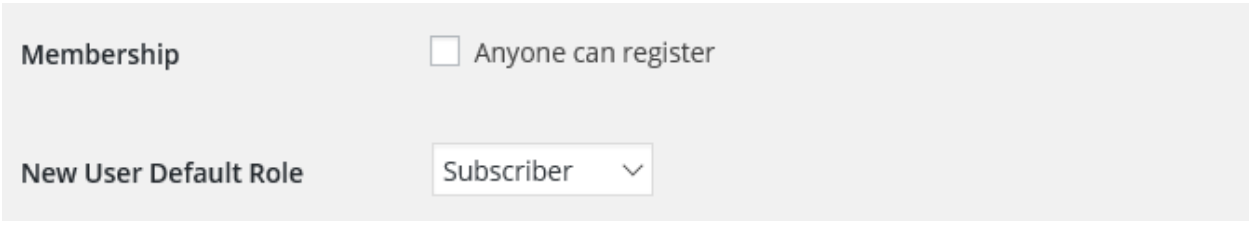
სურ. 5.5.1.2 საიტისფაილებისმისამართები

- **E-mail Address** -ველის მეშვეობით შესაძლებელი ელექტრონული ფოსტის ცვლილება; (იხ.სურ. 5.5.1.3)



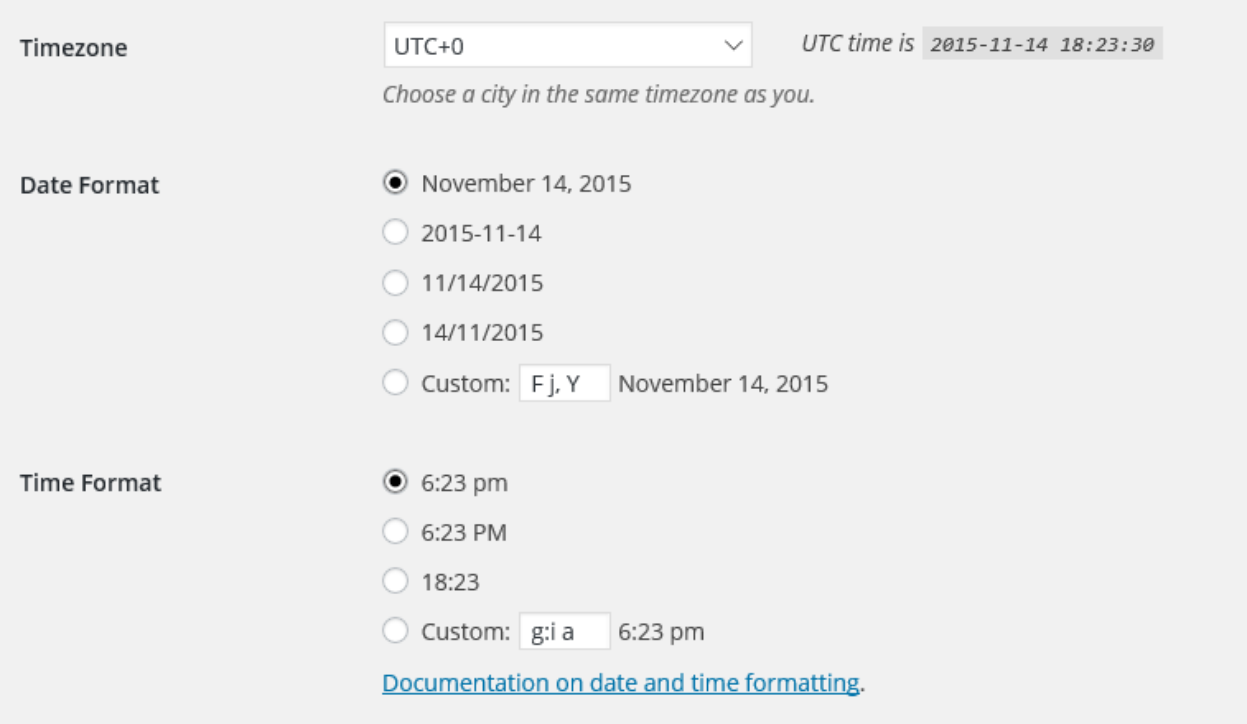
სურ. 5.5.1.3 ელ. ფოსტისმისამართისცვლილება

- **Membership – Anyone can register** - ველის მონიშვნა / გაატირურება გამოიწვევს შეზღუდვის მოხსნას რეგისტრაციის მსურველებისთვის - ყველას შეეძლება რეგისტრაცია;)(იხ.სურ. 5.5.1.4)
- **New User Default Role** - ამ ჩანართის მეშეობით შესაძლებელია საწყისი უფლებამოსილებების განსაძღვრა ყოველი ახალი მომხმარებლისათვის. მისი თითოეული პუნქტი განხილული იქნება უშუალოდ მომხმარებლის რეგისტრაციის პროცესის გარჩევისას.)(იხ.სურ. 5.5.1.4)



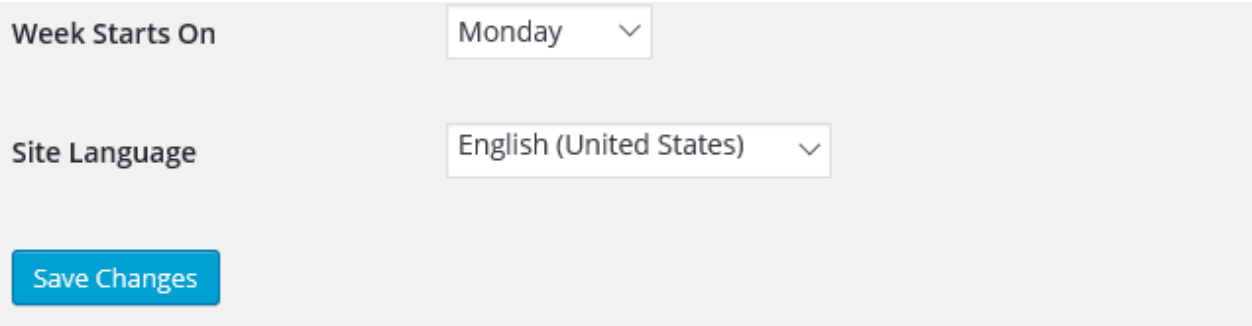
სურ. 5.5.1.4 რეგისტრაციისხილვადობადასაწყისიუფლებებისგანსაზღვრა

- **Timezone** - არჩეული დროის სარტყელი უზრუნველყოფს საიტზე გამოქვეყნებული სტატიის, გვერდის დროის პარამეტრების უზრუნველყოფას;(იხ.სურ. 5.5.1.5)
- **Data Format** - თარიღის ვიზუალიზაცია. (იხ. სურ. 5.4 - **IV -ბლოკი)** სტანდარტული 4 ვარიანტის ქვემოთ მოცემულია სასურველი არჩევითი ფორმატის შესაყვანი არეალი. თარიღის დაწვრილებითი ფორმატირების შესახებ ინფორმაცის გაეცანით მოცემულ ბმულზე https://codex.wordpress.org/Formatting_Date_and_Time (იხ.სურ. 5.5.1.5)
- **Time Format** - დროის ვიზუალიზაცია. სტანდარტული 4 ვარიანტის ქვემოთ მოცემულია სასურველი არჩევითი ფორმატის შესაყვანი არეალი. დროის დაწვრილებითი ფორმატირების შესახებ ინფორმაცის გაეცანით მოცემულ ბმულზე https://codex.wordpress.org/Formatting_Date_and_Time(იხ.სურ. 5.5.1.5)



სურ. 5.5.1.5 დროისსარტყლის, თარიღისადადროისფორმირება

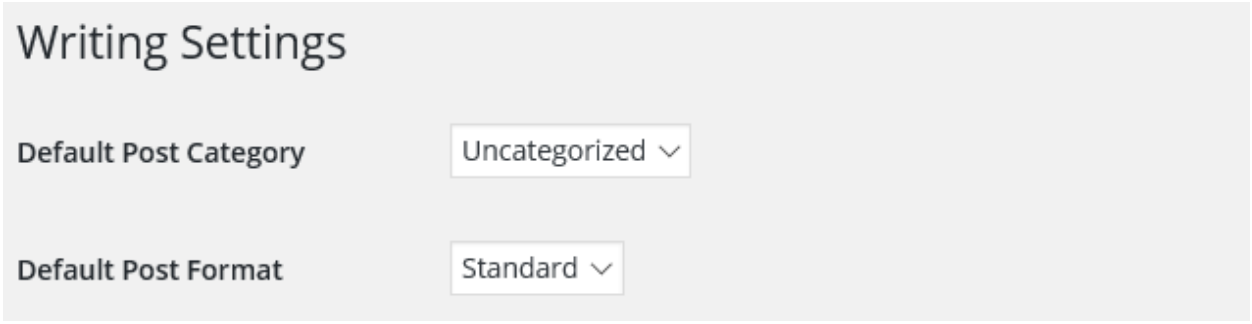
- **Week Start On** -კვირის საწყისი დღის მითითება;(იხ.სურ. 5.5.1.6)
- **Site Language** -საიტის სამართავი პანელის ენის არჩევა (იხ.სურ. 5.5.1.6)
- **Save Changes** - ღილაკის მეშვეობის ხდება დატანილი ცვლილებების შენახვა;



სურ. 5.5.1.6 კვირისსაწყისიდღისდასამართავიპანელისენისშერჩევა

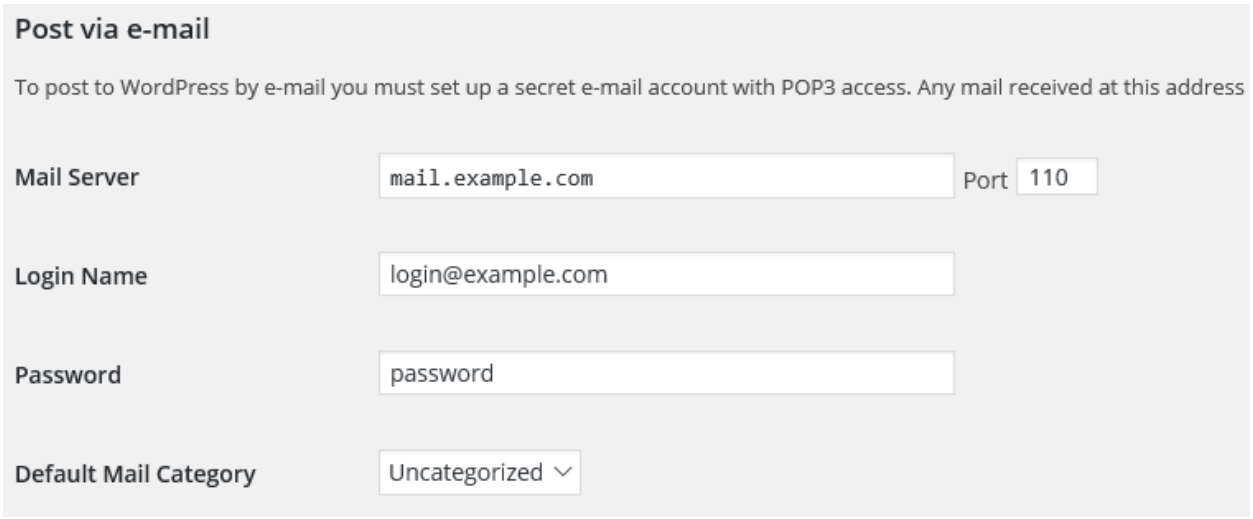
Writing - ადმინისტრირების პანელში დამატებული ინფორმაციის საწყისი პარამეტრების კონფიგურაცია.

- **Default Post Category** - საწყისი კატეგორიის განსაზღვრა. ყოველი ახალი დამატებული ჩანაწერის სტანდარტულ/საწყის კატეგორიად მითითებული იქნება ამ ველში არჩეული მნიშვნელობა. მინდინარე ეტაპზე მხოლოდ ჩანაწერების ერთი კატეგორია არსებობს - **Uncategorized,** ამიტომ ის საწყის მნიშვნელობადაა დაყენებული. (იხ.სურ. 5.5.2.1)
- **Default Post Format** - ჩანაწერების საწყისი ფორმატის განსაზღვრა. თითოეული ფორმატის მნიშვნელობაზე უშუალოდ ჩანაწერების დამატების გარჩევისას გავამახვილებთ ყურადღებას. (იხ.სურ. 5.5.2.1)



სურ. 5.5.2.1 ჩანაწერებისსაწყისიკატეგორიისადაფორმატისშერჩევა

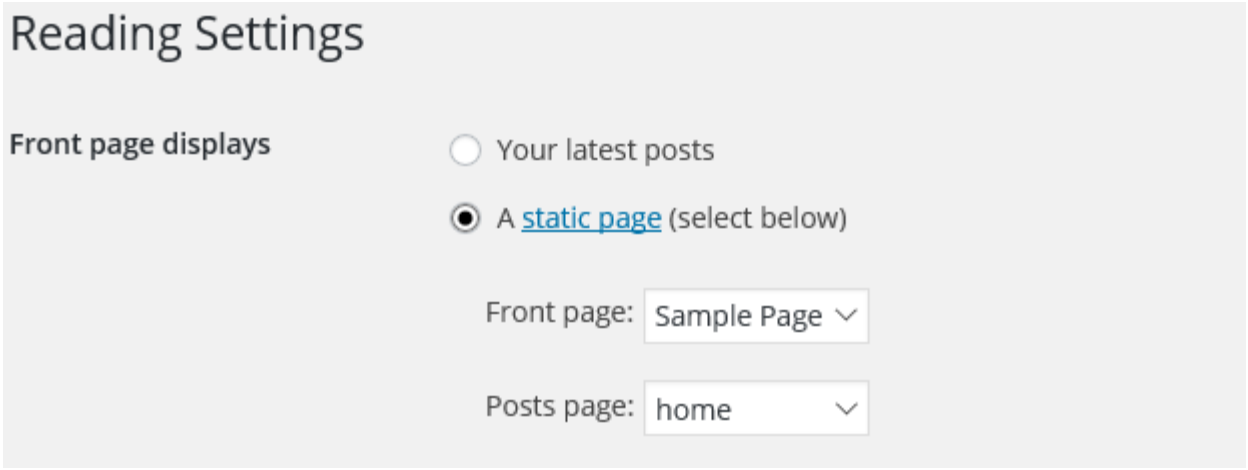
- **Post via e-mail -** ჩანაწერების გამოქვეყნება ელ. ფოსტის საშუალებით. იმისათვის რომ WordPress_ის ჩანაწერების გამოქვეყნება მოხდეს ელ. ფოსტის საშუალებით საჭიროა შეიქმნას დაცული ელ. ფოსტა POP3 წვდომით. POP3 წარმოადგენს სპეციალურ პროტოკოლს რომელიც უზრუნველყოფს საფოსტო პროგრამებსა დ ა საფოსტო ყუთს შორის კავშირს. ჩანაწერების ელ.ფოსტით გამოქვეყნებისათვის საჭიროა ზედმიწევნით ზუსტად შეივსოს შემდეგი ველები **Mail Server -** მეილ სერვერი, **Port -** პორტი, **Login Name -** მომხმარებლი, **Password -** პაროლი. **Default Mail Category -** შესაძლებელი მიეთითოს ჩანაწერების საწყისი კატეგორია. (იხ.სურ. 5.5.2.2)



სურ. 5.5.2.2 მეილსერვერისპარამეტრებისმართვა

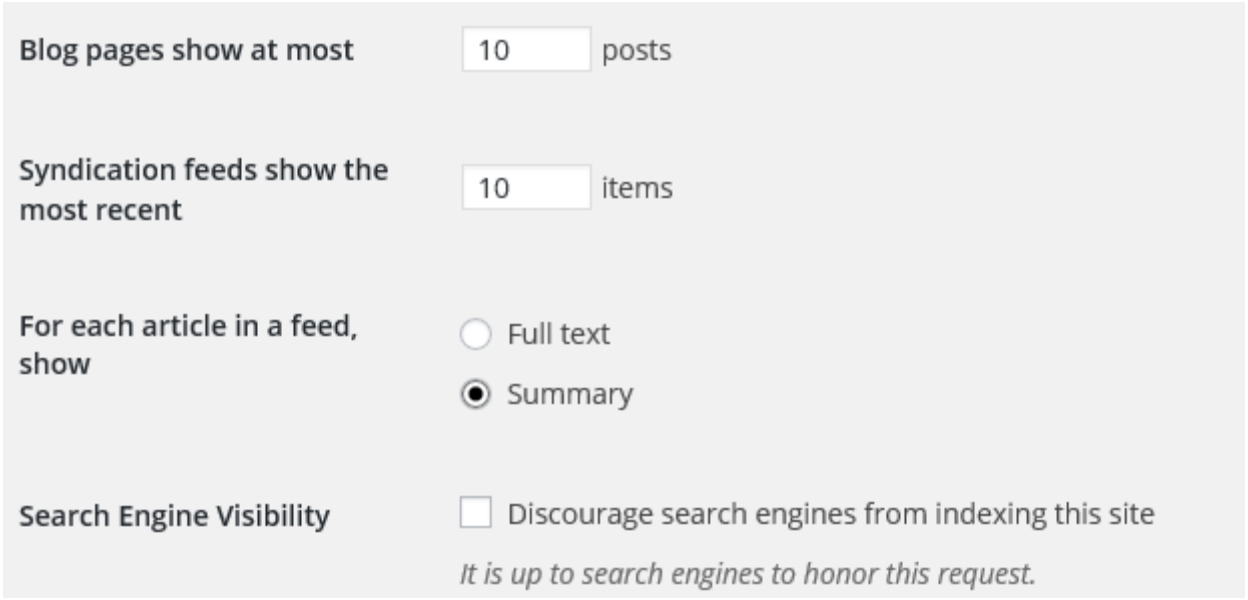
Reading – მენიუს პუნქტის მეშვეობით შესაძლებელია ვებ საიტის პირველი გვერდის შემცველობის კონფიგურაცია.(იხ.სურ. 5.8) კერძოდ:

- **Front page displays -** ჩანართი უზრუნველყოფს საწყის გვერდზე ჩანაწერების გამოტანას (იხ. სურ. 5.4 III - ბლოკი - ჩანაწერის გამოტანა). საწყის მნიშვნელობად არჩეულია **Your latest posts -** მნიშვნელობა, რაც ნიშნავს ეკრანზე უკანასკნელი პოსტების (პოსტი - ამ ცნებას ცოტა მოგვიანებით გავეცნობით) გამოტანას. ალტერნატიული საშუალებაა **A static page (select below)** - სადაც შესაძლებელია 1. **Front page**: საწყის არეალში (იხ. სურ. 5.4 III - ბლოკი) არა ყველა პოსტის არამედი კონკრეტული გვერდის ინფორმაციის გამოტანა. მაგ.: მიმდინარე სურათ 5.5.3.1_ზე მითითებულია Sample Page (default გვერდი). 2. Posts page: მიმდინარე ველში შესაძლებელია გვერდის შერჩევა რომელიც ხდება კონტეინერი უკანასკნელი უკანასკნელი პოსტებისა; მაგ.: მიმდინარე სურათ 5.5.3.1_ზე მითითებულია home (თქვენს შემთხვევაში აქ მხოლოდ Sample Page გამოჩნდება)



სურ 5.5.3.1 საწყისიგვერდზეჩანაწერებისგამოტანისკონფიგურაცია

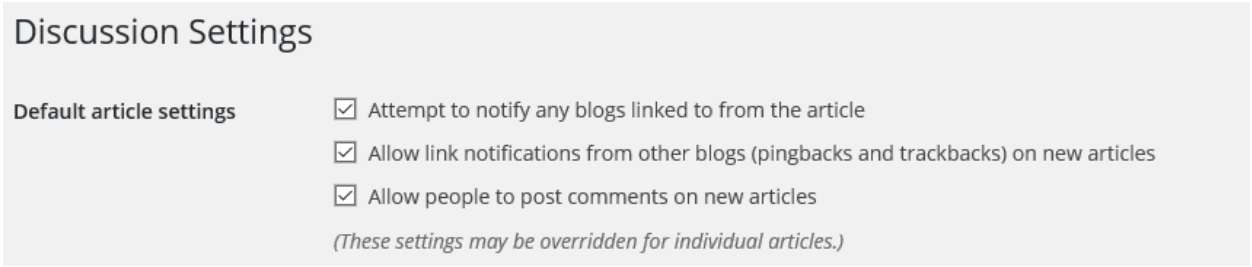
- **Blog pages show at most** - მიმდინარე არეალში განიშაზღვრება ხილულ რეჟიმში გამოჩენილი ჩანაწერების რაოდენობა. სურათ 5.5.3.2_ზე მოყვანილი მაგალითის შემთხვევაში ეს ჩანაწერები იქნება 10 ცალი. რა თქმა უნდა ეს არ არის სტატიკური 10 ჩანაწერის ჩვენების რეჟიმი. მეტი ჩანაწერის არსებობის შემთხვევაში გამოჩნდება დამატებითი ნავიგაციის ბმული და უზრუნველყოფს შემდეგ ჩანაწერებზე გადასვლას, ლოგიკით ყველა ჯერზე 10 პოსტი. (იხ.სურ. 5.5.3.2)
- **Syndication feeds show the most recent –** RSS _ში გამოტანილი უკანასკნელი ჩანაწერების რაოდენობის. მიმდნარე ეტაპზე მითითებულია 10 ჩანაწერი. **RSS** -(აბრევიატურასიტყვებისა Really Simple Syndication" ან "Rich Site Summary") მარტივი ენით ის წარმოადგენს XML ფორმატში შექმნილ სპეციალურ არხს, რომელიც უზრინველყოფს ამა თუ იმ საიტიდან ინფორმაციის მოწოდების უმარტივეს და უწყვეტ გზას... თუ გსურთ კონკრეტული საიტის მონიტორინგი, არ გინდათ გამოგრჩეთ რაიმე სიახლე და გამუდმებით არ გქონდეთ გახსნილი მიმდინარე ვებ გვერდი rss_ ის მეშვეობით შეგიძლიათ მიიღოთ ეს სიახლეები და მათზე გადასვლის შემდგომ გადამისამართდებით მთავარ საიტზე. (იხ.სურ. 5.5.3.2)
- **For each article in a feed, show** - მომხმარებლისთვის ჩანაწერის სრული, ან შემოკლებული ტექსტის მიწოდება RSS_ის მეშვეობით. **Full text**ველის გააქტიურების შემთხვევაში ჩანაწერი გადაიცემა სრული შემცველობით. **Summary**_ის მონიშვნის შემთხვევაში მხოლოდ მოკლე აღწერით(იხ.სურ. 5.5.3.2)
- **Search Engine Visibility** - საიტის ინდექსაციის გაუქმება საძიებოს სისტემებისათვის. (იხ.სურ. 5.5.3.2) Discourage search engines from indexing this site - ველის მონიშვნა აუკრძალავს საძიებო სისტემებს საიტის ინდექსაციას . ინდექსაცია - პროცესი როცა საძიებო სისტემა უზრუნველყოფს საიტის მონახულებას, ინფორმაციის განხილვას და ვებ გვერდის საკუთარ ბაზაში ჩამატებას.



სურ. 5.5.3.2 საიტისჩანაწერებისრაოდენობისადა RSS მათიგადაცემისპარამეტრებიმითითება

Discussion - ჩანართი უზრუნველყოფს ჩანაწერებისა და კომენტარების პარამეტრების რეგულირებას. როგორც დასაწყისში აღვნიშნეთ WordPress ძრავი ორინტირებულია ბლოგური ტიპის საიტებზე. აქედან გამომდინარე ჩანაწერებისა და კომენტარების მართვას საკმაოდ დიდი ყურადღება ეთმობა. ამ კონრეტულ ჩანართში ყურადღება სწორედ ამ მიმართულებების პარამეტრების რეგულაციებს ეთმობა.

- **Default article setting** - ჩანაწერების საწყისი მნიშვნელობები. პირველი 2 ველის პარამეტრების დასტური მისაღებია იმ შემთხვევაში თუ თქვენ ქმნით რეალურ ბლოგს. აწარმოებთ დიალოგს სხვა ბლოგერებთან არჩევთ საერთო თემებს. ეს ველები უზრუნველყოფს ბმულების გაცვლას. ამ ველების აქტივაციისას არსებობს რისკი - ბოტებმა დააგზავნონ სპამები, ამიტომ მათი გამოყენება უნდა მოხდეს მხოლოდ აუცილებლობის შემთხვევაში. მესამე ჩანართი უზრუნველყოფს კომენტარების სისტემის მოქმედებას. ველის გააქტიურება ნებას რთავს მომხმარებელს - გააკეთოს კომენტარი. (იხ სურ 5.5.4.1)



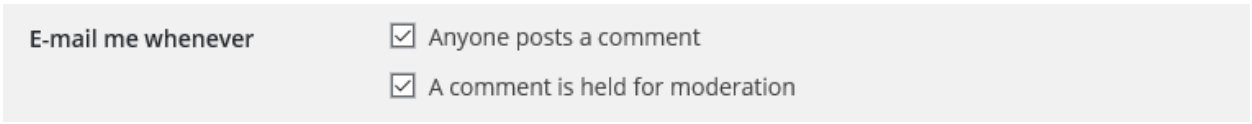
სურ. 5.5.4.1 ჩანაწერების გაცვლისა და კომენატერბის ნებართვის პარამეტრები.

- **Other comment settings –** კომენტარიების დამატებითი პარამეტრები. ამ არეალში გამოტანილია პუნქტები რომელთა მეშვეობითაც შესაძლებელია კომენტარების მართვა. თითოეული ველის მოკლე განმარტება შემდეგნაირია (იხ.სურ. 5.5.4,2) :
  - **Comment author must fill out name and e-mail** - კომენტარის ავტორმა უნდა მიუთითოეს სახელი და ელ. ფოსტა.
  - **Users must be registered and logged in to comment** - მომხმარებელი რეგისტრირებული უნდა იყოს წინააღმდეგ შემთხვევაში ეკრძალება კომენტარის დატოვება.
  - **Automatically close comments on articles older than ... days** - სტატიის განხილვა ავტომატუად დაიხურება ... დღის შემგომ;

- **Enable threaded (nested) comments** 5 **levels deep** - ნებადართული 5 დონიანი კომენტირება.
- **Break comments into pages with top level comments per page and the** last **page displayed by default. Comments should be displayed with the** older **comments at the top of each page** - ბოლო პუნქტი უზრუნველყოფს კომნტარების ხილვადობას, კონრეტული ჩანაწერის თუ რამდენი კომენტარი უნდა გამოჩნდეს. ასევე დასაშვებია ავირჩიოთ თუ რომელი კომენტარების ხილვადობა იქნება პრიორიტეტული - უკანასკნელის ან ძველის.

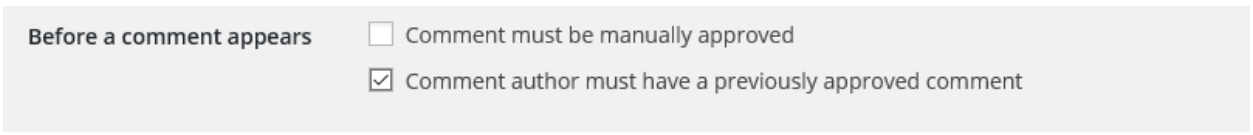
სურ. 5.5.4.2 კომენატერბის პარამეტრების მართვა

- **E-mail me whenever** – მიმდნინარე ჩანართის დანიშნულებაა წერილის გაგზავნა ადმინისტატორთან. წერილი მიუვა ადმინისტრატორს შემთხვევა 1) **Anyone posts a comment**- როდესაც ვინმე დატოვებს კომენტარს. 2) **A comment is held for moderation**– კომენტარი გამოჩნდეს მოდერაციის შემდეგ (მოდერაცია - ადმინისტრატორი ამოწმებს შინაარსს დამხოლოდ მისი დასტურის შემთხვევაში ხდება გამოქვეყნება) (იხ.სურ. 5.5.4,3).



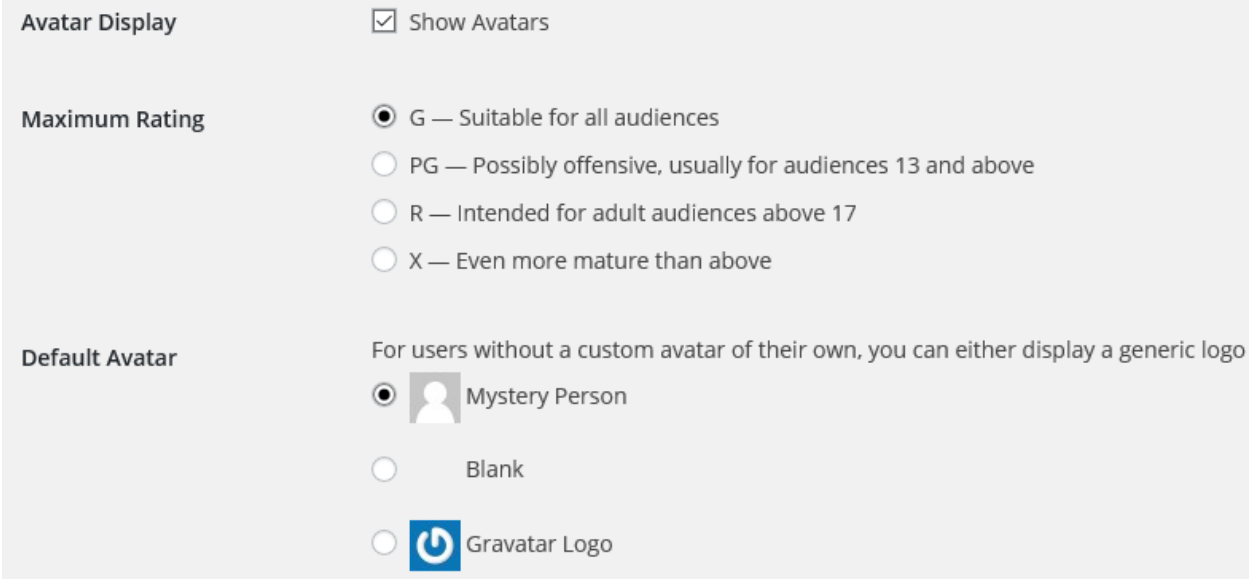
სურ. 5.5.4.3 ადმინისტრატორის ინფორმირება კონკრეტული ნაბიჯების შემდგომ

- **Before a comment appears** - კომენტარების გამოცენისათვის საჭიროა შე,დეგი პროცედურების გავლა: 1)**Comment must be manually approved** - კომენტარები გამოქვეყნებამდე გადის მოდერაციას; 2) **Comment author must have a previously approved comment** - ავტორი უნდა ჰქონდეს აქამდე რამოენიმე კონტროლირებული / დამოწმებული კომენტარები. (იხ.სურ. 5.5.4.4) რეალურად ამ ორი ველის გააქტიურება გამოწვეულია დაცვის გაუმჯობესების მიზნით, რათა არ მოხდეს სპამების გავრცელება. არის მეორე მხარეც მომხმარებელს სჩვევია თავისი კომენტარით ტკბობა ამიტომ სასურველია თ ამ დაბრკოლებას არ შევუქმნით. არსებობს პლაგინები რომლებიც უზრუნველყოფს კომენტარების ფილტრაცია - შემოწმებას.



სურ. 5.5.4.4 კომენტარების შემოწმება გამოქვეყნებამდე

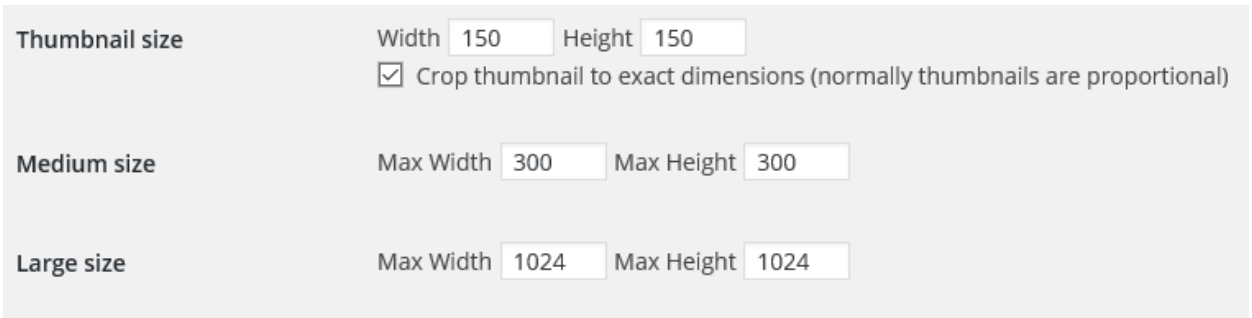
- **Comment Moderation , Comment Blacklist** - მიმდინარე ბლოკები გამოყოფილია კომენტარების უფრო დაწვრილებითი შემოწმებისთვის, შავ სიებში განთავსებისთვის. შესაძლებელია როგორც კონკრეტული მომხმარებლების IP_ის, ავატარის ან ელ. ფოსტის მიხედვით ბლოკირება ასევე სიტყვების ან სიტყვათა კომბინაციების მიხედვით.
- **Avatars** – არეალში შესაძლებელია მომხმარებლის გამოჩენა, რეიტინგის მიხედვით კლასიფიკაცია და ავატარის-სურათის შერჩება. (იხ.სურ.5.5.4.5)



სურ. 5.5.4.5 მომხმარებლის ავატარის ფორმირება

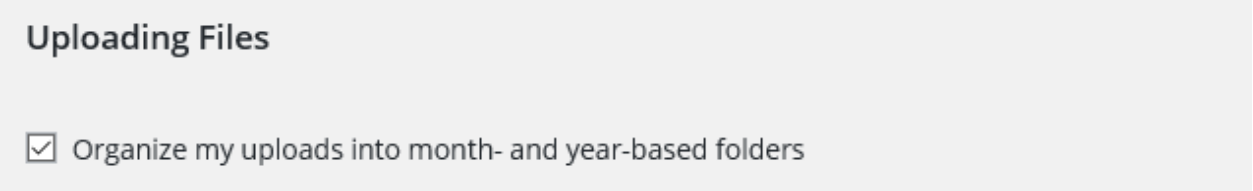
Media - ასატვირთი მულტიმედია ფაილების პარამეტრების კონფიგურირება შესაძლებელია მიმდინარე ბმული აქტივიზაციით. სტანდარტული პარამეტრების მიხედვით WordPress ახარისხებს (ჭრის) სურათს 3 ზომის მიხედვით : მინიატურა, საშუალო და დიდი ზომა . მიმდიანრე კატეგორიდან შესაძლებელია ამ ზომების პარამეტრების ცვლილება გამომდინარე საიტის დიზაინიდან

- **Thumbnail size** - მინიატურული სურათის ზომის ფიქსირება. ავტომატურად მითითებულია **Width** - სიგანე 150 / **Height** სიმაღლე 150; ამ კატეგორიაში შესაძლებელია Crop thumbnail to exact dimensions ველის მონიშვნის გაუქმება. ის უზრუნველყოფს პროპორციულ მოჭრას; (იხ.სურ. 5.5.5.1)
- **Medium size** - საშუაო ზომის სურათის პარამეტრები; ვებ გვერდზე გამოტანილი საშუალო ზომის სურათის პარამეტრების მაქსიმალური მნიშვნელობები მიმდინარე ველში მითითებული მაჩვენებლების იდენტური იქნება. მინიშნება - ამ შემთხვევაში არ არის აქსიომა რომ მოჭრილი სურათი აუცილებლად პროპორციული იქნება. ეს წესი მოქმედებს Large Size ზომის სურათებისთვისაც. (იხ.სურ. 5.5.5.1)
- **Large Size** - დიდი ზომის სურათი. ამ შემთხვევაში საწყის მნიშვნელობად 1024 პიქსელი არის მითითებული. (იხ.სურ. 5.5.5.1)



სურ. 5.5.5.1 ასატვირთი სურათის დაჭრილი ზომების პარამეტრების რეგულირება

- **Uploading Files** - ატვირთული ფაილის საქაღალდის დასახელება. უკვე მონიშნული ველი უზრუნველყოფს ატვირთული ფაილების სორტირებას თვისა და წლის მიხედვით დასათაურებულ საქაღალდეებში.(იხ.სურ. 5.5.5.2)



სურ. 5.5.5.2 ფაილების ჩალაგენა თვისა და წლის მიხედვით დასათაურებულ საქაღალდეებში

Permalinks - მიმდინარე ჩანართი უზრუნველყოფს გვერდების URL მისამართების ფორმირებას. არის რამოდენიმე ვარიანტი რომელსაც WordPress მზა სახით გვთავაზობს. შესაძლოა ამ მისამათების გენერირება მოხდეს სურვილისამებრ. სურ 5.5.6.1 მოცემულია მზა სახის URL შაბლონები.

მათგან პირველი **Default** - საწყისი მნიშვნელობით მინიჭებული მისამართია რომელიც სურათზე მოცემული ვიზუალისაა - **?p=123;**

Day and name - გააქტიურების შემთხვევაში მისამართის პორმირებისას გამოტანილი იქნება ჩანაწერის გამოქვეყნების თარიღი და სახელი ერთდროულად. პაქტიურად მიმზგავსებულია შემდეგი მნიშვნელობა **Month and name** განსხვავება მხოლოდ იმაშია რომ თარიღი გამოისახება დღის გარეშე, მხოლოდ თვე და წელი რა თქმა უნდა ასევე ჩანაწერის სახელთან ერთად.

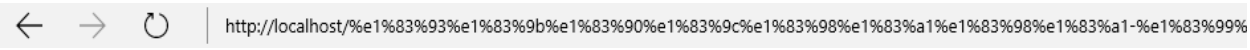
Numeric - მითითებული ველის აქტივაციისას URL მისამართი დაგენერირდება ჩანაწერის ID_ის მიხედვით. (ID წარმოადგენს იდენთიფიკატორს, უნიკალურ ციფრს რომელიც ენიჭება ყველა ჩანაწერს და ის არასოდეს მეორდება).

Post name - გადაეცემა სამისამართე პანელში ჩანაწერის სახელი.

უკანასკნელი ველი **Custom Structure** იძლევა საშუალებას მომხმარებელა თავად მოახდინოს მისთვის სასურველი მისამართის გენერირება. (იხ.სურ. 5.5.6.1). ვრცლად სამისამართე არეალის გენერირების შესახებ შეგიძლიათ გაეცნოთ მითითებულ მისამართზე - https://codex.wordpress.org/Using_Permalinks

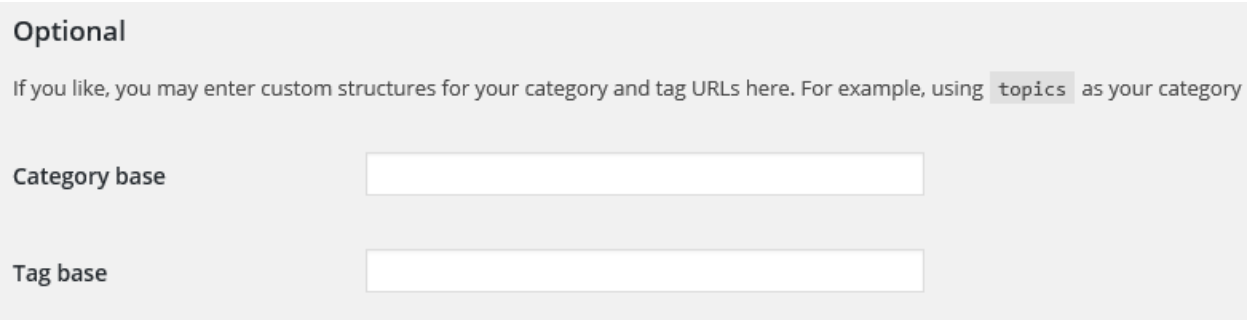
სურ. 5.5.6.1 მისამართების გენერეირება

გასათვალისწინებელია რომ ქართული პოსტების შემთხვევაში სათაურით გენერაციის დროს, URL მისამართი არც თუ ისე სახარბიელოდ ვიზუალიზდება (იხ.სურ. 5.5.6.2)



სურ. 5.5.6.2 არალათინური სათაურების გენერეირება სახელის მიხედვით

URL მისამართებში კატეგორიებისა და ტეგებისთვის შესაძლებელია დამატებითი განმასხვავებელი პრეფიქსების განსაზღვრა საკუთარი სურვილისამებრ. (იხ.სურ.5.5.6.3)



სურ. 5.5.6.3 კატეგორიების და ტეგებისთვის პრეფიქსების განსაზღვრა

6.2 დამატებითი პლაგინების (Plugins) დაყენება/გამოყენება

მიმდინარე პარაგრაფის თემატიკა

- პლაგინის არსი და დანიშნულება
- მართვის პანელთან მუშაობის ძირითადი საშუალებები
- პლაგინების გამოყენების მთოდები

პლაგინის არსი და დანიშნულება

პროგრამების საკმაოდ დიდი ნაწილი დღევანდელ დღეს იყენებს პლაგინებს. პლაგინი წარმოადგენს ძირითადი პროგრამული უზრუნველყოფის პროგრამულ დანამატს მისი ფუნცქიონალობის სრულყოფისათვის.მარტივად რომ ვთქვათ - პლაგინი ახდენს ინტეგრაციას მთავარ პორგრამასთან და ხდის მას მრავალფეროვანს, ცვლის მას უკეთესობისაკენ რაიმე კონკრეტული მიმართულებით.

პალგინის თავისებურებას წარმოადგენს ის რომ მათ დამოუკიდებლად მუშაობა არ შეუძლით, ის მხოლოდ დანამატია და იძლევიან შედეგს მხოლოდ მთავარ პროგრამასთან ინტეგრაციის შემდგომ. განსხვავებით ძირითადი პოგრამისაგან რომელიც დანამატის (პლაგინის) გარეშეც საკმაოდ კომფორტულად გრძნობს თავს და არ უჭირს მუშაობა.

ჩნდება კითხვა: და რა საჭიროა პალინი, როცა შეიძლება ყველა ფუნქციონალის ჩაშენება მოხდეს ძირეულ პროგრამაში მისი წარმოებაში ჩაშვებისთანავე?. ეს ხომ გაამარტივებს მომხმარებლისთვის ბევრ რამეს - აღარ გახდება საჭირო დანამატების მოძიება, ინტეგრაცია, პრამეტრების კონფიგურირება. რა თქმა უნდა ამ აზრს გააჩნია არსებობის საფუძველი, მაგრამ შევეცადოთ ჩავწვდეთ პლაგინის ძირითად არსს.

პლაფინს აქვს ორი ძირითადი ფუნქცია:

1. **პლაგინი ხელს უწყობს პროგრამის კომპაქტურობას და აჩქარებს მის მუშაობას;**
2. **პლაგინი აუმჯობესებს პროგრამას და ხელს უწყობს პროგრამის სრულყოფას ;**

პლაგინი ხელს უწყობს პროგრამის კომპაქტურობას და აჩქარებს მის მუშაობას

დავუშვათ საწყის ეტაპზე მოხდა სრულფასოვანი ფუნქციონალით დატვირთული პროგრამის შექმნა ის იქნება საკმაოდ მოცულობითი და მოითხოვს დიდ სივრცეს. ასევე ზედმეტი, მომხმარებლისათვის არამოხმარებადი, არასაჭირო ფუნციების მუშაობა წაიღებს დიდ რესურსს რაც თავისთავად უარყოფით გავლენას მოახდენს ამ პროგრამების სისწრაფეზე.შედეგად, მომხმარებელი მიიღებს დიდი რაოდენობით გამოუყენებელი მონაცემების და პროგრამების შენელებულ მუშაობას.

პლაგინი გამოყენება შესაძლებელს გახდის იმ კონკრეტული ფუნქციონალის დამატებისა რაც კონრეტულ მომხმარებელს გარკვეულ ეტაპებზე დასჭირდება. დანამატის პარამეტრების კონფიგურაცია თითიეული მომხარებლის ინდივიდუალური მოთხოვნის შესაფერისი იქნება, რაც გამოიწვევს პროგრამის მინიმიზებას მოთხოვნის შესაბამისად ეს კი უზრუნველყოფს ძირითადი პოგრამის ბევრად სწრაფ მუშაობას.

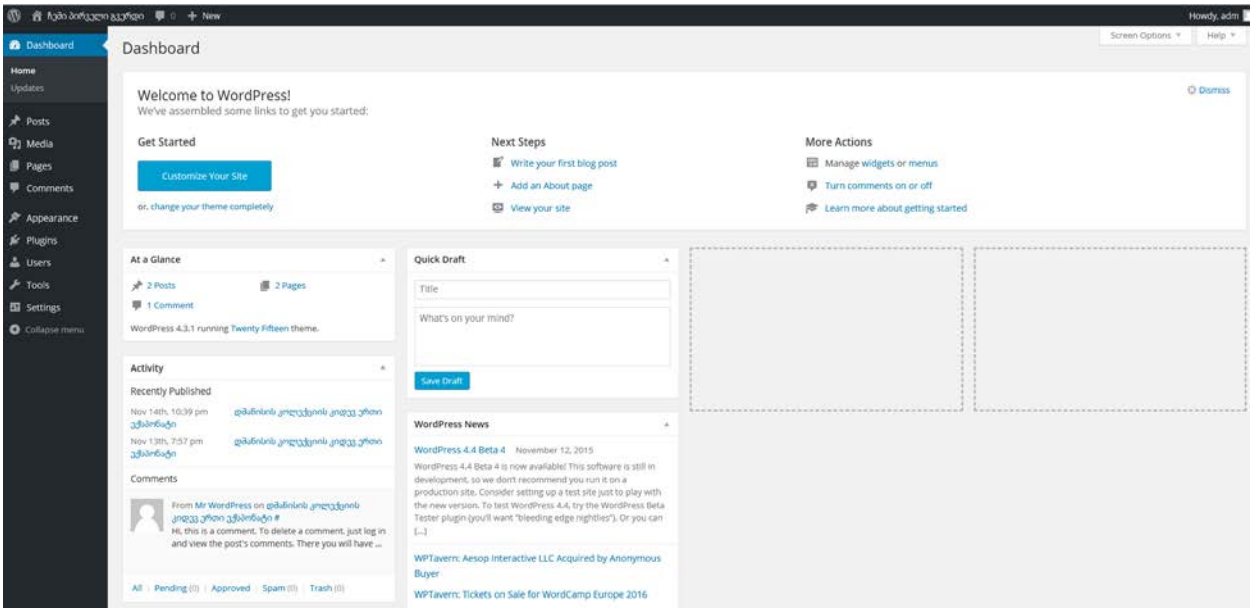
პლაგინი აუმჯობესებს პროგრამას და ხელს უწყობს პროგრამის სრულყოფას.

დანამატებზე მომუშავე პროგრამული უზრუნველყოფა, რომელიც არ ირღვევა შემდეგი პირობების დაშვებისას - დროის სხვადასხვა ეტაპზე კონკრეტული დანამატის გათიშვა ან ახლის გააქტიურება, ბევრად უფრო ორიენტირებულს ხდის მომხარებელზე. მომხმარებელი აღარ არის დამოკიდებული კონკრეტული მწარმოებლის პროდუქტის ახალ ვერსიაზე , მას დამოუკიდებლად ან პორგრამისტთა გუნდის დახმარებით შეუძლია დანამატების საშუალებით გაუმჯობესოს პროგრამის მუშაობა.

მართვის პანელთან მუშაობის ძირითადი საშუალებები

WordPress_ის სიმარტივე მის მრავალფუნქციურობასა და სამართავი პანელის სიმარტივეშია. მიმდინარე თავში განხილული იქნა მართვის პანელთან მუშაობის ძირითადი საშუალებები, რაც დაგარწმუნებთ თუ რამდენად იოლია მისი გამოყენება რეალურ მაგალითებზე დაყრდნობით. ფაქტობრივად თითოეული მოქმედება დაყვანილი ღილაკებზე მოქმედებით და დიდი გამოცდილება პროგრამული კუთხით საერთოდ არ არის საჭირო. ახლა კი უშუალოდ სამართავი პანელის შესახ.

ადმინისტრირების პანელში შესასვლელად აუცილებელია ავტორიზაციის გავლა (იხ.სურ. 4.5.7). რის შემდეგადაც მომხარებელი აღმოჩნდება სამართავი პანელის მთავარ გვერდზე (იხ.სურ. 6.1) გაითვალისწინეთ სურათები შეესაბამება მიმდინარე 4.3.1 ვერსიას და შეიძლება თქვენს შემთხვევაში იყოს მცირეოდენი ცვლილებები



სურ. 6.1 სამართავი პანელის მთავარი გვერდი

ადმინისტრირების პანელი რამოდენიმე ძირითადი ბლოკისაგან შედგება. ესენია:

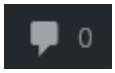
- სამათავი პანელის ზედა არეალი;
- სამართავი პანელის სანავიგაციო არეალი (მენიუ)
- ძირითადი ინფორმაციული არეალი;

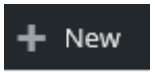
სამართავი პანელის ზედა არეალი

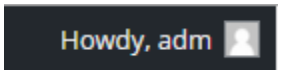
სამართავი პანელის ზედა არეალშიგამოტანილია სწრაფი წვდომის ღილაკები ღილაკები. მის მარცხენა მხარეს ჩალაგებული პიქტოგრამები:

-  - ბმულის მეშვეობით ვიღებთ ინფორმაციას WordPress_ის შესახებ. მის ქვე მენიუში არსებული პუნქტებით შეგვიძლია წვდომა CMS_ის მწარმორმოებლის ოფიციალური საიტთან, ძრავის დოკუმენტაციასთან, ფორუმსა და გამოხმაურებებთან. (ჩამონათვალი დალაგებულია მენიუს პუნქტების თანმიმდევრობით);

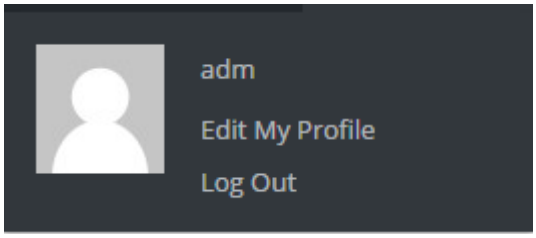
-  -ბმული უზრუნველყოფს საიტის მთავარ გვერდთან სწრაფ წვდომას / წვდომას პირდაპირ ადმინისტრირების პანელიდან;

-  - კომენტარებთან წვდომის სწრაფი ღილაკი.

-  - ახალი ჩანაწერის დამატების ღილაკი. მის ქვე მენიუში გამოტანილია ძირითადი პუნქტები სადაც შესაძლებელაი ახალი ჩანაწერის ფორმირება;

- 

- რეგისტრირებული მოხმარებლის ინფორმაციის მართვა; მასზე კურსორის მიტანისას შესაძლებელია როგროც რეგისტრირებული მომხმარებლის ინფორმაციის სამართავ არეალზე გადასხვლა (ამ ეტაპზე არ გავჩერდებით მომხარებლის არეალის გარჩევაზე ცოტა მოგვიანებით შევეხებით დაწვრილებით), ასევე მიმდინარე სესიის წყვეტა - სამართავი პანელის დატოვება/გამოსხვლა; (იხ.სურ. 6.1.1)

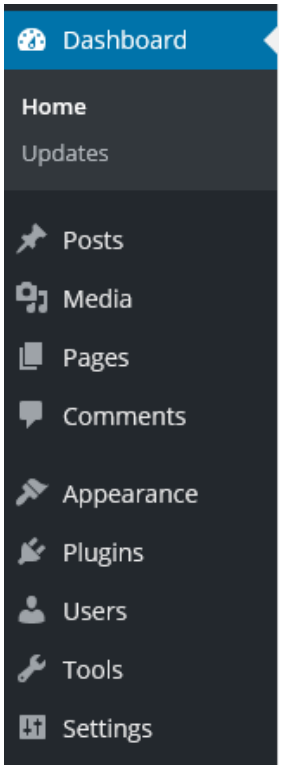


სურათი 6.1.1. რეგისტრირებული მომხმარებლის ნავიგაციის არეალი.

სამართავი პანელის სანავიგაციო არეალი (მენიუ)

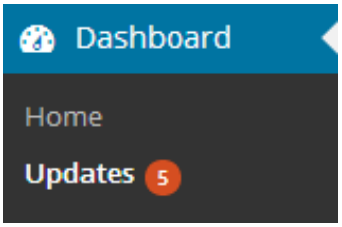
Dashboard

WordPress_ის ადმინისტრირების მარცხენა არეალში გამოტანილია სამართავი პანელის ნავიგაციის ბმულები. (იხ.სურ.6.2)



სურათი 6.2
ადმინ პანელის მენიუ

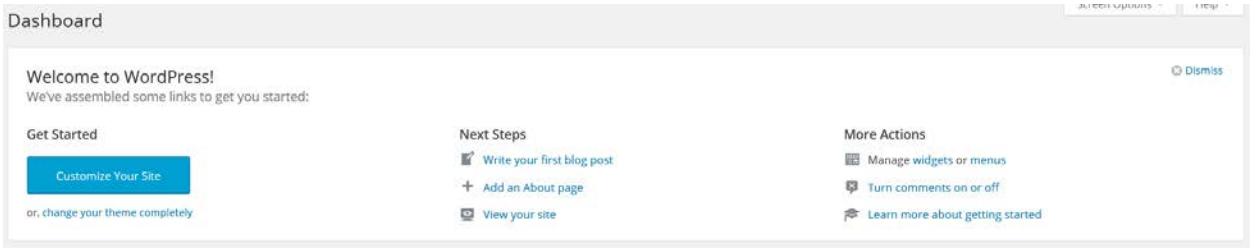
მათგან პირველი **Dashboard** წარმოადგენს კონსოლს რომელიც შეიცავს ინფორმაციას მთავარი გვერდისა და ძრავის ვერსიის ასევე კონკრეტული პლაგინების განახლებების შესახებ. (იხ.სურ. 6.2.1)



სურათი 6.2.1 ინფორმაცია განახლებების შესახებ

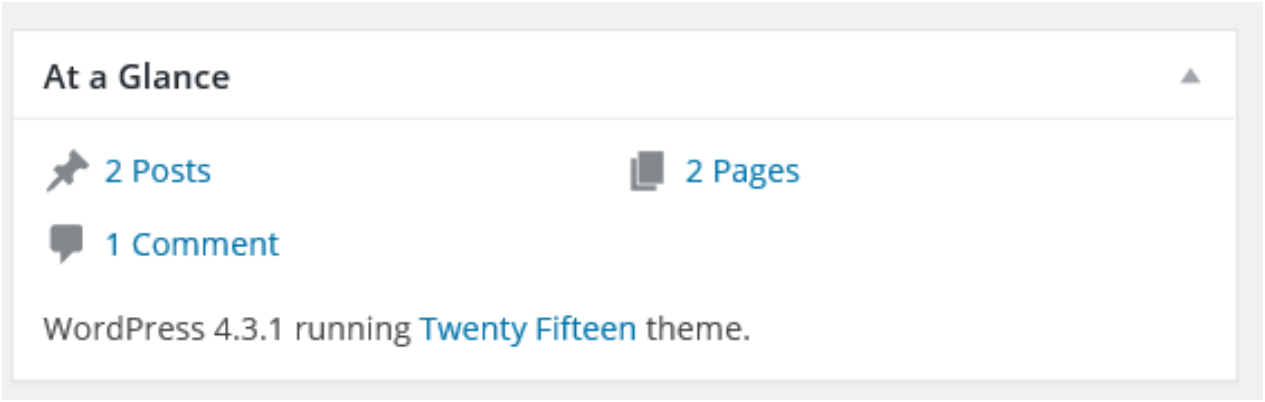
Dashboard -> home მენიუს აქტივაცია მოახდენს საიტის ადმინისტრირების პანელის მთავარ გვერდზე გადამისამართებას, რომელიც თავის მხრივ საინფორმაციო ანალიტიკურ გვერდს წარმოადგენს.

მიმდინარე ბლოკები აწვდიან სხვადასხვა ტიპის ინფორმაციას მოხმარებელს. მაგ.: **Welcome to WordPress!** - ბლოკი უზრუნველყოფს ბმულების მეშვეობით მცირეოდენი ცვლილებების დატანას საიტის ინფორმაციის ვიზუალურ გაფორმებაში (წარწერების, ფონის, ძირითადი ბლოკების ფერების ცვლილება), ჩანაწერების სწრაფ გამოქვეყნებაში, კომენტარების არეალის მართვაში. (იხ სურ 6.3.1)



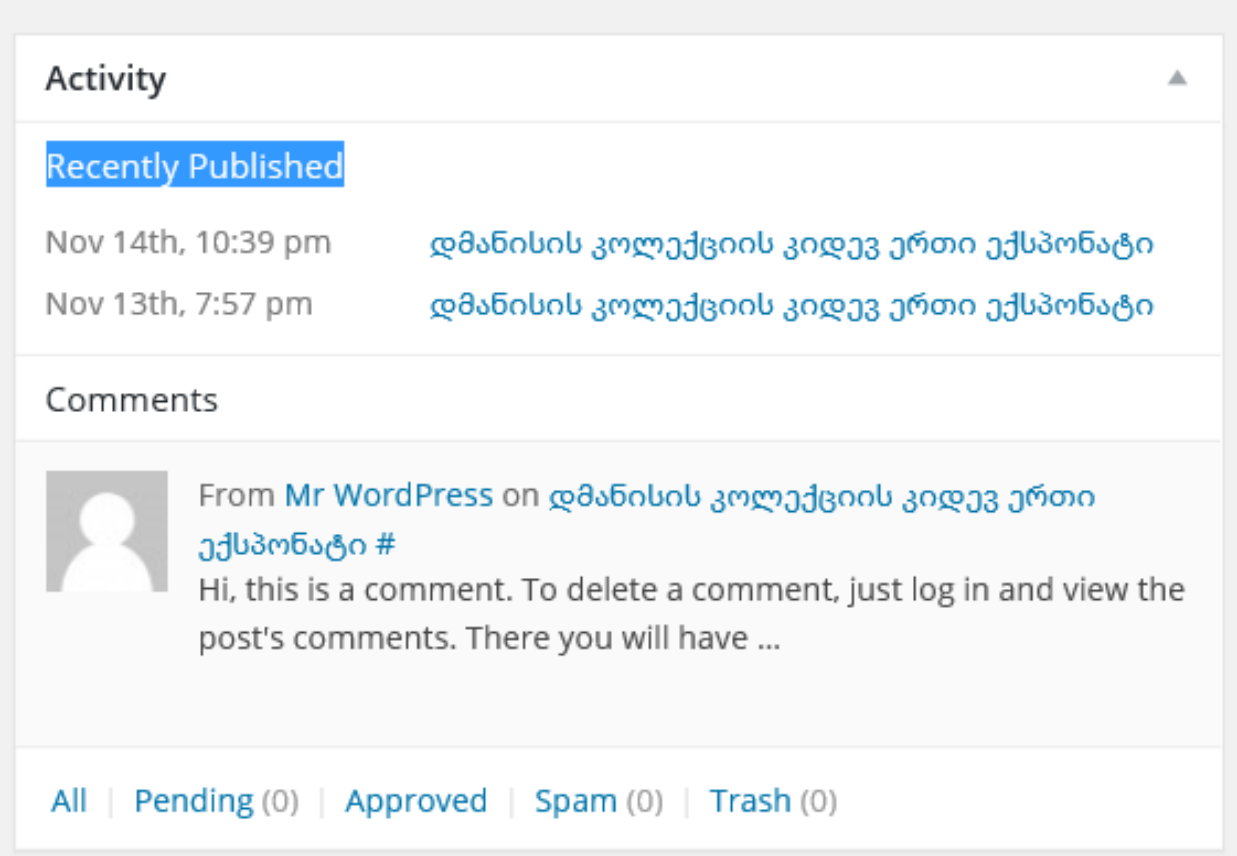
სურათი 6.3.1 მთავარი გვერდედიდან ცვლილებების შეტანა თემის იერსახეში

საიტზე არსებული ინფორმაციის სტატისტიკურ მონაცემებთან გაცნობა გვერდ **At a Glance** ჩანართშია შესაძლებელი. აქ გამოტანილია ჩანაწერების - **Posts,** გვერდების **-Pages,** კომენტარების **- Comment** რაოდენობა .(იხ სურ. 6.3.2)



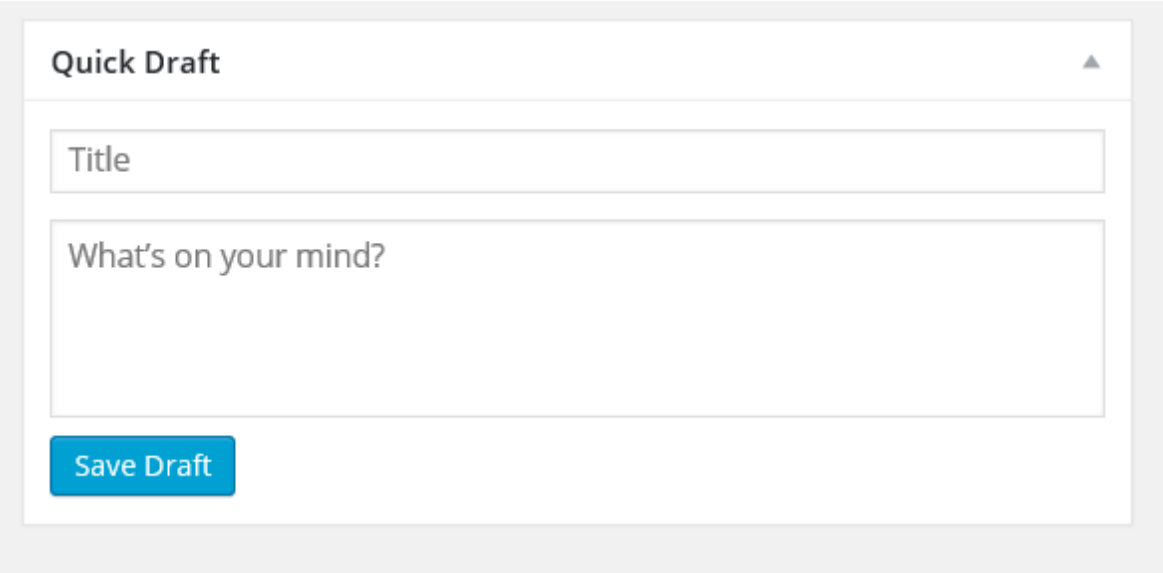
სურათი 6.3.2 საიტის ინფორმაციის რაოდენობრივი მაჩვენებელი

Activity – ბლოკი ახდენს ბოლო ქმედებების ვიზუალიზებას. ამ ბლოკში გამოტანილია უკანასკნელი განახლებები. გამოყვეყნებული სტაიტების სათაურები, ასევებ ბოლო კომენტარები კონკრეტული გვერდებისად ათემების შესაბამისად .(იხ სურ. 6.3.3)



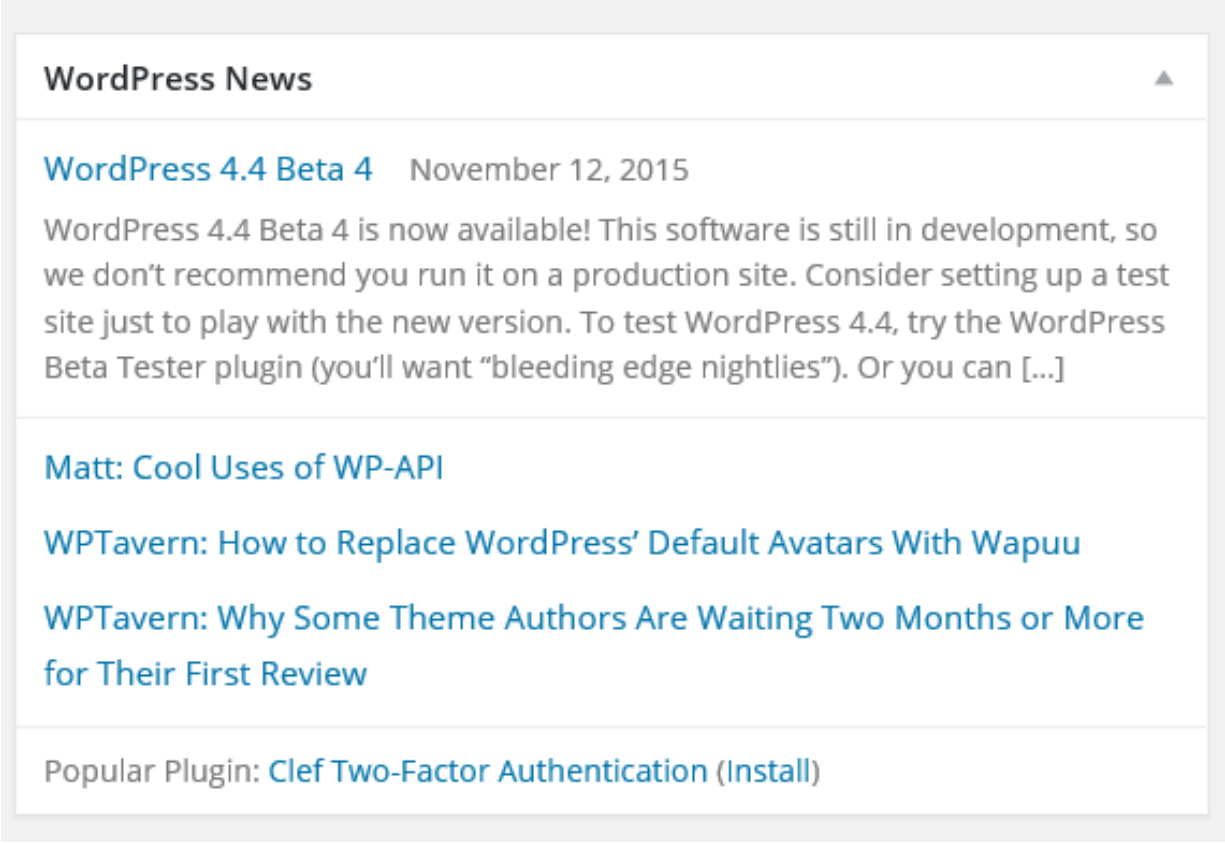
სურათი 6.3.3 უკანასკნელი აქტივობის წარმოჩენა

Quick Draft - არეალიდან უმარტივესად არის შესაძლებელი ჩანაწერის სწრაფი დამატება სათაურითა და ვრცელი ტექსტით. ის გადაინაცვლებს გამოუქვეყნებელი **Post**_ის ბლოკში გამოუქვეყნებელი სტატუსით. სურვილის შემთხვევაში ჩანაწერის აქტივაცია პრობლემას აღარ წარმოადგენს. (იხ სურ. 6.3.4)



სურათი 6.3.4 ჩანაწერის სწრაფი დამატება

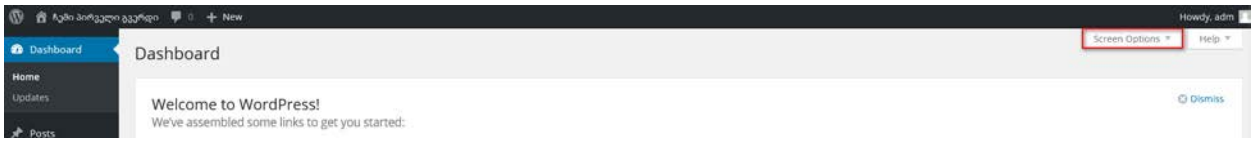
პიველივე გვერდზე შესაძლებელია WordPress_ის ირგვლივ მიმდინარე პროცესების კონტროლი ოფიციალური სიახლეების გაცნობის საშუალებით. **WordPress News** ბლოკი მუდმივ კავშირზე ამოყოფებს მოხმარებელს სიახლეების წვდომასთან. (იხ სურ. 6.3.5)



სურათი 6.3.5 WordPress_ის სიახლეების ბლოკი

მიმდინარე ბლოკების პოზიციონირება და სურვილისმაებრ დალაგება ძალიან მარტივად მაუსის კურსორის ჩაჭიდებით და გადატანითაა შესაზლებელი.

რა თქმა უნდა ამ ბლოკების ყოველდღიურ რეჟიმში მონიტორინგი შესაძლოა ადმინისთვის არ იყოს საინტერესო. მათი ხოლვადობის კონტროლი შესაძლებელია **Screen Option** ჩანართის საშუალებით. (ეს ჩანართი განლაბებულია ადმინისტრირების პანელის ზედა მარჯვენა არეალში მომხმარებლის ინფოს ქვევით იხ.სურ .6.3.6.) მიმდინარე ბმულზე ზემოქმედება გამოიტანს დამხმარე ფანჯარას სადაც გამოსახულია იმ ბლოკების ჩამონათვალი რომელიც მიმდინარე ეტაპზეა აქტივირებული (ხილულ რეჟიმში). მონიშვნის აქტივაცია / დეაქტივაციის საშუალებით შესაძლებელია ბლოკებისთვის ხილვადობა / გაქრობის ეფექტის რეგულირება. (იხ სურ 6.3.7)

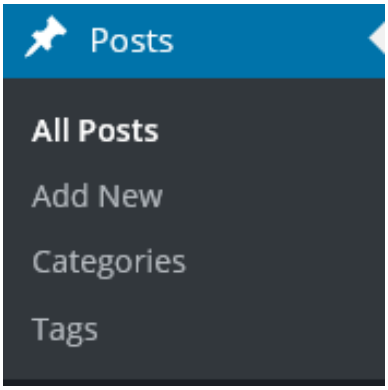


სურათი 6.3.6 ბლოკების ხილვადობის რეგულირება

სურათი 6.3.6 ბლოკების ხილვადობის აქტივაცია / დეაქტივაცია

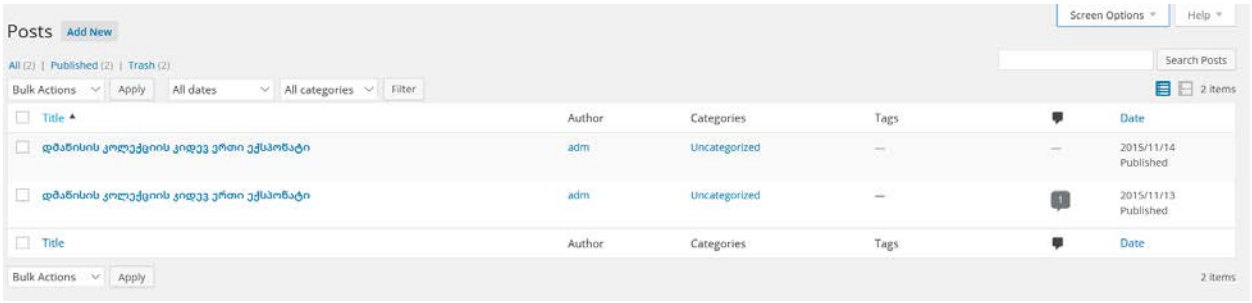
Posts

Posts ერთ-ერთი ყველაზე აქტუალური და ფართოდ გამოყენებადი ბლოკია WordPress_ის მენიუთაგან. პოსტები თავისი არსით წარმოადგენს ჩანაწერებს, სტატიას, სიახლეს. მიმდინარე პუნქტი აერთიანებს შემდეგ ბმულებს **All Posts, Add New, Categories, Tags.** (იხ. სურ. 6.4.1)



სურ 6.4.1 ჩანაწერების ბლოკი

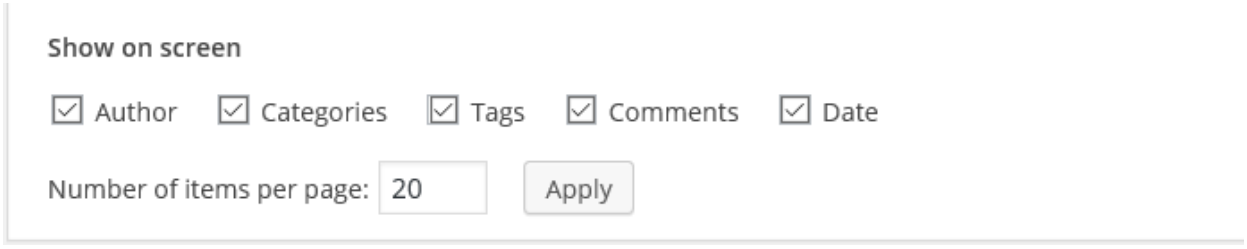
All Posts ბმულის გააქტიურება ავტომატურად Posts ჩანართზე ზემოქმედებისას ხდება. მასზე გადასვლისას საიტის ინფორმაციულ ნაწილში (ძირითად ბლოკში) გამოისახება სურათ 6.4.2 ნაჩვენები მოცემულობა. (იხ. სურ. 6.5.2) რომელიც თავისთავად მრავალი წვრილმანი ბლოკისაგან შედგება. შევეცადოთ თანმიმდევრულად გავყვეთ და გავარჩიოთ თითოეული ბმული თუ პიქტოგრამა რამეთუ საკმაოდ მნიშვნელოვანია თითოეული პუნქტის სრულყოფილად ათვისება.



სურათი 6.4.2 All Posts არეალი

სურათ 6.4.2_ზე ზედა მარცხენა კიდეში გამოტანილია ახალი ჩანაწერის ღილაკი >**Add New.** ამ ღილაკს სულ მალე დავუბრუნდებით.

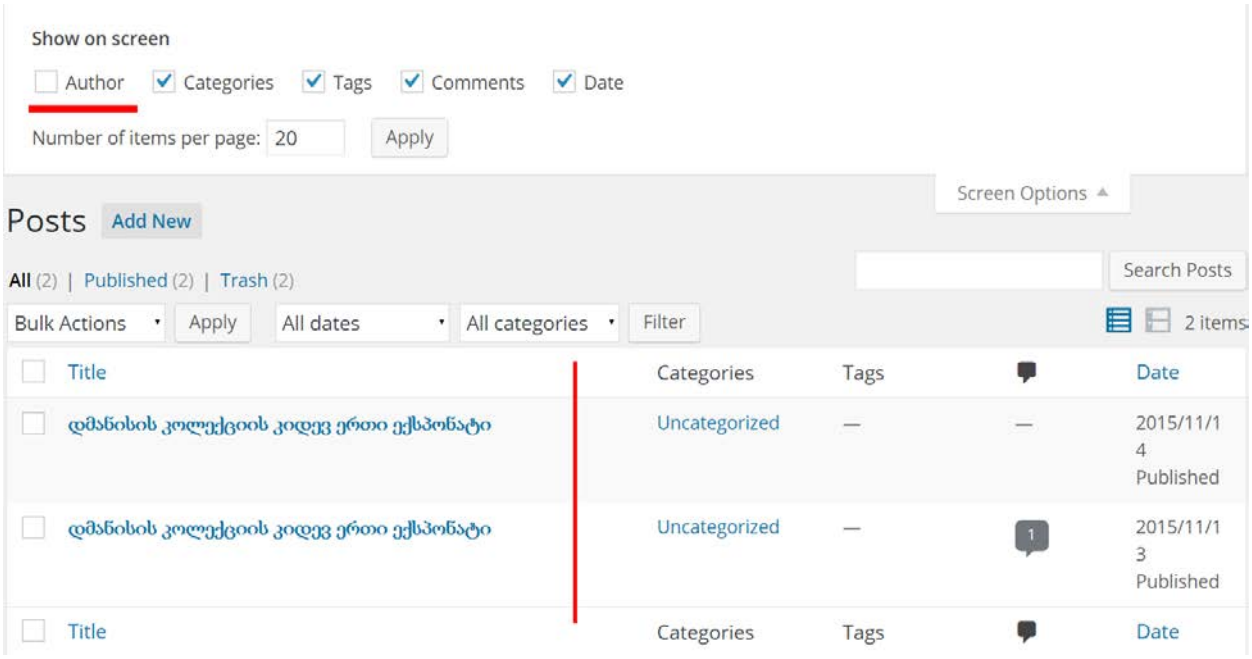
სურათ 6.4.2_ზე ზედა მარჯვენა კიდეში მითითებულია Screen Option ღილაკი რომელიც უზრუნველყოფს სხვადასხვა პარამეტრების რეგულირებას. მისი აქტივაციის შედეგად გაიშლება დამატები ველი პარამეტრებით. (იხ.სურ. 6.4.3) მიმდინარე ჩანართიდან ძირითად არეალში გამოტანილი ინფორმაციის პარამეტრების მართვა. (იხ.სურ. 6.4.4)



სურათი 6.4.3 პოსტების გამოტანის პარამეტრები სამართავ პანელში

სურათი 6.4.4 პოსტების გამოტანის ძირითადი არეალი

სურათ 6.4.3 ჩანართების აქტივაცია / დეაკტივაცია გამოიწვევს სურათ 6.4.3 ზე ნაჩვენები ბლოკების გამოჩენა / დამალვას. მაგ.: **Author** ღილაკის პასიურ რეჟიმში გადაყვანა გამოიწვევს ძირითადი ბლოკის ცვლილებას - კერძოდ დაიმალება ჩანაწერის ავტორი (მიმდინარე პროცესი გამოყოფილია წითელი ხაზებით) (იხ.სურ. 6.4.5)



სურათი 6.4.5 ჩანაწერის ავტორის დამალვა

ავტორის გარდა ძირითად ბლოკში შესაძლებელია ჩანაწერის კატეგორის, ტეგების, კომენტარებისა და მისი გამოქვეყნების თარიღის კონტროლი. შესაბამისად **Screen Option** პარამეტრები უზრუნველყოფს მათ ხილვადობის რეგულირებას.

მიმდინარე არეალის პირველი ბლოკი **Title** ახდენს თავად სიახლის სათაურის ვიზუალიზაციას. მისიდამალვა **Screen Option** ჩანართის მეშვეობით შეუძლებელია. ამ კონკრეტულ ფაქტთან დაკავშირებით კითხვაც არ უნდა გაჩნდეს რადგანაც ბლოკი რომელიც გამოძახებულია ჩანაწერების ინფორმაციის გაცნობისათვის მინიმუმ ამ ჩანაწერის სათაურს მაინც უნდა შეიცავდეს.

მიმდინარე გვერდზე ჩანაწერების რაოდენობის გამოტანის რეგულირება **Screen Option** უკანასკნელი ფუნქციის **„Number of items per page“** მეშვეობით ხდება, საწყის მნიშვნელობად 20 ჩანაწერია მითითებული. მისი ცვლილების შემთხვევაში აქტივაცია უნდა მოხდეს **Apply** ღილაკის მეშვეობით. (იხ. სურ. 6.4.3).

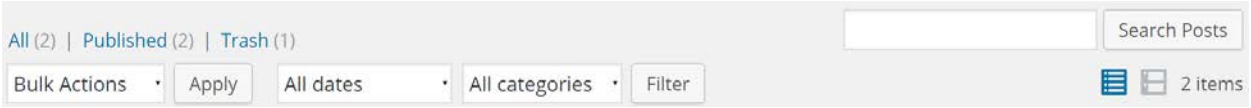
ჩანაწერების გამოტანის დამატებითი ფილტრაციისათვის გამოიყენება სურათ 6.4.6 ბმულებისა დაღილაკების ერთობლიობა.

- **All** - ჩანაწერების რაოდენობა;
- **Published** -გამოქვეყნებული ჩანაწერების რაოდენობა;
- **Trash**- სანაგვე ყუთში არსებული ჩანაწერების რაოდენობა. სრულ განადგურებამდე ჩანაწერი ინაცვლებს Trash განყოფილებაში. ეს ერთგვარი თავის დაზღვაა რათაა უნებურად არ წაგეშალოდ ღირებული ჩანაწერი. ამ ბლოკიდან შესაძლებელია ჩანაწერის აღდგენა. თუ ძირფესვიან განადგურებას დაგაწყვეტთ მოგიწევთ ამ ბლოკიდან მათი წაში.

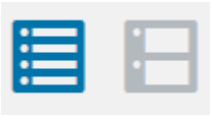
ასარჩევი არეალი **Bulk Action** გვაძლევს საძუალებას ერთ ან რამოდენიმე ჩანაწერზე ვიმოქმედოთ **Edit** - ჩანაწერის ცვლილება , **Move to Trash** - ჩანაწერის გადატანა სანაგვე ყუთში.

All dates და **All categories** მეშვეობით შესაძლებელია ჩანაწერების სორტირება კონკრეტული თარიღის ან კატეგორიის მიხედვით.

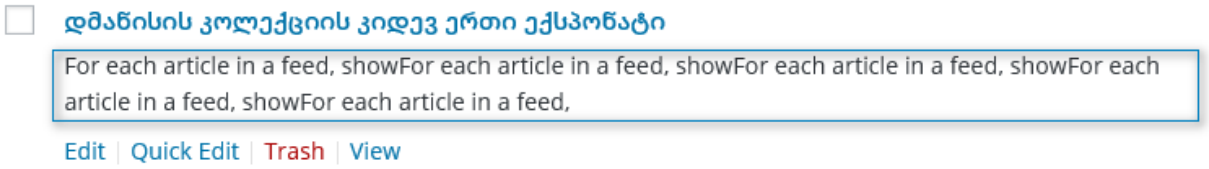
ასევე შესაძლებელია საკვანძო სიტყვის მეშვეობით კონკრეტული ჩანაწერის მოძიება, რაშიც მარჯვენა ზედა კიდეში გამოტანილი საძიებო ფორმა დაგეხმარებათ.



სურათი 6.4.6 ჩანაწერების ფილტრაცია



მიმდინარე პიქტოგრამების საშუალებით **Title** ჩანართში შესაძლებელი ჩანაწერის მხოლოდ სათაურის ხილვადობა(I პიქტოგრამა) , ჩანაწერის სათაურისა და მოკლე აღწერის გამოჩენა (II _პიქტოგრამა) (იხ .სურ. 6.4.7)



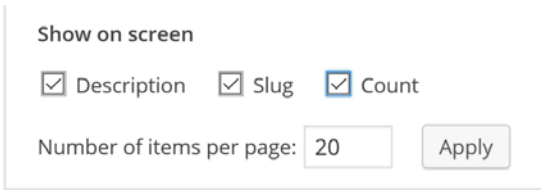
სურათი 6.4.7 ჩანაწერის მოკლე აღწერის გამოჩენა

ჩანაწერზე მაუსის კურსორის მიტანისას გამოდის დამატებითი სანავიგაციო ბმულები. რომლის მეშვეობითაც შესაძლებელი **Edit** - ჩანაწერის ცვლილება, **Quick Edit** - სწრაფი ცვლილება (რომელიც თავის თავში მოიაზრებს სხვადასვხა პარამეტრების ცვლილებას, მისი საშუალებით ჩანაწერის სრული ინფორმაციის ცვლილება ვერ ხერხდება, მხოლოდ : კატეგორია, ტეგები, სტატუსი, თარიღი, Slug. ამ უკანასკნელს დაწვირებით შევეხებით) , **Trash** - ჩანაწერის გადატანა სანაგვე ყუთში, **View** - ჩანაწერის გამოქვეყნებული სახე.

ახალი ჩანაწერის დამატებამდე გავარჩიოთ კატეგორიების დამატება. კატეგორიების პარამეტრების ცვლილება შესაძლებელია ჩანართ **Categories**_ზე გადასვლით (იხ. სურ. 6.4.1).

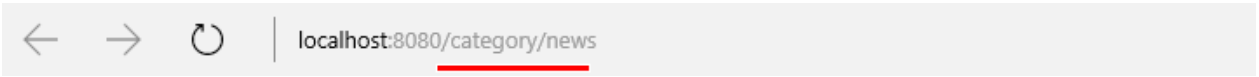
კატეგორია სამაოდ მნიშვნელოვანი დანამატია და ის გულისხმობს კატეგორიის დამატებას ჩანაწერების მომავალი სორტირებისათვის. მაგ. საიტი წარმოადგენს პიროვნების ბლოგს და ის წერს სტატიებს სხვადასვა მიმართულებით - ეხება პოლიტიკას, სპორტს, კულტურას და ა.შ. თითოეული ჩანაწერი შესაბამის კატეგორიებში დახარისხდება.

WordPress_ის სტრუქტურიდან გამომდინარე უკვე შეგექმნებპდათ ზოგადი წარმოდგენა თითოეული გვერდის ინტერფეისზე. წინამდებარე პუნტების გარჩევისას ვახსენეთ **Screen Option** ჩანართი. რა თქმა უნდა ამ კონკრეტულ ბლოკსაც გააჩნია ეს ველი (იხ. სურ. 6.5.1), რომლის პარამეტრები მიბმულია კატეგორიის გამოტანის ბლოკთან (იხ. სურ. 6.5.2). კატეგორიის პარამეტრები მუშაობს პოსტების პარამეტრების იდენტურად თავის ველებთან მიმართებაში.



სურათი 6.5.1 კატეგორიების პარამეტრები სურათი 6.5.2 კატეგორიების ჩამონათვლის არეალი

Description - მოკლე აღწერის ხილვადობის რეჟიმი, **Count** - მიმდინარე კატეგორიაში შემავალი ჩანაწერების რაოდენობა, **Slug** - წარმოადგენს კატეგორიის იმ დასახელებას რომელიც გვინდა გამოჩნდეს სამისამართე ველში. მაგ.: შევქმენით კატეგორია სიახლეები რომლის **Slug** - იქნება **news,** რაც სამისამართე პანელში შემდეგნაირად გამოისახება (იხ სურ 6.5.3). ძირითადად Slug შედგება ლათინური ასოებისაგან ან ციფრებისაგან, შესაძლებელია სიტყვათშორის გამოყოფად ავიღოდ დეფისები.

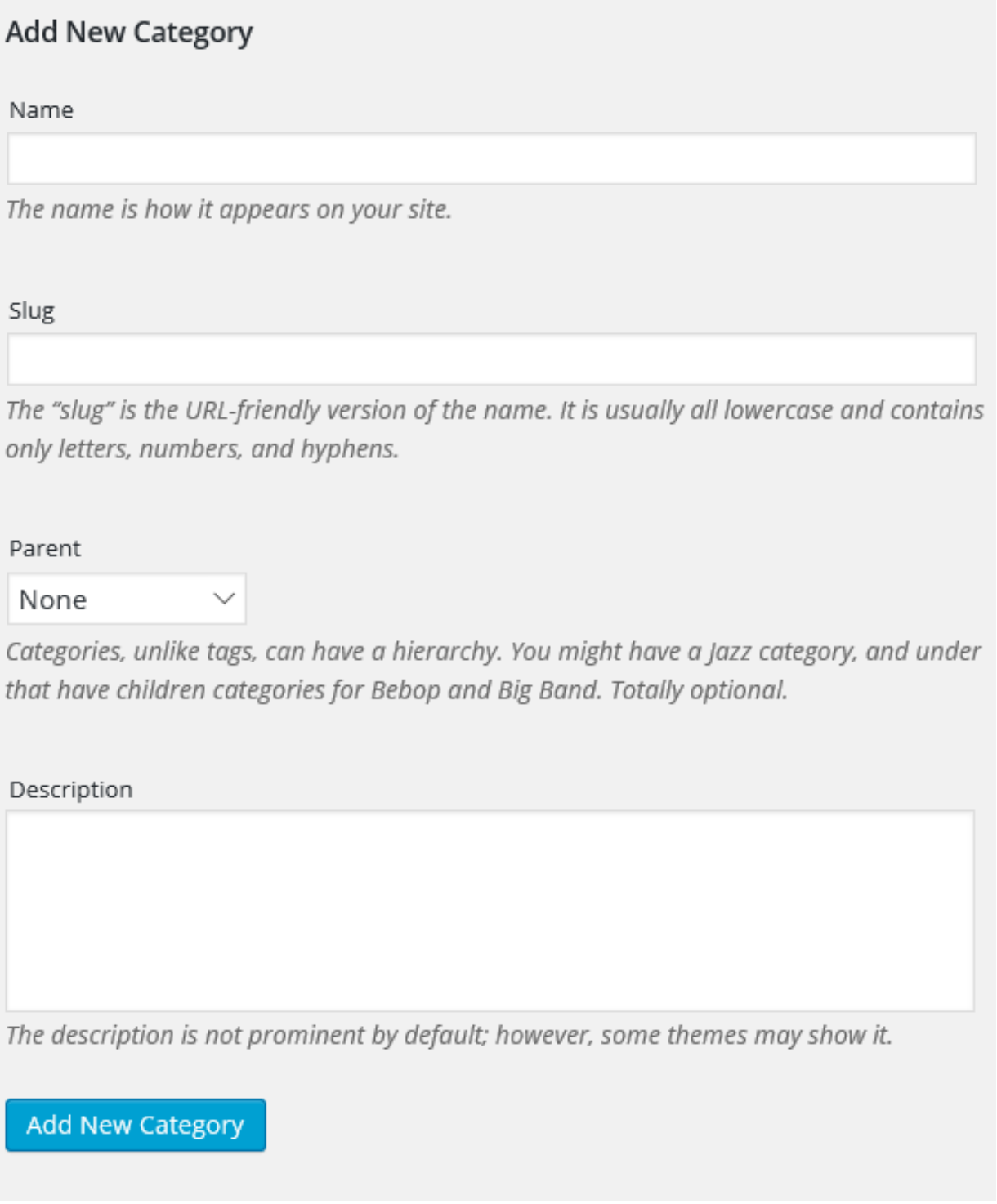


სურათი 6.5.3 კატეგორიის Slug

ახალი კატეგორიის დასამატებლად საჭიროა **Add New Category** - ბლოკის ველების შევსება (იხ.სურ. 6.5.4), სადაც :

- **Name** - წარმოადგენს კატეგორიის დასახელებას;
- **Slug** – Urlმისამართში კატეგორიის სახელს (ვისაუბრეთ ზედა აბზაცში სურ. 6.5.3);
- **Parent** - მიმდინარე ასარჩევ არეალში გამოდის ყველა არსებული კატეგორია, მისი მეშვეობით შესაძლებელია ახალ კატეგორიას რომელსაც დამატებაც ხორციელდება მიმდინარე ეტაპზე გავუწეროთ მშობელი კატეგორია. მაგ.: სპორტ კატეგორია შესაძლოა იყოს მშობელი ფეხბურთის, კალათბურთის ,რაგბის და ა.შ კატეგორიებისა. ამ შემთხვევაში ქვეკატეგორიებში (მაგ.: რაგბი) ჩანაწერის დამატება გამოიწვევს იმას რომ ჩანაწერი ავტომატურად გახდება მშობელი კატეგორიის კუთვნილებად (მოცემულ შემთხვევაში სპორტ კატეგორიის შემადგენელ ნაწილად);
- **Description** - საინფორმაციო ტიპის ველი, სადაც შესაძლებელია მიეთითოსკატეგორიის მოკლე აღწერა.

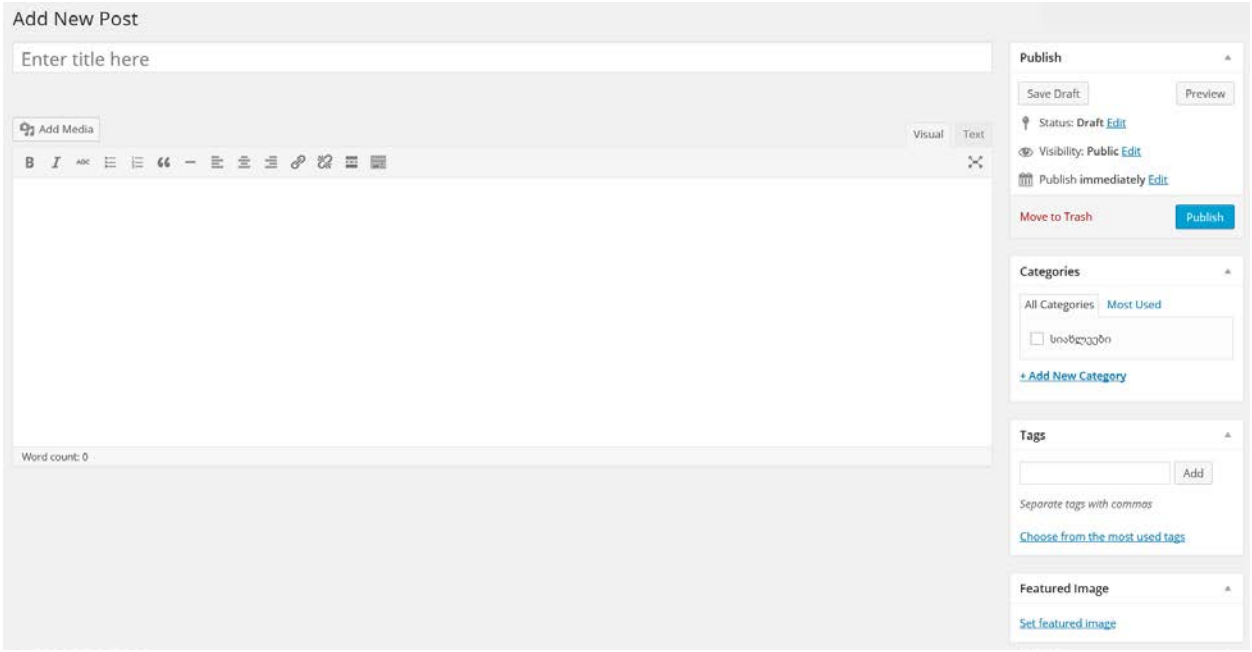
Add New Category - ღილაკის მეშვეობით ხდება შესაბამისი კატეგორიის დამატება.



სურათი 6.5.4 ახალი კატეგორიის დამატება

მენიუს შემდეგ ველ **Tags** _ების დამატება და პარამეტერბის ცვლილება სტანდარტულად ხორციელდება კატეგორიების და პოსტების მართვის მსგავსად. Tab_ები თავის მხრივ წარმოადგენს **Meta** ინფორმაციას, საკვანძო სიტყვების ერთობლიობას.

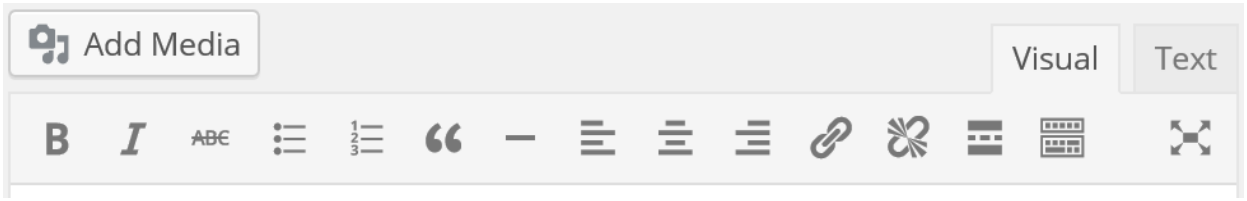
Posts - ველის ერთ ერთ მთავარ ბმულს **Add New** წარმოადგენს (იხ.სურ.6.4.1) რაც ახალი ჩანაწერის დამატებას გულისხმობს. მისი გააქტირუების შედეგად გამოისახება სურათ 6.5.5 ზე ნაჩვენები შედეგი.



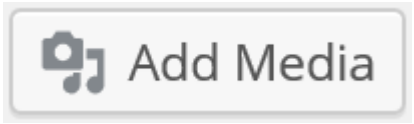
სურათი 6.5.5 ახალი ჩანაწერის დამატება

ძირითადი არეალი ეთმობა ჩანაწერის სათაურის და შემცველობის ველებს. **Enter title here** - ველი გამოყოფილია ჩანაწერის სათაურის შესაყვანად.

ვრცელი ბლოკი განკუთვნილია ჩანაწერის ვრცელი ტექსტის ჩასაწერად. სურათ 6.5.6 გამოტანილია იმ პიქტოგრამების და ღილაკების ერთობლიობა, რომელიც უზრუნველყოფს ჩანაწერის ვიზუალურ გაფორმებას და მასში სხვადასხვა ფაილების ინტეგრირებისათვის.



სურათი 6.5.6 ჩანაწერის მართვა



სურ 6.5.6.1

სურ 6.5.6.1 ღილაკის მეშვეობით შესახლებელია ფაილის ჩაგდება ჩანაწერის შემცველობაში.

ატვირთული ფაილი მაგ. სურათი განთავდება იმ პოზიციაზე სადაც მდებარეობდა კურსორი. სურათზე მაუსის კურსორის მიტანისას გადის მისი განლაგების ამსახველი ღილაკები რომელთა მოქმედებაში მოყვანაც სურათს გაუკეტებს შესაბამის ტექსტისმიერ სწორებას. (იხ.სურ. 6.5.7)

სურათი 6.5.7 სურათის მდებარების ცვლილება

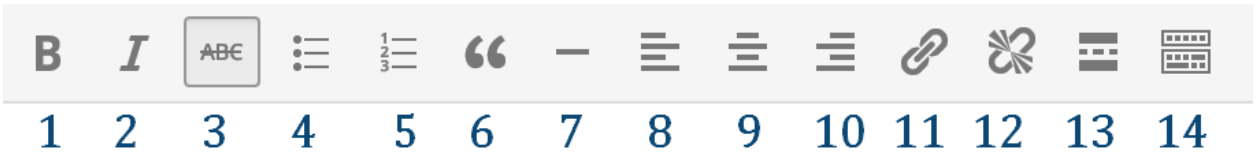
საქართველოს განათლებისა და მეცნიერების მინისტრი თამარ სანიკიძე უნგრეთში სამუშაო ვიზიტის ფარგლებში, უნგრეთის განათლების მიმართულების მინისტრს პალკოვიჩ ლასზლოს შეხვდა, შეხვედრას ასევე ესწრებოდნენ, განათლების საერთაშორისო ცენტრის დირექტორი წინო ჭელიძე და საქართველოს ელჩი უნგრეთში ზაზა კანდელაკი.

მოყვანილი პიქტოგრამები

საქართველოს განათლებისა და მეცნიერების მინისტრი თამარ სანიკიძე უნგრეთში სამუშაო ვიზიტის ფარგლებში, უნგრეთის განათლების მიმართულების მინისტრს პალკოვიჩ ლასზლოს შეხვდა. შეხვედრას ასევე ესწრებოდნენ, განათლების საერთაშორისო ცენტრის დირექტორი წინო ჭელიძე და საქართველოს ელჩი უნგრეთში ზაზა კანდელაკი.ორი ქვეყნის მინისტრს შორის საუბარი უმაღლეს განათლებას, განათლების დაფინანსებას და სტუდენტურ სესხებს შეეხო. უნგრეთს ყველა ამ მიმართულებით საკმაოდ დიდი გამოცდილება აქვს, საქართველო კი, მზად

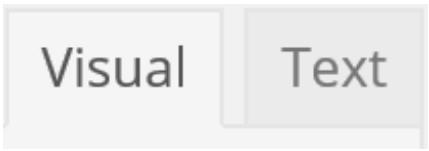
სურათ 6.5.6.2

რიგითობის მიხედვით შემდეგნაირად იშიფრება:



სურათი 6.5.6.2 ჩანაწერის გაფორმება

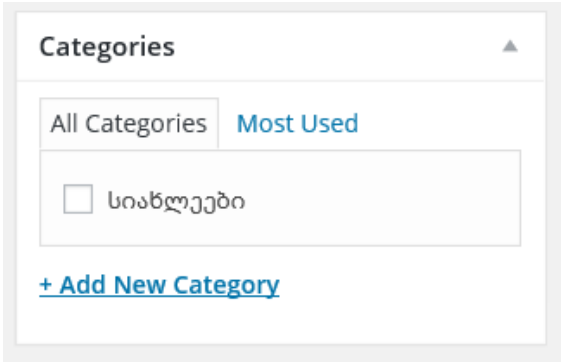
1. ტექსტის გამუქება;
2. ტექსტის დახრა;
3. გადახაზული ტექსტი;
4. ბულეტებით გაფორმებული სია;
5. დანომრილი სია;
6. ციტატა
7. ჰორიზონტალური ხაზი;
8. ტექსტის სწორება მარცხენა კიდესთან
9. ტექსტის ცენტრისმიერი სწორება
10. ტექსტის სწორება მარჯვენა კიდესთან
11. ბმულის დადება;
12. ბმულის მოხსნა;
13. ტექსტის წყვეტა - ტექსტის წყვეტა საკმაოდ ხრირად გამოყენებადი ფუნქციაა. მისი მეშვეობით შეგვიძლია გამოვყოთ ჩანაწერის ის არეალი, რომელიც გამოვა საიტზე სტატიის მოკლე აღწერის სახით.
14. უკანასკნელი პიქტოგრამის საშუალებით შესაძლებელია დამატებითი ფუნქციონალის გამოჩენა;



სურათ 6.5.6.3 მოყვანილი ჩანარტებით შესაძლებელია არეალის მნიშვნელობის განსაზღვრა - სტანდატული ტექსტისტვის ვირჩევთ **Visual**, html კოდს ვაგდებთ **Text** ბმულით აქტივირებულ არეალში.

სურ. 6.5.6.3

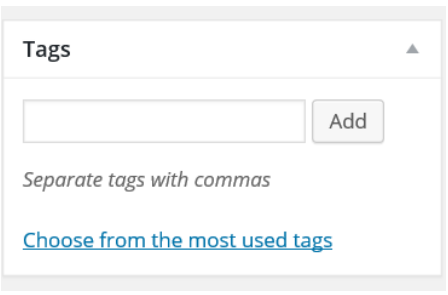
ჩანაწერის შევსების შემდგომ შესაძლებელია მისი კატეგორიის განსაზვრა. კატეგორიების შექმნა შესაძლებელია როგორც კატეგორიების ჩანართიდან ასევე სწრაფი ღილაკის საშუალებით **+ Add New Category.** (იხ.სურ. 6.5.8)



მიმდინარე ეტაპზე მოცემულია მხოლოდ სიახლეების კატეგორია. დასაშვებია რამოდენიმე კატეგორიის ერთდროულად მონიშვნაც

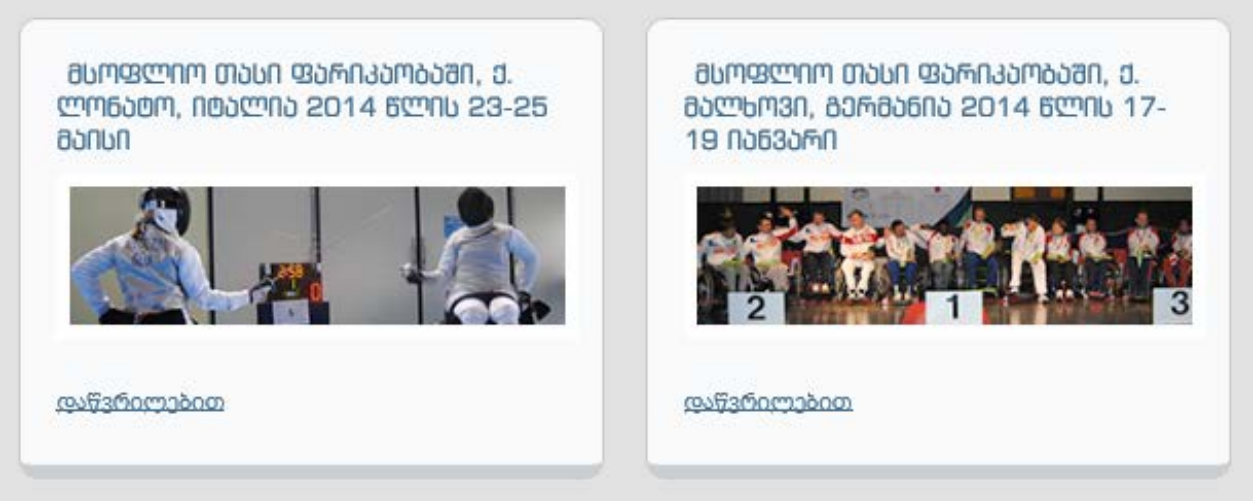
სურ. 6.5.8 ჩანაწერის კატეგორიის არჩევა

Tags - ბლოკი უზრუნველყოფს მიმდინარე ჩანაწერისთვის საკვანძო სიტყვების დამატებას. ჩამონათვალი ერთმანეთისაგან უნდა გამოიყოს მძიმის მეშვეობით. (იხ.სურ.6.5.9) მეტების დამატებისას



გაითვალისწინეთ რომ ინფორმაცია შეესაბამებოდეს ჩანაწერის შემცველობას. სურ. 6.5.9
Tag_ ების დამატება

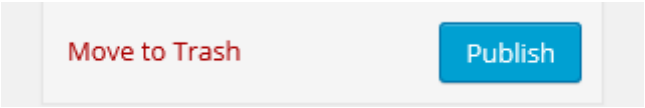
Featured Image - ბლოკის უზრუნველყოფს ჩანაწერზე საპრეზენტაციო -მინიატურა სურათის მიბმას. წინამდებარე აბზაცებში განხილულ იქნა სურათის ჩამატება ჩანაწერის შიგთავსში, მიმდიანრე ბმულის მეშვეობით შესაძლებელია კონკრეტულ ჩანაწერს გაუკეთდეს სურათი რომელიც მხოლოდ პირველ გევრდზე სტატიის პრეზენტაციისას გამოისახება და უკვე სრულ ვერსიაში ის აღარ იქნებაბ ხილული. ეს მეთოდი ხშირად გამოყენებადია. სტატიების გაფორმება სტანდარტული ზომის სურათებით. (იხ.სურ. 5.6.10)



სურათი 6.5.10 ჩანაწერი მინიატურით

ჩანაწერის ინფორმაციის დამატებისა და სასურველი ველების აქტივაციის შემდგომ დარჩა საბოლოო ეტაპი - **Publish** ღილაკზე ზემოქმედებით ჩანაწერის გამოქვეყნება.

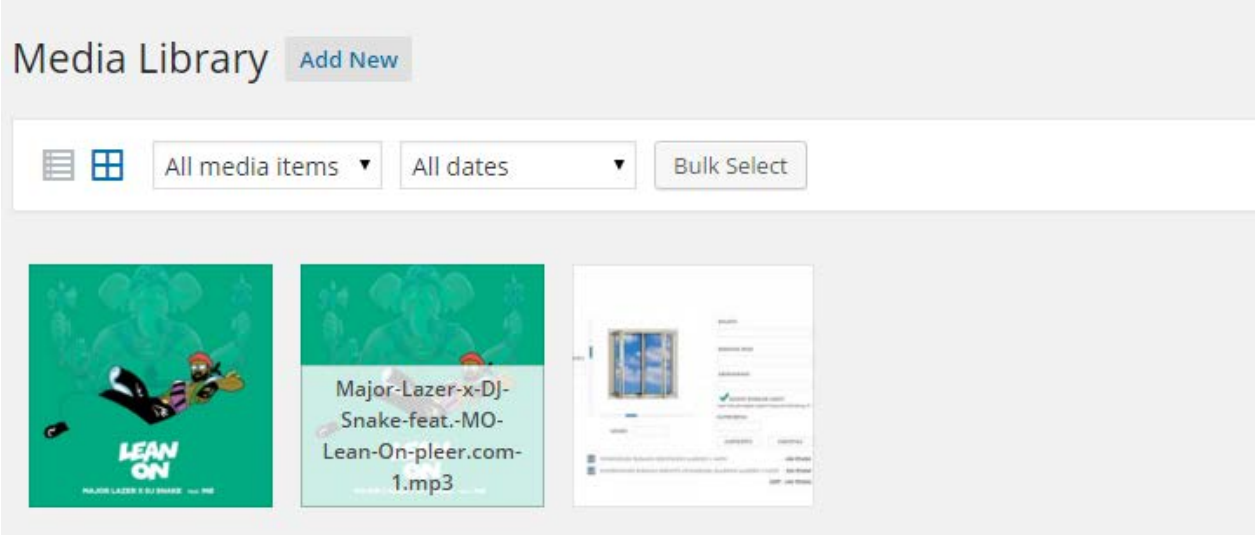
სტატიის უარყოფა მისი სანაგვეში გადატანითაა შესაძლებელი - **Move to Trash.** (იხ.სურ.. 6.5.11).



სურათი 6.5.11 ჩანაწერის წაშლა/გამოქვეყნება

Media

Post_ების შემდეგი ჩანართი **Media** (იხ.სურ. 6.2) საკმაოდ მნიშვნელოვანია. ის წარმოადგენს სურათებისა და მულტიმედია ფაილების არქივს. მიმდინარე ბლოკის საშუალებით შესაძლებელია როგორც ახალი ფაილების დამატება ასევე საიტზე არსებული ფაილების ხილვა, ძიება, სორტირება ფაილის ტიპისა და ატვირთვის დროის მიხედვთ. (იხ. სურ. 7.1) ასევე მათი წაშლა.



სურ.7.1 - Media ჩანართის ბლოკი - ატვირთული ფაილები და ფილტრაციის არეალი

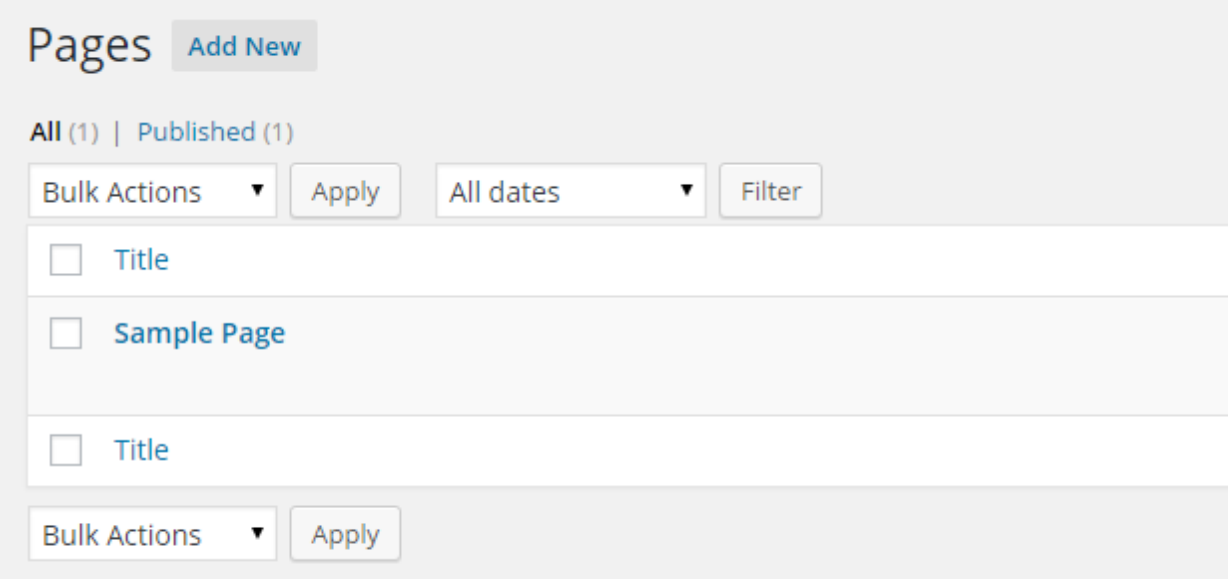
Pages

Post_ების მსგავსი სამართავი სისტემა გააჩნია **Pages** ჩანართს. (იხ.სურ.8.1) Pages_წარმოადგენს სტანდატული ტიპის გვერდების ერთობლიობას. მის ვიზუალზე დაკვირვებით შესაძლოა გვერდების პოსტებტან სხვაობა საწყის ეტაპზე საერთოდაც ვერ იქნას აღქმული, მაგრამ მათი შინაარსი ფუნდამენტურად განსხვავდება ერთმანეთისაგან.

გვერდი წარმოადგენს სტატიკურ ინფორმაციასთან მიმსგავსებულ ტიპს. მაგ.: საიტზე სანავიგაციო პანელში მენიუს პუნქტებად ხშირია „კომპანიის შესახებ“, „კონტაქტი“ და ა.შ. ბმულების გამოყენება, რომელთა დანიშნულებაა ერთგვაროვანი გვერდის გახსნა, სადაც გამოისახება კონკრეტული ინფორმაცია. მიმდინარე შემთხვევებში კომპანიის შესახებ წარმოადგენს ტექსტს, ხოლო კონტაქტი შესაძლოა შედგებოდეს საკონტქტო ფორმისა და მისამართების ერთობლიობისაგან.

რაც შეეხება პოსტებს - ჩანაწერები გაიგივებულია აქტიურ ფაზასთან. მათი ცვალებადობა ხდება ინტენსიურად. მაგ.: სიახლეების გვერდი. მისი გახსნის შემთხვევაში ერთი სტატიკური ინფორმაციის მატარებელი გვერდის მაგივრად გამოდის პოსტების ჩამონათვალი, შესაძლოა იყოს მოკლე აღწერები და შემდგომ მოხდეს გადამისამართება უშუალოდ ვრცელ სტატიაზე.

გვერდების სამათავი სისტემის დეტალურად გავლა არაფრის მომცემია გამომდინარე იქიდან რომ ის ფაქტობრივად პოსტების მართვის ანალოგურია (გადახედეთ ჩანართ **Posts**).



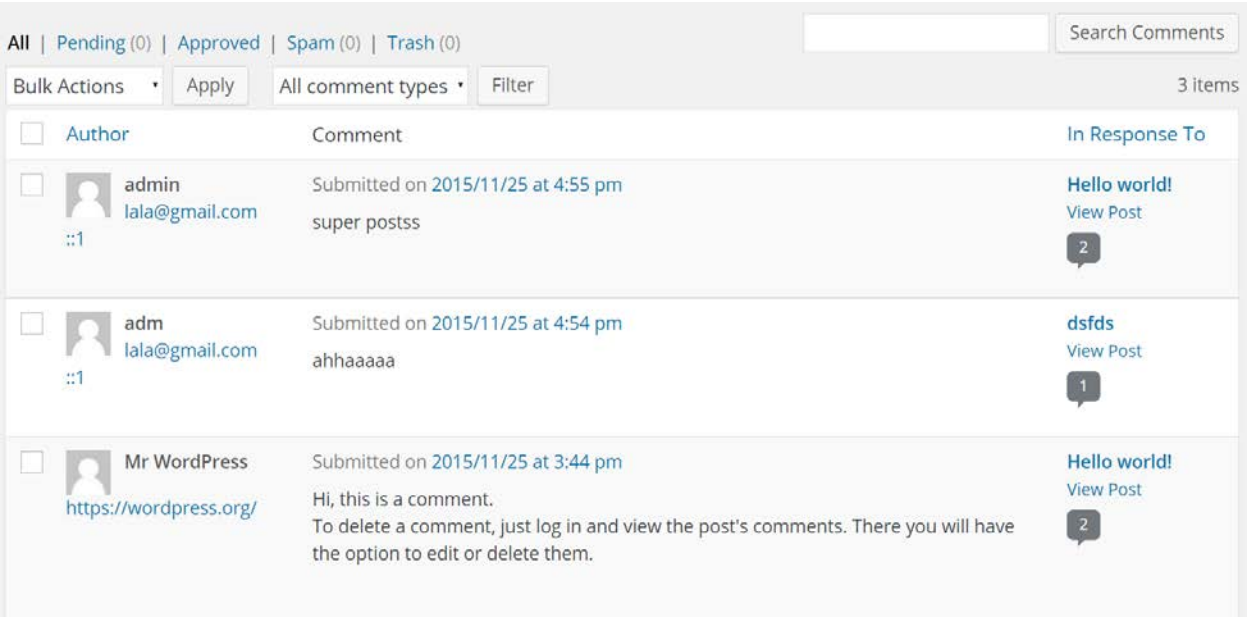
სურ. 8.1 - Pages - გვერდების მართვის ბლოკი

Comments

WordPress_ის გაცნობის შესავალშივე აღვნიშნეთ რომ იგი ორინეტირებული იყო ბლოგური ტიპის საიტების შექმნაზე. როგორც იცით ბლოგი თავის ტავში სტატიების ერთობლიობას გულისხმობს.

ბლოგერისათვის საკმაოდ მნიშვნელოვანი თუ რა გამოხმაურება მოჰყვება მის ამა თუ იმ ჩანაწერს. ასევე შესაძლოა კონკრეტული ტემა განხილვის საგნად იქცეს უშუალოდ მომხმარებლებს შორის.

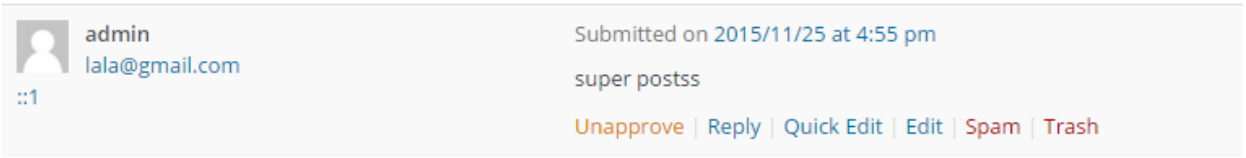
ყოველივე ამის გათვალისწინებით არ უნდა იყოს გასაკვირი რომ WordPress CMS აღჭურვილია კომენტარების ავტომატური სისტებით. ჩანართი **Comments** უზრუნველყოფს კომენტარების მონიტორინგს და გამოიყენება მათ სამართავად. მიმდინარე ბმულის აქტივაცია გვაძლევს სრულ საიას საიტზე დაწერილი კომნატერბის შესახებ. (იხ. სურ. 9.1)



სურ. 9.1 - კომენტარების სია

კომენტარების ბლოკი სხვადასხვა ინფორმაციის მატარებელია. აქ არის ავტორების ჩამონატვალი - Author სვეტში, უშუალოდ კომენტარები - Comments სვეტში და ჩანაწერები რომელთაც ეკუთვნით კონკრეტული კომენტარები.

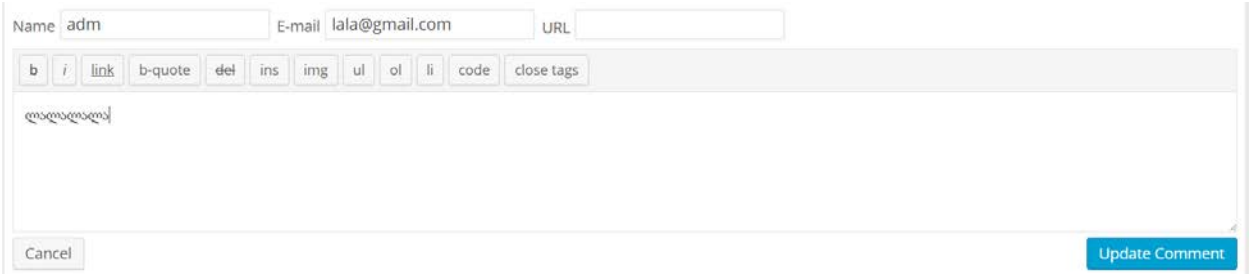
კომენტარების რედაქტირებისათვის საჭიროა მათზე კურსორის მიახლოვება რაც უზრუნველყოფს დამატებითი მენიუს გამოჩენას რედაქტირებისათვის (იხ.სურ.9.2)



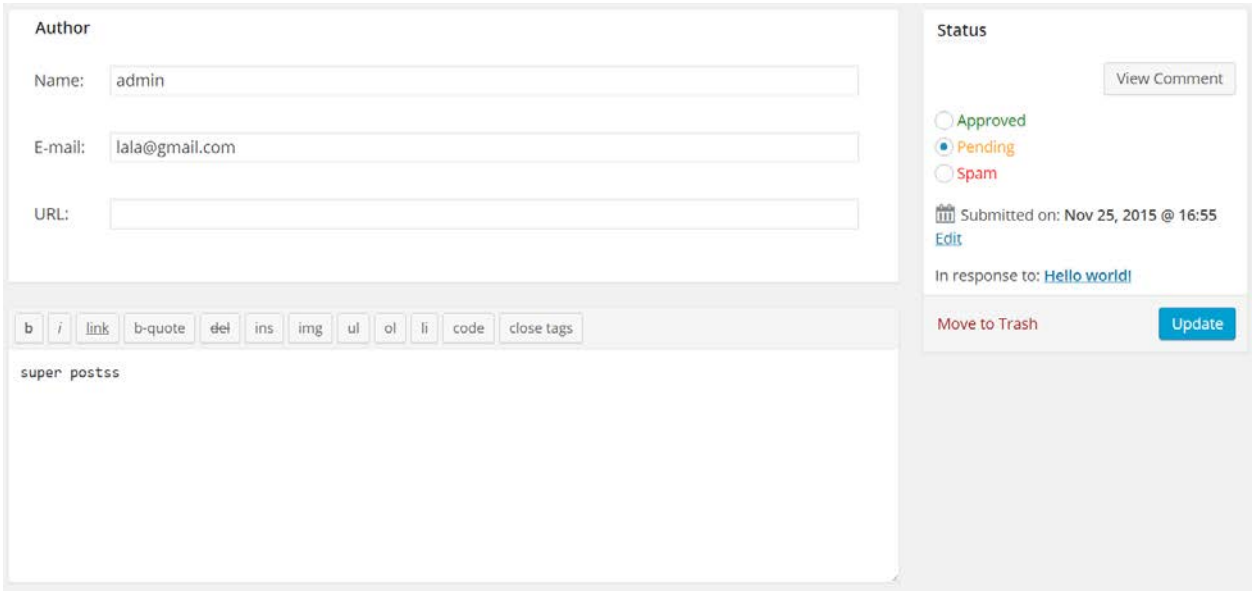
სურ. 9.2 - კომენტარების რედაქტირება

მენიუს ველები:

- **Unapprove** - კომენტარისმოდერაცია - ნების დართვა;
- **Replay** - კომენტარისთვის პასუხის გაცემა
- **Quick Edit** - სწრაფი რედაქტირება (იხ სურ 9.2.1). კომენტარის სწრაფი რედაქტირების ველში შესაძლებელია კომენტარის ავტორის, ფოსტის, URL მისმართისა და უშუალოდ კომენტარის შემცველობის ცვლილება. მაგრამ მის სტატუსის კონტროლი მიმდინარე ველებიდან ვერ ხერხდება
- **Edit** - კომენტარის რედაქტირება (იხ.სურ. 9.2.2). კომენტარის სრული რედაქტირება იძლევა, როგორც კომენტარის დამტოვებლის პარამეტრების და კომენტარის შემცველობის, ასევე სტატუსისა და გამოქვეყნების თარიღის ცვლილების შესაძლებლობას
- **Span**- კომენტარის გადატანა სპამებში;
- **Trash** - კომენტარის გადატანა სანაგვე ყუთში

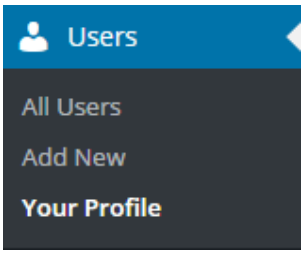


სურ. 9.2.1 - კომენტარების სწრაფი რედაქტირების ველი



სურ. 9.2.2 - კომენტარების სრულფასოვანი რედაქტირების ველი

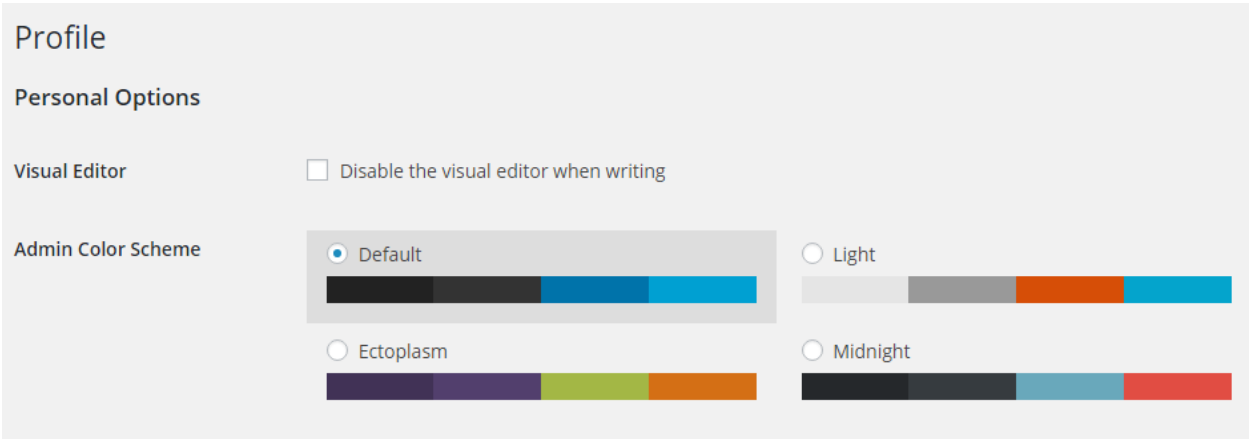
სამართავი სისტემა საშუალებას გვაძლევს მოვახდინოთ სხვადასხვა რაოდებოს მოხმარებლების დამატება საიტის სამართავად. მენიუს პუნქტ **Users**_ის მეშვეობით შესაძლებელია როგორც დარეგისტრირებული მომხმარებლების მონაცემების მონიტორინგი , ასევე ახალი მოხმარებლის დამატება. რა თქმა უნდა მიმდინარე ფუნქციონალთან ყველა მომხარებელს როდი აქვს წვდომის უფლება და ის მხოლოდ ადმინისტრატორების წილხვედრია. საკუთარი ანგარიშის ცვლილება ნებისმიერი უფლებები მქონე მომხმარებელს შეუძლია.



მიმდინარე ანგარიშის ცვლილება შესაძლებელია **Users >Your Profile** მისამართის აქტივაციით. ეკრანზე გამოსული შედეგი ველთა ერთობლიობა უზრუნველყოფს მიმდინარე მოხმარებლის სხვადასვხა მონაცემების ცვლილებას.

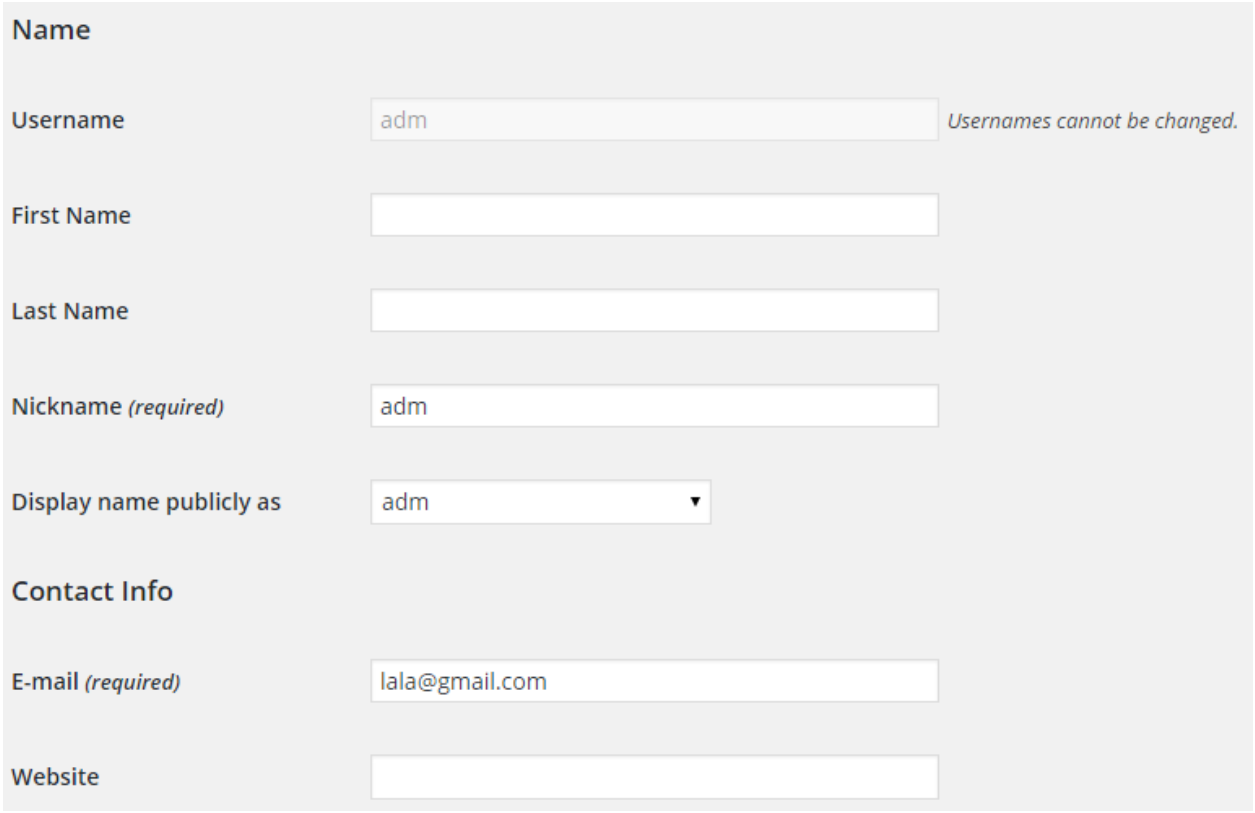
პერსონალური მონაცემების პარამეტრებიდან გამოსაყოფი ველებია:

- **Visual Editor** - მონიშვნის აქტივაცია გამოიწვევ მიმდინარე მომხმარებლისთვის სხვადასხვა ჩანართში ვიზუალური ედიტორის დამალვას. მაგ.: ახალი გვერდის ინფორმაციის დამატებისას გამოყოფილი ველი (textarea) არ იქნება ვიზუალური რედაქტორთ გაფორმებული. მასში მხოლოდ html ბრძანებების განთავსების შესაძლებლობა იქნება. (უმჯობესია ეს ველი დატოვოთ გაუაქტიურებელ მდგომარეობაში) (იხ.სურ. 10.1)
- **Admin Color Scheme** - ადმინისტრირების პანელის დიზაინი. შესაძლებელია ადმინისტრირების პანელის დიზაინი შეირჩეს თქვენს გემოვნებასთან მიმსგავსებული ფერთა გამისაგან. WordPress გთავაზობთ რამდენიმე შაბლონურ ვარიანტს სწორედაც რომ მიმდინარე არეალში. (იხ.სურ. 10.1)



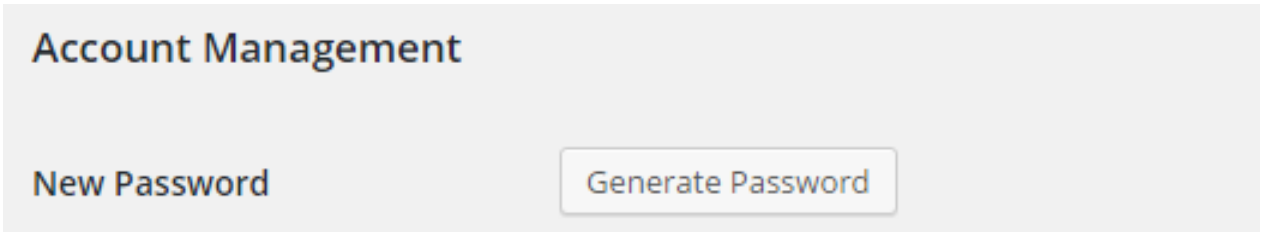
სურ.10.1- მომხამებლის პერსონალური პარამეტრები- სამართავი პანელის ფერთა გამის შერჩევა

- **Name** - ბლოკის ველთა ერთიანობა ემსახურება მომხამრებლის პირადი ინფორმაციის ცვლილებას. მომხარებლის **Username**_ის ცვლილება არ შეიძლება, შესაძლებელია **Nickname** ველში ახალი სახელის შერჩევა და უკვე ველ **Dispay name publicly as** ახალი nickname აქტივაცია. რაც თავისთავად ამ მომხარებელს მიანიჭებს ახალ სახელს და მისი ნებისმიერი ქმედების შედეგად ავტორად სწორედა ახალი nickname_ით იქნება მოხსენიებული. მიმდინარე ველებში **E-mail** - ელ. ფოსტის და Website -ვებ გვერდის ცვლილება (იხ. სურ.10.2)

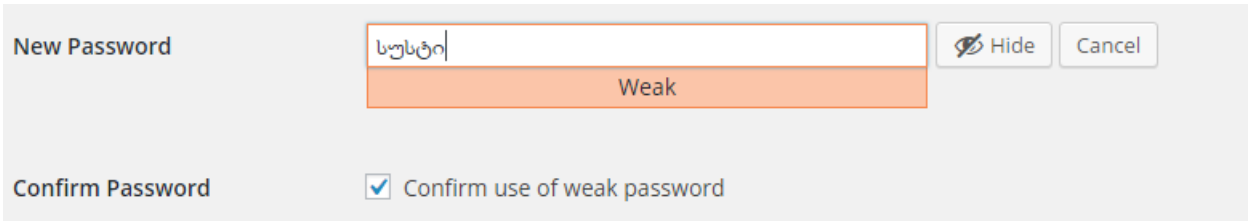


სურ.10.2- მომხამებლის პირადი სახელის,ფოსტისა და ვებგვერდის კომფიგურაცია

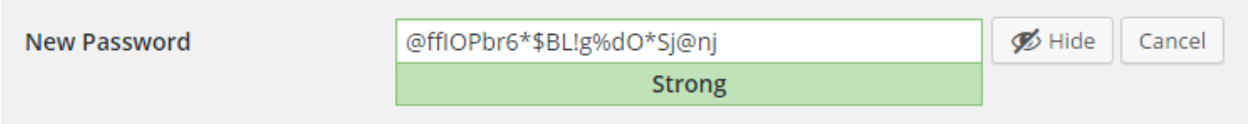
- **New Password** - ახალი პაროლის მინიჭება Generate Password. (იხ.სურ10.3) უმჯობესია ახალი პაროლი გენერირებული იყოს სხვადასხვა რეგისტრის ასოებისა და ციფრების ერთობლიობისაგან. მიმდირე ეტაპზე პაროლის სირთულის აღსანიშნათ ფერების გამაც დაგეხმარებათ. (წითელი_სუსტი, ყვითელი _საშუალო, მწვანე _ მისაღებად დაცული.). სუსტი პაროლის შემთხევაში WordPress_ის დაცვლის სისტემა გაფრთხილებთ და ავტომატურ დასტურს არ გაკეთებინებთ. თუ მაინც გსურთ პაროლის ძალაში დატოვება მოგიჭევთ დამატებით ველ **Confirm Password** დათანხმება. (იხ.სურ10.4) იდეალური შემთხვევა იქნება თუ დაიტოვებთ ალბათობით მონიჭებულ პაროლს. (იხ.სურ10.5)



სურ.10.3- ახალი პაროლის მინიჭება



სურ.10.4- სუსტი პაროლის შემთვევაში პასუხისმგებლობის საკუთარ თავზე აღება



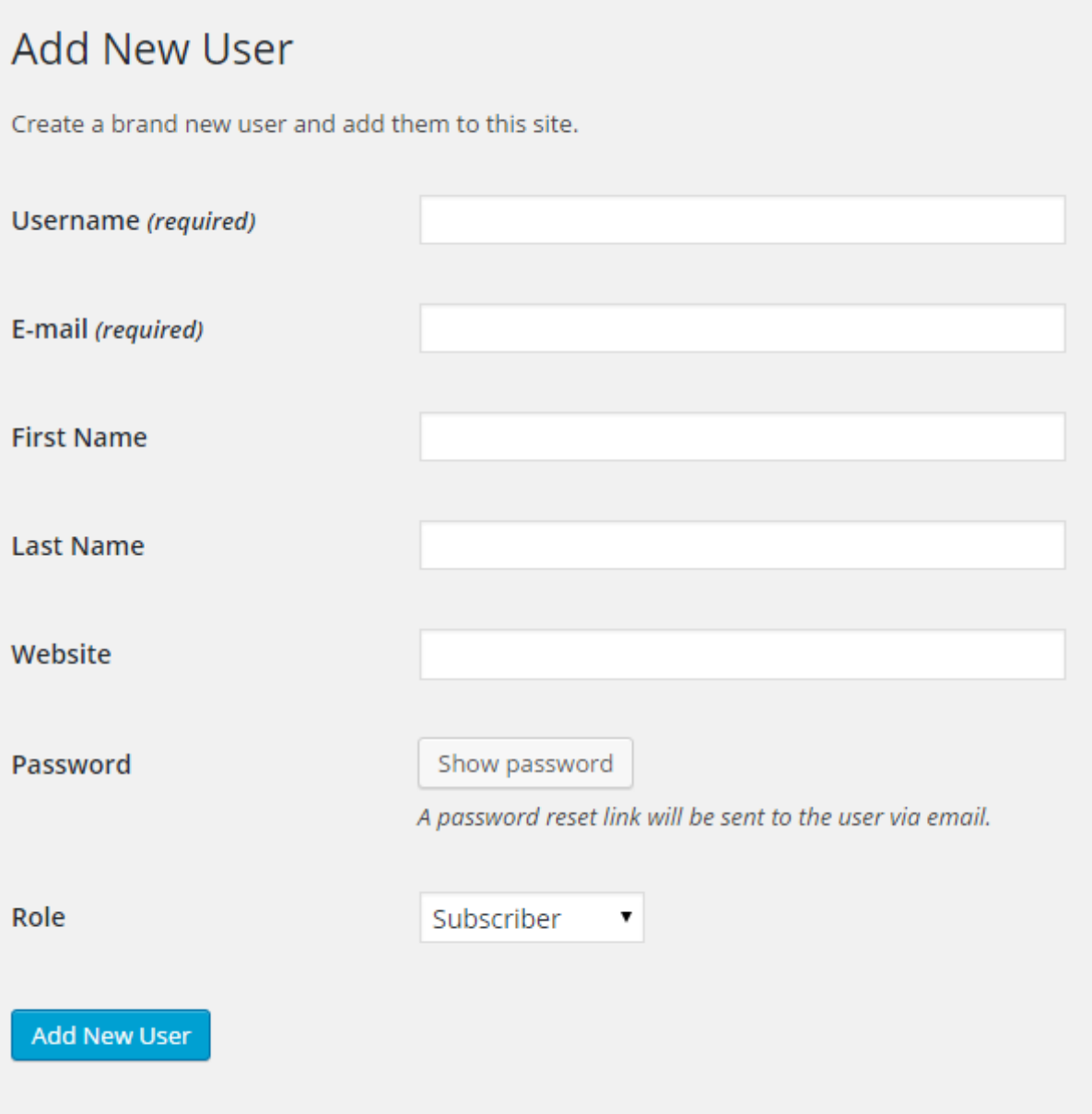
სურ.10.5- ალბათობით გენერირებული პაროლი

მიმდინარე ანგარიშის ცვლილების გარდა როგორც აღვნიშნეთ შესაძლებელია მომხამრებლების მონიტორინგი და ახალი მომხმარებლის დამატება. მაგრამ ეს მხოლოდ ადმინისტრატორის პრეროგატივაა.რეალურად აქამდე განხილული ველებიდან ყველასთან წვდომა მხოლოდ ადმინის ფუნქციებში შედის. პირველი მომხმარებელი რომელიც WordPress_ის ინსტალაციის დროს დავარეგისრირედ (იხ. სურ. 4.5.5) სწორედაც რომ ადმინისტრატორის პრივილეგიებით აღიჭურვა.

ახალი მომხმარებლის რეგისტრაციისათვის საჭიროა შემდეგი მისამართის აქტივაცია **Users >Add New.**

გამოსულიშედეგი ფაქტობრივად იდენტურია Profile_ის რედაქტირებისა. ერთი განსხვავებით უკანასკნელი ველი **Role** მოხსენიებულია ახალი მომხმარებლის დამატებისა და სხვა მომხარებლების რედაქტირების დროს. სწორედაც ეს ველი განსაზღვრავს მომხმარებლის პრივილეგიას. (იხ.სურ.10,6)

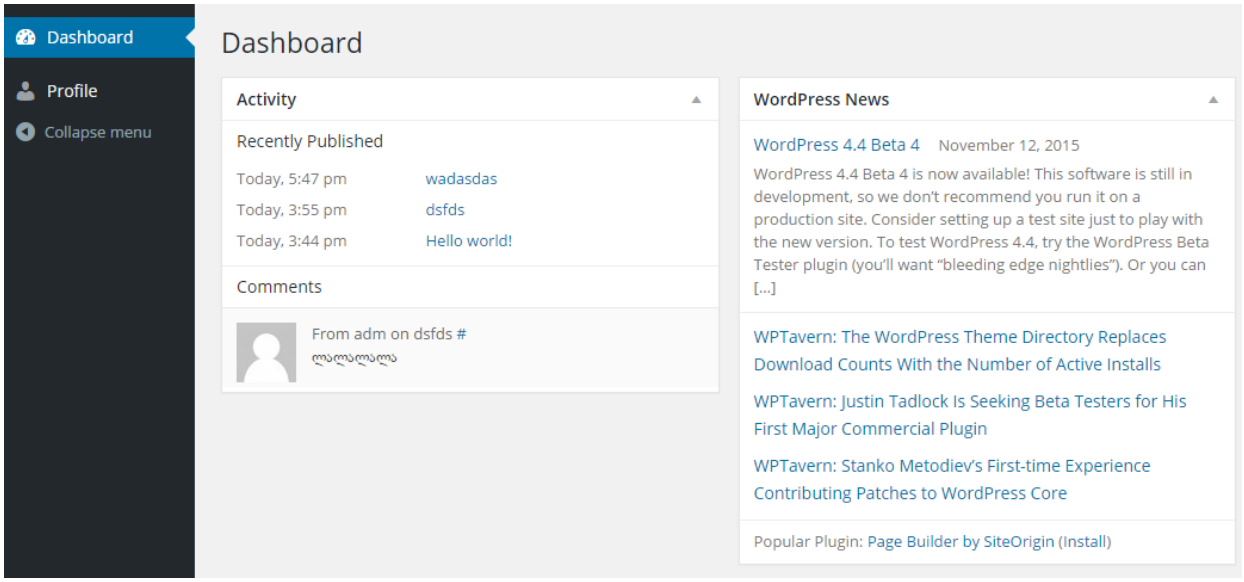
დაიმახსოვრეთ ახალი მომხარებლის დამატება და სხვა მომარებლების ცვლილების შესაძლებლობა აქვს მხოლოდ და მხოლოდ ადმინისტრატორის პრივილეგიის მქონე მომხხარებელს



სურ.10.6- ახალი მომხმარებლის დამტება

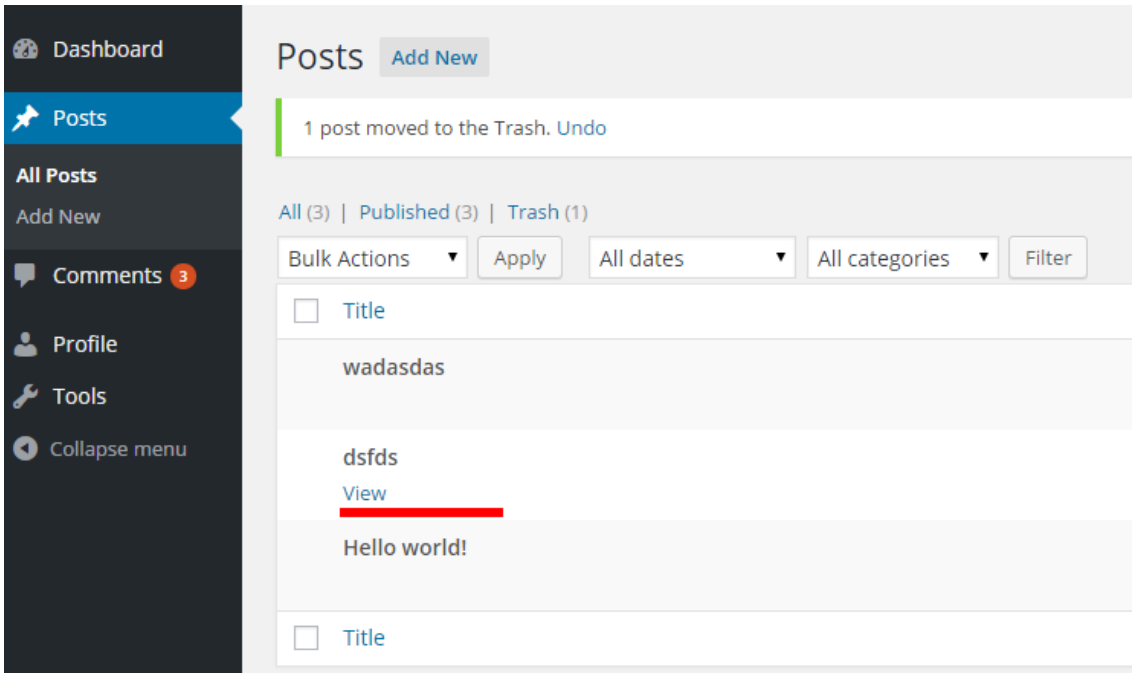
Role - პრივილეგიის პუნტები მნიშვნელობების მხრივ შემდეგნაირად გამოიყურება:

- **Subscriber** - მომხარებელი. მის უფლებეში მხოლოდ კონსოლის (Dashboard)პირველი გვერდის გაცნობა და საკუთარი პროფილის რედაქტირება შეუძლია.(იხ.სურ. 10.7.1) ერთადერთი შესაძლებლობა თუ ამას შესაძლებლობა შეიძლება ეწოდოს შემდეგია -მის მიერ საიტზე დატოვებული კომენტარის ავტორად მისი სახელი გამოქვეყნდება.



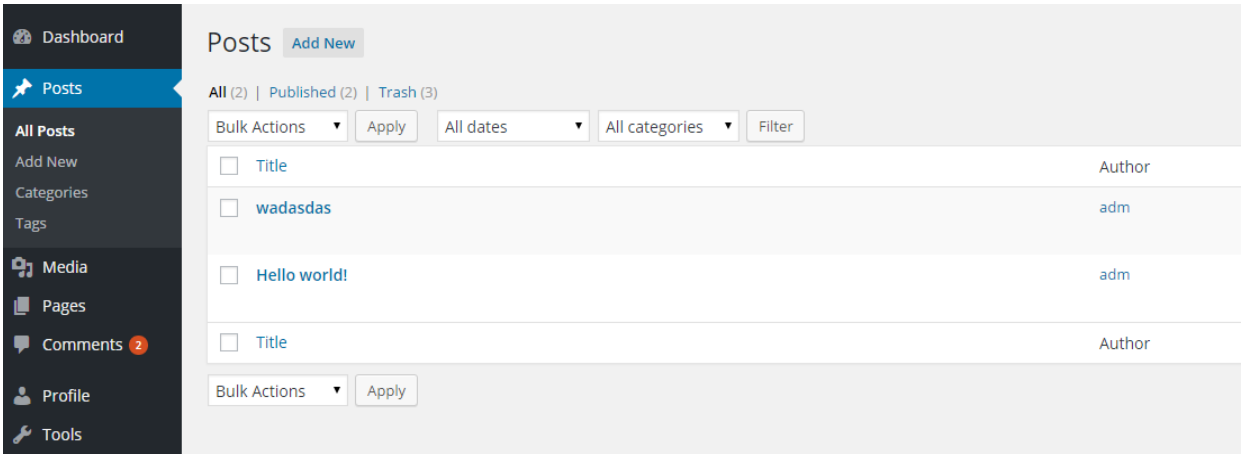
სურ.10.7.1- Subscriber - მომხმარებლის ადმინისტრირების პანელი

- **Contributor –** ამ პრივილეგიის მქონე მომხმარებელს შედარებით ფართო შესაძლებლობები გააცნია უფლება აქვს დაამატოს ჩანაწერები - პოსტები, ოღონდ მათი გამოქვეყნება მხოლოდ ზედა პრიორიტეტის მქონე მომხმარებლის დამოწმების შემდეგ იქნება შესაძლებელი. მას აქვს უფლება წაშალოს მხოლოდ თავის მიერ გამოსაქვეყნებლად გამზადებული სტატიები(იხ.სურ. 10.7.2)



სურ.10.7.2- **Contributor** - მომხმარებლის ადმინისტრირების პანელი

- **Author -** ავტორის უფლებების მქონე მომხმარებელი. მას შეუძლია ჩანაწერების გამოქვეყნება სხვების უკითხავად. საკუთარი ჩანაწერების სრული ფუნქციონალით მართვა. მედიაფალილებთან ურთიერთობა.მას არ გააჩნია წვდომა გვერდების Pages ჩანართთან;
- **Editor -** წარმოადგენს მომხმარებელს რომელიც აღჭურვილია დამატების , რედაქტირებისა და წაშლის ფუნქციებით. მის კომპეტენციას სცდება მომხმარებლების დამატება, ახალი მოდულებისა დაყენება და CMS _ის განახლებება(იხ.სურ.10.7.3)



სურ.10.7.2- **Editor** - მომხმარებლის ადმინისტრირების პანელი

- **Administrator** - ადმინისტრატორი. მას გააჩნია ყველანაირი უფლება .

პლაგინების გამოყენების მთოდები

პლაგინები - ეგრეთ წოდებული დანამატები საკმაოდ ხშირად გამოყენებადი ელემენტებია WordPress_ის ძრავისათვის და ისინი დიდ ზეგავლენას ახდენენ CMS მრავალფუნქციონალურაზე.

WordPress_ის ძირითად პრიორიტეტად სწორედაც პლაგინების მარტივი ხელმისაწვდომობა სახელდება. ინტერნეტ სივრცეში დიდი რაოდენობით მოიპოვება სხვადასხვა დანიშნულების და შესაძლებლობის დანამატები, რაც ყველაზე მთავარია მათი უმეტესი ნაწილი სრულიად უფასოა. უშუალოდ WordPress_ის ოფიციალურ ვებ გვერდზეც შესაძლებელია დანამატების მოძიება. იხილეთ ბმული https://wordpress.org/plugins/

არსებობს პლაგნების საიტზე განთავსების ორი ძირითადად გზა:

1. შესაძლებელია დანამტის მოძიება უშუალოდ ადმინისტრურების პანელიდან;
2. შესაძლებელია დანამატის დაყენება ფაილური სტრუქტურიდან;

დანამტის მოძიება უშუალოდ ადმინისტრურების პანელიდან

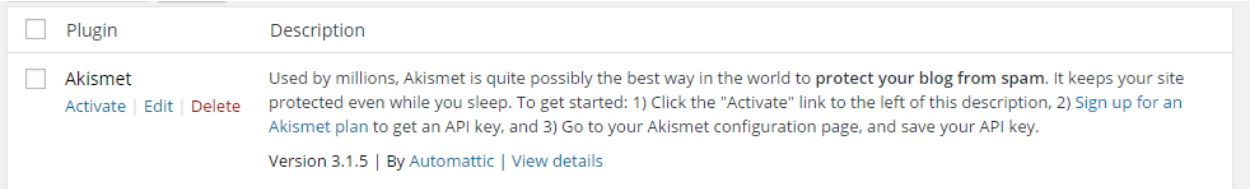
WordPress_ის სამართავი სისტემის მენიუს პუნქტიხ **Plugins** უზრუნველყოფს საიტზე არსებული დამატებების მართვას. (იხ.სურ. 6.2)

მითითებული ბმულის აქტივაცია გამოიტანს საიტზე არსებულ ყველა დანამატს. (იხ.სურ. 11.1) შესაძლოა დანამატები როგორც პასიურ ასევე აქტივირებულ მდგომარეობაში იმყოფებოდეს. პლაგინები რომელთა მომხამრების საჭიროება მიმდინარე ეტაპზე არ არსებობს ან ყენდება სატესტო რეჟიმში შესაძლოა გაითშოს რათა არ მოხდეს ერთმანეთისათვის ხელის შეშლა.

პლაგინები რა თქმა უნდა დიდ დახმარებას უწევს დეველოპერს და საკმაოდ უადვილებს საქმეს მაგრამ ერთგვარ თავსატეხსაც წარმოადგენს. გამომდინარე დანამატების ხელმისაწვდომობის ფართო არეალისა არსებობს ალბათობა უშუალოდ პლაგინიდან მოხდეს შეღწევადობა საიტის სამართავ პანელში ან მონაცემტა ბაზაში რაც გამოიწვევს მის დაზიანებას (ეგრედ წოდებული საიტის გატეხვა). ამიტომ დანამატების მოძიება დაყენებისას ზედმიწევნით გაარჩიეთ სხვადასხვა წყაროებზე (იქნება კომენტარები,

ფორუმები თუ ანოტაცია) დაყრდნობით რაიმე ხარვეზი ხო არ შეინიშნება კონკრეტული დანამატის ფუნქციონირებისას.

ასევე რეკომენდირებულია : მომქმედ საიტზე არ დატოვოთ გაუატიურებელი დანამატი რადგანაც ის წარმოადგენს პოტენციურ საფრთხეს. თუ დანამატი გჭირდებად ის შესაბამისად უნდა იყოს აქტივირებული, თუ არ გჭირდებათ უმჯობესია წაიშალოს.

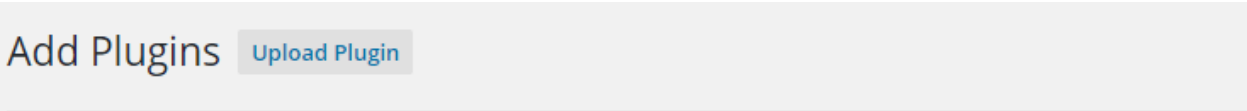


სურ. 11.1 პლაგინების ძირითადი არეალი

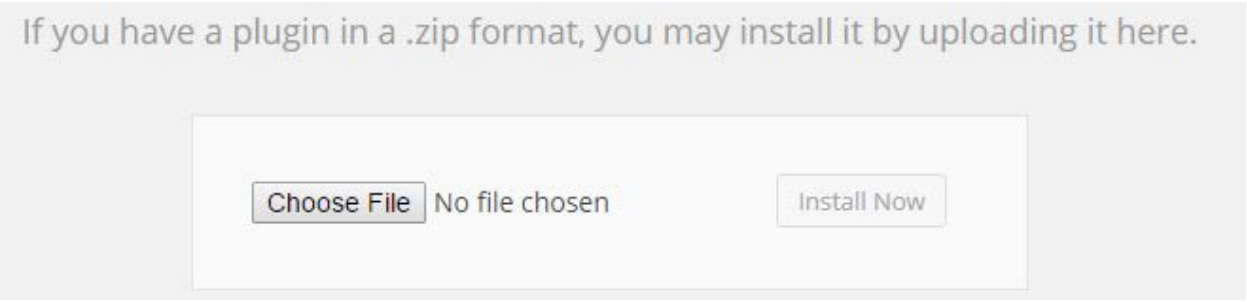
სურათ 11.1_ზე ნაჩვენებია WordPress_ის საწისი დანამატი Akismet. ეს კონკრეტული პლაგინი უზრუნველყოფს საიტზე არსებული კომენტარებიდან სპამების გამოვლენას. თუ საიტი არ არის ორიენტირებული კომენტარების სისტემაზე (ამდაგვარი ტიპის საიტები ძირითად შემთხვევაში ბლოგებია) მიმდინარე პლაგინი საერთოდ არ არის საჭირო და შესაძლებელია მისი წაშლა **Delete** ბმულის გამოყენებით. დანამატის აქტივაციისათვის გამოიყენება ბმული **Activate.** ბმული **Edit** გამოიყენება პლაგინის კოდის ცვლილებისთვის. ამ უკანასკნელის მხოლოდ მცირეოდენი ახსნით შემოვიფარგლოთ გამომდინარე იქიდან რომ მისი ცვლილება პროგრამირების ცოდნას მოითხოვს, ამიტომ დეტალებში შესვლა მიმდინარე ეტაპზე სცდება კომპეტენციის ფარგლებს.

ახალი პლაგინის დასაყანებლად მნიშვენელოვანია კონკრეტულად იმ კატეგორიის დანამატის მოძებნა რა პროცესის შესრულებაც გვიწევს. მაგ.: თუ გვსურს საიტის საკონტაქტო გვერდიდან მოხდეს წერილის გაგზავნა ფორმების მეშვეობით საჭიროა მოძიებულ იქნას კონტქტის პლაგინი და არა გალერეის.

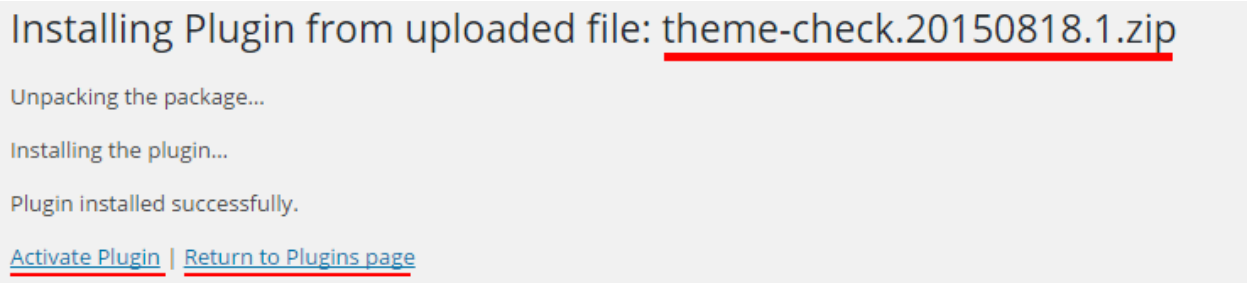
იმისათვის რომ მოხდეს ახალი პლაგინის მოძიება საჭიროა **Plugins**>**Add New** ჩანართზე გადახვლა. ახალი პლაგინის დამატებისას შესაძლოა პლაგინი მოძიებულ და ჩამოწერილ იქნას ინტერნეტიდან. ძირითად შემთხვევაში პლაგინი .zip ფორმატში იქნეა . მის დასამატებლად საჭიროა ჩანართ **Upload Plugin** აქტივაცია (იხ.სურ. 11.2), რომელიც თავის მხრივ გვთავაზობს ფაილის ასატვირთ ფორმას. (იხ.სურ. 11.2.1). ამ გზით დაყენებული ჩანართი საწყის ეტაპზე დეაქტივირებული ამიტომ ეკრანზე არ აისახება საჭიროა მისი აქტივაცია უშუალოდ მოქმედებაში მოსაყვანად **Activate Plugin** (იხ. სურ. 11.2.2). **Return to Plugins page**-ბმულის მეშვეობით შესაძლებელია პლაგინების გვერდზე დაბრუნება.



სურ. 11.2 პლაგინის ფაილაის ატვირთვის არეალის გამოძახება

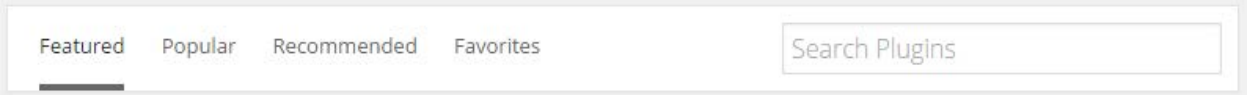


სურ. 11.2.1 პლაგინის ასატვირთი ფორმა



სურ. 11.2.2 პლაგინის ატვირთვის წარმატებით დასრულებისას მიღებული შედეგი

შესაძლოა მოძიება დაყენების პროცესი მოხდეს ერთდორულად. თუ საწყის ეტაპზე არ გაგვაჩნია კონკრეტული პლაგინი მისი მოძიება შესაძლებელია **Plugins > Add New** საძიებო ველის (**Search Plugins**) მეშვეობით (იხ. სურ. 11.3). ახალი დანამატის დაყენებისას შესაძლოა გათვალისწინებულ იქნას Wordpress_ ის ფილტრი-პალგინების სორტირება.(იხ. სურ. 11.3).



სურ. 11.3 პლაგინის მოძიება

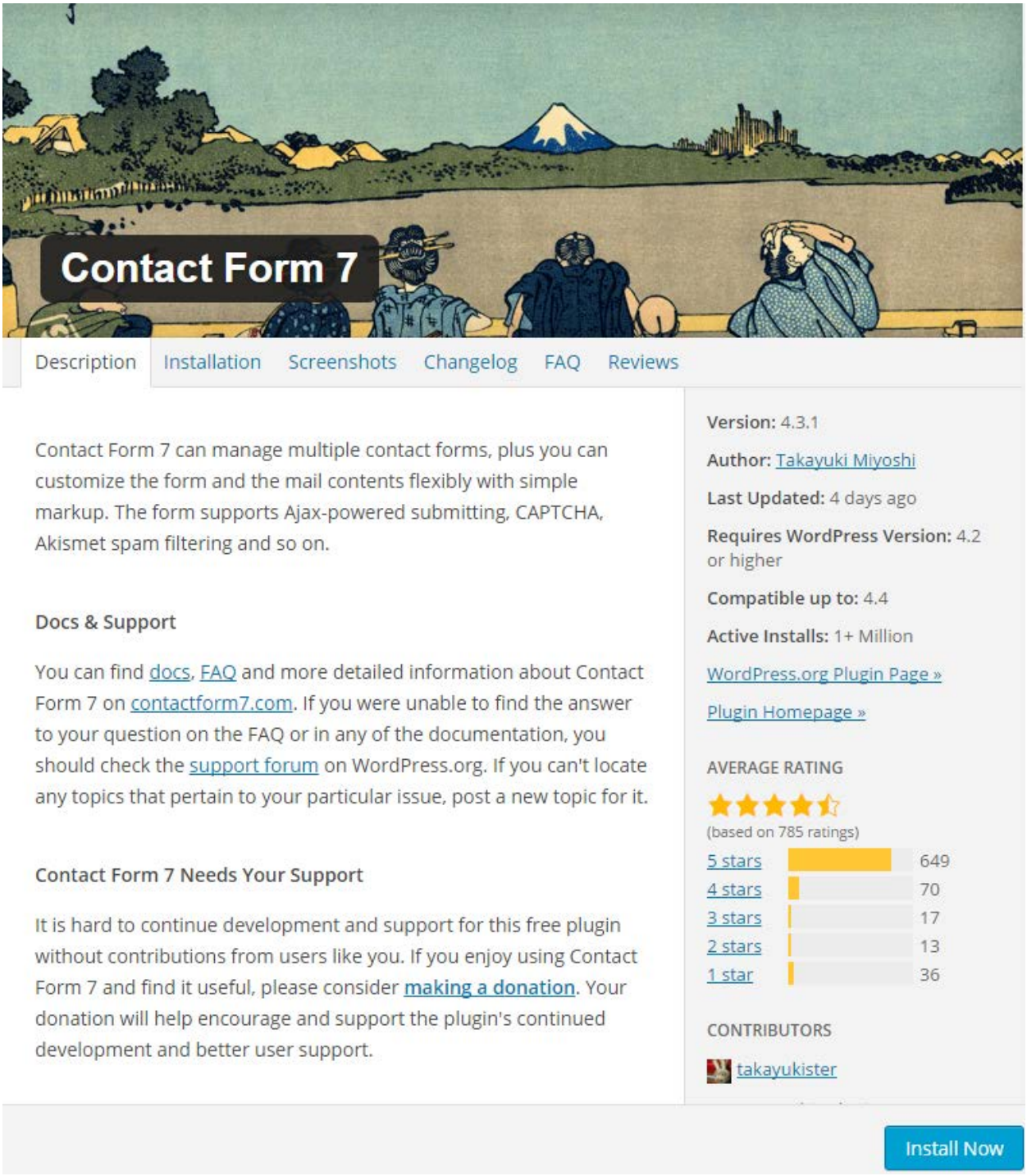
- **Featured** - რჩეული პლაგინების ჩამონათვალი;
- **Popular** - პოპულარული / ყველაზე ხშირად გამოყენებადი ფლაგინები;
- **Recomended** - რეკომენდირებული დანამატები;
- **Favorite** - ფავორიტი / ჩვენს მიერ სროტირებული დანამატები;

მაგალითისათვის გავარჩიოთ რამოდენიმე პლაგინი. როგორც აღვნიშნეთ ვთქვათ ვეძებთ საკონტაქტო ფორმას , რომელიც უზრუნველყოფს წერილის გაგზავნას უშუალოდ საიტიდან. ამისათვის ლოგიკურად საძიებო ველში ვკრეფთ **contact form** ან უბრალოდ **Contact.** ძიების დასრულების შემდეგ საიტი გვთავაზობობს იმ პლალგინების ერთობლიობას რომელიც მიმსგავსებულ იქნა საძიებო სიტყვასთან. ამ არქივიდან უნდა შეირჩეს ჩვენთვის სასურველი დანამატი. როგორც აქამდე აღვნიშნეთ უმჯობესი იქნება პლაგინის დაყენებამდე ამომწურავად გავეცნოთ მის შესახებ გამოხმაურებებს. არ გამორიცხოთ შემთხვევა - შესაძლოა პლაგინი საწყის ეტაპზევე იყოს დაზიანებულ მდგომარეობაში.

მოძიებული დანამატებიდან შევარჩიოთ ერთ ერთი მაგ.: **Contact Form 7.** (იხ.სურ 11.4)

ნებისმერ პლაგინის ანოტაცია შესაძლებელია მივიღოთ **More Details** ბმულზე გადასვლით (იხ.სურ. 11.4.1) ასევე კონკრეტული დანამატის ხარისხიანობაზე მეტყველებეს ვარსკვლავებით გამოხატული რეიტინგი.

სურ. 11.5 მოძიებული დანართი



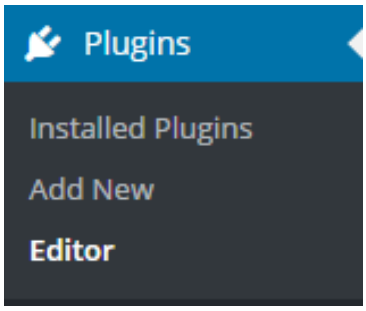
11.4.1Contact Form 7 ანოტაცია

Contact Form 7 ანოტაცია წარმოდგენელილია სურათ 11.4.1_ზე. ძირითადად სხვადასხვა დამატებების მენიუები მიმსგავსებულია ერთმანეთთან. ფაქტობრიავდ ყველგანაა პლაგინის მოლე აღწერა- **Description,** ინსტალაციის მეთოდები -**Installation**,

შესაძლოა ნაჩვენები იყოს სურათებად თუ როგორ გამოიყურება მუშა მდგომარეობაში ესა თუ ის კონკრეტული პლაგინი - **ScreenShots,** ასევე ინფორმაცია ვერსიათა უახლესი განახლებებს შორის არსებულ ცვლილებეზე -**Changelog,** აქტუალური ხშირად დასმული შეკითხვები - **FAQ,** კომენტარები / განხილები - **Reviews.**

ამ დოკუმენტაციის გაცნობის შემდეგ შესაძლებელია თამამად მიიღოთ გადაწყვეტილება გსურთ თუ არა ამა თუ იმ პლაგინის მოხმარება.

პლაგინების თითქმის 100% თან ახლავს ანოტაცია სადაც გაშიფრულია თითოეული დეტალი. მისი პარამეტრების რეგულირებიდან უშუალოდ საიტზე წარმოჩენამდე. სასურველია წინასწარ ამ კონკრეტული ფაილის **readme.txt** გაცნობა. ამისათვის შევდივართ **Plugins > Editor .** (იხ. სურ. 11.5)

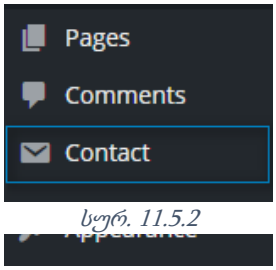


სურ.11.5 ფაილური სტრუქტურა

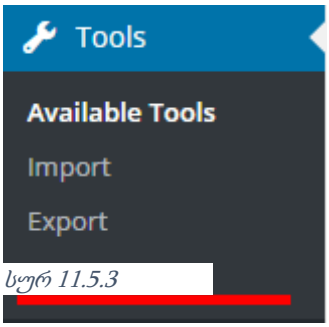
მარჯვენა ზედა არეალში **Select plugin to edit** შესაძლებელი უშუალოდ იმ პლაგინის არჩება რომლის ანოტაციის გაცნობაცაა სასურველი. **Plugin Files** ბლოკში გააქტიურდება არჩეული დანამატის ფაილები. **Readme.txt** ფაილი გვაწვდის ამომწურავ ინფორმაციას მიმდინარე პლაგინის დაყენების, აქტივაციის და მისი გამოძახების შესახებ. (იხ სურ. 11.5.1)

სურ.11.5.1 პლაგინის readme.txt

დაყენებული პლაგინი ადმინისტრირების პანელში შეიძლება სხვადასხვა ადგილას აისახოს. ძირითად შემთხვევაში დანამატი განთავსდება WordPress_ის სამართავი პანელის მენიუს ერთ ერთ პუნქტად (იხ.სურ.11.5.2)



სურ. 11.5.2



სურ 11.5.3

შესაზლებელია დაყენებული პლაგინი განთავსდეს **Tools** მენიუს ქვეპუნქტად. მაგ.: სურათ. 11.5.3 _ ზე ნაჩვენებია პლაგინ **WP-DB Backup** მდებარეობა.

შესაძლებელია პლაგინი განთავსდეს WordPress_ის **Settings** მენიუს ქვეპუნქტად

შესაძლებელია დაყენებული პლაგინი საერთოდ არ გამოჩნდეს ადმინისტრირების პანელში და ისე იმუშაოს. ამიტომ სავალდებულოა ზედმიწევით დაწვრილებით გაეცნოთ Readme.txt ფაილს.

პლაგინის დაყენებისა და აქტივაციის შემდგომ შესაძლებელია მასთან წვდომა და პარამეტრების რეგულირება. რა თქმა უნდა ეს ყოველი ინდივიდუალურია პლაგინისდამიხედვით ამიტომ რომელიმე

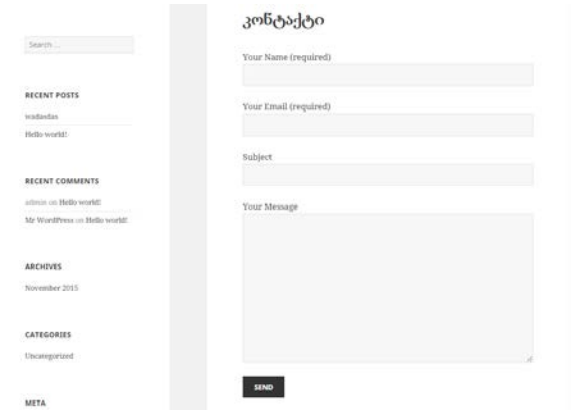
კონკრეტული მაგალითის განხილვა ვერანაირ შედეგს ვერ მოიტანს. თან საინტერესოა ყურადღების გამახვილება იმაზედ, რომ შესაძლოა თუნდაც მიმდინარე დანამატის პარამეტერების რეგულირების ცოდნა ამ ტექსტის გარჩევისას უკვე ისე იყოს მოძველებული, რომ ახალ ვერსიასაც კი არ შეესაბამებოდეს, არათუ სხვა პლაგინის პარამეტრებს.

ერთი მნიშვნელოვანი დეტალი რაც არ უნდა დატოვოთ ყურადღების მიღმა ეს არის პლაგინის გამოძახება სასურველ ადგილას. რა თქმა უნდა ესეც ინდივიდუალური მაგრამ შედარებით სტრუქტურიზებული. არსებობს რამდენიმე ძირითადი გზა პლაგინის გამოსაძახებლად:

1. პლაგინის გამოძახება შეიძლება შემოკლებული კოდის (**Shortcode**) მეშვეობით. მისი მოძიება შეიძლება დანამატის readme.txt ფაილში. **Shortcode** - შესაძლოა ხელმისაწვდომი იყოს უშუალოდ პლაგინის პარამტერების რეგულირების შემგომ. ჩვენს მიერ დაყენებული პლაგინის შემთხვევაში მისი პარამეტრების გამოძახებისას გამოსულ ინფოს თან ახლავს Shortcode (იხ.სურ. 11.6.1) პლაგინი გამოიძახება მას შემდეგ რაც მონიშნულ კოდს გავააქტიურებთ (ჩავაკოპირებთ) იმ გვრდზე, რომლის გამოძახებისასაც გვსურს რომ გაიხსნას ამ შემთხვევაში საკონტაქტო ფორმის ველები. (სურ. 11.6.2.1 – 11.6.2.2)

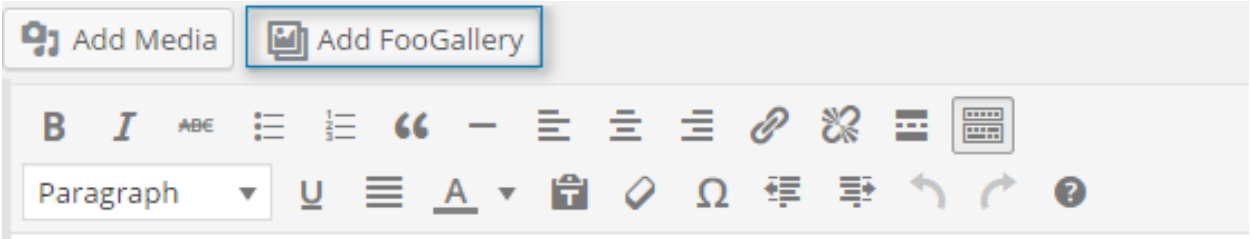
სურ. 11.6.1 საკონტაქტო ფორმის გამოსაძახებელი Shortcode

სურ. 11.6.2.1 Shortcode ჩასმა გვერდის ველში



სურ.11.6.2.2 საიტზე გამოშული შედეგი

2. შესაძლებელია პლაგინის დაყენებისას მისი გამოძახების ღილაკი ავტომატურად გამოჩნდეს **Posts** და **Pages** ჩანართის ველებში და მათი აქტივაცია მხოლოდ მაუსის მარტივი კლიკით მოხდეს. სურათ 11.6.3 ნაჩვენებია ერთ ერთი პლაგინი (Foo Gallery) რომელიც ავტომატურად აყენებს გამშვებ ღილაკს გვერდებისა (**Pages**) და ჩანაწერების (**Posts**) ჩანართზე.



სურ. 11.6.3 Foo Gallery პლაგინის გამოსაძახებელი ღილაკი

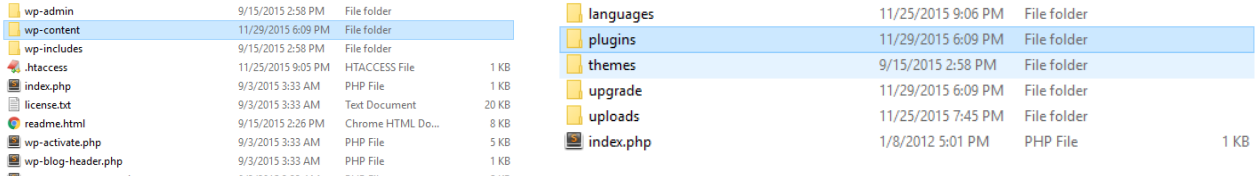
ზემოთ მოყვანილი მაგალითები ის ძირითადი გზებია რომელთაც პლაგინების უმრავლესობა იყენებს გამოძახების აქტივაციისათვის. არსებობს ასევე Shortcode_ბი რომელთა გამოყენება შესაძლებელია უშუალოდ php კოდში ჩასმით. ამ საკითხს დროებით არ ვეხებით გამომდინარე იქიდან რომ ფაქტობრივად კოდთან ურთიერეთობა ამ ეტაპზე არ გაგვირჩევია.

კომპანიების მიერ შექმნილი პლაგინები ამ ძირითად პრინციპებს სრულებით ემორჩილებიან, მაგრამ პლაგინის დაწერა შესაძლებელია პროგრამირების მცოდნე ნებისმიერი პიროვნების მიერ და რამდენად მოექცევა ეს კონკრეტული დაწესებულ ჩარჩოებში ეს რა თქმა უნდა არა კონტროლირებადია.

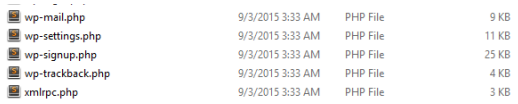
ამიტომ თუ პლაგინს ახლავს Readme.txt ფაილი აუცილებელია პლაგინის გამოყენებამდე დაწვირლებით გაარჩიოთ იგი დაწვრილებით.

დანამატის დაყენება ფაილური სტრუქტურიდან

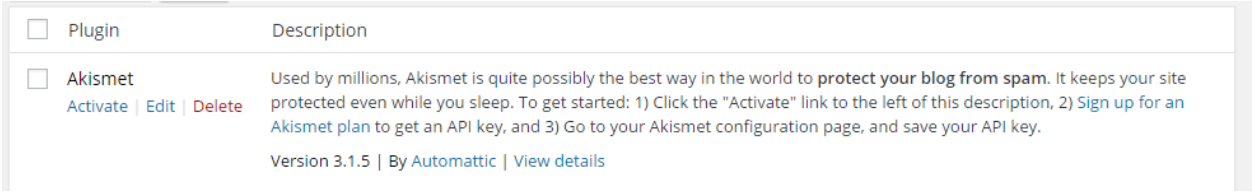
პლაგინების დაყენების პირველი გზა იმედია წარმატებით აითვისეთ. შესაძლებელია ასევე პლაგინის დაყენება უშუალოდ ფაილური სტრუქტურიდან. ამისათვის გვესაჭიროება პლაგინი. მისი მოძიება შესაძლებელია როგორც ოფიციალურ საიტზე ასევე ნებისმიერ საძიებო სისტემაში. ძირითად შემთხვევაში ფაილი იქნება .zip ფორმატში. საჭიროა მოხდეს ფაილის განარქივება. გადავიდეთ Xampp_ის საქაღალდე **htdocs > wp-content >plugins** და მოვახდინოთ პლაგინის საქაღალდის განთავსება. (იხ.სურ 11.7)მიმდინარე საქაღალდეში ჩანს ყველა ინსტალირებული პლაგინი (აქტიურიც და არააქტიურიც)



სურ.11.7 WordPress_ის ფაილური სტრუქტურა - პლაგინის განთავსება



დამატებული პლაგინი ადმინისტრირების პანელში არააქტივირებული იერსახით მოგევლინებათ ჩანართ **Plugins**_ში. მის აქტივაციისთვის მივმართოთ ნაცად ხერხს > პლაგინის აქტივაციას (იხ. სურ. 11.8)



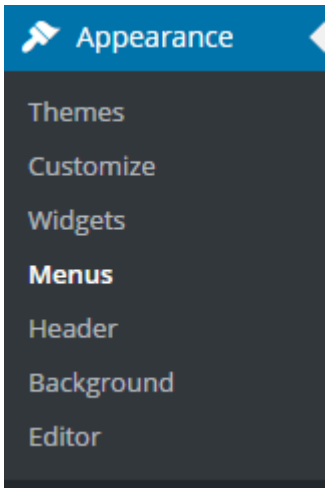
სურ.11.8 დეაქტივირებული პლაგინი

6.3 WEB გვერდის დიზაინის აწყობა კონკრეტული CMS-ის მოთხოვნების მიხედვით

მიმდინრე პარაგრაფის თემატიკა

- საიტის დიაზინის აწყობა კონკრეტული CMS-ის მოთხოვნების მიხედვით

WordPress_ის სამართავი მენიუს ერთ ერთ მნიშვნელოვან პუნქტს წარმოადგენს **Appearnce** ჩანართი (იხ.სურ. 12.1). ამ ბლოკის ქვეპუნქტების რაოდენობა და ნაირსახეობა დამოკიდებულია აქტიური თემის ფუნქციონალზე. **Themes - თემები** ველი უზრუნველყოფს მიმდინარე ეტაპზე არსებული თემების ვიზუალურ წარმოჩენასა და მართვას. მიმდინარე ჩანართიდან შესაძლებელია როგორც ახალი თემების მოძიება/ დაყენება ასევე არსებული თემებიდან არჩევა/ აქტივაცია. პლაგინების მსგავსად ძრავი შეიძლება შეიცავდეს თემების ერთობლიობას მაგრამ დანამატებისაგან განსხვავებით თემა აქტიურ მდგომარეობაში შეიძლება იყოს მხოლოდ ერთი. (პლაგინების შემთხვევაში ერთდროულად განუსაზღვრელი რაოდენობის დანამატის აქტივაციაა შესაძლებელი).



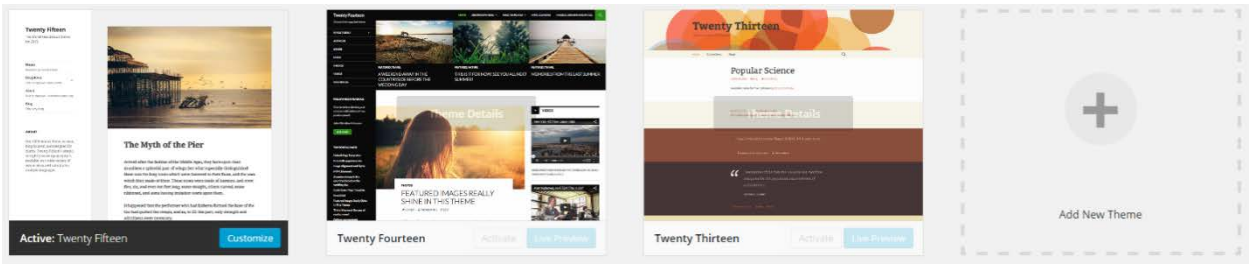
სურ.12.1 Apparance ჩანართი

WordPress_ის ერთ ერთი დამსახურება მრავალფეროვანი თემების დიდი არქივია. თემა პასუხისმგებელია საიტის ვიზუალურ გაფორმებაზე, ამიტომ დიდი სიზუსტითა და გემოვნებით უნდა მოხდეს საიტისთვის თემის მისადაგება.

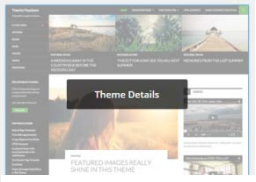
როგორც აღვნიშნეთ WordPress თემების საკმაოდ დიდ კოლექციას ფლობს. მარტო ის რად ღირს რომ ინსტალაციის დროს პირდაპირ 3 საწყის თემას სტავაზობს მომხმარებელს. რა თქმა უნდა ძირითად შემთხვევაში საწყისი თემებიდან მომხმარებელი არც ერთს ირჩევს (მათი სიმარტივის გამო), მაგრამ ეს სრულებით არ აკნინებს შემქმნელების მცდელობას შესთავაზონ მომხმარებლს მრავალფეროვანი დიზაინი.

თემების ჩამოწერა / ინსტალაცია შესაძლებელია როგორც საძიებო სისტემებიდან ასევე WordPress_ის ოფიციალური მისამართიდან https://wordpress.org/themes/ . შევეცადოთ სხვადასხვა გზებით განვახორციელოთ თემების მოძიება დაყენება.

Themes პუნქტის გააქტიურებით მიიღება სურათ 13.1 ნაჩვენები შედეგი, სადაც ასახულია CMS ყველა არსებული თემა. ამ ეტაპზე აქტიურია პირველი თემა. დანარჩენ თემებს აქვთ გადახედვის - **Live Preview** და აქტივაციის **Activate** ღილაკი. ასევე წინა პლანზე გამოტანილია თემის დასახელება.



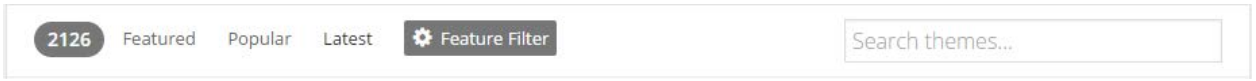
სურ. 13.1 საიტის თემები



სურ.13.2თემის დეტალური განხილვა

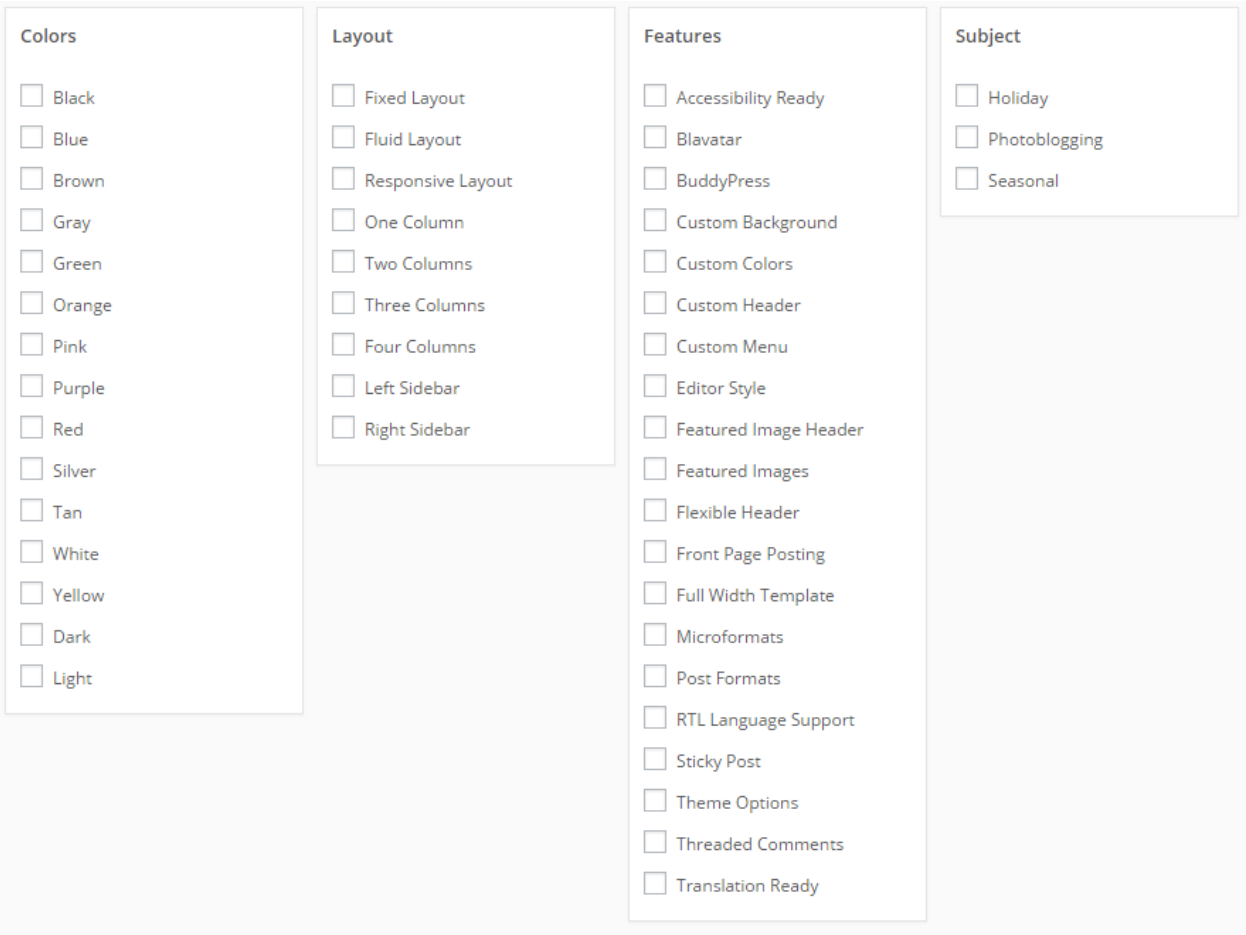
Theme details-წარმოაჩენს თემის დეტალური ინფორმაცია: თემის სახელს, ვერსიას, შემქმნელს, შაბლონის მოკლე აღწერას, გამოყენებულ ფერებთა გამას. აქვეა თემის წასაშლელი ბმული.

Add New Theme ღილაკის საშუალებით შესაძლებელია ახალი თემის მოძიება დაყენება. თემის საძიებო არეალი ფაქტობრივად იდენტურია პლაგინის ძიების ველისა მხოლოდ ერთი პატარა საკმაოდ მნიშვნელოვანი დეტალით გამოირჩევა. (იხ.სურ. 13.3).



სურ.13.4 ძიების ფილტრი

Feature Filter - გაფართოებული ფილტრი ძიებისთვის. აქ არის თემის ძიება კონკრეტული ფერების , საიტის სტრუქტურის, ფუნქციონალისა და კატეგორიის მიხედვით (იხ.სურ. 13.4)

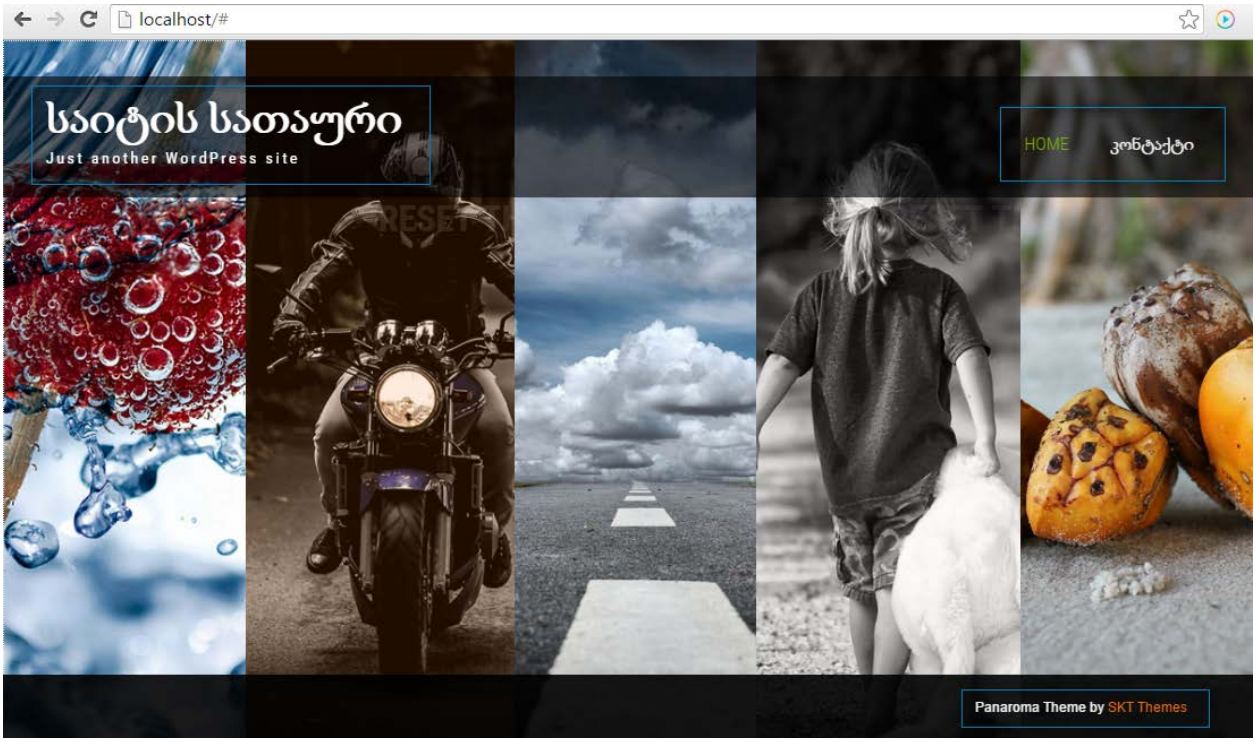


სურ. 13.4 თემის ძიების გაფართოებული შესაძლებლობები

სასურველი პარამეტრების მონიშვა უზრუნველყოფს თქვენი კატეგორიის თემის მიმსგავსებული ვერსიის მოძიებას. თემა შეარჩიეთ იმ ბლოკური სტრუქტურისა როგორი სტილის საიტიც დაგჭირდებათ.

არსებობს თემის დაყენების სხვა გზაც. ისევე როგორც პლაგინების შემთხვევაში შესაძლებელი თემაც მოძიებულ იქნას ალტერნატიული წყაროდან , საერთოდაც არ იყოს ინტრენეტ სივრცეში განთავსებული (ვინმეს მიერ იყოს ხელით შექმნილი), ან ნებისმიერი საძიებო სისტემიდან მოვახდინოთ მისი ჩამოწერა. ამ შემთხვევაში მივმართავთ გზას - ახალი თემის დამატების ბმულის აქტივაცია **Add New > Upload Theme** და თემის ფაილის მითითება. შესაძლებელია ასევე ამოარქივებული თემის საქაღალდის განთავსება WordPress_ის ფაილური სტრუქტურის **wp-content > themes** არეალში. Xampp_ის შემთხვევაში WordPress დაყენებულია **htdocs** საქაღალდეში და წინამდებარე მისამართის გავლა სწორედაც ამ კონკრეტულიდან უნდა მოხდეს.

სატესტო რეჟიმში მოძიებულ იქნა ვიზუალურად განსხვავებული თემა სახელწოდებით ‘Panaroma’. მისი აქტივაციის შემდგომ საიტის მთავარი გვერდი იყებს სახეცვლილ იერს. სურათ 13.5 მოცემულია Panaroma თემის გააქტიურების შედეგად მიღებული ვებ გვერდი.



სურ. 13.5 ერთ ერთი თემის აქტივაციის შედეგად საიტის დიზაინის ცვლილება

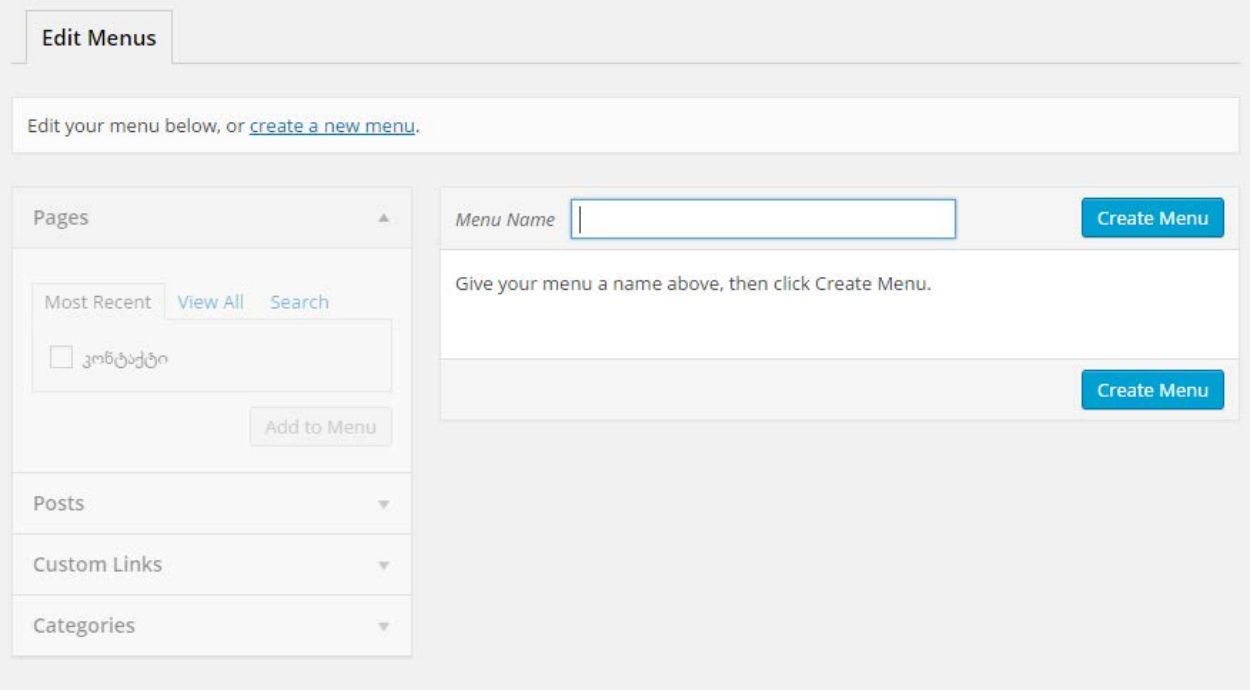
ეს კონკრეტული თემა პიველ გვერდზე გამოსახავს 3 ბლოკს: საიტის სათაურს, მენიუს და ინფორმაციას სატის შემქმნელის შესახებ. ასევე მთავარი შემცველობა დაყოფილია და წარმოჩენელია ბლოკების სახით.

მიმდინარე თემა გამოყენებულ იქნა იმისათვის რომ წარმომეჩინა თუ რამდენად რადიკალურად შეიძლება შიეცვალოს არსებული თემა ერთი მარტივი ღილაკის მოქმედებაში მოყვანის შემდგომ. ამასთან მიმდინარე თემა არ ზღუდანს რამოდენი მნიშვნელოვანი პარამეტრების მართვას.

როგორც წინამდებარე აბზაცებში აღვნიშნეთ თემის ცვლილება იწვევს WordPress_ის ადმინისტრირების პანელის მენიუ **Appearance** ჩანართის ქვეპუნქტების ცვლილებასაც. მიმდინარე თემას აქვს წვდომა რამოდენიმე მნიშვნელოვან ჩანართთან როგორებიცაა **Menus** და **Widgets.** ეს ორი ჩანართი შესანიშნავი საშუალებაა მარტივი გზებით საიტის ფუნქციონალის გასაზრდელად.

Appearance > Menus - პუნქტი საშუალებას იძლევა იმართოს საიტზე არსებული მენიუები. ავტომატურ რეჟიმში ყოველი დამატებული გვერდი (**Pages** ჩანართიდან) ვებ გვერდზე მენიუს პუნქტად ემატება. ეს რა თქმა უნდა ძირითად შემთხვევებში არახელსაყრელია. შესაძლოა საიტზე არსებობდეს გვერდები რომელთა დანიშნულება საერთოდ არ არის მენიუს პუნქტად გამოჩნდეს, არამედ ისინი გაერთიანებულნი არიან „ზოგადი გვერდის“ ცნების ქვეშ. ეს ყოველივე ნიშნავს იმას რომ არსებობს გვერდი და მისი გაცნობა ხდება სხვა გვერდიდან ან გვერდის ინფორმაციიდან ბმულით გადამისამათების შემდეგ.

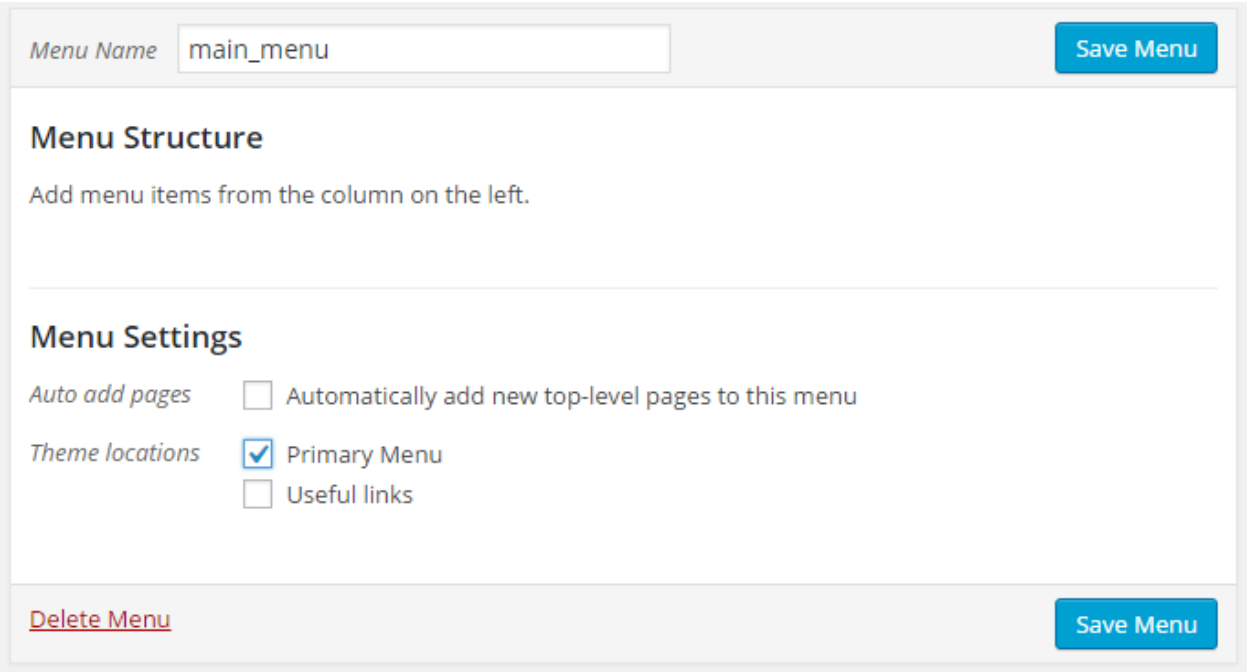
Appearance > Menus ხელსაყრელია როგორც მენიუთა კლასიფიკაციის, ასევე სასურველი სანავიგაციო ბმულებისაგან ახალი მენიუს შესაქმნელად. სურ. 13.6 ნაჩვენებია ახლი მენიუს შესაქმნელი მოცემულობა.



სურ 13.6 მენიუების სამართავი სივრცე

სურათის 13.6 მოცემულობიდან გამომდინარე საიტზე არ არის არც ერთი მენიუ და მოქმედებს ჩვენს მიერ უკვე ახსნილი მეთოდი ყველა შედგომი დამატებული გევრდი ავტომატურ რეჟიმში იქცევა მენიუს პუნქტად.

Create Menu საშუალებით შესაძლებელია ახალი მენიუს დამატება. ახალი მენიუ **Menu Name** ველში შეყვანილი სახელწოდების იქნება. (იხ სურ 13.6.1)



სურ.13.6.1 მენიუს შექმნა

მენიუს სახელწოდება უნდა შედგებოდეს ლათინური ანბანის ასოთა თანწყობისაგან, სახელი უნდა წარმოადგენდეს ერთ მთლიანობას.დასაშვებია ორსიტყვიანი სახელის დარქმევისას სიტყვათა შორის ბმად გამოყენებულ იქნას ქვედატირე. უმჯობესია თუ სახელი ლოგიკურად იქნება შერჩეული.

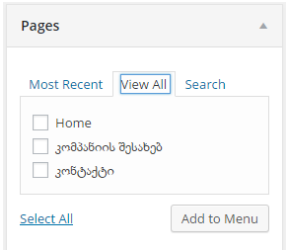
თუ მენიუს მოვუნიშნავთ პუნქტ **Primary Menu** მენიუ ავტომატურად გადაიქცევა მთავარ მენიუდ და ის გამოისახება იქ სადაც თემაში ნაგულისხმევია ძირითადი მენიუს გამოტანა. მიმდინარე თემაში

მარცხენა ზედა არეალში (იხ. სურ 13.5) პირველი ჩანართის - **Automatically add new top-lavel pages-to this menu** დამოწმება ყველა ზედა დონის გვერდს ავტომატურად დაამატებს მენიუს პუნქტად.

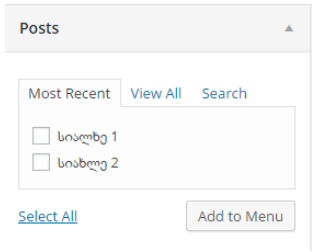
ახალი მენიუს დამატების დასასრულებად საჭიროა მენიუს შევსება ბმულებით და დამოწმება **Save Menu** ბრძანების გამოყენება.

მენიუს პუნქტებად ინფორმაციის ჩამატება შესაძლებელია მარჯვენა ბლოკიდან სადაც გამოტანილია სხვადასხვა ერთეულები. მაგ.: **Pages, Posts, Custom Links, Categories.** ამ ბლოკებში გამოდის ინფორმაცია რომელაც შეიცავენ უშუალოდ მიმდინარე ჩანართები. მიმდინარე ეტაპზე:

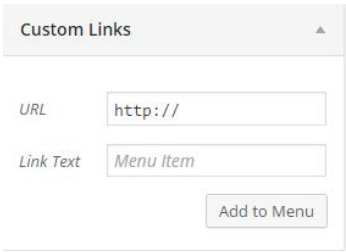
- **Pages** - შეიცავს შემდეგ გვერდებს Home, კომპანიის შესახებ და კონტაქტი; (იხ.სურ. 13.7.1)
- **Posts -** სიახლე1 , სიახლე2 ; (იხ.სურ. 13.7.2)
- **Custom Links -** ზოგადი ბმულები (ამ ჩანართს დაწვრილებით განვიხილავთ **)** ; (იხ.სურ. 13.7.3)
- **Categories** - სიახლეები, სპორტი (იხ.სურ. 13.7.4)



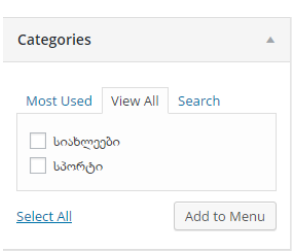
სურ.13.7.1



სურ.13.7.2



სურ.13.7.3

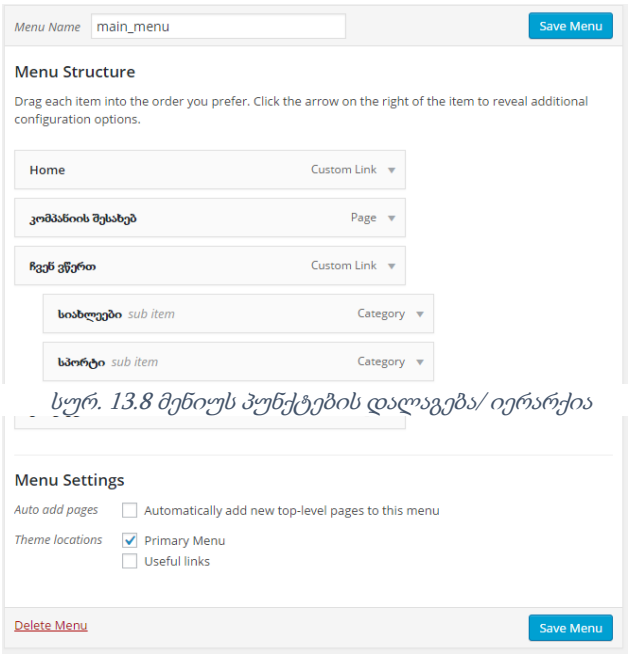


სურ.13.7.4

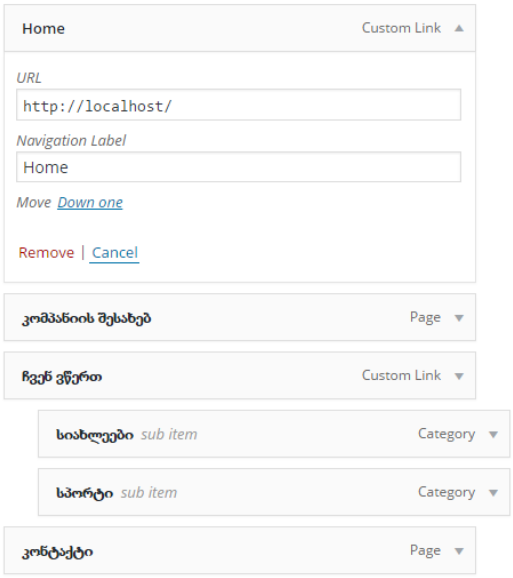
საკმარია ნებისმიერი ჩანართების მონიშვნა **Add Menu** და ის უკვე წაროადგენს მენიუს პოტენციურ პუნქტს.

რაც შეეხება **Custom Links** ბლოკს აქ მცირეოდენ განსხვავებული სიტუაციაა. პირველ ველ URL -ში მიეთითება ის მისამართი თუ სად უნდა გადავიდეს მენიუს ეს პუნქტი, ხოლო მეორეში სახელწოდება. მაგ.: საიტის მენიუს ერთ ერთ პუნქტზე ხელის მიტანისას ამ პუნქტს არ აქვს პუნქცია ახალი გვერდის გახსნის მას ევალება ქვე პუნქტების ჩამოშლა. **URL** მისამართში გაიწერება # (html_იდან უნდა გახსოვდეთ თუ რას ნიშნავს) ხოლო **Link Text**-ში მენიუს პუნქტის სახელწოდება.

WordPress_ში მენიუს პუნქტების სურვილისამებრ დალაგება ძალიან მარტივი პროცესია. მხოლოდ მაუსის ჩაკიდებით და გადატანთ შესაძლებელია რიგითობის მინიჭება. იერარქიაც საკმაოდ მარტივადაა გადაწყვეტილი მაუსის კურსორის მოკიდვითა და პუნქტის ნაბიჯებით შეწევა იწვევს იერარქიის წარმოქმანს. (იხ.სურ13.8)



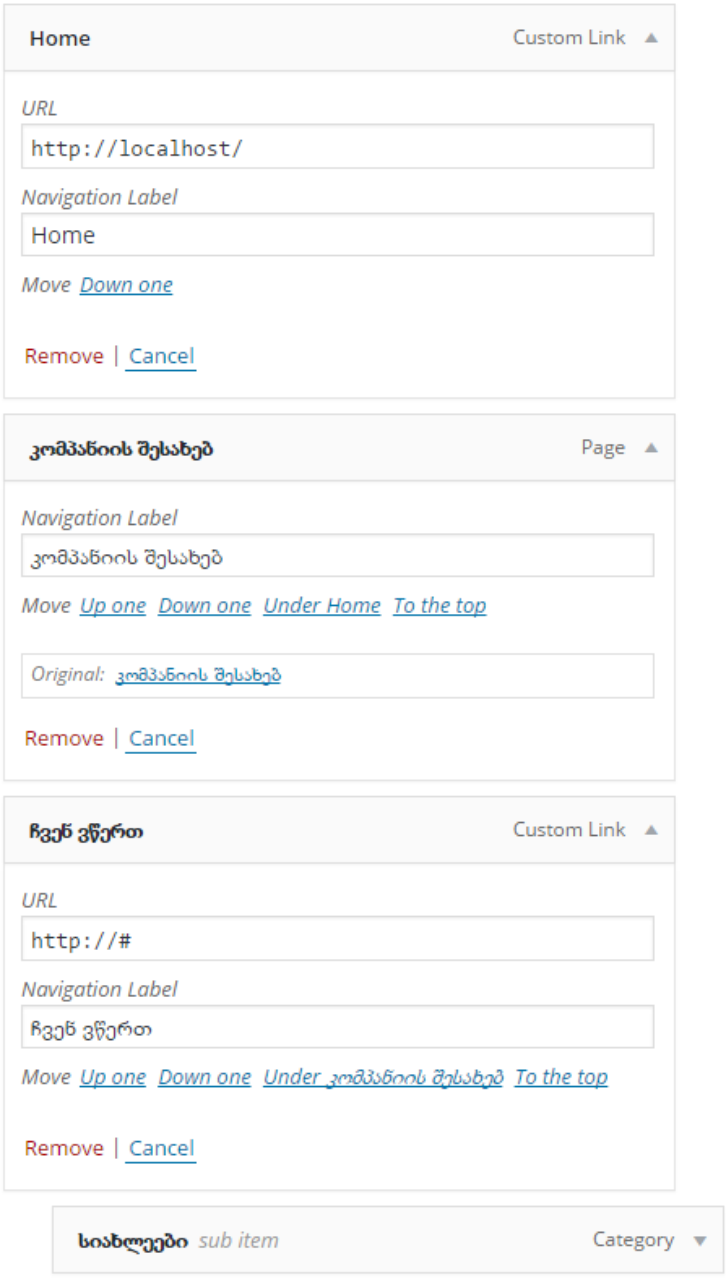
სურ. 13.8 მენიუს პუნქტების დალაგება/ იერარქია



სურ.13.9 მენიუს პუნქტის პარამეტრები

შესაძლებელია უკვე დამახსოვრებულ მენიუს პუნქტებში ცვლილებების შეტანა, ამისათვის საკმარია საჭირო პუნქტზე მაუსის დაკლიკვა. პუნქტი გადავა ცვლილების რეჟიმზე (იხ.სურ. 13.9)მაგ.: home-მთავარი გვერდი ავტომატურად ლათინურად არის მოხსენიებული. მთავარ ან საწყის გვერდად მისი მოხსენიებისთვი საჭიროა ვიმოქმედოთ მასზე და ჩავასწოროთ **Navigation Label**. ასევე შეიძლება მისი URL მისამართის ცვლილება. უმჯობესია საწყისი გვერდის URL_ს არ შეეხეთ. მიმდინარე პარამეტრებიდან შესაძლებელია არასასურველი პუნქტის წაშლა **Remove** ბრძანებით.

WordPress მენიუს პუნქტების URL მისამართის ცვლილების შესაძლებლობას მხოლოდ მთავარ გვერდთან მიმართებაში (Home) და **Custom Links** ბლოკიდან დამატებული მენიუს ველის შემთხვევაში გავძლევს. გვერდების, პოსტების და კატეგორიების შემთხვევაში მიმდინარე პარამეტრი დამალულ მდგომარეობაშია, რადგანაც აზრს კარგავს ძირითადი მიდგომა, თუ კონკრეტულ გვერდზე გვინდა გადასვლა საჭიროა ამ გვერდის მისამართი იყოს უცვლელი თუ ხელით შევეცდებით რაიმეს შეტანას მაშინ ის გვერდს კი არა ზოგად ლინკს (Custom Links) წარმოადგენს



სურ.13.10 მენიუს პუნქტების პარამეტრები- ვრცლად

13.11_ზე მოყვანილია განხილული თემის მაგალითი.

სურათ 13.10 ზე ნათალად არის ნაჩვენები თითოეული პუნქტის პარამეტრებთან წვდომის შესაძლებლობა. მაგალითის მიხედვით კარგად ჩანს რომ სახელწოდების ცვლილება ნებისმიერ პუნქტზეა შესაძლებელი. URL_ს მხოლოდ მთავარი და ზოგადი ლინკების შემთხვევაში ვეხებით.

საკმაოდ საინტერესოა შემდეგი ფაქტიც რომ შესაძლოა გვერდს სხვა სათაური გააჩნდეს და მენიუში სხვა სახლით გამოვიტანოთ. **Navigation Label** ველი მხოლოდ მენიუს პუნქტის სახლეს ცვლის და არ ეხება ძირითადი გვერდის სათაურს. და რაში გვჭირდება სხვადასხვა სახელი?

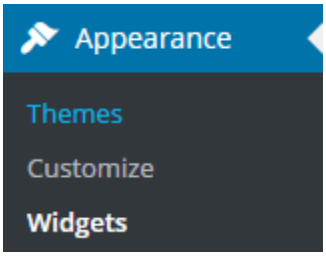
მაგ.: მენიუს პუნქტი **„ ჩვენ შესახებ „** საკმაოდ გავრცელებულ ბმულს წარმოადგენს, დამკვეთების უმრავლესობა ითხოვს ამ ტექტსის გამცენას ძირითად მენიუში. მაგრამ უშუალოდ ბმულზე გადასვლისას - გვერდის დასახელებად უპირატესობას ანიჭებენ არა **„ჩვენ შესახებ“** არამედ ტექსტ **„კომპანიის შესახებ“.** ამ შემთხვევაში მენიუს ფორმირებისას გვერდ „კომპანიის შესახებ“ Navigation Label ველის საშუალებით შევუცვლით სახელს და დავარქმევთ „ჩვენ შესახებ“ (რა თქმა უნდა ჩვენს მიერ არწერილი შედეგის მისარებად ეს ერთადერთი გზა არაა, მაგრამ სიმარტივის თვალსაზრისით არაფერი დაეწუნება.) . სურათ

სურ. 13.11 გვერდის სათაურის ცვლილება მენიუს პუნქტში Navigation Label _ის მეშვეობით

მენიუს ფორმირების და პუნქტების პარამეტრების რეგულირების შემდეგ ვებ გვერდზე გამოდის სურათ 13.12 ნაჩვენები მენიუ. როგორც ატყობთ პუნქტები იმ თანმიმდევრობითაა დალაგებული როგორც ადმინისტრირების პანელში მივუთითეთ, ასევე Custom Link_ის მეშვეობით შექმნილი ბმული **„ჩვენ ვწერთ“** ასრულებს მხოლოდ ზოგად მოვალეობას და ის არ მისამართდება არსად, მას ავალია ქვეკატეგორიებს შემცველობა(სიახლეები, სპორტი).

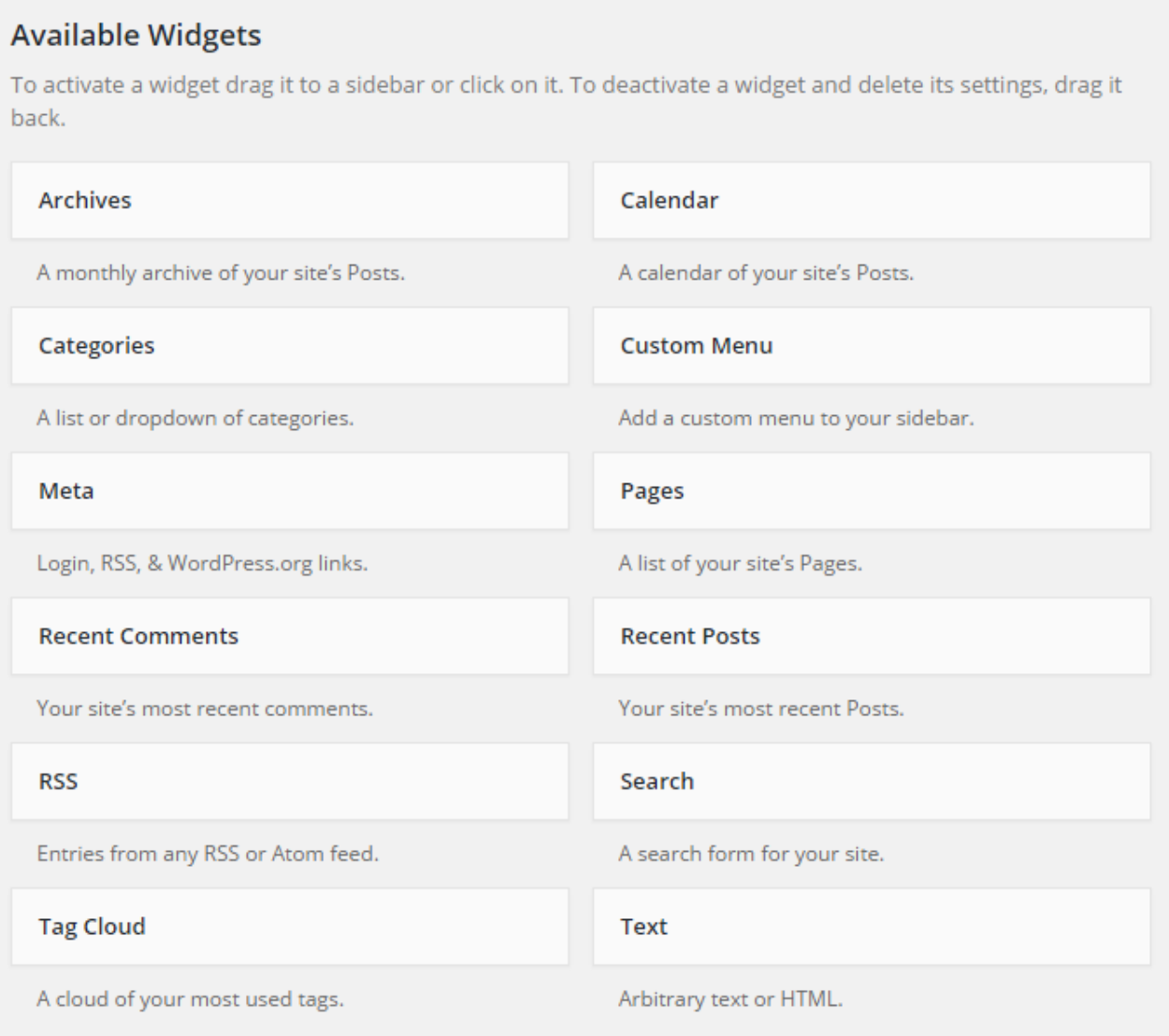
სურ. 13.12 ადმინ პანელში შექმნი მენიუს ვიზუალიზაცია საიტის მთავარ გვერდზე

ძირითადი მენიუს დამატების შემდეგ შეეცადეთ და დაამატეთ მორიგი მენიუ რომელსაც მოგვიანებით გამოვიყენებთ. უბრალოდ მიმდინარე ეტაპზე ის არსად აისახება. ჩემს შემთხვევაში ეს იქნაბ **testmenu ,** რომელიც შეიცავს ორ პუნქტს **სიახლეები** და **კონტაქტი.** (ამ მენიუს მოგვიანებით დავუბრუნდებით)



Appearance > Widgets ბლოკი ერთ ერთი მნიშვნელოვანი ჩანართია WordPress_ის თემის შესაძლებლობების გაფართოებისათვის. ზოგადად ვიჯეტები წარმოადგენს რაიმე პროგრამული კოდის - ფუნქციონალის ერთობლიობას.

ვიჯეტების არეალის აქტივაციისას ნათლად დაინახავთ უკვე არსებულ (default) აქტიურ პროგრამულ ჩანართებს. (იხ.სურ.14.1) შესაძლებელია ისეთი პლაგინების მოძიება დაყენება რომელიც გაამდიდრებს widgets ბლოკს და მას უფრო მეტ დატვირთვას შესძენს ეს თავისთავად გამოიწვევს საიტის დატვირთავს სასარგებლო პროგრამული დანამატებით.

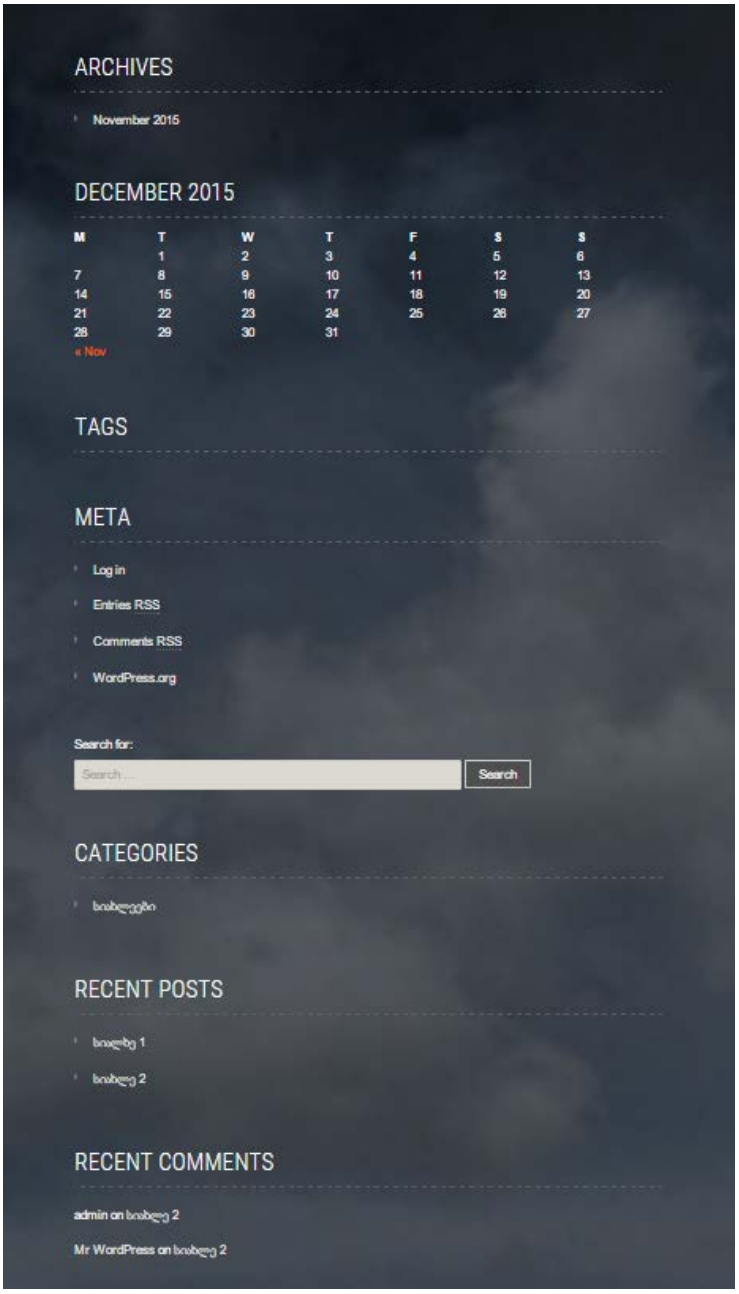


სურ. 14.1 default ვიჯედები

საწყის ვიჯეტების აქტივაციის შედეგად საიტზე მიღებული შედეგი :

- **Archives** - გამოისახება ჩანაწერების არქივი დახარისხებული თვეების მიხედვთ;
- **Calendar** -გამოვა კალენდარი, რომელიც შეიცავს ინფორმაციას ჩანაწერებზე, კონკრეტული დღის ბმულზე გადასვლისას გააქტიურდება მიმდინარე დღის სიალხეები.
- **Categories** -არსებული კატეგორიები;
- **Custom Menu** - ზოგადი მენიუ, სწორედ ამ პუნქტის გააქტიურების შემდეგ შესაძლებელაი ჩვენს მიერ შექმნილი სატესტო მენიუს **testmenu** გამოტანა **sidebar** ბლოკში;
- **Meta** – ადმინისტრირების პანელში, rss ლენტაზე, WordPress ოფიციალურ საიტზე შესასვლელი ბმულების ერთობლიობა
- **Pages** - გამოდის გვერდების სათაურები , შესაძლებელია ყველას ერთდროულად ან შერჩევით გვერდების გამოტანა.
- **Recent Comments** - უკანასკნელი კომენტარები;
- **Recent Posts** -უკანასკნელი ჩანაწერები / პოსტები;
- **Rss** -აისახება სიახლეების გამოწერის არხი (საწყის თავებში დეტალურადა აახსნილი RSS_ ის მნიშვნელობა)
- **Search** -საძიებო ფორმა;

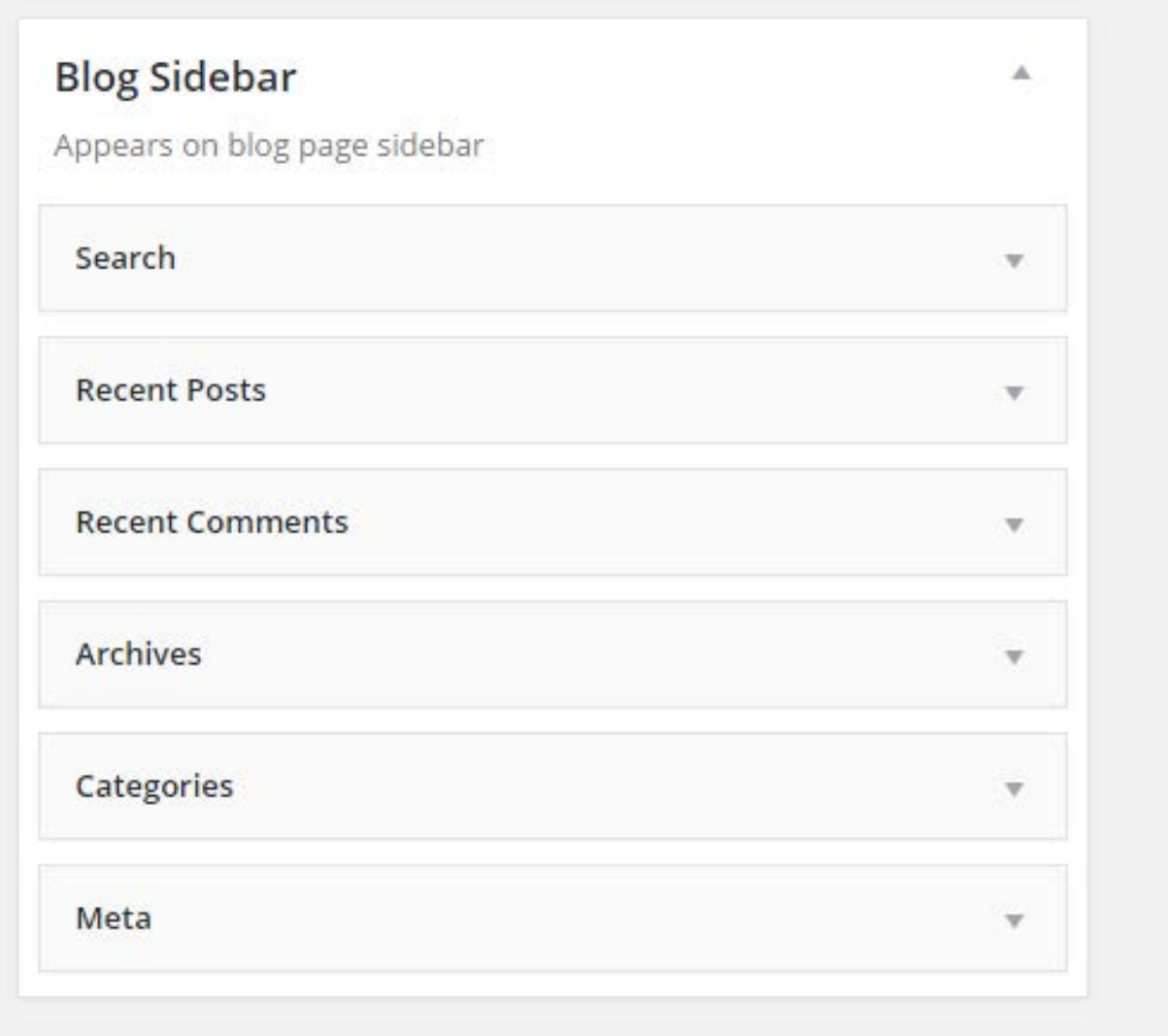
- **Tag Cloud** -ტეგების ღრუბელი. მის მისაღებად საჭიროა საკვანძო სიტყვები Tag_ები გამოყენებული იყოს ჩანაწერებში / პოსტებში. წინააღმდეგ შემთხვევასი არანაირირი ღრუბელი არ გამოისახება
- **Text** - სტანდარტული ტექსტუალური ინფორმაციის გამოტანა, მაგ.: ტელეფონის ნომრის



სურ 14.2 აქტივირებული ვიჯეტები საიტის Sidebar_ში

გააქტიურებული დანამატების ერთობლიობა საიტის შემცველობაში შემდეგნაირად გამოისახება. ჩვენ შემთხვევაში მოცემულ მაგალითზე ნაჩვენებია გააქტიურებული არქივი, კალენდარი, ტეგების ღტულები რომელიც ცარიელია და შესაბამიად არასრულყოფილადაა ასახული, მეტა ბმულები, საძიებო ფორმა, კატეგორიები, უკანასკნელი პოსტები და უკანასკნელი კომენტარები. მათი პარამეტრების ცვლილება (ხილული რაოდენობები, სათაურები და ა.შ) შესაძლებელია ადმინ პანელის Widget ჩანართიდან (იხ.სურ. 14.2)

სურათ 14.1 ზე მოცემული ველები იძლევა საშუალებას კონკრეტული მოდულები გამოყენებულ იქნას საიტის გარკვეულ სივრცეში. თითოეული თემა ინდივიდუალურად განსაზთვრავს თუ რამდენი მართვადი ბლოკი შეიძლება გააჩნდეს მას და ასევე თუ სად იყოს ისინი განთავსებული ავტომატურ რეჟიმში. თუ იმ მოსაზრებას უგულველბელყოფთ, რომ შესაძლებელია დიზაინის სტრუქტურაში ჩარევა და ბლოკების გადაადგილება, მაშინ ვრჩებით მხოლოდ დაყენებული თემის შემოთავაზებული სივრცეების იმედად. ვიჯეტეის არეალის მარჯვენა მხარეს გამოტანილია მსგავსი არეები. მიმდინარე თემა გთავაზობთ ერთ ბლოკს **Blog Sidebar,** რომელიც საწყის ეტაპზე შევსებულია სხვადასხვა ვიჯეტებით. (იხ.სურ.14.3)



სურ.14.3 ზოგადი ინფორმაციის არეალი

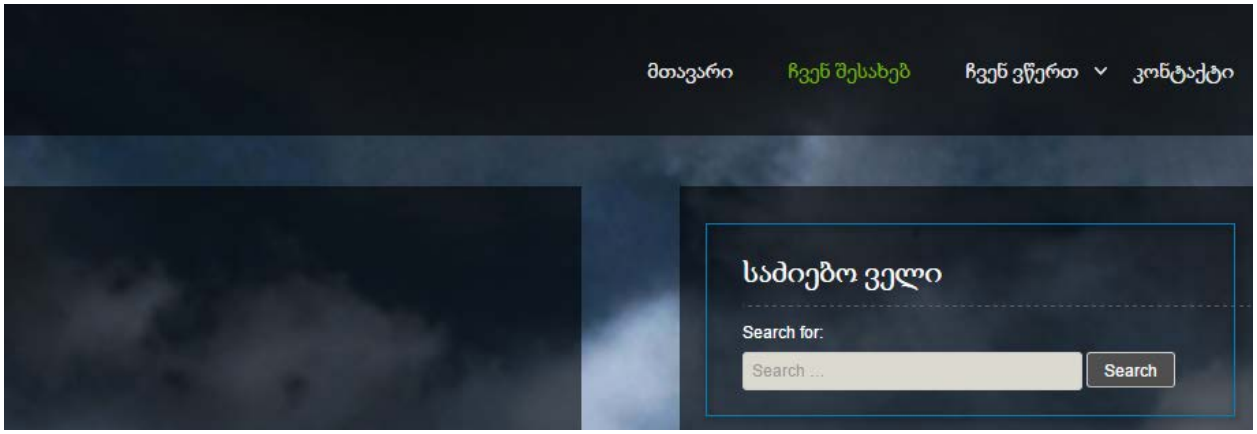
ვიჯეტების დამატება **Sidebar** ბლოკში საკმაოდ მარტივი პროცესია და მაუსის მარტივი მოძრაობით ჩაავლე-გადაიტანე ხორციელდება.

Sidebar -წარმოადგენს დამატებით არეალს - ბლოკს გამოყოფილს სხვადახვა სახის ინფორმაციის გამოსატანად . ძირითად შემთხვევებში საიტს შესაძლოა გააჩნდეს მარცხენა, მარჯვენა Sidebar_ები , როგორც ცალ ცალკე ასევე ორივე ერთად . არსებობს შესაძლებლობა საიტს ქონდეს ქვედა Sidebar. ეს ყოველივე გამომდინარეობს საიტის სტრუქტურიდან გამომდინარე.

რაც შეეხება ვიჯეტების პარამეტრების კონფიგურაცია , მათი გააქტიურების - მათზე მაუსის დაკლიკების შედემდეგ არის შესაძლებელი. გამომდინარე იქიდან რომ ვიჯეტები განსხვავებულ მოქმედებებ ახორციელებნ მათი პარამეტრები ერთმანეთისაგან განსხვავებულია და განხილვა არაფერს მოგცემს იმდენად ინდივიდუალური მიდგომა სჭირდება თითოეულ მათგანს. საეთო რაც გააჩნია ძირითად შემთხვევაში ეს სასათაურე ველია **Title** , რომლის მეშვეობითაც ხდება ვიჯეტის დასათაურება და ისე წარმოჩენა საიტზე. (იხ.სურ.14.3.1) ასევე შესაძლებელია **Delete** ღილაკის გამოყენებით ვიჯეტის Sidebar ბლოკიდან ამოშლა.

სურ.14.3.1 საძიებო ველის ვიჯეტის დასათაურება

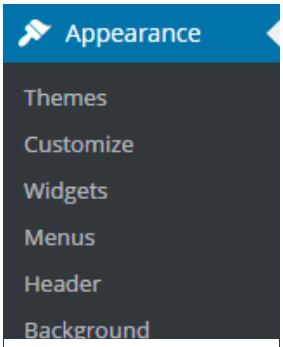
სურათ 14.3.1 მოცემულობიდან გამომდინარე საიტზე გამოსული ძიების ფორმა დასათაურებული იქნება ტექსტით - **"საძიებო ველი"** (იხ.სურ.14.4)



სურ.14.4 საძიებო ველის სათაურის ცვლილება Widgets Title_ს მეშვეობით

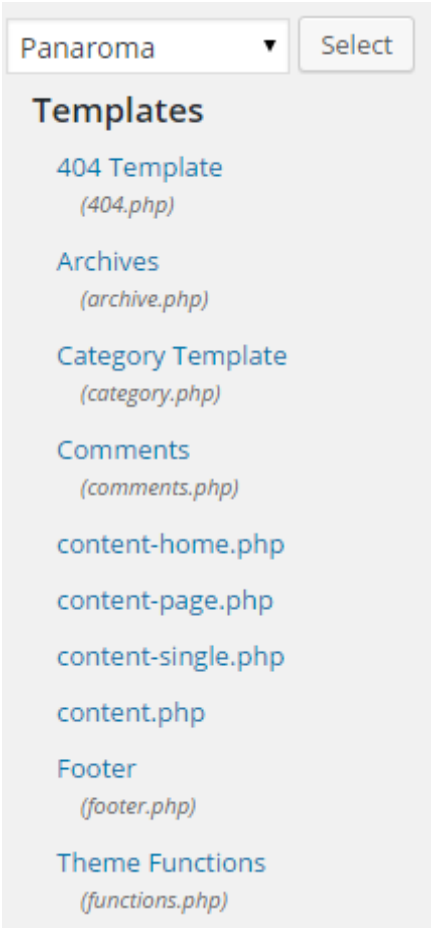
საიტის ინფორმაციული სტრუქტურის სრულყოფის შემდგომ შესაძლებელია მოინდომოთ საიტის დიზაინის ცვლილება, რაც ძირითად შემთხვევაში შესაძლოა ფონებისა, ფერების, ფონტის ტიპების ცვლასთან იყოს დაკავშირებული. ძირითადი ბლოკური ბადის ცვლილება მზა თემაში საკმაოდ შრომატევადია და დიდი ალბათობით არც ღირს წვალებად., როგორც საწყის ეტაპზე აღვნიშნეთ - „ მოიძიეთ სასურველი კატეგორიის და სტრუქტურის თემა“ თქვენს მიერ შერჩეული საიტის ბლოკური სქემა უკვე უნდა ესატყვისებოდეს პირვანდელ მოთხოვნილებებს.

ადმინისტრირების პანელიდან თემის ფაილურ სტრუქტურასტან წვდომისათვის გამოიყენება **Apprearance > Editor** ჩანართი (იხ.სურ.15.1)

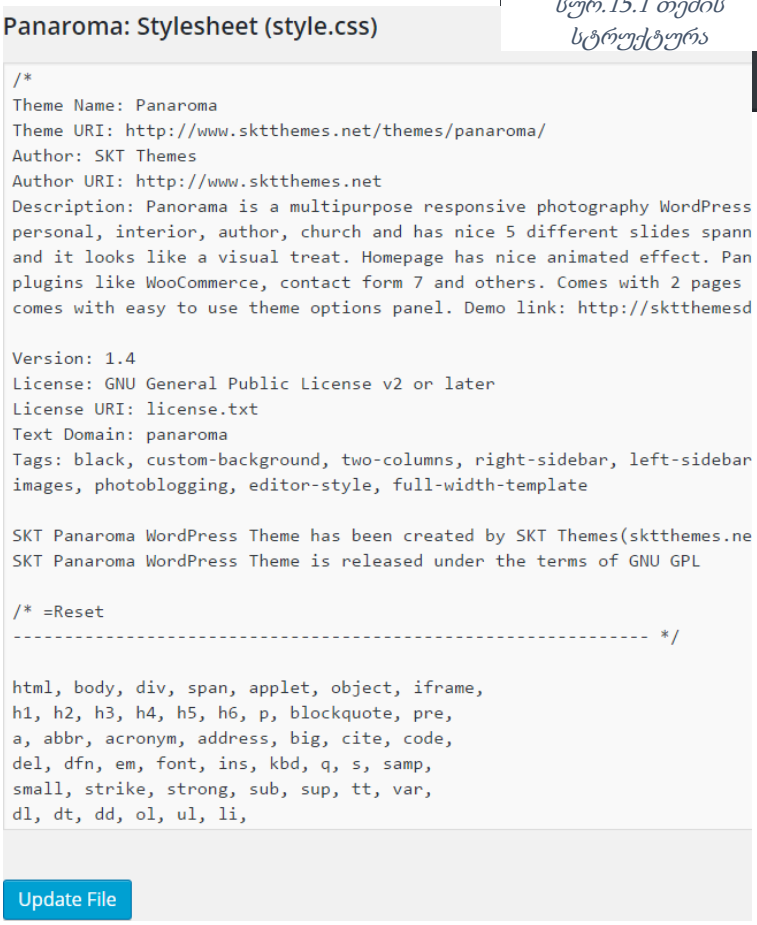


სურ.15.1 თემის სტრუქტურა

მიმდინარე ბმულის გააქტირუების შედეგად შესაძლებელია აქტიური თემის ფაილებთან ურთიერთობა. (იხ.სურ. 15.2) ეს გზა საკმაოდ ხელსაყრელია ჰოსტზე განთავსენული საიტის შემთხვევაში , რადგანაც ფაილებთან ურთიერთობისათვის არ დაგჭირდებათ დადამატებითი რედაქტორებისა და FTP ანგარიშის მონაცემების ცოდნა.



სურ 15.2 თემის ფაილები



სურ 15.3 რედაქტირების არეალი

Apprearance > Editor ჩანართის გამოძახება გამოიტანს აქტიური თემის ფაილებს. ძირითად რედაქტირების ველში (იხ.სურ.15.3) საწყის ფაილად გახსნილია style.css სტილების ფაილი. სტილების ფაილის ზედა არეალი დათმობილია თემის სხვადასხვა ინფორმაციის შესანახად. ის კომენტარის სახით არის გამოყოფილი და უმჯობესია თავი შეიკავოთ მისი წაშლისაგან თუ გსურთ თემის მიმდინარე ვიზუალით ისარგებლოთ. მიმდინარე ბლოკში განთავსებული ინფორმაცია ემსახურება თემის სახელის, მწარმოებლის, ვერსიის, მოკლე აღწერისა და ა.შ შენახვას, რაც შემდგომში გამოისახება თემის მოძიება/ დაყენების პროცესში თემის დეტალური გარჩევის ჩანართში (**Theme details** იხ.სურ **13.2**).

თუ განიზრახავთ თემის სტილების კორექტირებას აუცილებელია Style.css ფაილში დაიტანოთ ცვლილებები. თემები ხშირად განიცდიან განახლების პროცესს რაც იწვევს თემის ფაილების სრულ განახლებასაც. თქვენს მიერ style.css ფაილში დატანილი ცვლილებები შესაძლოა ფუჭი აღმოჩნდეს და გარკვეული პერიოდის შემდეგ საიტის თემამ დაიბრუნოს პირველადი სახე. (თუ თემის ვერსია განახლდა).

ეს რა თქმა უნდა არ შედის თქვენს ინტერესებში ამიტომ სასურველია საწყის ეტაპზევე იქნას თადარიგი დაჭერილი. არსებობს რამოდენიმე გზა თემის განახლების თავიდან ასარიდებლად:

1. თემას გადაერქვას სახელი - style.css ფაილში პირველივე ხაზზე წარმოდგენილია თემის დასახელება - **Theme Name.** მოახდინეთ თემის სახელის გადარქმევა. (იხ.სურ. 15.4)
2. შესაძლოა მხოლოდ პირველი პროცედურს წარმატებით შესრულების შემთხვევაში არ მიიღოთ სასურველი შედეგი. ურიანი იქნებოდა **Theme URI, Author** და **Author URI** ველების ცვლილება.(იხ.სურ. 15.4)
3. ყოველი განახლების შემდეგ იცვლება თემის ვერსია. წამოჭრილი პრობლების გადასაწყვეტად ზედა 2 პუნქტთან ერთად მიზანშეწონილია თემის ვერსიის ცვლილებაც. მაგ.: მიმდინარე ეტაპზე თემისვერსია **Version: 1.4,** შესაძლებელია ვერსიის ისეთი დიდი მნიშვნელობის გაწერა რომელსაც ფაქტობრივად ვერასდროს მიაღწევს მიმდნარე თემა, ვთქვათ **140**. (იხ.სურ. 15.4)
4. არსებობს პლაგინების გამოყენების შესაძლებლობა ამ პრობლების გადასაჭრელად.

```
/*
Theme Name: Panaroma
Theme URI: http://www.sktthemes.net/themes/panaroma/
Author: SKT Themes
Author URI: http://www.sktthemes.net
Description: Panorama is a multipurpose responsive photography WordPress theme suitable for c
different slides spanning the whole homepage. One can showcase whatever they want on the home
pages with blog, pages, compatibility with plugins like WooCommerce, contact form 7 and other
easy to use theme options panel. Demo link: http://sktthemesdemo.net/panoramapro/

Version: 1.4
License: GNU General Public License v2 or later
License URI: license.txt
Text Domain: panaroma
Tags: black, custom-background, two-columns, right-sidebar, left-sidebar, responsive-layout,
template
```

სურ. 15.4 თემის სტილური ფაილის პარამეტრების ცვლილება

ლოკალურ სერვერზე არსებული WordPress_ის შემთხვევაში (ჩვენ შემთხვევაში) ფაილურ სტრუქტურასთან წვდმა შესაძლებელეია აგრეთვე სერვერის ძირეული საქაღალდიდან (**htdocs) > wp-content > themes > აქტივირებული თემის საქაღალდის გამოძახება.** თქვენს წინაშეა თემის ფაილურის სტქუქტურა. Style.css_უზრუნველყოფს თემის სტილებთან წვდომას.

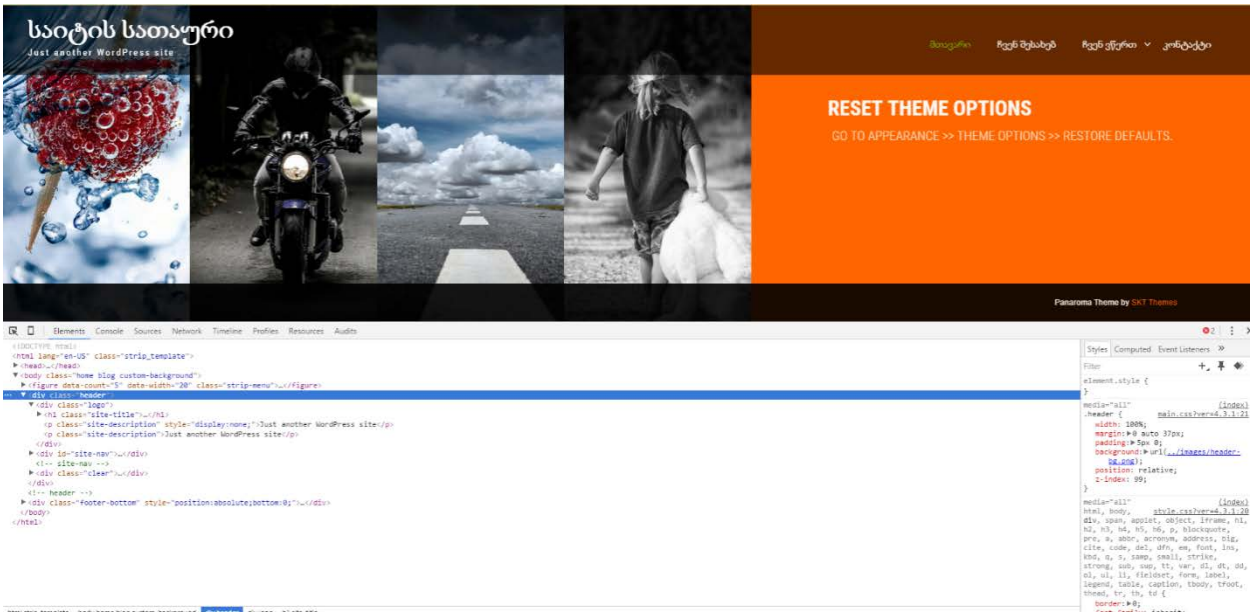
თემის დანარჩენ ფაილებს მოგვბიანებით - შემდეგ ქვეთავში გაეცნობით. სადაც დეტალურად იქნება განხილული ახალი თემის შექმა. ეს რა თქმა უნდა საშუალებას მოგცემთ შექმნათ როგორც საკუთარი თემა, ასევე დაარედაქტიროდ უფრო ვრცელი შესაძლებლობებით უკვე არსებული/მოძიებული/დაყენებული თემა.

ამ ეტაპზე მხოლოდ მარტივ სტილურ ცვლილებებს შვეეხოთ. მაგ. დავალება აზრი მდგომარეობს შემდეგში: შეცვალეთ მიმდინარე თემის ზედა არეალის ფონისა და ფონტების ფერები, ფონტის ზომები, ჩამოსაშლელი მენიუს ფონის ფერი. მივიღოთ სურათზე 16.1 ნაჩვენები შედეგი. გაითვალისწინეთ დავალების შესასრულებლად დამაკმაყოფილებელია html- css ცოდნა და სრულიად არ მოგეთხოვებად რაიმე პროგრამირების ენის ცოდნა.

სურ 16.1 დავალების შედეგად მიღებული შედეგი

გავყვეთ ეტაპობრივად.

საწყის ეტაპზე საჭიროა გამოავლინოთ იმ ბლოკების მიმმართველები/ სელექტორები (id ან class), რომელთა სტილებიც უნდა ცვალოთ. Inspect element - ის მეშვეობით (F12) ან ბრაუზერის რომელიმე დამატების (add ons) შეეცადეთ კოდის შიგთავსში შესვლას. ბლოკის მონიშვნის შედეგად html არეალში (სურ.16.2 მარცხენა ქვედა კიდე) გამოიყოფა ბლოკის კოდი, რითიც შესაძლებელია სელექტორის დადება. სურათ 16.2 ზე ნაჩვენები შედეგით მიმმართველი არის Class = “header”, შესაბამისად Css (სურ.16.2 მარჯვენა ქვედა კიდე) ჩანართი წარმოადგენს აქტიური ელემენტის სტილებს და მიგითითებთ მის მდებარეობაზე (ჩვენ შემთხვევაში თემის ძირითად საქაღალდეში Css საქაღალდე -> main.css ფაილის 21 ხაზი)



სურ16.2 ბლოკის სელექტორის და სტილების მდებარეობის განსაზღვრა

Css სტილების ადგილმდებარეობის განსაღვრის შემდეგ ანაცვლებთ უშუალოდ იმ ჩანაწერს რომლის ცვლილებაც გსურთ. (იხ სურ. 16.3 ბლოკისთვის შავი ფონის თეთრით ჩანაჩვლება) ეტაპობრივად გადადიხართ ყველა იმ ბლოკზე რომელთა ცვლილებაცაა დაგეგმილი. პარალელურ რეჟიმში ამოწმეთ მიღებული შედეგი რათა რაიმე მცირეოდენი უზუსტობის გამო საერთად არ დაირღვას პირვანდელი სახე და ვერც სასურველი შედეგი მიიღოთ.

```
21  .header{ width:100%; margin:0 auto 37px; padding:5px 0; background: rgba(255,255,255,.5); position:relative; z-index:99; color: #000; }
```

სურ. 16.3 header ბლოკისათვის შავი გამჭირვალე ფონის თეთრი ფონით ჩანაცვლება

ამ პრიმიტიული მოქმედებების შედებად თქვენ მიიღებთ სასურველ შედეგს. გაითვალისწინეთ შეცვლილი სტილები დაიკარგება და საიტი დაიბრუნებს პირვალნდელ სახეს თუ არ მოახერხებთ თემის განახლებების დეაქტივიზებას. (განხილულია რამოდენიმე აბზაცის წინ)!!!

6.4 ვებგვერდის სტრუქტურის ან შემცველობის/შინაარსის შეცვლა, კონკრეტული კლიენტის მოთხოვნის შესაბამისად

მიმდინარე პარაგრაფის თემატიკა

- კლიენთთან ურთიერთობის ძირითადი საკითხები
- კლიენტის მოთხოვნის გათვალისწინება
- ვებგვერდის სტრუქტურის ან შემცველობის/შინაარსის შეცვლა

საიტის აწყობა არც თუ ისე მარტივი საქმეა (რა თქმა უნდა თუ მხედველობაში მივიღებთ სტანდარტებით აწყობილ ვეგ საიტს), მაგრამ დამკვეთთან ურთიერთობა უფრო მეტ სირთულესთან არის დაკავშირებული.

ძირითად შემთხვევაში დამკვეთი წარმოადგენს ფინანსურ რესურსის მქონე ობიექტს, რომელიც ვერ ერკვევა ვებ ინდუსტრიაში. თქვენ ვალდებულნი ხართ საწყის ეტაპზე მოუწყოთ მოკლე მიმოხილვა ვებ ტექნოლოგიებში. ისევე როგროც სხვა სფეროებში მიმდინარე მიმართულებაშიც საკმაოდ მაღალი კონკურენციის პირობებია. შესაბამისად შეკვეთის მისაღებად გიწევთ თქვენი უპირატესობის დამტკიცება. ეს შეიძლება იყოს მომსახურების მაღალი ხარიხი, პროდუქციის შესრულების მინიმალური დროითი რესურსი, მცირე ბიუჯეტი და ა.შ.

დამკვეთთან ურთიერთობის გასაადვილებლად თქვენი იდეების სრულყოფილად მიტანაში დაგეხმარებათ ის ცოდნა და გამოცდილება, რაც გაგაჩნიათ. ყოველივე ეს რეალიზებული უნდა იყოს პროექტებში. დაიმახსოვრეთ თქვენი პროექტები თქვენი სავიზიტო ბარათია. დიდი არქივის ქონის შემთხვევაში საერთოდ არ გიწევთ არაფრის მტკიცება დამკვეთის გადმოსაბირებლად, ყველაფერზე მეტყველებს ნამუშვართა ხარისხი.

ხშირ შემთხვევაში თავად დამკვეთის მხრიდან ხდება იმ არქივის გამოთხოვნა თუ რა შეგიქმნიათ, რათა საწყის ეტაპზევე გაიჩინოს წარმოდგენა პიროვნებაზე, რომელთან საქმის დაჭერასაც აპირებს ობიექტი. გაითვალისწინეთ დამკვეთი ყოველთვის ცდილობს მიიღოს ინფორმაცია თქვენს შესახებ სხვადასხვა წყაროებიდან ამიტომ შეეცადეთ მაქსიმალური რესურსი გაიღოთ თითოეული შეკვეთის კეთილსინდისიერად შესრულებაში - ყოველივე ისევ თქვენს სამომავლო ინტერესებშია.

რაც შეეხება უშუალოდ დამკვეთთან ურთიერთობას იყავით გულწრფელი ვებ ინდუსტრიის შესახებ ინფორმაციის მიწოდებისას. ხშირია დამკვეთის მხრიდან ისეთი კითვები, რომელიც შესაძლოა მოკლებული იყო არსს, შეეცადეთ კომპეტენციის ფარგლებში ამომწურავად უპასუხოთ ყველა შეკითხვაზე და არ დატოვოთ უყურადღებოდ. მაგ.: უთხარით იმის შესახებ, რომ საიტზე სადაც

განთავსებულია ერთჯერადად ინფორმაცია მხოლოდ კომპანიის შესახებ (ბმული - ჩვენ შესახებ) და ვთქვათ საკონტაქტო გვერდი, ეს საიტი განწირულია საძიებო სისტემებში მოსაძებნად. ასევე ნუ შეიქმნის ილუზიას მხოლოდ ამ ტიპის საიტით ინტერნეტიდან ფულის შოვნასა და გამდიდრებაზე.

მოუსმინეთ დამკვეთს თუ რა სურს და მხოლოდ ამის შემდეგ შესთავაზეთ და მიაწოდეთ თქვენი იდეები. თუ სურვილები რადიკალურად განსხვავდება იმ მოცემულობისგან, იმ სტანდარტებისაგან, რაც მსოფლიო ბაზარზეა შეეცადეთ აუხსნათ. ამ შემთხვევაში ყოველთვის იფიქრეთ იმაზე , რომ ხარისხიანი პროდუქტის მიღება მხოლოდ დამკვეთის ინტერესებში არ არის და ის შემდგომში თქვენს სავიზიტო ბარათს წარმოადგენს.

შესთავაზეთ დამკვეთს რამოდენიმე ალტერნატიული ვარიანტი. მიეცით საშუალება ამოირჩიოს მისთვის საურველი დიზაინი. არ დააყენოთ დილემის წინაშე „ან ეს ან არაფერი“. ყოველთვის გახსოვდეთ თქვენ არ ხართ ერთადერთი ვისაც მსგავსი მომსახურების შეთავაზება შეუძლია.

რაც შეეხება უშუალოდ საიტის იერსახეს დეტალურად გაიარეთ დამკვეთთან თითოეული ნიუანსი. შეეცადეთ მიხვიდეთ ოპტიმალურ ვარიანტამდე. საქმის გაიოლების მიზნით ნურასოდეს ეტყვით უარს რთული პროგრამული ნაწილის შესრულებაზე.

არასოდეს დაიწყოთ საიტის აწყობა დამტკიცებული დიზაინის გარეშე. ხშირად მოგიწევთ მსვლელობისას შეცვლა და ამან შეიძლება ცუდი გავლენა იქონიოს დამკვეთთან ურთიერთობაში. გახსოვდეთ საიტი პირველ რიგში დამკვეთის კუთვნილებაა და შეეცადეთ მაქსიმალურად მოარგოდ მას.

საიტის სტუქტურისა და დიზაინის გაცნობა /დამოწმების შემდეგ მნიშვნელოვანია შეარჩიოთ ტექნოლოგია ვებ გვერდის ასაწყობად. როგორც აღვნიშნეთ არსებობს უამრავი CMS რომელთა გამოყენებითაც შესაძლებელია საკმაოდ მარტივად გადაიჭრას ესა თუ ის რთული პროგრამული ნაწილი. ხშირ შემთხვევაში ხდება შემდეგი: დამწყები ვებ დეველოპერი CMS-ს იყენებს მხოლოდ იმ მზა სახით როგორსაც თავად კომპანია და ორგანიზაცია გვთავაზობს. ეს რა თქმა უნდა მარტივი და მოსახერხებელია , მაგრამ არაა დიდად მოქნილი. გამომდინარე იქიდან რომ დამკვეთს შეიძლება საერთოდ განსხვავებული სტრუქტურისა და დიზაინის საიტი სურდეს, არ უნდა ვაიძულოთ იმ ტიპის საიტის შეთავაზება რომლის “თემასაც“ მარტივად მივაკვლიეთ ინტერნეტში. ხდება მარტივი პროცედურა: დამკვეთი გთხოვთ რაიმე ბლოკის გადაადგილებას და .... ზუსტადაც და... რადგანაც ხშირია შემთხვევა ბლოკის გადაადგილებისას გიწევთ სტრუქტურის დარღვევა. CMS შედგება უამრავი ფაილისაგან. ხდება ამ ფაილებში ძიება , არევა, დარევა და იღებთ არასრულყოფილ შედეგს, დიდი დროითი და ფიზიკური რესურსის ხარჯვის შემდეგ.

დამეთანხმებით უმჯობესი იქნებოდა თქვენს მიერ აწყობილი html-css საიტის ერთი ხელის მოსმით CMS_ზე გადაწყობა ყოფილიყო შესაძლებელი, მაგრამ ჩემზე კარგად მოგეხსენებად ესე მარტივადაც არ არის საქმე. შემდეგ ქვეთავში შევეცადოთ ავხსნათ როგორაა შესაძლებელი საკუთარი სტატიკური საიტის გადაწყობა WordPress_ის ძრავზე რათა ის გახდეს დინამიური.

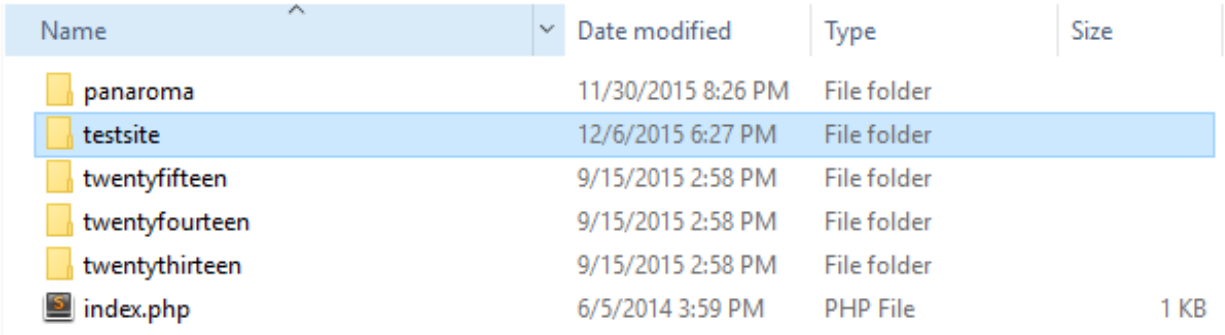
WordPress_ის თემის შექმნა

WordPress_ის თემის შესაქმნელად საჭიროა გავიგოთ ძირითადი ფაილური სტრუქტურა, რომელიც ამ CMS_ის თემას გააჩნია. ნებისმიერი სახის დოკუმენტაცია WordPress_ის შესახებ შეგიძლიათ მოიძიოთ ოფიციალური საიტის **Support** > **documentation** ჩანართში. სრული მისამართი https://codex.wordpress.org/Main_Page

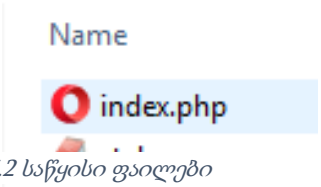
დოკუმენტაციის მიხედვით WordPress_ის თემის ძირითად ფაილებს წარმოადგენს:

- **style.css** -სტილების ფაილი
- **index.php** - მთავარი გვერდის ფაილი

ამ ორი ფაილის გამოყენებით სასურველ საიტს რა თქმა უნდა ვერ მივიღებთ. მარგამ გავყვეთ ეტაპობრივად.



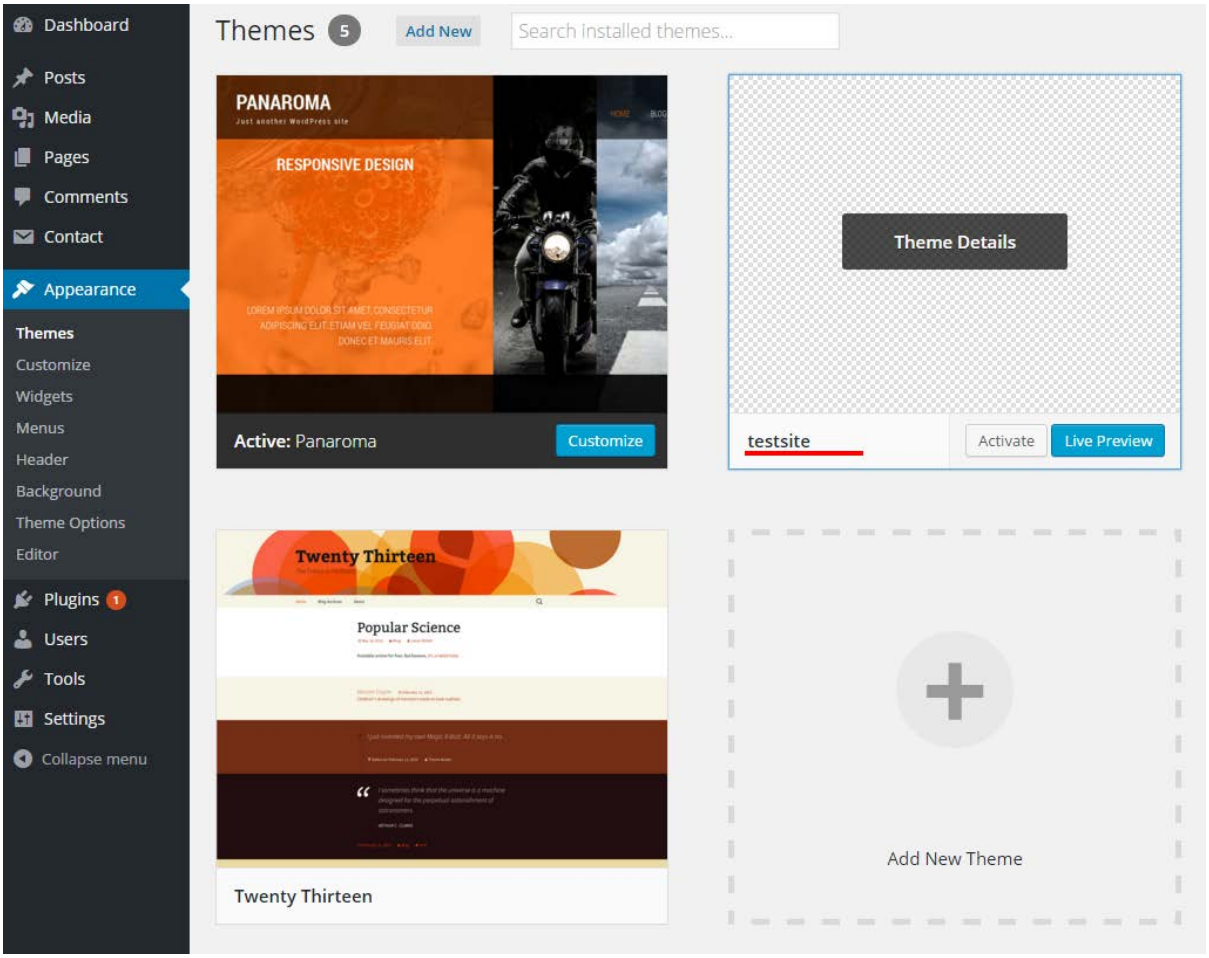
სურ. 17.1 ახალი თემის საქაღალდე



სურ.17.2 საწყისი ფაილები

უპირველეს ყოვლისა გავამზადოთ საქაღალდე ახალი თემისათვის WordPress_ის ძრავის **themes** ჩანართში ჩვენ შემთხვევაში - **testsite**. მასში შევქმნათ **style.css** და **index.php** ფაილები. (იხ.სურ17.1-17.2)

ამ ნაბიჯის წარმატებით დასრულება განაპირობებს ადმინისტრირების პანელის **Appearance>themes** ჩანართში ახალი თემის გამოჩენს, რომელიც დასათაურებულია ავტომატურად საქაღალდის სახელწოდებით და მასზე დეტალური ინფორმაცია არ მოიპოვება (იხ.სურ. 17.3)

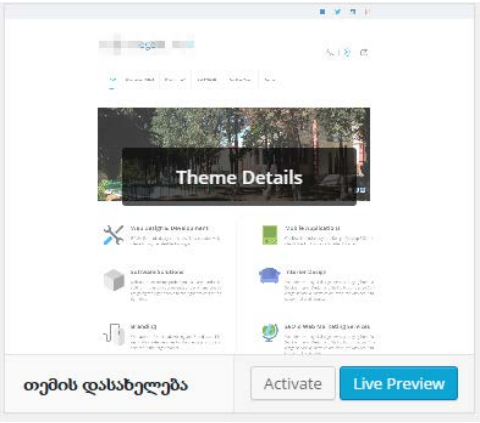


სურ.17.3 ანონიმური თემა

იმისათვის რომ **Theme Details** ვიზუალურად სრულყოფილი სახე მიიღოს საჭიროა სტილების ფაილი გავაფორმოთ სურათ 17.4 ნაჩვენები მოცემულობით. თითოეული სტრიქონი შეავსეთ შესაბამისი მნიშვნელობით

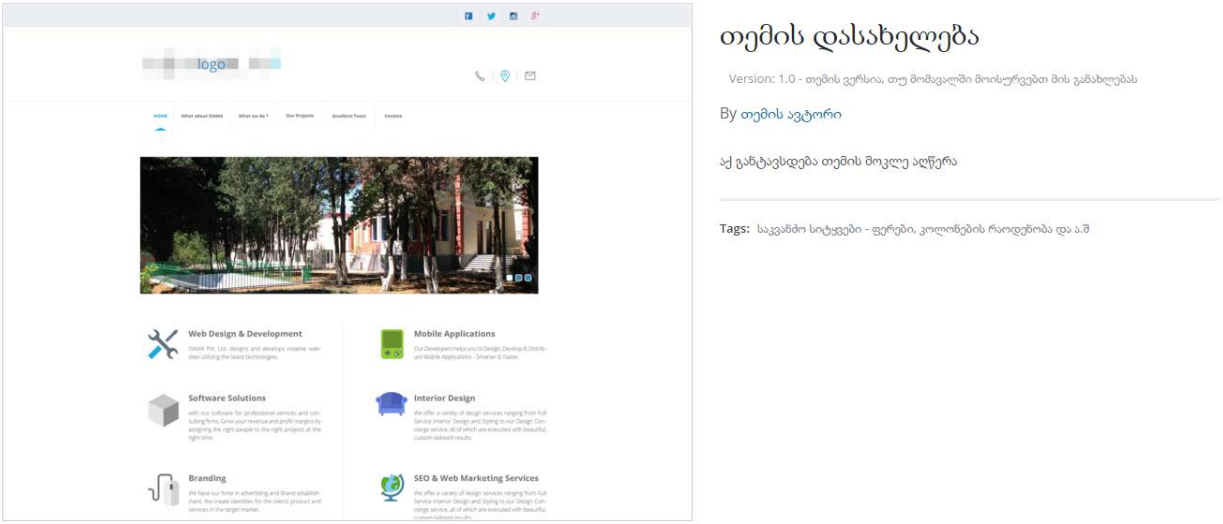
```
/*
Theme Name: თემის დასახელება
Theme URI: თემის  მისამართი ( ლინკი )
Author: თემის ავტორი
Author URI:  ავტორის მისამართი ( ლინკი )
Description: აქ განტავსდება თემის მოკლე აღწერა
Version: 1.0  - თემის ვერსია, თუ მომავალში მოისურვებთ მის განახლებას
License: ლიცენზია
License URI: ლიცენზიის მისამართი
Tags: საკვანძო სიტყვები - ფერები, კოლონების რაოდენობა და ა.შ

Text Domain: დომენის სახელი
*/
```

სურ17.4 სტილების ფაილის გაფორმება

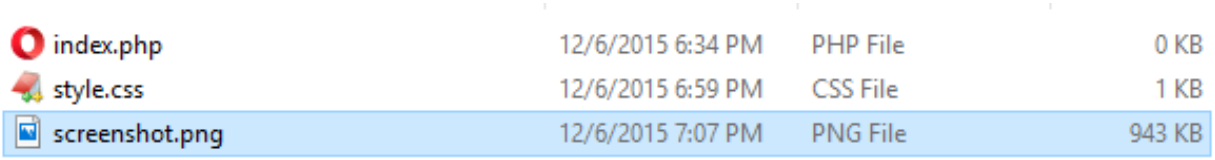


ფაქტობრივად თემის ვიზუალი ადმინისტრირების მხარეს დასრულებულია, ერთადერთი რაც დარჩა მინიატურის დაყენებაა. როგორც ატყობთ სურ 17.3 თემა სათაო სურათის გარეშეა წარმოდგენილი. ამისათვის შექმენით საიტის საპრეზენტაციო გვერდის მინიატურა სახელით **screenshot.png**. (იხ.სურ.17.5) **Theme Details** სრულყოფილი სახე ნაჩვენებია სურათ 17.6_ზე

სურ.17.5 თემის მინიატურა



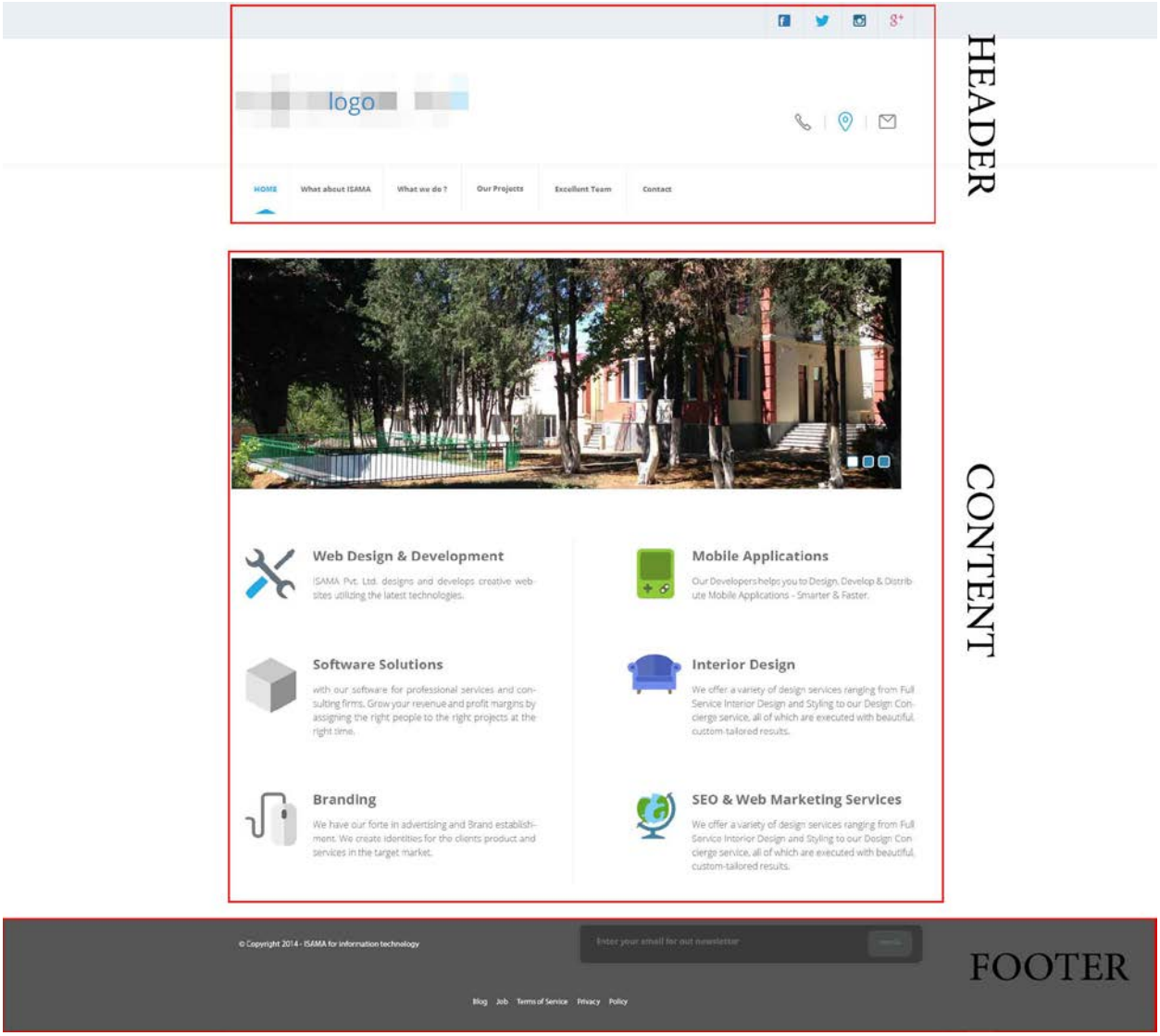
სურ.17.6 თემის დეტალური ინფორმაცია



ფაილური სტრუქტურა screenshot.png ჩამატების შემდგომ

როგროც ატყობთ საწყის ეტაპზე გაცხადებულ ფაილებს (style.css, index.php) დაემატა screenshot.png ფაილი. გზადაგზა შევეცდებით იმ ფაილების დამატებას, რომელიც მნიშვნელოვანია თემის სრულყოფისათვის.

სურათ 17.7ზე მოცემულია საიტი რომელიც თემის აწყობასაც შევეცდებით. ჩვენი მიზანი WordPress_ის ფუნქციონალის შესწავლაა ამიტომ ყურადღება გადატანილი იქნება html_შაბლონის გადაწყობაზე. სტილების ფაილის დეტალური გარჩევა არ იგეგმება.



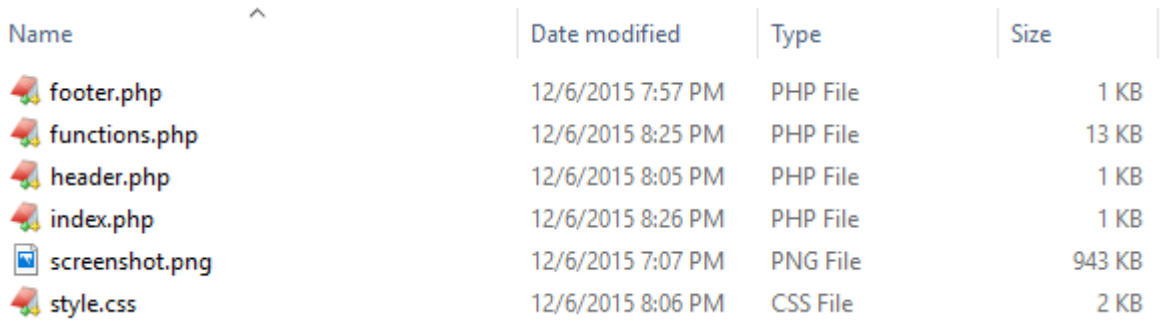
სურ.17.7 საიტის შაბლონი

საიტისძირითადიბლოკებიგამოყოფილიაწითელიჩარჩოთი.(იხ.სურ.17.7) ზოგადადყველასაიტიგარკვეულიბლოკურისტრუქტურითხასიათდება. უმრავლესშემთხვევაშისაიტებიიყოფა 3 დიდბლოკად:

- ესარისსაიტისთავი -**header**
- საიტისმთავარიინფორმაციულიბლოკი - **content**
- საიტისძირი- **footer**

საიტისგვერდებსშორისნავიგაციისდროსზედა - header და ქვედა -footer ბლოკები უცვლელ მდგომარეობაშია. იცვლება მხოლოდ content ბლოკის შემცველობა.

გამომდინარე იქდან, რომ მხოლოდ index.php ფაილი არასკმარისია რადგანაც ვცდილობთ მრავალგვერდიანი ფუნქციონალური საიტის აწყობას შევეცადოთ დავამატოთ ფაილების ერთობლიობა რომელიც დაგვჭირდება. ამ ლოგიკის მიხედვით ცალკე გამოვყოთ საიტის ზედა და ქვედა ბლოკები და მათი კოდები ჩავალაგოთ შესაბამისად **header.php** და **footer.php** ფაილებში. თემის ფუნქციონალის გასაფართოებლად დაგვჭირდება **function.php** ფაილი უმჯობესია თუ მასაც გაითვალისწინებთ. ფაილური სტრუქტურა გაფართოვდება (იხ სურ 17.8)



Name	Date modified	Type	Size
footer.php	12/6/2015 7:57 PM	PHP File	1 KB
functions.php	12/6/2015 8:25 PM	PHP File	13 KB
header.php	12/6/2015 8:05 PM	PHP File	1 KB
index.php	12/6/2015 8:26 PM	PHP File	1 KB
screenshot.png	12/6/2015 7:07 PM	PNG File	943 KB
style.css	12/6/2015 8:06 PM	CSS File	2 KB

სურ 17.8 განახლებული ფაილური სტრუქტურა

header.php ფორმირება

header.php ფაილი შევსებული იქნება იმ გამოყოფილი ინფორმაციით, რომელიც ყველა გვერდზე ვრცელდება-შესაბამისი html გაფორმებით. ჩვენ შემთხვევაში გამოყოფილი ბლოკი მინიშნებით Header. (იხ.სურ.17.7) header.php შეგიძლიათ გააფორმოთ სურათ 17.9 მსგავსად.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4     <meta charset="<?php bloginfo( 'charset' ); ?>">
5     <title><?php wp_title( '|', true, 'right' ); ?></title>
6      <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" />
7     <?php wp_head(); ?>
8  </head>
9
10 <body>
```

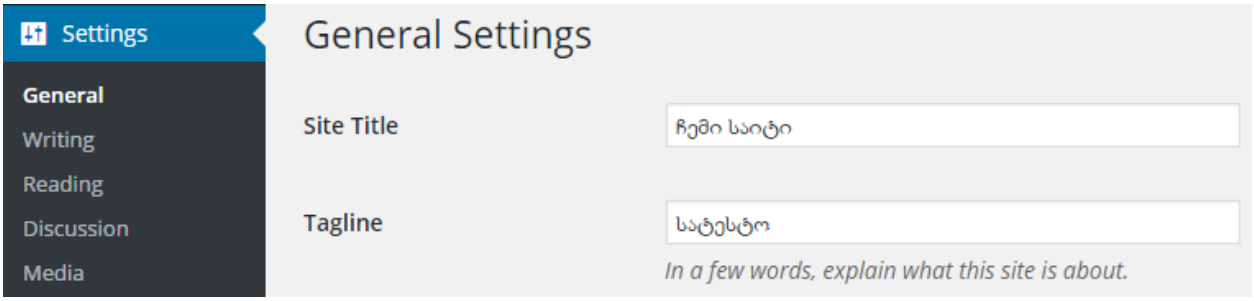
სურ. 17.9 header.php გაფორმება

4 - სტრიქონიდან html-კოდში მცირეოდენ განსხვავებული ბრძანებები გამოისახება, რომლებთანაც შეხება შესაძლოა აქამდე არ გქონიათ. **<?php ?>** -ეს ბრძანებები ემსახურება პრორამირების ენა php და წარმოადგენს მის გამხსნელ და დამხურავ ბრძანებეს. მათ შორის მოთავსებული ერთეულებს ფუნქციები ეწოდებათ. ამ ეტაპზე ფუნციათა არსის ცოდნა ნაკლებად მნიშვნელოვანია მთვარია გაიგოთ არსი თუ რომელი ფუნქცია რა მოქმედებას ასრულებს.

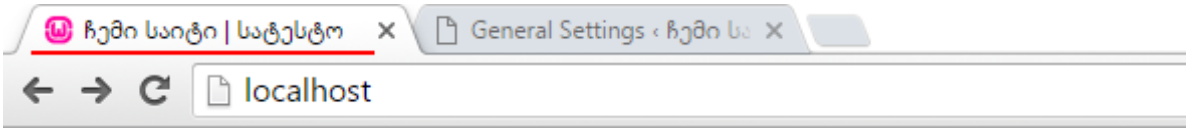
შევეცადოთ განვმარტოთ უცნობი ველები:

4 ხაზზე არსებული ფუნქცია **bloginfo("charset")** პასუხისმგებელია ვებ გვერდის კოდირებაზე. რა თქმა უნდა არავითარ პრობლემას არ წარმოადგენს ხელით utf-8 კოდირების განსაზღვრა, უბრალოდ გამომდინარე იქიდან, რომ ძრავი არის უნივერსალური , მას შეუძლია აიღოს ნებისმიერი კოდირება რომელიც განესაზღვრება მონაცემთა ბაზას WordPress_ის რეგისტრაციის დროს (ჩვენს შემთხვევაში utf8 - სურ.3.5 database - collate)

5 ხაზი wp_title() ფუნქცია - უზრუნველყოფს დინამიური title გამოტანას ვებ გვერდისათვის. ის შედგება საიტის სატაურისაგან და იმ გვერდის სათაურისგან რომელიც გახსნილია კონკრეტულ შემთხვევაში. (იხ.სურ. 17.10)



ადმინისტრირების პანელიდან საიტის სათაურის და განსაზღვრების დამატება



სურ.17.10 wp_title ფუნქციით მიღებული შედეგი

აღსანიშნავია, რომ მიმდინარე ფუნქცია არ იმუშავებს თუ function.php ფაილში არ განსაზღვრავთ შემდეგ ფუნქციას. (იხ.სურ.17.11) არსებობს მრავალი ნაირსახეობა მიმდინარე ფუნქციის უბრალოდ ეს ვერსია მეტად კომპაქტურია. შესაძლებელია მიმდინარე ფუნქციის შესახებ ინფორმაციის მოძიება შემდეგ მისამართზე https://developer.wordpress.org/reference/functions/wp_title/#source-code

```php
<?php

function testsite_wp_title( $title, $sep ) {
    global $paged, $page;

    if ( is_feed() )
        return $title;

    // Add the site name.
    $title .= get_bloginfo( 'name', 'display' );

    // Add the site description for the home/front page.
    $site_description = get_bloginfo( 'description', 'display' );
    if ( $site_description && ( is_home() || is_front_page() ) )
        $title = "$title $sep $site_description";

    // Add a page number if necessary.
    if ( ( $paged >= 2 || $page >= 2 ) && ! is_404() )
        $title = "$title $sep " . sprintf( __( 'Page %s', 'aisi' ), max( $paged, $page ) );

    return $title;
}
add_filter( 'wp_title', 'testsite_wp_title', 10, 2 );

?>
```

სურ, 17.11 wp_title ფუნქციის დამატება გამოძახება function.php ფაილში

6 ხაზზე (სურ.17.9) ხდება სტილების შემოტანა style.css ფაილის მეშვეობით **bloginfo('stylesheet_directory')** - ფუნქცია უზრუნველყოფს style.css ფაილის მისამართის ავტომატურ გაწერას. თქვენს მიერ გავლილი მასალიდან ინფორმირებულნი ხართ, რომ სტილების ფაილთან დაკავშირებას სჭირდება სრული მისამართი ფაილამდე მაგ.: http://localhost/wp-content/themes/testsite/style.css საიტის გადატვირთვა სხვა მისამართზე გამოიწვევდა საიტის მისამართის ცვლილებას და სტილების აღარ აღიქმებოდა საიტზე ან მოგიწევდათ მისამართის ჩანაცვლება. მიმდიანრე ფუნქცია ამ ყოველივეს ახდენს ავტომატურად. ამიტომ მისი გამოყენებაც რეკომენდებულია.

7 - სტრიქონზე გამომახებული ფუნქცია **wp_head() _** ფუნქციას უმჯობესია არ შეეხოთ და დატოვოთ მუშა მდგომარეობაში. შესაძლოა ერთი შეხედვით საერთოდ ვერ დაადგინოთ მისი მუშაობის პრინციპი, არაფერი შეიცვალოს საიტზე მისი წაშლის შემთხვევაში, მაგრამ ის საკმაოდ მნიშვნელოვანი დატვირთვის მატარებელია. მისი მოკლე დახასიაათება - **wp_head()** ფუნქციაეგრედწოდებული შუამავლის როლს ასრულებს სხვადასხვა ფუნქციებთან მიმართებაში და ახდენს მათ ასახვას საიტზე.მისი გათიშვის შემთხვევაში შესაძლებელია როგორც საიტზე გამოყენებულ ფუნქციებს ასევე იმ plugin_ების მუშაობას შეემთხვას საფრთხე რომლებიც იდეურად დამოკიდებულები არიან ამ ფუნქციაზე.

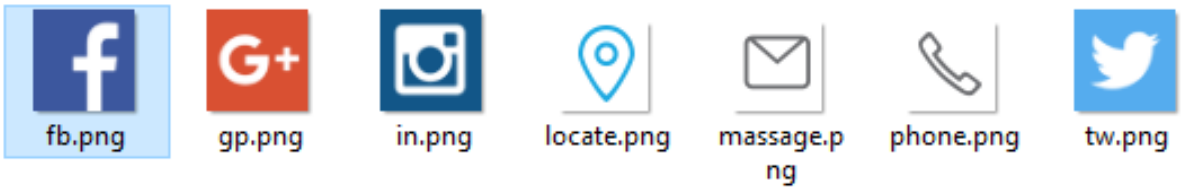
Wordpress_ის ზედა ნაწილი ფაქტობრივად გავარჩიეთ. ახლა გადავინაცვლოდ უშუალოდ შემცველობაზე. არსებობს უამრავი ფუქცია რომელთა გამოყენებითაც შესაზლებელია საკმაო ფუნქციონალის მიმატება გახდეს შესაძლებელი. ჩვენ ძირითად მეთოდებს შევეხებით . ეს ყოველივე მოგცემთ საშუალებას გაიგოთ მიდგომა რომელიც ამ ძრავს გააჩნია და მომავალში თავად გაიღრმავოდ ცოდნა.

დავალება 17.7 შესაბამისად შევქნათ ზედა ველი - შევიტანოთ სოციალური ქსელების ლოგოტიპები, header.php_ში უნდა განვათავსოთ ასევე ლოგო, საკონტაქტო ინფორმაციის ბმულები და საიტის მთავარი მენიუ.

საიტზე არსებული სურთების შესანახად ფაილურ სტრუქტურაში იქმნება დამატებითი საქაღალდე. სასურველია მისი სახელი დამკვიდრებული პრაქტიკიდან ავიღოთ და შევურჩიოთ სტანდარტულად **image** ან **images**. მიმდინარე საქაღალდეში ამ ეტაპზე განვათავსებ სოციალური ქსელის, სამისმაარტე ბმულების ლოგოტიპებს, ფაილური სტრუქტურა მიირებს შემდეგ სახეს

Name	Date modified	Type	Size
images	12/6/2015 10:11 PM	File folder	
footer.php	12/6/2015 7:57 PM	PHP File	1 KB
functions.php	12/6/2015 9:20 PM	PHP File	13 KB
header.php	12/6/2015 10:08 PM	PHP File	1 KB
index.php	12/6/2015 10:17 PM	PHP File	1 KB
screenshot.png	12/6/2015 7:07 PM	PNG File	943 KB
style.css	12/6/2015 10:16 PM	CSS File	2 KB

თემის ფაილური სტრუქტურა მიმდინარე ეტაპზე



Images საქაღალდეში არსებული სურათების ერთობლიობა

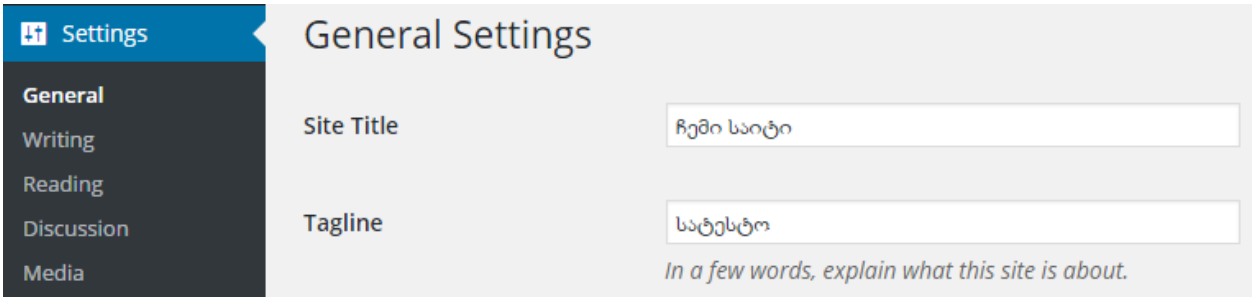
არსებული სურათების საიტზე განთავსება ნაჩვენებია 17.12_ზე. დაუკვიდით გამოყოფილ არეალს. ფუნქცია **bloginfo("template_url")** უზრუნველყოფს გზის განსაზღვრას კონკრეტულ ფაილამდე, ჩვენ შემთხვევაში სურათამდე.

<img src="**<?php bloginfo("template_url")?>**/images/fb.png" > სურათის ან ნებისმიერი ფაილის შეტანისას აუცილებელია გამუქებული ფუნქციის გამოყენება რათა მოხდეს რეალური წვდომა ფაილის მდებარეობასთან. საგულისხმოა ასევე „/“ რომელიც აუცილებლად თან უნდა სდევდეს ფუნქციას, რადგანაც ეს მისამართი კონკრეტულ საქაღალდეს წარმოადგენს

```
<body>
<div id="topheader">
   <div class="center">
      <nav class="social">
         <a href="http://facebook.com"><img src="<?php bloginfo("template_url")?>/images/fb.png" width="28" height="28" border="0" alt=""></a>
         <a href="http://twitter.com"><img src="<?php bloginfo("template_url")?>/images/tw.png" width="28" height="28" border="0" alt=""></a>
         <a href="https://www.instagram.com"><img src="<?php bloginfo("template_url")?>/images/in.png" width="28" height="28" border="0" alt=""></a>
         <a href="https://plus.google.com"><img src="<?php bloginfo("template_url")?>/images/gp.png" width="28" height="28" border="0" alt=""></a>
      </nav>
   </div>
</div>
```

სურ.17.12 სურათის შეტანისას მართებული src-ს გაწერა

საიტის სათაურისა და მოკლე მიმოხილვა რეგულირდება ადმინისტრირების პანელის **Setting > General . Site Title** საიტის სათაურს წარმოადგენს. მისი ვიზაულიზება საიტის თემაში შესაძლებელია ფუნქცია **<?php bloginfo('name'); ?>**მეშვეობით, ხოლო Tagline -**<?php bloginfo('description'); ?>**(იხ.სურ17.13)



ადმინისტრირების პანელიან საიტის სახელის და აღწერის მართვა

```
<h1><a href="<?php echo esc_url( home_url( '/' ) ); ?>" rel="home"><?php bloginfo( 'name' ); ?></a></h1>
<h2><?php bloginfo( 'description' ); ?></h2>
```

სურ 17.13 საიტის სახელის და Tagline გამოტანა

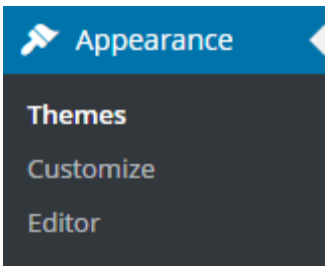
ხშირ შემთხვევაში სიტის სახელი წარმოადგენს ბმულს რომელიც გადამისამართდება პირველ გევრდზე ამისათვის თუ დაუკვირდებით სურათ 17.13 ჰამოიყენება ფუნქცია **<?php echo esc_url(home_url('/')); ?>**რომელიც განთავსებულია ბმული ტეგის სამისამართე არეალში.

საიტის შენების პროცესი წარმატებით მიმდინარეობს. მიმდინარე შედეგი გამოსახულია სურათ 17.14

სურ 17.14 მიმდინარე ეტაპისათვის მიღებული შედეგი

სამისამართე ლოგოტიპები პირობითადაა შეტანილი (მსგავსად სოციალური იკონებისა) გამომდინარე იქიდან, რომ მათი მნიშვნელობა ამ ეტაპზე არ ატარებს დიდ პროგრამულ დატვირთვას.

გადავერთოთ header_ის დამამთავრებელ ფაზაზე, ერთ ერთ მნიშვნელოვან ბლოკზე სანავიგაციო - მენიუს არეალზე. ავტომატურ რეჟიმში მენიუ ახალდამატებული გევრდების ერთობლიობაა. იმისათვის, რომ თემას გააჩნდეს ფუნქციონალი მენიუს შექმნისა საჭიროა function.php მორიგი პროგრამული კოდის დამატება.



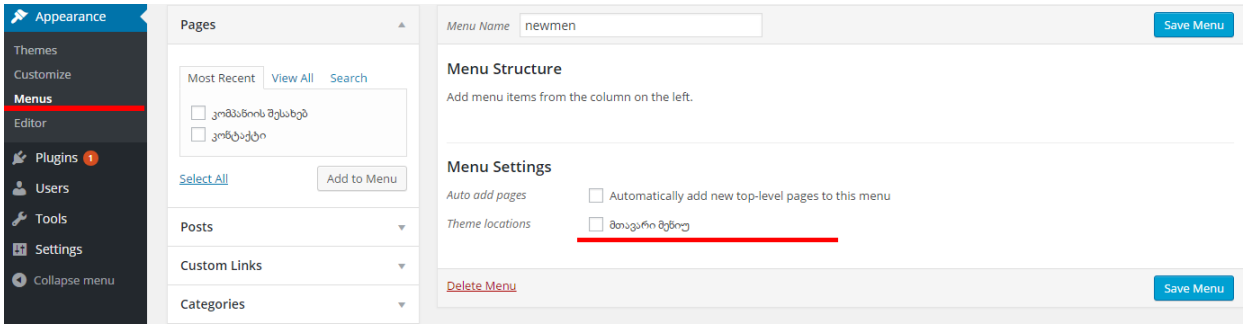
მიმდინარე ეტაპზე **Appearance** ჩანართი მხოლოდ **Themes, Customize** და **Editor** ქვემენიუებს შეიცავს. როგორც ვიცით მენიუს ჩანართი ერთ ერთი უმნიშვნელოვანესია თემისათვის. ამიტომ მოახდინეთ function.php მენიუს გამოძახების ფუნქციის დამატება. (იხ.სურ.17.15)

register_nav_menu('primary', __('მთავარი მენიუ', 'testsite')) ფუნქცია უზრუნველყოფს Apprearance ბლოკში მენიუს ბმულის დამატებას. register_nav_menu ფუნქციაზე სრული ინფორმაციის მისაღებად ეწვიეთ შემდეგ ბმულს https://codex.wordpress.org/Function_Reference/register_nav_menu

```
25 register_nav_menu( 'primary', __( 'მთავარი მენიუ', 'testsite' ) );
```

სურ. 17.15 მენიუს დამატება ადმინისტრირების პანელში

register_nav_menu() - ფუნქციის გააქტურებისას ადმინისტრიერბუს პანელში მიღებული შედეგი ნაჩვენებია სურათ 17.16_ზე. ყურადღება გაამახვილეთ მარჯვენა კიდეში გამოყოფილ ტექსტზე. ეს ის ტექსტია რომელიც მენიუს რეგისტრაციისას მიეთითა function.php ფაილის register_nav_menu ფუნქციაში.



სურ 17.16 register_nav_menu() ფუნქციის დამატებით მიღებული შედეგი

მენიუების შექმნის და მათგან ერთ ერთის "მთავარ მენიუდ" ქცევა არ ნიშნავს, რომ ეს მენიუ ავტომატუად განლაგდება სასურველ ადგილას. საჭიროა მენიუს გამოტანის ფუნქციის გააქტიურება იმ ადგილას სადაც გვსურს მისი ასახვა.

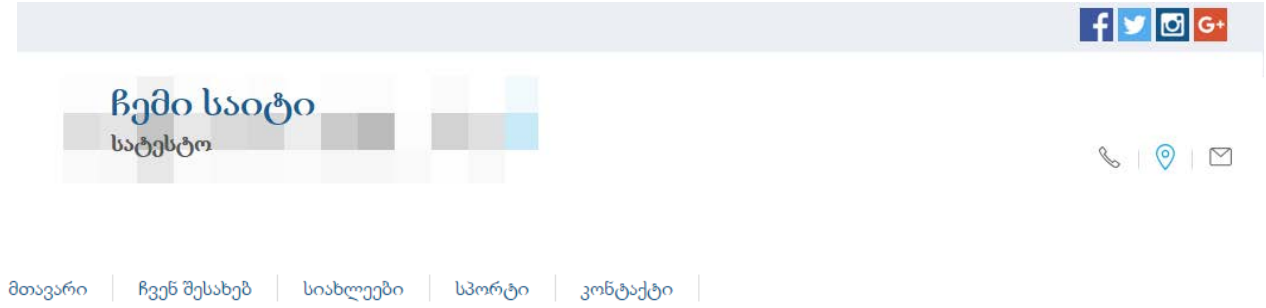
მაგალითის შესაბამისად მენიუ გამოტანილ უნდა იყოს header.php ფაილში სასურველ ადგილას შემდეგი ფუნქციის გამოყენებით **wp_nav_menu()** (იხ.სურ.17.17).დოკუმენტაცია მიმდინარე ფუნქციის შესახებ განთავსებულია შემდეგ მისამართზე https://codex.wordpress.org/Function_Reference/wp_nav_menu

```
22 <?php wp_nav_menu( array( 'theme_location' => 'primary', 'menu_class' =>
'nav-menu', 'menu_id' => 'primary-menu' ) ); ?>
```

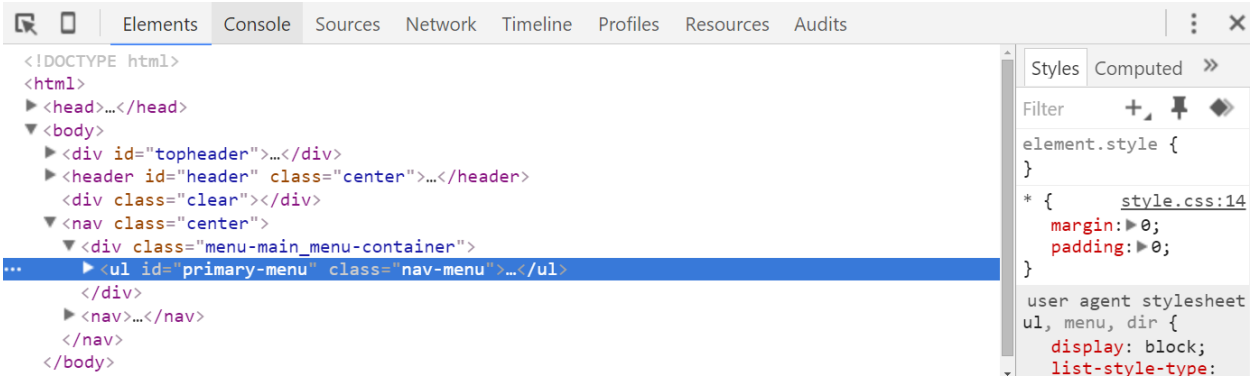
სურ. 17.17 საიტზე მენიუს გამოტანა

გაითვალისწინეთ სურათზე მოცემული ფუნქცია განლაგებულია ერთ ხაზზე. სურათზე ნაჩვენები ატრიბუტები არასრული ჩამონათვალია, სრული ინფორმაციისთვის შეგიძლიათ გაეცნოთ წინამდებარე აბაცში მოცემულ ბმულს. menu_class ატრიბუტი ანიჭებს შექმნილ მენიუს კლასს, ხოლო menu_id მენიუს სელექტორი id_ია. სურათ 17.19 _ ზე ნაჩვენებია სელექტორებით გაფორმებული **ul** ტეგი.

ფინალური შედეგი ფორმირებული header.php ფაილისა იხილეთ სურათ 17.18_ზე.



სურ 17.18 header.php გამოსული მენიუ



სურ 17.19 wp_nav_menu() ფუნქციით მინიჭებული class და id სელექტორები

მოცემული დავალების საწყისი ეტაპი წარმატებით შეასრულეთ. დასრულდა საიტის ზედა ნაწილის ფორმირება. გაითვალისწინეთ მიღებული header.php ფაილი მხოლოდ მოცემულობის შესაბამისად იქნა აწობილი, ის არ წარმოადგენს უნივერსალურ ფაილს და ყველა სტილისა და ფუნციონალის შესაბამის საიტს არ მიესადაგება.

index.php ფორმირება

index.php ფაილი პასუხისმგებელია საიტის მთავარი გვერდის ვიზუალიზებაზე. მოცემული მაგალითის შემთხვევაში მთავარი გვერდზე გამოტანილია 6 სიახლე. შესაძლებელია პირველი გვერდი უფრო მეტად ფუნქციური იყოს და შესაბამისად მხოლოდ ის მიდგომა, რომელსაც გავარჩევთ არ იქნება მორგებადი ყველა დიზაინზე. ეს არის კონკრეტული დავალების გადაჭრის ერთი რიგითი გზა.

Header.php არსებული კოდების ერთობლიობის შემოსატანად index.php გვერდზე გამოიყენება ფუქცია **<?php get_header(); ?>.** არსებული ფუნქციის გამოყენება დასაშვებია ყველა იმ ფაილში, სადაც დაგჭირდებათ header ფაილში არსებული კოდების ერთობლიობის გამოყენება. ეს საკმაოდ მოსახერხებელია და თავიდან აგარიდებთ კოდის დუბლირებას. (იხ.სურ. 18.3).

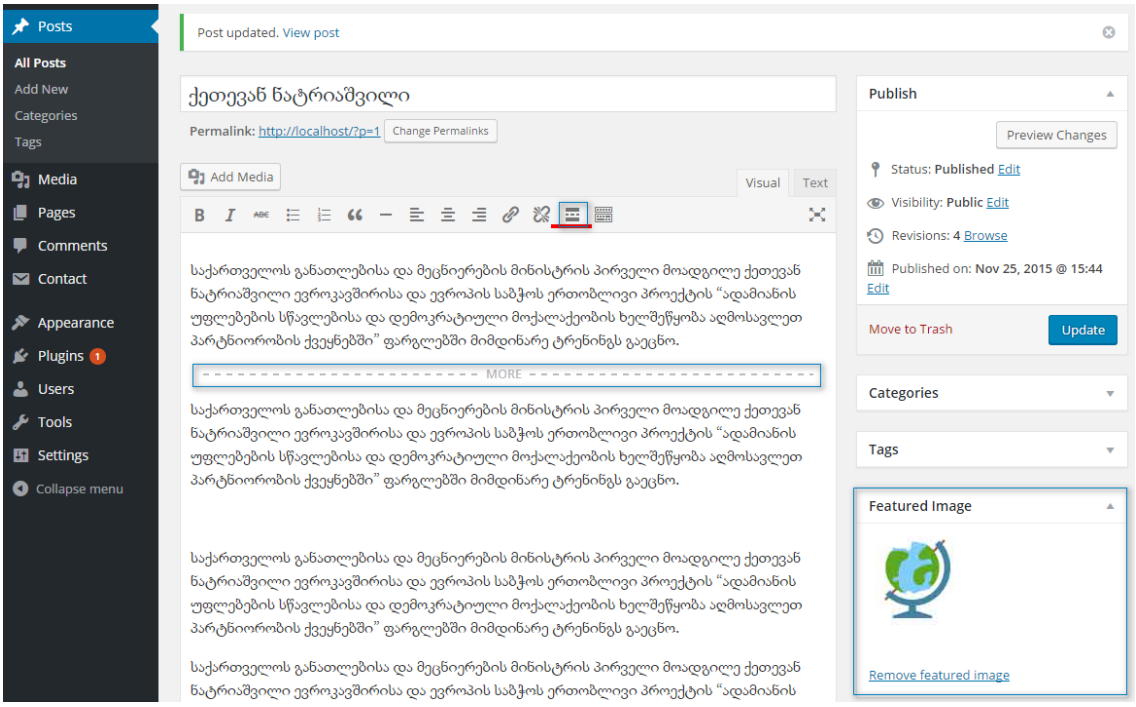
სიახლეების გამოსამახებლად საჭიროა ჯერ ეს სიახლეები შევქმნათ ადმინისტრირების პანელის **Posts>Add New** ჩანართიდან. პოსტების დამატება სტანდარტული პროცესია, რომელიც განხილულია **სამართავი პანელის სანავიგაციო არეალი** _ის ქვეთავში **Posts**.

ერთი მნიშვნელოვანი დეტალი. ახალი თემის შემთხვევაში პოსტების მინიატურის დასამატებელი ბლოკი (**Featured Image**) ფიზიკურად არ არსებობს. მის შესაქმნელად function.php ფაილში უნდა მოხდეს სურათ 18.1 ნაჩვენები ფუნქციების დამატება

```
28 add_theme_support( 'post-thumbnails' );
29 set_post_thumbnail_size( 604, 270, true );
```

სურ. 18.1 პოსტებისათვის მინიატურის ფუნქციონალის დამატება

სურათ 18.2_ზე მოცემულია ახალ იჩანაწერის დამატების მეთოდი. მისი გამოყენებით დაამატეთ მინიმუმ 7 ჩანაწერი.



სურ 18.2 ახალი ჩანაწერის დამატება

როგორც სურათის მოცემულობიდან ჩანს ახალი ჩანაწერის დამატებისას ის დასათაურდა, შეივსო მოცულობითი / ინფორმაციული ტექსტით, გაფორმდა მინიატურით. ყურადღება გაამახვილეთ გამოყოფილ ---- MORE---- ველზე, რომელიც უზრუნველყოფს ჩანაწერიდან იმ ნაწილის გამოყოფას, რომელიც უნდა გამოჩნდეს მისი გამოქვეყნებისას, ხოლო სრული ტექსტი აისახება ჩანაწერის დეტალური ნახვისას. ---- MORE ---- წყვეტის მიღება შეიძლება ფუნქციური ღილაკის მეშვეობით - გამოყოფილია წითლი ხაზგასმით.

ჩანაწერების დამატების შემდეგ აუცილებელია რაოდენობის განსაზღვრა, თუ რამდენი გამოისახოს კონკრეტულ გვერდზე ან კატეგორიაში. შედეგის მიღება შესაძლებელია ადმინისტრირების პანელის

Setting > Reading ბლოკში **Blog pages show at most** - რაოდენობის ცვლილებით . საწყის მნიშვნელობად ასახულია 10.

დარჩა უკანასკნელი ეტაპი - პროგრამული კოდის მეშვეობით ავსახოთ ისინი სასურველ ადგილას, მაგალითის შესაბამისად საიტის მთავარ გვერდზე - index.php ფაილში.

```
1  <?php get_header(); ?>
2   <?php
3     if(have_posts()):
4     while ( have_posts() ) : the_post(); ?>
5        <article class="news">
6              <?php the_post_thumbnail( 'thumbnail' ); ?>
7              <h3><a href="<?php the_permalink() ?>"><?php the_title(); ?></a></h3>
8              <div class='exc'><?php the_content('ვრცლად'); ?></div>
9        </article>
10    <?php
11    endwhile;
12    endif;
13    ?>
```

სურ. 18.3 ჩანაწერების გამოტანა

სურათ 18.3_ის **პირველ სტრიქონზე** გამოტანილია სწორედ ის ფუნქცია (**get_header()**), რომელზეც არც თუ ისე დიდი ხნის წინ ვისაუბრეთ - header.php ფაილის შემცველობის ინტეგრირება index.php ფაილის საწყის სტრიქონზე.

პირობითი ოპერატორი **if(have_posts())** ამოწმებს - არსებობს თუ არა ჩანაწერები მონაცემთა ბაზაში - **სტრიქონი 3**. დადებითი მნიშვნელობის შემთხვევაში სრულდება შემდ ხაზზე არსებული მოცემულობა. უარყოფითი შედეგი ტოვებს ყოველგვარ მოქმედებეს 12 ხაზის ჩათვლით და კოდს კითხვადობა გრძელდება 13 სტრიქონიდან.

While() - ციკლი **(ხაზი 4)** უზრუნველყოფს ჩანაწერების იმ რაოდენობის გამოტანას რამდენიც მითთებული იყო ადმინისტრირების პანელიდან.

სტრიქონი 5 - 9 იკვრება ბლოკური სტრუქტურა თუ რომელ ტეგებში უნდა იყოს მოთავსებული კონკრეტული ინფორმაცია. როგორც სურათიდან ჩანს დაიბეჭდება გარკვეული რაოდენობა ჩანაწერების (ჩვენ შემთხვევაში 6) და თითოეული განთავსებული იქნება article - ტეგში, კლასით news.

<?php the_post_thumbnail('thumbnail'); ?> - ფუნქცია უზრუნველყოფს მცირე ზომის მინიატურს გამოტანას. მისი ზომებისა და პარამეტრების შესახებ დაწვრილებითი ინფორმაციის მისაღებად იხილეთ ბმული

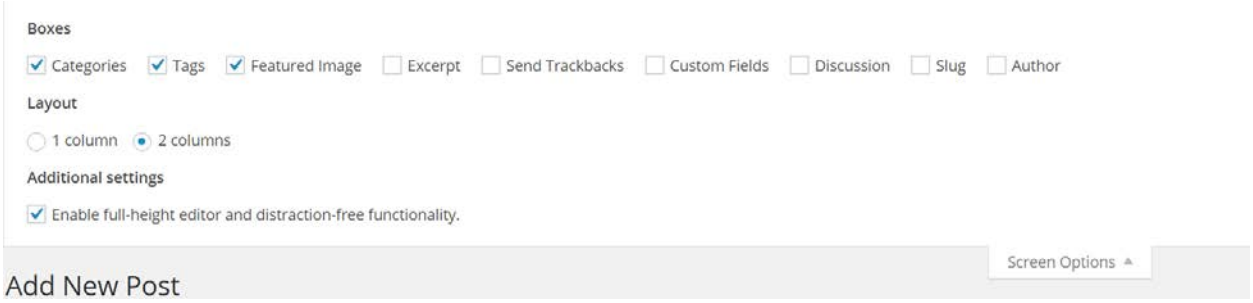
https://developer.wordpress.org/reference/functions/get_the_post_thumbnail/

<?php the_title(); ?> - ფუნქცია უზრუნველყოფს პოსტის სათაურის გამოტანას. მისი დეტალური გარჩევისათვის გაეცანით ბმულს

https://developer.wordpress.org/reference/functions/the_title/

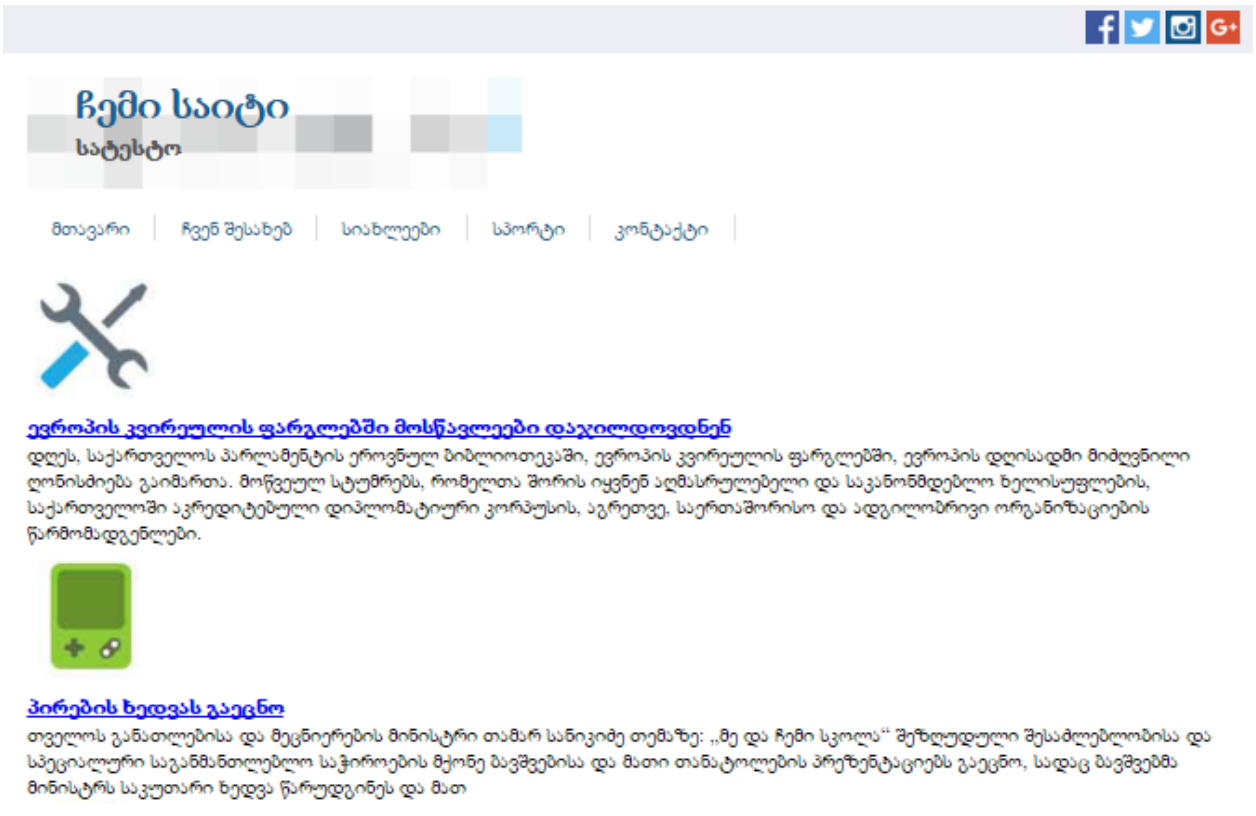
<?php the_permalink() ?>- ფუნქციის მეშვეობით შესაძლებელია ბმულის დადგენა. მიმდინარე მაგალითზე ის განთავსებულია <a href=””> ტეგში რაც უზრუნველყოფს მასზე მოქმედებისას პოსტის გადამისამართებას სრულ მისამართზე - ჩანაწერის დეტალური ნახვა.

<?php the_content(“ვრცლად”); ?> - გამოაქვს ---MORE--- წყვეტის გამოყენებამდე არსებული ტექსტი. მისი ალტერნატივა შესაძლოა იყოს **the_excerpt()**. მოცემული ფუქციისმეშვეობით შესაძლებელია ასევე ცალკე ბლოკიდან - **excerpt** გამოვიტანოთ ის განსხვავებული ინფორმაცია რომელიც შეიძლება უძღვოდეს პოსტს. ამ ბლოკის აქტივაცია ხდება ადმინისტრირების პანელიდან Posts>Add New ჩანართის Screen Option ბმულის Excerpt ბლოკის მონიშვნით (იხ.სურ.18.4)



Exerpt ბლოკის აქტივაცია

გაწეული სამუშაოს შედეგად საიტზე გამსული შედეგი ასახულია სურათ 18.5_ზე. სტილებით გაფორმება თქვენთვის მომინდვია. საიტის ინფორმაციის რაოდენობიდან გამომდინარე ეკრანზე ასახულია მხოლოდ 2 ჩანაწერი. ზედა ნაწილი გაფორმებულია header.php ფაილით. შემცველობა შედგება ჩანაწერებისაგან. შედეგი შესაძლოა დადებთად შეფასდეს.



index.php ფაილის სრულყოფის პროცესი - ჩანაწერების გამოტანა

საიტის პირველი გვერდის საბოლოო სრულყოფისათვის დარჩენილია საიტის ძირის - footer _ის ფორმირება.

footer.php ფორმირება

Header_ის მსგაავსად თითქმის ყველა საიტს გაჩნია ძირი - footer, რომელიც თავის მხრივივ, როგორც მთავარ ასევე შიდა გვერდებზეც უცვლელი სახით გამოისახება. ეს დებულება რა თქმა უნდა არ წარმოადგენს აპრიორს, უბრალოდ იმდენად ხშირია ფაქტობრივად 100%_იან მაჩვენებელს უტოლდება.

საიტის footer წარმოადგენს არეალს, სადაც აისახება სხვაადასხვა სახის ინფორმაცია. უმეტესწილად მიმდინარე ველი დათმობილია საიტის შემქმნელისა და საავტორო უფლებების (copyright) ვიზუალიზებისთვის, მაგრამ ეს იმდენად სპეციფიური ბლოკია შესაძლოა მასში მრავალი სხვა ინფორმაციის გამოტანაც მოხდეს. მაგ.: თუ საიტი მოცულობითია მომხმარებლისათვის ნაკლები დაბრკოლების შესაქმნელად ძირითადი მენიუს მსგავსი ბმულები წარმოუჩინოთ - რათა სხვა პუნქტზე გადასასვლედ არ მოუწიოს საიტის ზედა ნაწილამდე სქროლვა. შესაძლოა ასახოთ სასარგებლო ბმულები, საკონტაქტო ინფორმაცია, ტექსტური შეტყობინების ფორმის ველები და ა.შ.

საიტის ძირის თითქმის ყველა შემთხვევაში პირველი გვერდის იდენტურად გამოისახება შიდა გვერდებზე. როგორც header.php ფაილის განხილვისას ვახსენეთ ის პროგრამული კოდი, რომელიც უმეტეს გვერდებზე იდენტურად გამოისახე უმჯობესია მოთავსებულ იქნას რაიმე ფაილში და საჭიროებისამებრ გარკვეულ ადგილებში იმპორტირებული, რათა არ მოხდეს კოდი დუბლირება. ამ თეზისის გასამყარებლად WordPress ძრავში ფაილური სტრუქტურის ერთ ერთი მნიშვნელოვანი ელემენტს წარმოადგენს **footer.php ,** რომელიც სწორედ საიტის დამამთავრებელი ბლოკის ფორმირებას ახდენს.

მოცემული მაგალითის შემთხვევაში (იხ. სურ 17.7) ნათლად არის გამოყოფილი საიტის footer. შევეცადოთ მოვახდინოთ footer.php ფუნქციონალით დატვირთვა. მაგალითის შესაბამისად ვებ გვერდის footer აერთიანებს: საავტორო უფლებებს (copyright), საძიებო ფორმას და მენიუს. დავალების მცირეოდენ გასართულებლად დავუშვათ შემთხვევა, რომ ამ ბლოკის მენიუ არ წარმოადგენს ძირითადი ნავიგაციის იდენტურ ვარიანტს.

გამომდინარე იქიდან რომ საიტი ერთ მთლიან ორგანიზმს წარმოადგენს თუ მის header.php_ში სავალდებულოა Doctype და შესაბამისად html-head-body კონსტრუქციის გახსნა footer.php _ში აუცილებელია body , html ტეგების დახურვა;

რეკომედირებულია body დამხურავი ტეგის წინ გამოიძახოთ **<?php wp_footer() ?>**ფუნცქცია.მას ფაქტობრივად ანალოგიური დატვირთვა აქვს footer.php ფაილისათვის, რაც ფუნქცია wp_head()_ს header.php_ისათვის. მოცემული ფუნქცია wp_head() განმარტებულია წინამდებარე აბზაცებში (header.php ფორმირება) გაეცანით მას, რათა წარმოდგენა შეგექმნათ **wp_footer()**_ზე.

```
6     <?php wp_footer(); ?>
7  </body>
8  </html>
```

სურ.19.1 foot.php ძირითადიელემენტები

სტანდარტული footer.php აუცილებელი დამამთავრებელი ელემენტები ნაჩვენება სურათ 19.1.

დამატებით ფუნქციების გამოყენება უშუალოდ footer_ის სტრუქტურასა და კონკრეტულ დავალებაზეა დამოკიდებული.

ჩვენი მაგალითის შემთხვევაში Copyright სტატიკურ მდგომარეობაშიც, რომ გამოისახოს html ტეგების მეშვეობით არაფრერი დაშავდება. უმჯობესია ყურადღება გაამახვილოთ დანარჩენ მოდულებზე.

მაგ.: footer_ის მენიუ პირობისამებრ განსხვავებულ სანავიგაციო ბლოკს წარმოადგენს და header.php ფაილში არსებული მავარი მენიუს გამოტანის ფუნქციის ერთი-ერთში კოპირება და ჩასმა ვერ მოხერხდება. საწყის ეტაპზე საჭიროა მეორე მენის დამატება უშუალოდ ადმინისტრირების პანელის **Appearance>Menus** ჩანართიდან. ჩემს შემთხვევაში მიმდინარე მენიუ სახელწოდებად გამოყენებულ იქნება **ft_menu** პუნქტებად > გვერდები „კომპანიის შესახებ“ და „კონტაქტი“.

სურათ 19.2 მოცემულია wp_nav_menu() ფუნქცია სადაც მენიუსთან წვდომა ხდება მისი სახელის მიხედვით. მიმდინარე ფუნქცია განხილულია header.php_ის შექმნის ქვეთავში ძირითადი მენიუს ფორმირებისას. იქვეა წარმოდგენილი ბმული ოფიციალურ გვერდზე ამ კონკრეტული ფუნქციის შესახებ. თუ მას ვრცლად გაეცანით შეგექმენებოდათ წარმოდგენა მენიუს გამოტანის სხვადასხვა გზებზე. პარამეტრებიდან ერთ-ერთია წვდომა მენიუსთან სახელის მიხედვით. (იხ.სურ.19.2)

```
5    <?php wp_nav_menu( array( 'menu' => 'ft_menu', 'menu_class' => 'custom-menu', 'menu_id' => 'footer-menu' ) ); ?>
```

სურ. 19.2 მენის გამოზახება სახელის მიხედვით

დანარჩენი პარამეტრები თქვენთვის უკვე ნაცნობია და შესაძლებელაი სურვილისამებრ მათი გამოყენება.

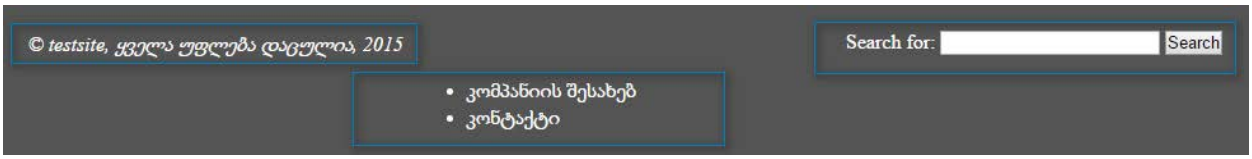
ამ კონკრეტულ მაგალითზე საიტის footer_ში გამოტანილია **საძიებო ფორმა**. მისი რეალიზაციისათვის დაგჭირდებათ მორიგი ფუნქციის მოძიება. სრულყოფილი ინფორმაცია ძიების ფუნქციის შესახებ შეგიძლიათ იხილოთ ბმულზე
https://codex.wordpress.org/Function_Reference/get_search_form . როგორც ატყობთ მისი გამოძახება არანაირ სირთულეს არ წარმოადგენს საჭროა სასურველ ადგილას მიუთითოდ - **<?php get_search_form(); ?>.**(იხ.სურ.19.3 - **სტრიქონი 3).** არსებული პუნქცია გამოიტანს ფორმას საწყისი სტრუქტურით, მისი ცვლილებისათვის გაეცანით მოყვანილ ბმულს დაწვრილებით.

footer.php დასრულებულია. ამოიწურა ყველა ის ტექნიკური დავალება, რომელიც მიმდინარე ფაილის ფუნქციონალით დატვირთვას ეხებოდა. footer.php საბოლოო სახე ნაჩვენებია სურათ 19.3_ზე.

```
1    <footer class="main-footer">
2      <em> © testsite, ყველა უფლება დაცულია, 2015 </em>
3      <div id="mysearch"> <?php get_search_form(); ?> </div>
4      <nav class="footer_nav">
5        <?php wp_nav_menu( array( 'menu' => 'ft_menu', 'menu_class' =>
6        'custom-menu', 'menu_id' => 'footer-menu' ) ); ?>
7      </nav>
8    </footer>
9    <?php wp_footer(); ?>
10 </body>
11 </html>
```

სურ. 19.3 მაგალითის შესაბამისი footer.php

საიტზე მიღებული ბლოკის ვიზუალური იერსახე გამოსახულია სურათ 19.4_ზე.



სურ.19.4 საიტზე მიღებული შედეგი

სურათზე ნაჩვენები შედეგის მიხედვით საძიებო ფორმის სრტუტურა შესაძლოა ყველასთვის მისაღები არ იყოს. მაგ. მარტივი მიზეზი - ძიების ღილაკი ლათინური ასოებითაა წარმოდგენილი, ასევე მინიშნება ძიების ველზე - **Search for** შესაძოა საერთოდ არ გჭირდებოდეთ. როგორც ძიების ფუნქციების გამოძახებისას აღვნიშნეთ ფორმა საწყისი მნიშვნელობით წარმოადგენს შემდეგ ვიზაულს.

შესაძლებელია function.php ფაილში სურათ 19.5_ზე მოცემული ფუნქციის გამოძახება, რაც საშუალებას მოგცემთ ფორმის ველების მნიშვნელობების საწყისი მნიშვნელობიდან გახადოთ მართვადი. (იხ.სურ.19.5). ფუნქციის მოძიება/კოპირება შესაძლებელია ბმულიდან, რომელზეც ყურადღება გამახვილდა ძიების ფორმის გარჩევისას.

სურათ 19.5_ზე გამოყოფილია სტრიქონები, რომელთა ინფორმაციის ცვლილება შესაძებელია დაგჭირდეთ არაინგლისურენოვალი საიტის შემთხვევაში. თუ ფორმის ველის მიმთითებელი საერთოდ არ გჭირდებად 33-ხაზზე არსებული <label> ტეგის მნიშვნელობის წაშლა არავითარ პრობლემას წარმოადგენს და არანაირ დაბრკოლებას არ შეუქმნის ფორმას მუშაობაში

```
31 function my_search_form( $form ) {
32   $form = '<form role="search" method="get" id="searchform" class="searchform" action="' . home_url( '/' ) . '" >
33   <div><label class="screen-reader-text" for="s">' . __( 'Search for:' ) . '</label>
34   <input type="text" value="' . get_search_query() . '" name="s" id="s" />
35   <input type="submit" id="searchsubmit" value="'. esc_attr__( 'Search' ) .'" />
36   </div>
37   </form>';
38
39   return $form;
40 }
41 add_filter( 'get_search_form', 'my_search_form' );
```

სურ.19.5 ფორმის ელემენტებთან წვდომისათვის function.php _ში ფუნქციის დამატება

მოძიებული ინფორმაციის გამოტანა ამ ეტაპზე შეუძლებელია, გამომდინარე იქიდან, რომ ფაილი სადაც უნდა მოხდეს შედეგის ჩვენება ფიზიკურად არ არსებობს.

თემის ფაილურ სტრუქტურაში საჭიროა search.php ფაილის დამატება. სწორედ ის უზრუნველყოფს გვერდის ფორმირებას, სადაც გამოვა მოძიებული ინფორმაცია. (იხ.სურ. 20.1) არ დაგავიწყდეთ მიმდინარე ფაილში **header** და **footer** ფუქციების შეტანა რათა არ დაირღვას საიტის სტრუქტურა! (1 და 12 სტრიქონები)

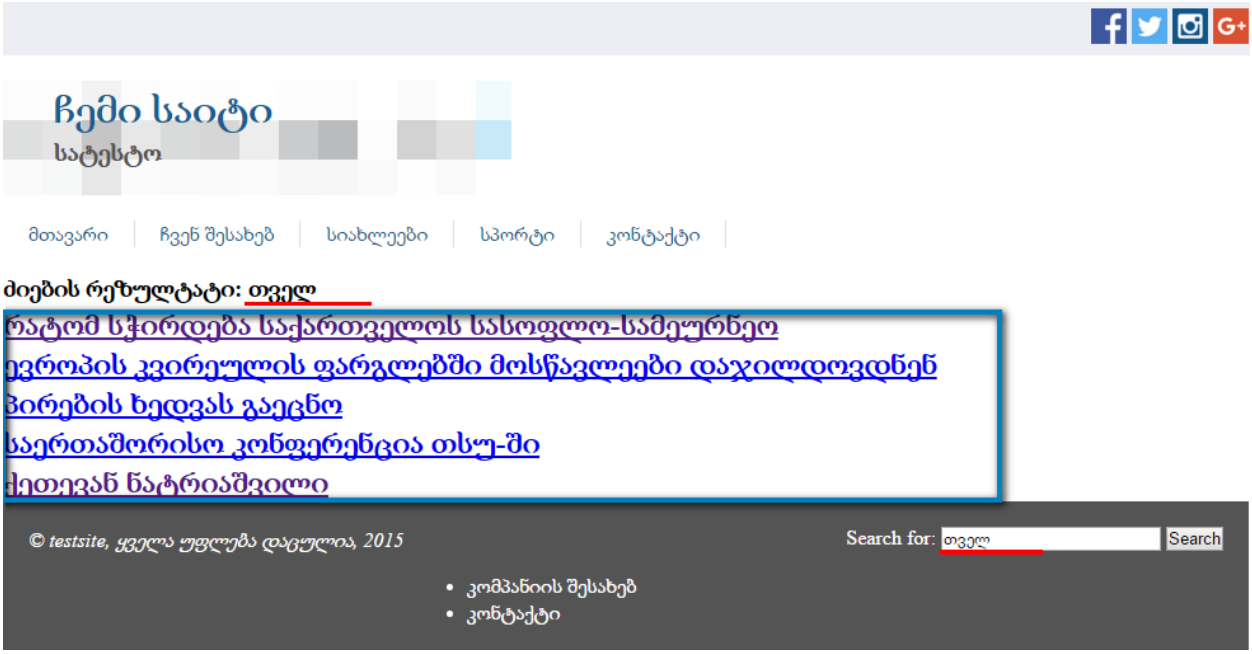
```php
<?php get_header();  ?>
<div class="container">
      <?php if ( have_posts() ) : ?>
        <header class="page-header">
          <h1 class="page-title"><?php printf( __( 'ძიების რეზულტატი: %s', "testsite" ), get_search_query() ); ?></h1>
        </header>
        <?php while ( have_posts() ) : the_post(); ?>
              <h2><a href="<?php the_permalink() ?>"><?php the_title(); ?></a></h2>
        <?php endwhile; ?>
      <?php endif; ?>
   </div>
<?php get_footer(); ?>
```

სურ. 20.1 საძიებო ფაილის search.php - მოწყობა

სურათ 20.1 გამოყოფილი სტრიქონებიდან 5 - ხაზზე გამოტანილი ფუნქცია უზრუნველყოფს საძიებო სიტყვის ან სიტყვათა თანწყობის გამოტანას. 7-9 სტრიქონებზე განთავსებული php ბრძანებები (ციკლის ოპერატორი) თქვენთვის უკვე ნაცნობია. ის გამოყენებული იყო index.php ფაილის სრულყოფისას და თავის თავში გულისხმობდა „წინაწარ განსაზღვრული ჩანაწერების“ გამოტანას. ციკლი search.php დოკუმენტშიც იდენტურად მუშაობს. მოცემულობის თანახმად საიტზე აისახება ყველა გვერდის სათაური, რომლის ნებისმიერი ტექსტუალური ველი (სათაური, მოკლე აღწერა, ვრცელი ტექსტი) შეიცავს საძიებო სიტვის მსგავს სიტყვათა ერთობლიობას.

რა თქმა უნდა შესაძლებელია ძიების შედეგად, არამხოლოდ სათაური, არამედ მოკლე აღწერა - **the_excerpt()**, ვრცელი ტექსტი - **the_content()**, მინიატურა - **the_post_thumbnail('thumbnail')** ან სხვა მნიშვნელობებიც წარმოვადგინოთ, ამისათვის საჭიროა შესაბამისი ფუნქციების ჩამატება 7-9 სტრიქონების დიაპაზონში.

ძიების რეზულტატი ასახულია სურათ 20.2_ზე.



სურ.20.2 ძიების რეზულტატი

ყოველივეს ზედმიწევნით ზუსტად შესრულებით შეიქმნება მინიმალური შესაძლებლობების მქონე თემა-შიდა გვერდებითა, კატეგორიებით და საძიებო გვერდით.

WordPress_ს ავტომატიზებული აქვს ნავიგაცია თუნდაც შეზღუდული ფაილური სტრუქტურიდან გამომდინარე. უმჯობესი იქნება მოკრძალებულ რჩევას თუ გაითვალიწინებთ მცირეოდენ დანამატს დაიტანთ თემაში ფაილების სახით. ეს უფრო მეტ შესაძლებლობას მოგცემთ იმოქმედოთ გასხვავებული სტილებით, რათა არ იყოს თითოეული გვერდის 100% იანი მსგავსება.

როგორც ცნობილია ადმინისტრირების პანელიდან შესაძლებელია Posts, Pages, Categories ინფორმაციული ბლოკების მართვა. შესაბამისად საიტზე გამოტანილი შემცველობა შესაძლებელია ვიზუალურად განხვავებულად იქნას წარმოდგენილი.

WordPress ძრავის შემთხვევაში category.php, page.php, single.php ფაილები გამოიყენება ზედა აბზაცში ჩამოთვლილი ბლოკების ინფორმაციის წარმოსაჩენად.

category.php

მაგ. სიახლეები კატეგორიაში შესვლისას საიტის დიზაინი არ უნდა ემთხვეოდეს მთავარი გვერდის იერსახეს. აქამდე არსებული მოცემულობით განსხვავებულად წარმოჩენა ფაქტობრივად შეუძლებელი იქნებოდა. ამიტომ სასურველია თუ შეიქმნება category.php სადაც მოხდება უკვე ნაცნობი ბრძანებებით სასურველი ჩონჩხის ფორმირება. (იხ.სურ.21.1)

ერთადერთი უცნობი დეტალი შეიძლება შესაძლოა 5 სტრიქონზე განთავსებული ფუნქცია იყოს. მიმდინარე ფუქცია რაღაცით მსგავსია search.php ფაილის ჩანაწერისა. ის უზრუნველყოფს კატეგორიის სათაურის გამოსახვას.

```
1  <?php get_header();  ?>
2  <div class="container">
3      <?php if ( have_posts() ) : ?>
4        <header class="page-header">
5          <h1 class="page-title"><?php printf( __( '%s', "testsite" ), single_cat_title( '', false )  ); ?></h1>
6        </header>
7        <?php while ( have_posts() ) : the_post(); ?>
8              <article class="newsIn fix">
9                <h3><a href="<?php the_permalink() ?>"><?php the_title(); ?></a></h3>
10               <div class="clear"> </div>
11                   <a href="<?php the_permalink() ?>"><?php the_post_thumbnail( 'thumbnail' ); ?></a>
12               <div class="clear"> </div>
13               <a href="<?php the_permalink() ?>" class="more">დაწვრილებით</a>
14             </article>
15
16       <?php endwhile; ?>
17     <?php endif; ?>
18   </div>
19 <?php get_footer(); ?>
```

სურ. 21.1 category.php ფაილის მოწყობა

7 – 14 სტრიქონებში ხდება სასრუველი ბლოკური ჩონჩხის ფორმირება კატეგორიის ვიზუალიზებისათვის.სურათ 22.2_ზე წარმოდგენილია პიველი index.php გვერდზე გამოსული სიახლეების და უშუალოდ მენიუს პუნქტ სიახლეებზე გადასვლისას გამოტანილი ინფორმაცია. დიზაინს შორის სხვაობის გარდა აშკარაა, რომ მთავარ გვრდზე სიალხე გამოდის მოკლე აღწერით ხოლო სიახლეების კატეგორიაში გადასვლისას სიხალე მხოლოდ სურათით და სათაურით გვევლინება. ეს ერთგავრი მაგალითია, რომელიც წარმოაჩენს category.php ფაილის საჭიროების მნიშვნელობას.

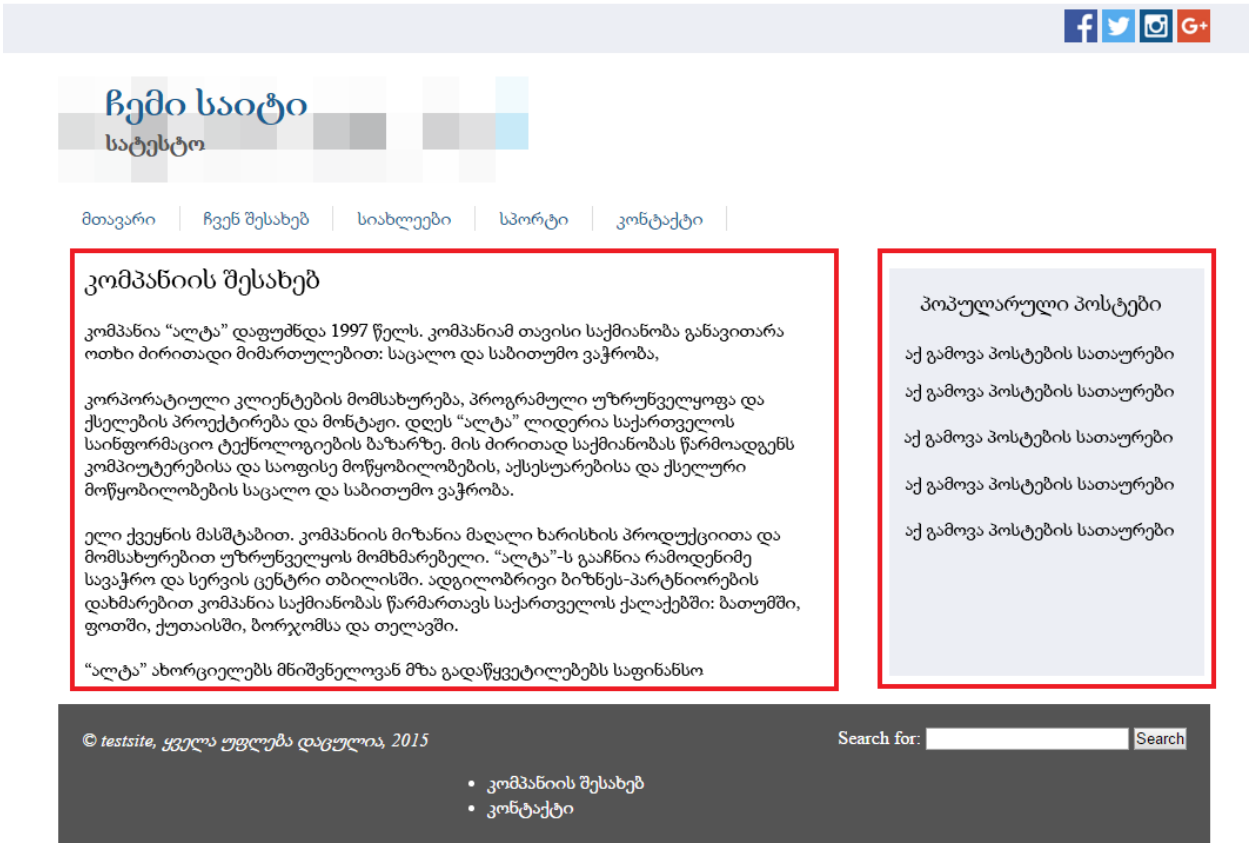
შესაბამისად თქვენზეა დამოკიდებული რამდენად საჭიროდ მიიჩნევთ ამ ფაილის შექმნას, ფაქტია მისი არასებობის შემთხვევაშიც შესაძლებელია ინფორმაციის გამოტანა საიტზე.

სურ.22.2 category.php ფაილითკატეგორიების მართვა.

page.php

თემის ფაილურ სტრუქტურაში page.php ფაილის შექმნით გეძლევათ შესაძლებლობა დამატებითი სტრუქტურა განუსაზღვროთ ადმინისტრირების პანელის ჩანართ **Pages**_დან ფორმირებული გვერდებებს. არ არის ლოგიკას მოკლებული მოსაზრება - შიდა გვერდის ბლოკთა სტრუქტურა განსხვავდებოდეს კატეგორიის, პოსტების ან სულაც მთავარი გვერდის ვიზუალისაგან. სწორედ ამ იდეის რეალიზება შესაძლებელია page.php ფაილის შექმნით.

მისი შემცველობის გაფორმება უშუალოდ საიტის დიზაინსა და თქვენს გემოვნებაზეა დამოკიდებული. შევეცადოთ page.php ფაილის შექმნა მაგალითზე (სურ. 23.1) ნაჩვენები მოცემულობიდან გამომდინარე.



სურ.23.1 შიდა გვერდის page.php მოცემულობა

სურათ 23.1_ის მოცემულობის თანახმად შიდა გვერდის ინფორმაცია არ ვრცელდება ვებ გვერდის 100%ზე. ის აკუმულირებულია მარცხენა მხარეს. ხოლო მარჯვენა მხარეს შემოტანილია ბლოკი **პოპულარული პოსტებით.**

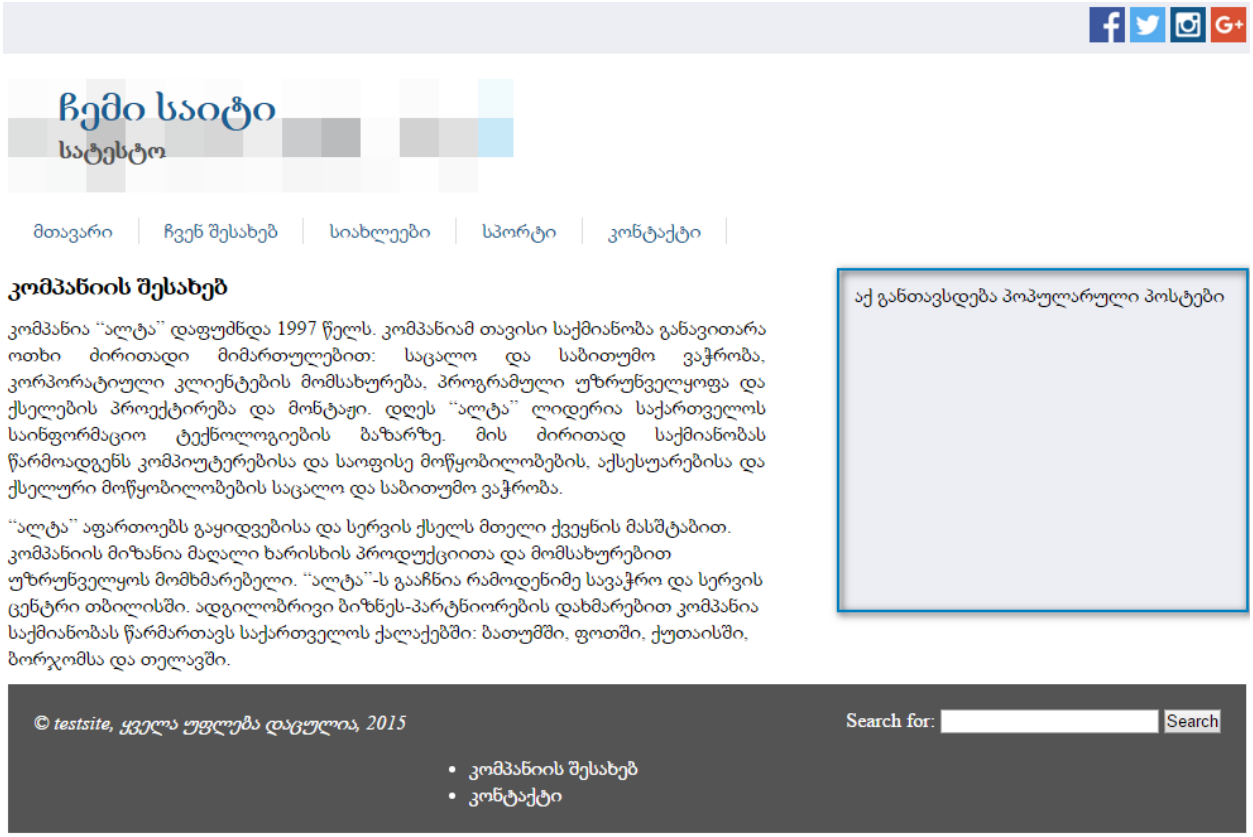
თქვენს საიტის არ გააჩნია ფუნქციონალი არსებული დავალების შესასრულებლად, ამიტომ მოვიქცეთ შემდეგნაირად. საწყის ეტაპზე თემის ფაილურ სტრუქტურაში მოვახდინოთ **page.php** ფაილის შექმნა. ფაილში განვსაზღვროთ ბლოკები გვერდის ინფორმაციული შემცველობისა და დამატებითი მოდულის (სურ.23.1 პოპულარული პოსტები) გამოსატანად. (იხ.სურ.23.2)

```
1  <?php get_header();  ?>
2  <div class="container">
3        <?php if ( have_posts() ) : ?>
4           </header>
5           <?php while ( have_posts() ) : the_post(); ?>
6                 <article class="page">
7                       <h1><?php the_title(); ?></h1>
8                       <div class="clear"> </div>
9                          <div class="information"><?php the_content(); ?><div>
10                      <div class="clear"> </div>
11                </article>
12          <?php endwhile; ?>
13       <?php endif; ?>
14       <section class="popPosts">
15       აქ განთავსდება პოპულარული პოსტები
16       </section>
17    </div>
18 <?php get_footer(); ?>
```

სურ. 23.2 page.php ფაილის შემცველობა

როგორც სურათ 23.2_დან ხედავთ ძირითადი ინფორმაციული ბლოკის ასახვას 3-13 სტრიქონებში მოთავსებული ჩანაწერები უზრუნველყოფს. ბლოკის html ჩონჩხის გარდა ამ სტრიქონებში გამოყენებულია თქვენთვის უკვე კარგად ნაცნობი მრავალჯერ გამოყენებული პირობითი ოპერატორი **if** და ციკლის ოპერატორი **while.** გვერდის სათაურის გამოტანას უზრუნველყოს როგორც აქამდე არსებულ ფაილებში **the_title()** ფუნქცია , ხოლო შემცველობას ასახავს **the_content()** ფუნქცია. დანარჩენი ჩონჩხის ფორმირება უკვე html ტეგების დამსახურებაა.

საიტზე მიღებული შედეგი ნაჩვენებია სურათ 23.3_ზე.



სურ. 23.3 სურათ 23.2ზე ნაჩვენები ბრძანებების მეშვეობით მიღებული შედეგი

დავალების დასმისას აღინიშნა რომ მარჯვენა ბლოკში სასურველია გამოტანილი იყოს პოპულარული პოსტები. ზოგადად მარჯვენა არეალი გავითვალისწინოთ ცვლადი ინფორმაციის გამოსატანად, ანუ დავუშვათ, ბლოკი განსაზღვრული იქნება სხვადასხვა ვიჯეტების გამოსატანად. (ვიჯეტები დაწვრილებით განხილულია **Appearance >widgets** ჩანართის განხილვისას). ვიჯეტების გარჩევისას მათ მნიშვნელობაზე საკმაოდ ვრცლად ვისაუბრეთ. ისინი უზრუნველყოფენ საიტის მრავალფუნქციურობას. რა თქმა უნდა შესაძლებელია პოგრამული კოდის ხელით დაწერა და პოპულარული პოსტების გამოტანა, მაგრამ ეს ყოველივე დიდ დროით რესურსს წაიღებს მაშინ როცა არსებობს მზა პლაგინი არსებული პრობლემის მოსაგვარებლად.

პლაგინის დაყენება არანაირ სირთულეს არ წარმოადგენს. თუ ყურადღებას გაამახვილებთ ახალი თემის პირობებში არ გაგაჩნით წვდომა **Appearance > widgets** ჩანართთან. დროა გავიხსენოთ მივიწყებული ფაილი function.php და მოვახდით მასში **sidebar** დამატების ფუქციის გააქტიურება, რაც ავტომატურად გამოიწვევს ვიჯეტების ჩანართის გააქტიურებას (იხ.სურ. 23.4)

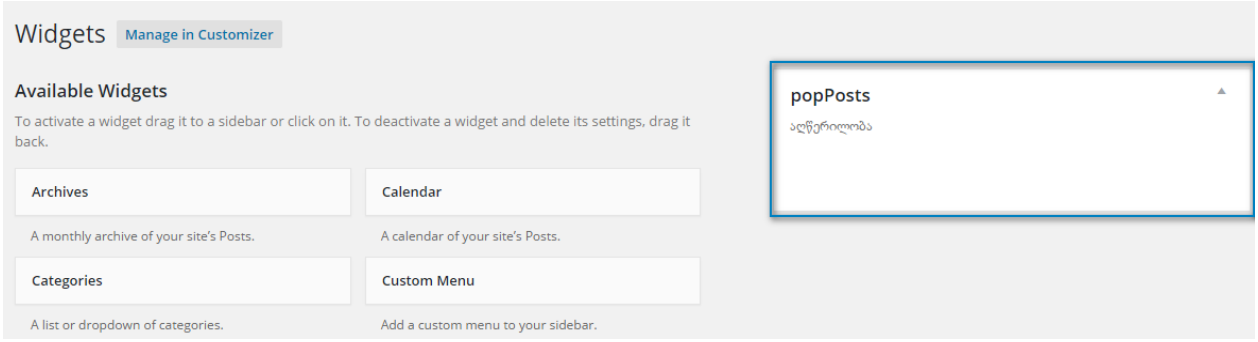
```
43 function testsite_widgets_init() {
44    register_sidebar( array(
45       'name'          => __( 'popPosts', 'testsite' ),
46       'id'            => 'common',
47       'description'   => __( 'აღწერილობა', 'testsite' ),
48       'before_widget' => '<aside id="%1$s" class="widget %2$s">',
49       'after_widget'  => '</aside>',
50    ) );
51 }
52 add_action( 'widgets_init', 'testsite_widgets_init' );
```

სურ.23.4 ახალი sidebar დამატება

ფუნქცია **register_sidebar()** უზრუნველყოფს ახალი სივრცის გამოყოფას. (იხ.სურ.23.5) მას პარამეტრად გადაეცემა მასივი (array()) სხვადასხვა მნიშვნელობებით. მაგ.: **name**- სახელი, **id** -იდენთიფიკატორი, **description** -მოკლე აღწერილობა, **before_widgets** - ვიჯეტის გამოტანისას მისი განსათავსებელი არეალი ანუ გამხსნელი ტეგი თავისი ატრიბუტებით, **after_widget** - დამხურავი ტეგი. მიმდინარე ფუნქციის შესახებ სრული ინფორმაციის მისაღებად ეწვიეთ მითითებულ მისამართს

https://codex.wordpress.org/Function_Reference/register_sidebar

ფუნქციის გააქტიურება იწვევს ადმინისტრირების პანელში ახალი არეალის დამატებას და თუ ის პირველი რეგისტრირებული sidebar_ია ააქტირუებს **Apprearance>widgets**.



სურ.23.5 ახალი sidebar არეალის დამატება

თუ გსურთ საიტის ფუნქციონალის მიზნით სხვადასხვა sidebar ბლოკების რეგისტრაცია დააკოპირეთ 44-50 სტრიქონებში მოთავსებული ფუნქცია **register_sidebar()**, შეცვალეთ სახელები და იდენთიფიკატორი id.

sidebar__ის რეგისტრაცია არ ნიშნავს მის ავტომატურ გამოსახვას საიტზე. საჭიროა გამოიყენოთ ფუნქცია, რომელის მეშვეობითაც გამოიძახებთ სასურველ Sidebar_ს საჭირო ადგილას.

არსებობს sidebar_ის გამოძახების ფუნქცია **dynamic_sidebar()**, რომელსაც არგუმენტად უნდა გადაეცეს ან სახელი ან იდენთიფიკატორი. ორივე შემთხვევაში გამოძახების ფუნქცია იმუშავებს იდენტურად (იხ.სურ. 23.6)

```
<?php dynamic_sidebar( 'popPosts' ); ?>
<?php dynamic_sidebar( 'common' ); ?>
```

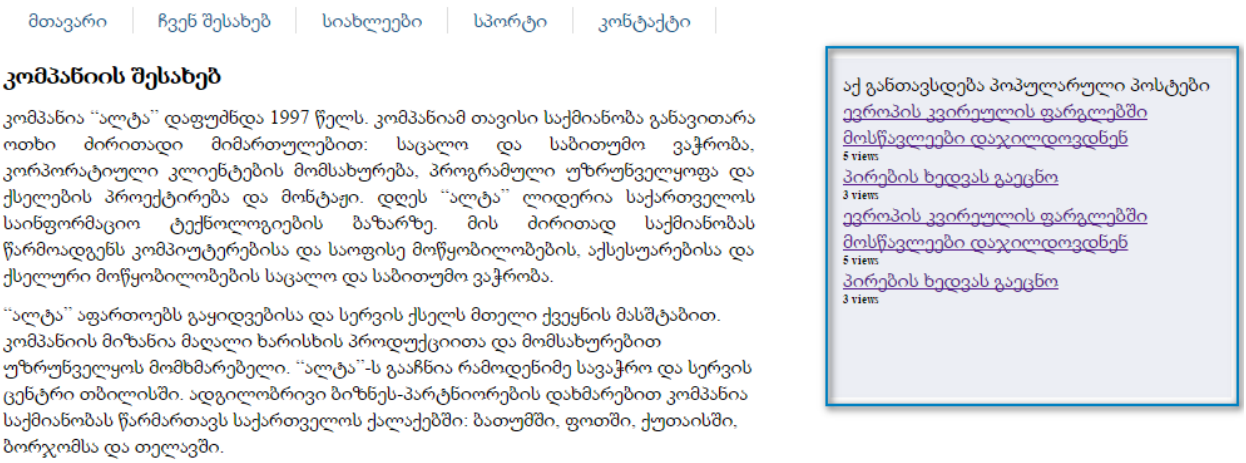
სურ.23.6 sidebar გამოძახების ფუნქცია

ფუნქციის შესახებ სრულფასოვანი ინფორმაციის მოძიება შესაძლებელია შემდეგ ბმულზე https://codex.wordpress.org/Function_Reference/dynamic_sidebar

პოპულარული პოსტების პლაგინის აქტივაციის შემდეგ, მას მოყვება ვიჯეტების ფუნქციონალი. ვიჯეტის დამატებით **popPosts** Sidebar_ში და მისი გამოძახებით page.php ფაილში (იხ.სურ. 23.7) მიიღება დავალების შესაბამისი საბოლოო შედეგი (იხ.სურ.23.8)

```
2  <div class="container">
3      <?php if ( have_posts() ) : ?>
4        </header>
5        <?php while ( have_posts() ) : the_post(); ?>
6            <article class="page">
7                <h1><?php the_title(); ?></h1>
8                <div class="clear"> </div>
9                  <div class="information"><?php the_content(); ?><div>
10               <div class="clear"> </div>
11           </article>
12       <?php endwhile; ?>
13     <?php endif; ?>
14     <section class="popPosts">
15     აქ განთავსდება პოპულარული პოსტები
16     <?php dynamic_sidebar( 'popPosts' ); ?>
17     </section>
18   </div>
19 <?php get_footer(); ?>
```

სურ.23.7 დასრულებული page.php. გამოყოფილი სტრიქონი - "popPosts" sidebar_ის გამოძახება



სურ 23.8 საიტის page გვერდების სტრუქტურა - დასრულებული იერსახე

როგორც page.php ფაილის შექმნის საწყის ეტაპზე აღვნიშნეთ საიტმა შეიძიანა მეტი ფუნქციონალური დატვირთვა. მიმდინარე ფაილი გეხმარებათ კონრეტული დავალების შესაბამისად თქვენთვის სასურველი ბლოკთა სქმების აწყობასა და საიტის უფრო მოქნილი ჩონჩხის გენერირებაში.

single.php

single.php მორიგი ფაილია, რომელიც დაგეხმარებათ პოსტების შიდა გვერდის სტრუქტურის ფორმირებაში. მიმდინარე ფაილის დეტალურ განხილვას აზრი არა აქვს, გამომდინარე იქიდან, რომ იგი იქმნება page.php ფაილის მსგავსად. თქვენს მიერ დაგროვილი ცოდნა მოგცემთ საშუალებას თავად დააგენერიროდ ეს კონკრეტული ფაილი თქვენი სურვილისამებრ. შესაძლოა მიმდინარე ფაილში სრულებით არ გჭირდებოდეთ ერთი sidebar, მასსში გამოიტანოთ სხვადასხვა ფუნქციების ერთობლიობა.

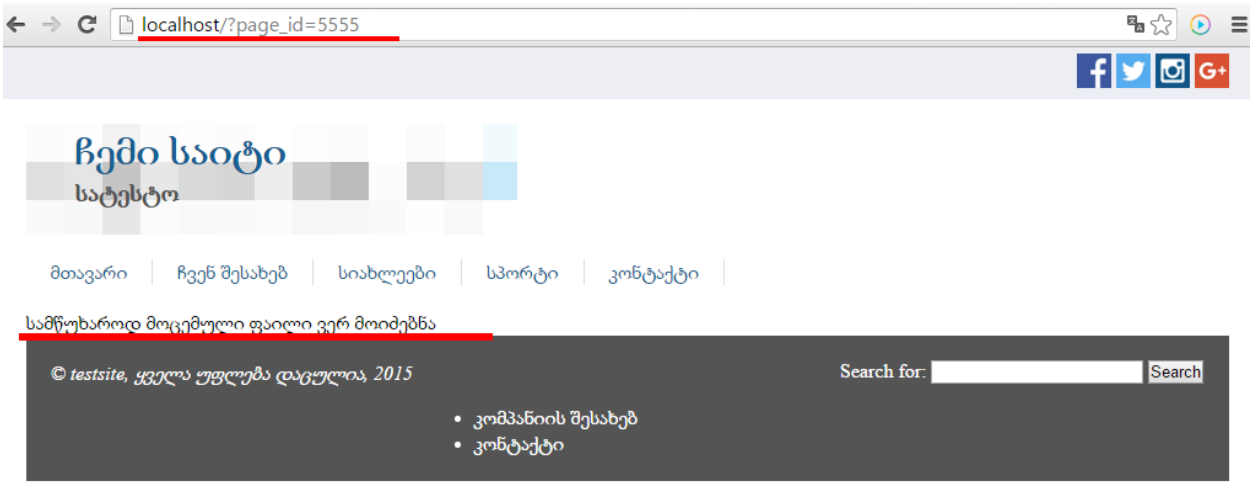
დაიმახსოვრეთ single.php ფაილი ასახავს პოსტებს ეკრანზე. ანუ ამ ფაილის მეშვეობით შესაძლებელია გამოიტანოთ ადმინისტრირების პანელის **Posts** ჩანართში არსებული ერთი კონრეტული ჩანაწერის ინფორმაცია.

404.php

სასურველია თუ თქვენი თემის ფაილურ სტრუქტურას მიამატებთ 404.php ფაილს, რომელიც გამოიძახება საიტზე დაშვებული შეცდომითი ბრძანების შედეგად და იქნება საინფორმაციო სახის. მაგალითისათვის სამისამრთე პანელში მოვითხოვოთ ისეთი გვერდის გამოძახება, რომელიც საერთოდ არ არსებობს ვებ გვერდზე და მინიშნების სახით გამოვიდეს შეტყობინება - „სამწუხაროდ ასეთი გევრდი არ მოიძებნა“. ეს მინიშნება სწორედაც რომ გამოტანილი იქნება 404.php ფაილში.(იხ.სურ. 24.1 - 24.2)

```
1 <?php get_header(); ?>
2    <h1>სამწუხაროდ მოცემული ფაილი ვერ მოიძებნა</h1>
3 <?php get_footer(); ?>
```

სურ.24.1 ფაილი 404.php



სურ.24.2 შეცდომის ასახვა საიტზე 404.php ფაილის მეშვეობით

როგორც სურათ 24.2 მაგალითზეა ნაჩვენები სამისამართე პანელში მოვითხოვეთ გვერდი id_ით **5555,** ბაზაში არ აღმოჩნდა არსებული მისამართის მქონე გვერდი და გამოიტანა შესაბამისად მინიშნება.

შეჯამება

WordPress_ წაროადგენს საკმაოდ მრავალფუნქციურ ძრავს. მისი მეშვეობით შესაძლებელია არამხოლოდ მარტივი ბლოგების, არამედ სრულფასოვანი სოლიდური საიტის შექმნა მცირე დროითი რესურსის ხარჯზე. როგორც ზემოთ განხილული ინფორმაციიდან ჩანს WordPress_ი საკმაოდ მოთხოვნადია მსოფლიო ბაზარზე, ეს თავისთავად საფრთხის შემცველია, გამომდინარე მის მიმართ დიდი მოთხოვნისა არსებობს დიდი საფრთხე CMS_ის არაკეთილმოსურნეთა მიერ განხორციელებული კიბერ შეტევების სიმრავლისა. გაითვალისწინეთ მეგობრული რჩევა და მაქსიმალურად დაიცავით საიტი მსგავსი თავდასხმებისაგან, გაეცანით დაწვრილებით სხვადასხვა საიტებზე შემოტავაზებებს და რჩევებს თავდაცვასთან დაკავშირებით.

თუ ყველაფერს ზედმიწევნით ზუსტად გაარჩევთ საფრთხე ფაქტობრივად მინიმუმამდე იქნება დაყვანილი და თქვენ მიიღებთ საუკეთესო მრავალფუნქციურ ძრავს.

საიტის აწყობა არ ნიშნავს იმას, რომ ის უკვე ნებისმიერი მომხმარებლის მიერ იქნება წვდომადი. საჭიროა საიტის განთავსება „ინტერნეტ სივრცეში“. (ეს ტერმინი დიდად არ შეესაბამება რეალურ განმარტებას უბრაკოდ უკეთ გასაგებია). განსათავსებლად საჭიროა რამოდენიმე პროცედურის გავლა:

- უნდა გქონდეთ დომენი;
- საჭიროა ჰოსტინგი;
- მნიშვნელოვანია ჰოსტინგისა და დომენის ბმა;
- და ბოლოს საჭიროა საიტის ფაილების ატვირთვა ჰოსტინგზე;

Domain - დომენი საიტის სახელწოდებას რომლის მეშვეობითც ხდება მომხმარებლის მიერ საიტის მოძიება. მაგ.: google.com, gmail.com, emis.ge და ა.შ. მომხმარებლის მიერ სამისამართე პანელში საიტი სახელის შეყვანისას კომპიუტერი უკავშირდება **DNS** (Domain Name System) სერვერს. გამომდინარე იქიდან რომ მანქანური ენა ვერ აღიქვამს ტექსტუალურ ბრძანებებს დომენეს რეალურ რეჟიმში შეესაბამება რიცხვითი სიმბოლიკა. ე.წ. **IP** მისამართი.

მომხმარებლის მიერ დომენის მოთხოვნისას საქმეში ერთვება DNS სერვერი რომელიც იღებს მოთხოვნას ტექსტუალურ ფორმატში და აბრუნებს რიცხვით მისამართს. მაგ.: google.com = 213.157.222.187 შეესაბამება შემდეგი IP მისამართი . სატესტო რეჟიმში შეგიძლიათ სამისამართე პანელში შეიყვანოთ მიმდინარე IP და ნახოთ, თუ რომელი საიტის გამოძახება მოხდება.

Hosting - ჰოსტინგი წარმოადგენს სერვერზე გამოყოფილ სივრცეს, სადაც ხდება საიტის განთავსება.

DNS სერვერბის მისამართები მოცემულია უშუალოად ჰოსტინგ კომპანიების მიერ, რომელთაგანც თქვენ ქირაობთ ჰოსტს და მათი შესაბამის ადგილას გაწერის შემთხვევაში ჰოსტინგი და დომენი დაუკავშირდებიან ერთმანეთს.

რაც შეეხება ფაილების ატვირთავს მას ვრცლად შევეხებით შემდგომ ქვეთავებში.

7.1 ვებსერვერის FTP ანგარიშისა და მონაცემთა ბაზების გამართვა

მიმდინარე პარაგრაფის თემატიკა

- ჰოსტინგის მართვის პანელიდან FTP ანგარიშის შექმნა
- ჰოსტინგის მართვის პანელიდან FTP მართვა
- ჰოსტინგის მართვის პანელიდან ბაზების შექმნა
- ჰოსტინგის მართვის პანელიდან ბაზების მართვა
- ჰოსტინგის მართვის პანელიდან ბაზების დაკავშირება

საიტის ჰოსტინგის თქვენს მმართველობაში გადმოსვლის შემდგომ შესაძლებელია მასზე სხვადასხვა მანიპულაციების ჩატარება. ჰოსტინგთან მომხამრებლის წვდომისათვის საჭიროა მომხარებელი და პაროლი, რომელსაც გამოყოფს ჰოსტინგ კოპანია. სხვადასხვა ჰოსტი შესაძლოა მოიხმარდეს განსხვავებულ სამართავ პანელს. საკმაოდ მძლავრ სამართავ პანელად ითვლება cPanel, ასევე არის რამოდენიმე ფართოდ გავრცელებული ჰოსტინგის სამარტავი პანელი Directadmin, ISPmanager, Usermin და ა.შ.

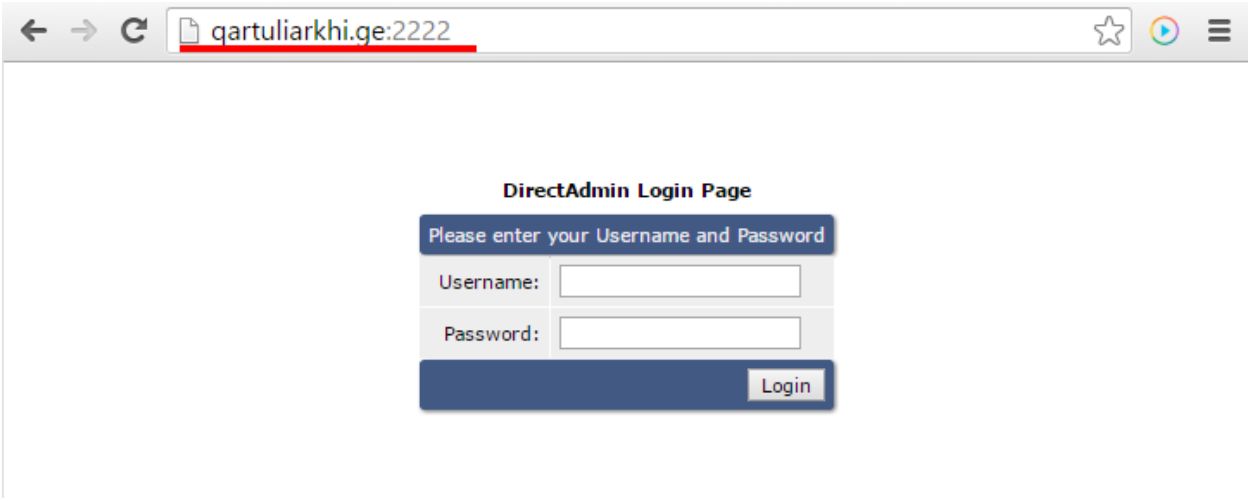
ქართული ჰოსტინგ კომპანიები ძირითადად cPanel_ს და DirectAdmin_ს მოიხმარენ.

cPanel

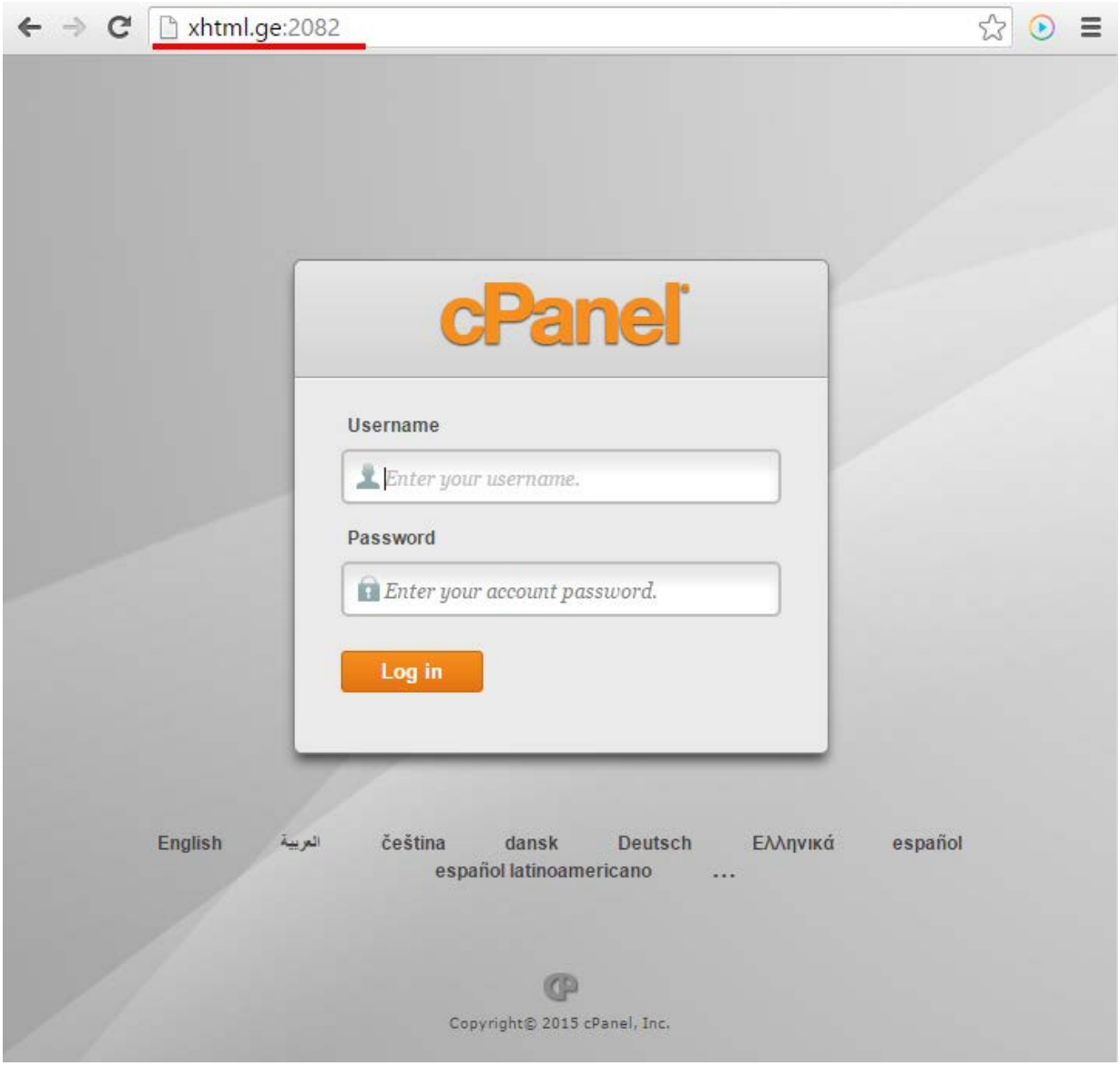
DirectAdmin

ჰოსტინგის მართივის პანლზე მოსახვედრათ უნდა აკრიფოთ საიტის დომენი : სამართავი პანელის პორტი. მაგ.: cpanel_ის პორტი 2082_ია ხოლო DirectAdmin 2222.

(იხ.სურ.25.1 – 25.2).



სურ25.1 საიტის DirectAdmin მართვის პანელის გამოძახება



სურ25.2 საიტის მართვის პანელის cPanel_ის გამოძახება

მომხმარებლის Username და Password განსაზღვრულია საწყის მნიშვნელობად ჰოსტინგის პროვაიდერის მიერ, მათი ცვლილება შესაძლებელია ავტორიზაციის შემდგომ.

ავტორიზაციის გავლის შემდეგ თქვენს წინაშე გამოისახება სხადასხვა დიზაინის და შესალებლობების მქონე სამართავი პანელი. რა თქმა უნდა ძირითადი ფუნქცონალი ფაქტობრივად ერთნაირია ორივე სამართავ პანელში , ყოველ შემთხვევაში თემებები, რომელსაც ჩვენ შევეხებით.

რა არის FTP?

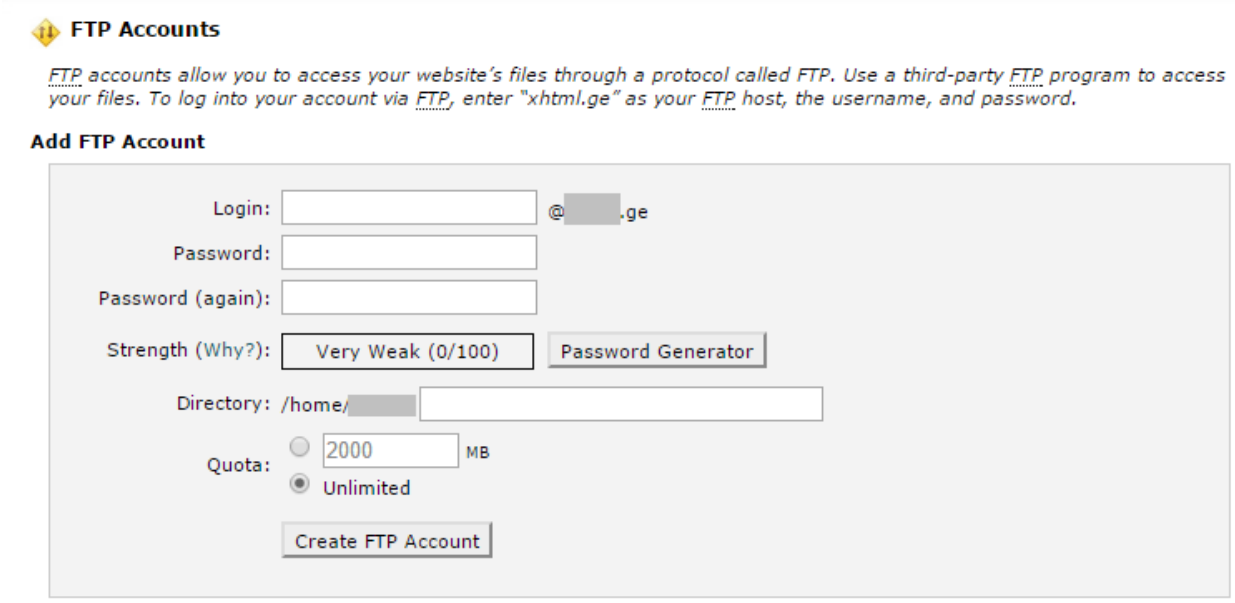
File Transfer Protocol - ფაილების გადაცემის პროტოკოლი. ეს არის წესი რომელიც არეგულირებს ფაილების, შესაბამისად ინფორმაციის გაცვლას. უფრო მარტივ ენაზე ftp გამოიყენება ფაილების გადასაცემად ინტერნეტში, ფაილების გადასატანად ჰოსტინგზე.

ახალი ftp არხის გახსნის შემთხვევაში თქვენ უზრუნველყობთ მომხმარებლის რეგისტრაციას, რომელსაც წინასწარ განსაზღვრული უფლებების შესაბამისად შეეძლება ჰოსტინგის კონკრეტულ არეალთან წვდომა და მასში ფაილების მართვა.

ftp ანგარიშის შექმნა cPanel სამართავი პანელის მაგალითზე - ვიძახებთ განყოფილება **Files**_ში **FTP Accounts** ჩანართს(იხ.სურ. 25.3)

სურ 25.3 FTP ანგარიშებზე გადასვა

მოხდება ახალი მომხარებლის ანგარიშის დამატებაზე გადამისამართება (იხ.სურ. 25.4)

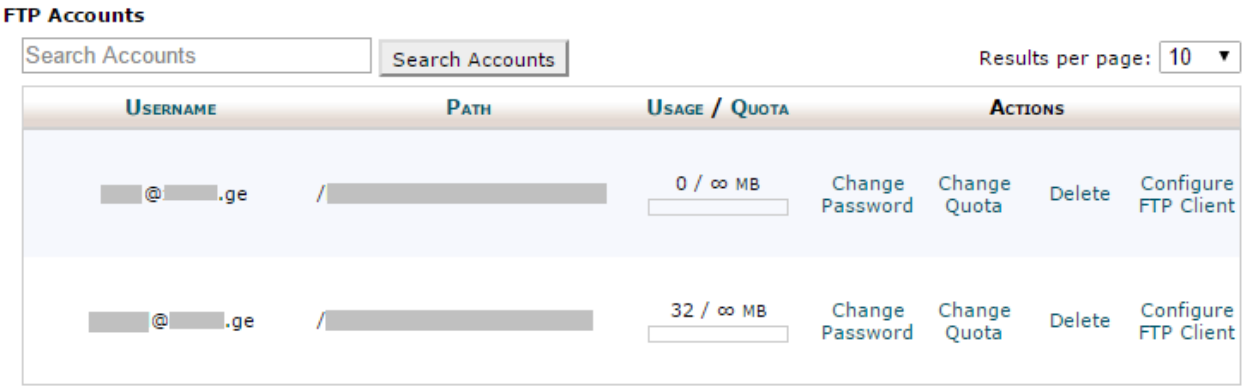


სურ,25.4 ახალი ftp ანგარიშის დამატება

შესაბამის ბლოკში უნდა მიეთითოს:

- Login - ახალი ანგარიშის სახელი;
- Password - პაროლი;
- Password (again) - გაიმეორეთ პაროლი;
- Strength - წარმოაჩენს პაროლის სიძლიერეს - Password Generator_ის მეშვეობით შესაძლებელია რთული პაროლის დაგენერირება;
- Directory- კონკრეტული მომხარებლისთვის თუ რომელ საქაღალდესთან იქნება წვდომა შესაძლებელია, რა თქმა უნდა შესაძლებელია იყოს ძირეული საქაღალდე public_html, ან რომელიმე კონკრეტული საქაღალდე.
- Quota - ატვირთული ინფორმაციის რაოდენობის განსაზვრა MB_ში. შესაძლებელია დავტოვოთ Unlimited- ულიმიტო, ან განვსაზღვროთ კონკრეტული სიდიდე.
- Create FTP Account - ახალი ანგარიშის დამატება.

ახალი FTP ანგარიშის დამტების ბლოკის ქვემოთ განთავსებულია ყველა გახსნილი FTP ანგარიში (იხ.სურ. 25.5)



სურ 25.5 FTP ანგარიშები და მათი მართვა

სურათ 25.5 ზე მოცემულია 2 FTP ანგარიში. თითეული თავისი სახელით, წვდომის საქაღალდით (მიმდინარე მაგალითზე დამალულია ეს სტრიქონები), ასევე მითიტებულია თითოეული ანგარიშის ლიმიტი - Quota და არსებული რაოდენობა (32/∞MB). ერთ ერთ მომხამრებელს ულიმიტო სივრციდან ათვისებული აქვს 32MB.

აქვეა შესაძლებელი პროლის შეცვლა - **Change Password,** ლიმიტის ცვლილება - **Change Quota,** ანგარიშის წაშლა **-Delete.**

დაიმახსოვრეთ FTP ანგარიშის წაშლა არ ნიშნავს ავტომატურად იმ საქაღალდის და ფაილების განადგურებას, რომელიც მის მიერ იყოს შექმნილი / შევსებული.

მონაცემთა ბაზები

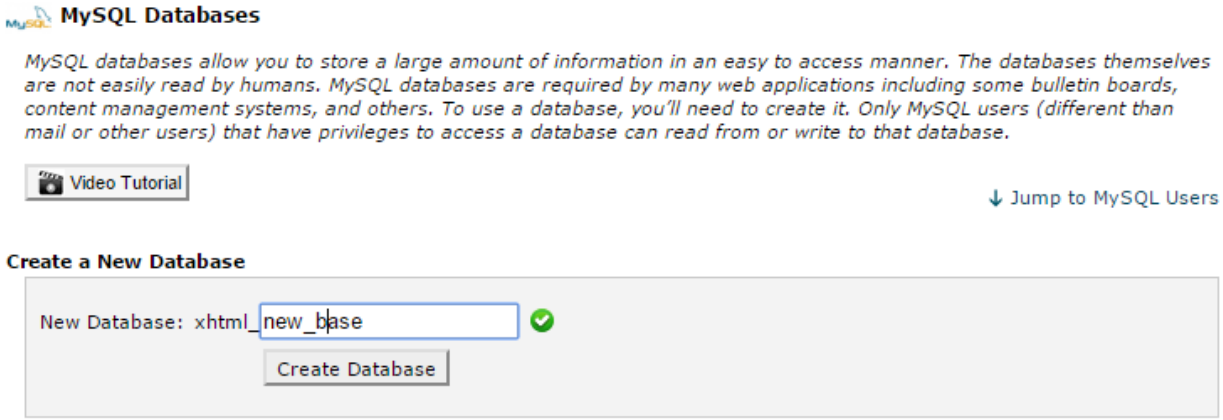
მონაცემთა ბაზა წარმოადგენს ინფორმაციის შემნახველი ცხრილების ერთობლიობას.

მონაცემთა ბაზის შექმნა და არსებულის მართვა cPanel_ში საკმაოდ მარტივი პროცესია - საჭიროა გამოიძახოთ **Databases** განყოფილებიდან ჩანართი **MySQL Databases** (იხ.სურ.25.6)

სურ 25.6 მონაცემთა ბაზების მართვა

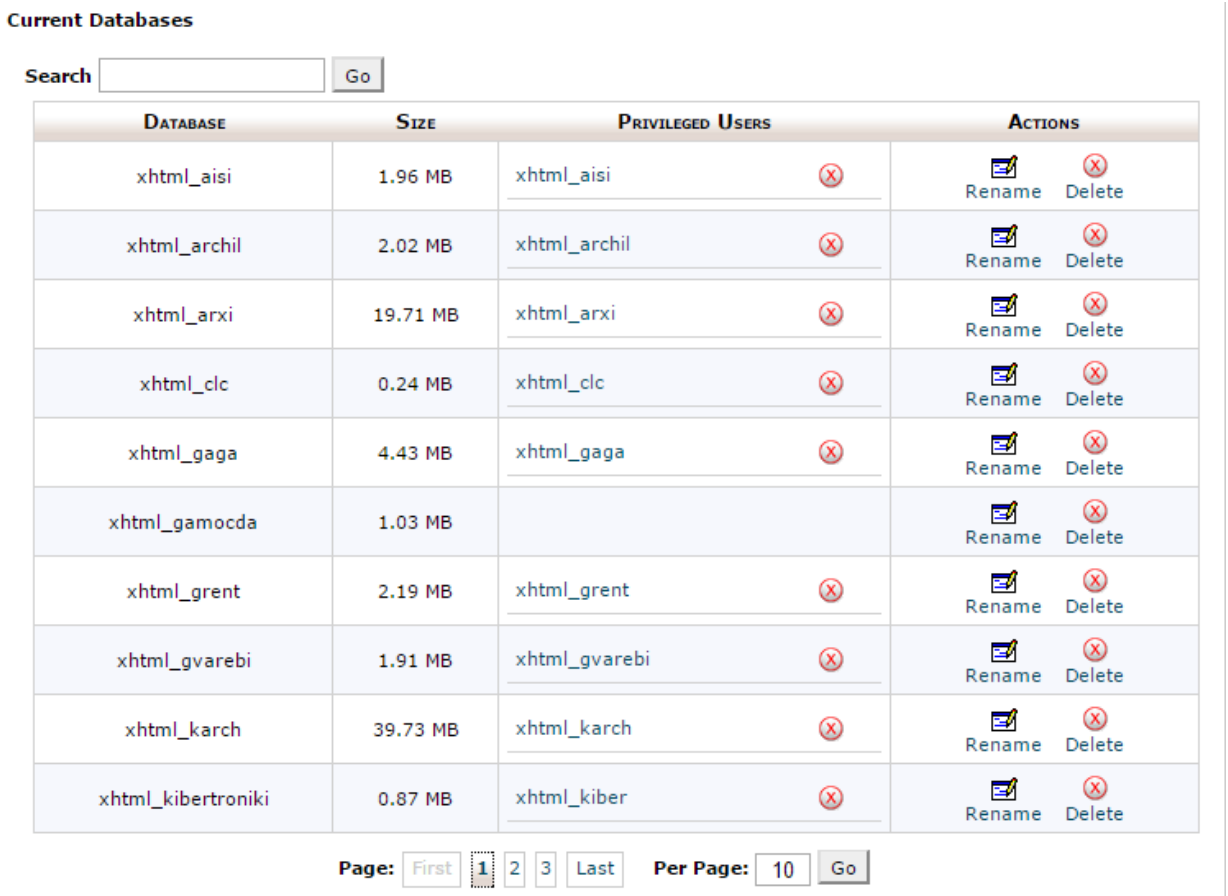
ახალი მონაცემთა ბაზის დამატება შესაძლებელია **Create a New Database** ბლოკის მეშვეობით. სადაც **New Database** მიეთითება ახალი მონაცემთა ბაზის სახელი. რეკომენდირებულია მონაცემთა ბაზის სახელი შედგებოდეს ლათინური ანბანის ანოსებისაგან, სახლი იყოს უწყვეტ ფორმატში დაწერილი, 2 და მეტ სიტყვიანი მონაცემთა ბაზის სახელები გამოყავთ ქვედა ტირით, სასურველია სიტყვების რაოდენობა არ აჭარბებდეს 5 სიტყვას.

Create Database - ღილაკის მეშვეობით ხდება ახალი ბაზის დამატება. (იხ.სურ 25.7)



სურ 25.7 ახალი მონაცემთა ბაზის დამატება

უკვე რეგისტრირებული მონაცემთა ბაზები წარმოდგენილია **Current Databases** ბლოკში.


Current Databases

Search Go

Database	Size	Privileged Users	Actions
xhtml_aisi	1.96 MB	xhtml_aisi	Rename Delete
xhtml_archil	2.02 MB	xhtml_archil	Rename Delete
xhtml_arxi	19.71 MB	xhtml_arxi	Rename Delete
xhtml_clc	0.24 MB	xhtml_clc	Rename Delete
xhtml_gaga	4.43 MB	xhtml_gaga	Rename Delete
xhtml_gamocda	1.03 MB		Rename Delete
xhtml_grent	2.19 MB	xhtml_grent	Rename Delete
xhtml_gvarebi	1.91 MB	xhtml_gvarebi	Rename Delete
xhtml_karch	39.73 MB	xhtml_karch	Rename Delete
xhtml_kibertroniki	0.87 MB	xhtml_kiber	Rename Delete

Page: First 1 2 3 Last Per Page: 10 Go

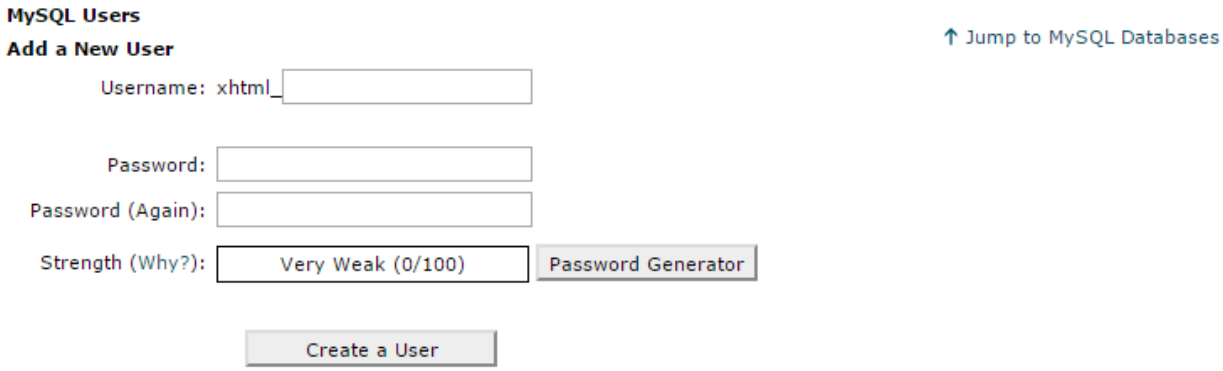
სურ.25,8 არსებული მონაცემთა ბაზები

მიმდინარე ბლოკი აღჭურვილია საძიებო ფორმით, რომელიც უზრუნველყოფს ბაზების მარტივად ძიებას. ასევე ცხრილით სადაც გამოტანილია მონაცემთა ბაზები სხვადასხვა სამართავი პარამეტრებით და ცხრილის ბოლოში გამოტანილია ბაზების ვიზუალიზაციის ფილტრი. სურათ 25.8 მოცემულობით თითიეულ გვერდზე გამოტანილია 10 ბაზა და სულ შექმნიალია 3 გვერდი.

თითიეული ბაზის გასწვრივ გამოტანილია სხვადასხვა პარამეტრები:

- **Database** - ბაზის სახელი;
- **Size**- ბაზის ზომა;
- **Privileged Users** - წევრთა პრივილეგიები (შემდგომ პუნქტში დაწვრილებით განვიხილავთ);- შესაძლებელია მომხმარებლის წაშლა- უფლებელის ცვლილება;
- **Actions** - აერთიანებს ბაზის სახელის გადარქმევისა - Rename და წაშლის - Delete ფუნქციონალს

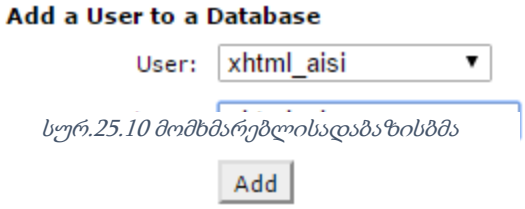
მონაცემთა ბაზას სჭირდება მომხმარებელი რომელსაც ექნება წვდომა მიმდინარე ბაზის ფუნქციონალთან. შესაძლოა ერთ კონკრეტულ მომხამრებელზე რამოდენიმე მონაცემთა ბაზა იყოს დამოკიდებული. მომხმარებლებზე სრული ინფორმაცია წარმოდგენილია **MySQL Users** ბლოკში. თუ ახალი user_ის რეგისტაცია გვჭირდება მონაცემთა ბაზასთან მომავალი დაკავშირებისათვის მივმართავთ მიმდინარე ბლოკის **Add a New User** (ჩანართს იხ.სურ. 25.9)



სურ. 25.9 მონაცემთა ბაზის ახალი მომხმარებლის დამატება

ახალი მომხარებლის დამატება სტანდარტული პროცესია რომელიც ზედა თავებშია განხილული და მასზე დეტალურად აღარ ღირს გაჩერება

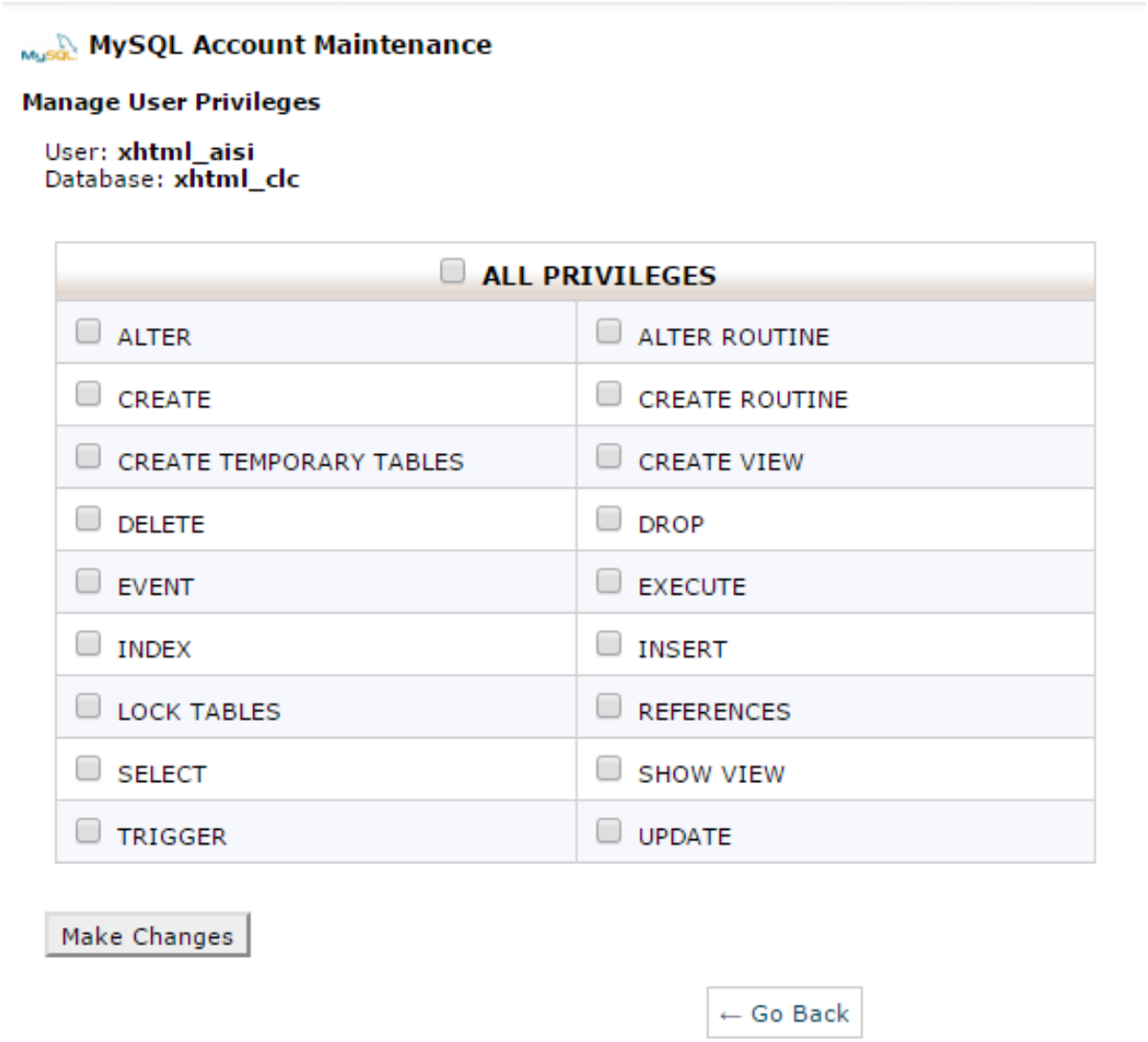
არსებული მომხმარებლის ბმა კონკრეტულ მონაცემთა ბაზასთან შესაძლებელია შემდეგ ბლოკ **Add a User to a Database** _ში სადაც საჭიროა აიჩიოთ მომხმარებელი და ბაზა. **Add** ღილაკის გამოყენებით დაადასტურეთ პროცესის გაგრძელება. (სურ. 25.10)



სურ.25.10 მომხმარებლისადაბაზისბმა

სურათ 25.10 მოცემულობის თანახმად მომხმარებელი **xhtml_aisi** ხდება მმართველი **xhtml_clc** მონაცემთა ბაზისა.

Add ღილაკი საბოლოო აკორდი არ არის ბმის დასრულებისა. ის გადაგანაცვლებთ შემდეგ ნაბიჯზე, სადაც უნდა მოხდეს მიმდინარე მომხმარებლისათვის ამ კონკრეტული ბაზის მიმართ გამოყენებული უფლებების განსაზღვრა (იხ.სურ.25.11)



2სურ. 25.11 მომხამრებლის უფლებების განსაზღრა

ALL PRIVILEGES - მომხმარებლისთვის ყველა უფლების მინიჭება. მიმდინარე პუნქტების აქტივაციისას მომხამრებელს ეძლევა სხვადასხვა უფლებები. მაგ.: მხოლოდ შექმნის CREATE, ცვლილების ALTER, წაშლის DELETE, ინფორმაციის ამორების - SELECT, განახლების - UPDATE და ა.შ.

Make Changes -ღილაკი საბოლოო უფლებებს ანიჭებს მომხმარებელს და ხელს უწყობს პროცესის დასრულებას.

არსებული მომხმარებლების პარამეტრების მართვა შესაძლებელია **Current Users** ჩანართიში არსებული **Actions** ბლოკის მეშვეობით (იხ.სურ.25.12), სადაც :

- **Set Password** - ახალი პაროლის მინიჭება;
- **Rename**- მომხამრებლის სახლელის ცვლილება
- **Delete** - მომხმარელი წაშლა;

Current Users

Users	Actions
xhtml_aisi	Set Password Rename Delete
xhtml_archil	Set Password Rename Delete
xhtml_arxi	Set Password Rename Delete
xhtml_clc	Set Password Rename Delete

სურ.25.12 არსებული მომხამრებლების ცხრილი

7.2 ფაილების ატვირთვა სერვერზე

მიმდინარე პარაგრაფის თემატიკა

- ვებსერვერთან წვდომა FTP კლიენტის მეშვეობით
- ვებსერვერზე საქაღალდეების შექმნა
- ვებსერვერზე ფაილების ატვირთვა
- ვებსერვერზე საქაღალდეების უფლებების მართვა (chmod)

ჰოსტინგზე განტავსებული საიტის ფაილური სტრუქტურის სამართავად ფართოდ გამოიყენება სხვადასხვა პროგრამული უზრუნველყოფა. მცირეოდენი შესწორების შესატანად არაკომფორტულია ყველა ჯერზე ფაილურ სტრუქტურასთან კავშირის დამყარება ჰოსტინგის სამარტავი პანელის მეშვეობით. ხშირ შემთხვეაში ვებსერვერთან წვდომის გასამარტივებლად მოიხმარენ FTP კლიენტს.

FTP კლიენტი წარმოადგენს პროგრამულ უზრუნველყოფას (პროგრამას) მარტივი სამართავი სტრუქტურით. მსოფლიოში მრავალი ამდაგვარი პროგრამული უზრუნველყოფა არსებობს და დეტალურად მათი დადებითი და უარყოფითი მხარეების გაცნობა უშუალოთ მათი მოხმარების წინ უნდა გაარჩიოთ. არსებობს ფართოდ მოხმარებადი FTP კლიენტები, როგორებიცაა მაგ.: **FileZilla**, **Total Commander, WinSCP, SmartFTP**და ა.შ .

მათი მუშაობის პრინციპი ფაქტობრივად ერთგვაროვანია , აქედან გამომდინარე ერთი კონკრეტული FTP კლიენტერბის მუშაობის არსის ცოდნა ნიშნავს დანარჩენზე წარმოდგენის ქონას. ჩვენ შემთხვევაში განხილული იქნება FTP კლიენტი **FileZilla.**

FileZilla კლიენტის მოძიება / დაყენება საკმაოდ მარტივი პროცესია. მიჰყევით სურათ 26.1 – 26.3 მოყვანილ ინსტრუქციას და ყოველგარი დაბრკოლების გარეშე სულ რამოდენიმე წუთში პროგრამული უზრუნველყოფა იქნება თქვენს განკარგულებში.

საწყის ეტაპზე საჭიროა პროგრამის მოძიება. FileZilla ფართოდ გავრცელებული რესურსია, ამიტომ მისი მოძიება საძიებო სისტემებში სირთულეს არ წარმოადგენს. სასურველია გამოყენებულ იქნას ოფიციალური საიტი პროგრმაის ჩამოსაწერად. (იხ. ბმული https://filezilla-project.org/) (იხ.სურ.26.1)

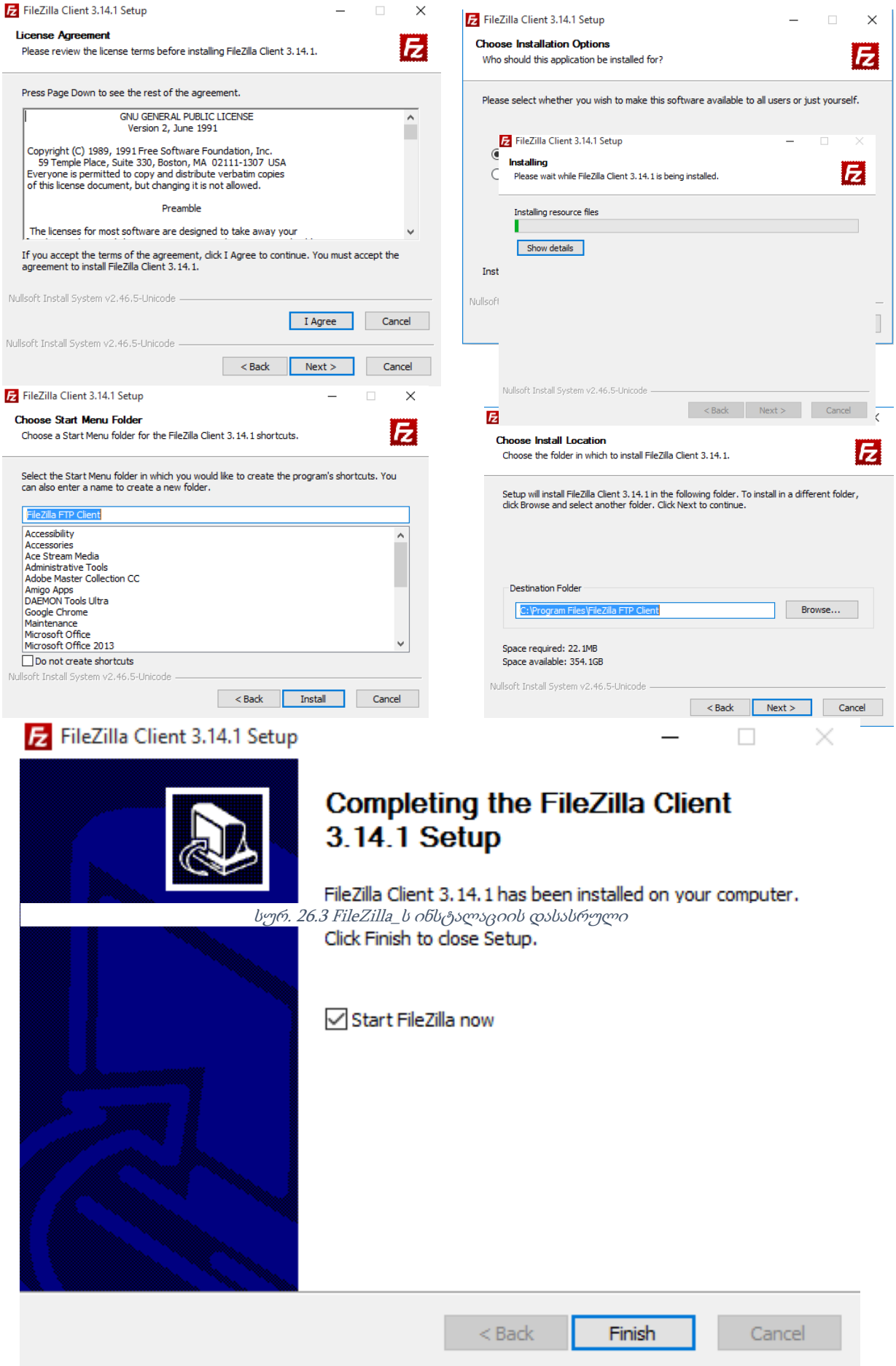
სურ. 26.1 FileZilla _ოფიციალური ვებ გვერდი

სურათ 26.1 ზე გამოყოფილი ბლოკი უშუალოდ FTP კლიენტის ჩამოსაწერ ბმულზე მოახდენს გადამისამართებას. (იხ.სირ 26.2).

ვსურ 26.2 FileZilla პროგრამის ჩამოწერა

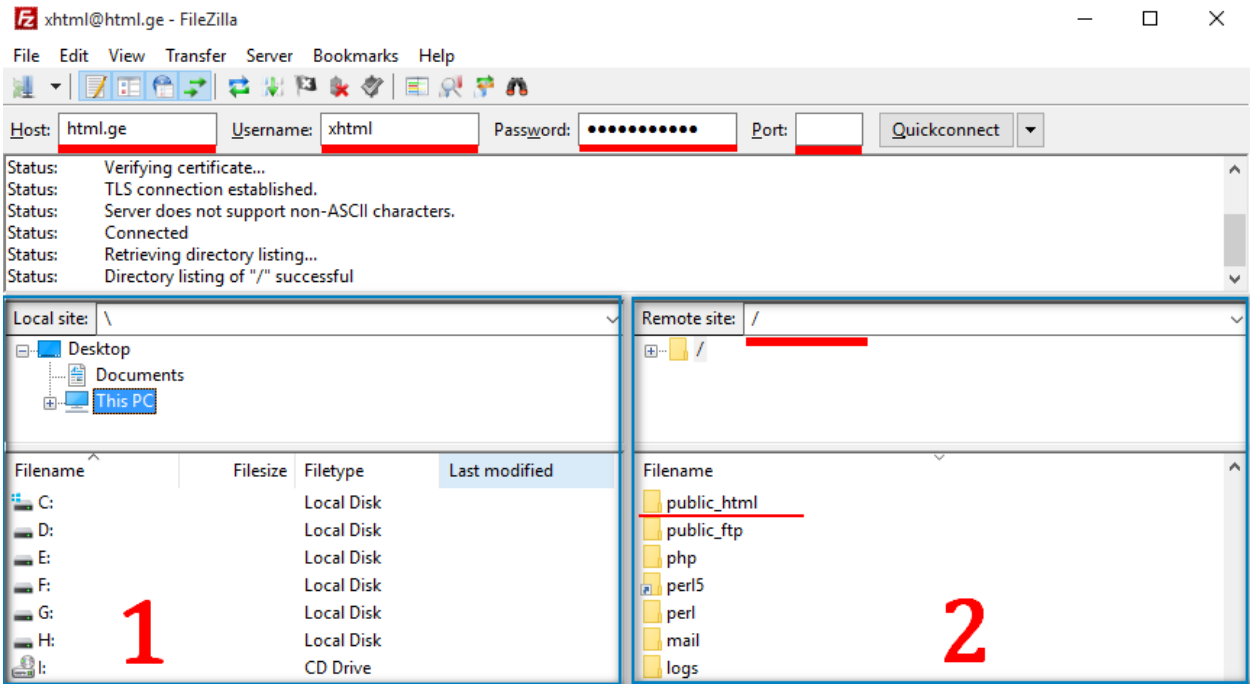
სურათ 26.2_ზე გამოყოფილი ბმულის აქტივაცია და პროგრამა თქვენს ხელთაა...

პროგრამა გამშვები ფაილის სახით ჩამოიწერება და დამატებით მანიპულციებს არ საჭიროებს მხოლოდ გააქტიურება გევალებათ. შემდგომი ნაბიჯები ნაჩვენებია სურათ 26.3_ზე



სურ. 26.3 FileZilla_ს ინსტალაციის დასასრული

ინსტალაციის დასრულების შემდეგ შეგიძლიათ გამოიძახოთ პროგრამა და მოხდინოთ ვებ სერვერთან დაკავშირება. (იხ.სურ. 26.4)



სურ.26.4 ვებ სერვერთან კავშირის დამყარება

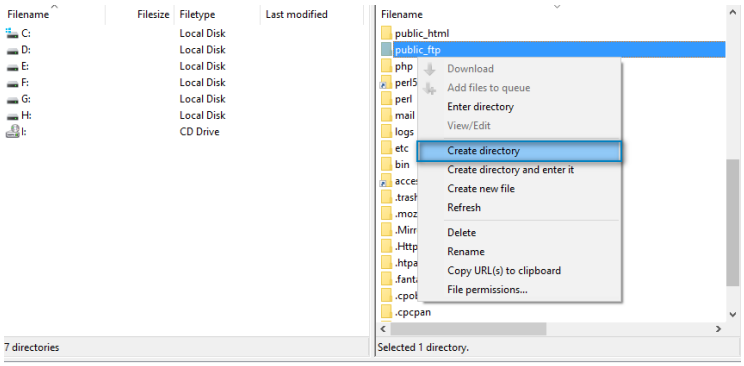
სურათ 26.4 _ზე გამოყოფილია ის მნიშვნელოვანი ბლოკები, რომლებიც უშუალოდ მონაწილეობენ სერვერზე ფაილების ატვირთვაში. საწყის ეტაპზე სერვერთან კავშირის დასამყარებლად საჭიროა შემდეგი პარამეტრების ზედმიწევნით ზუსტად შესრულება:

- **Host** - ჰოსტის მისამართი (მაგალითი იხილეთ სურათზე);
- **Username** - მომხმარებლის სახელი;
- **Password** - პაროლი
- **Port** - პორტის მისამართი. **FTP**-ს შემთხვევაში default მნიშვნელობა **21**, ხოლო **SFTP**შემთხვევაში **22.**

კავშირის წარმატებით დამყარების შემდეგ პროგრამის **მე_2 ბლოკი** (იხ.სურ. 26.4) შეივსება ფაილური სტრუქტურით.

ახალი FTP_ს დამატებისას ჩვენს მეირ მოხსენიებული იყო **Directory** ჩანართი. სწორედ იმ ფაილთა სტრუქტურასან ექნება წვდომა დამაკავშირებელ მომხარებელს. სურათ 24.6 ზე მოცემულ შემთხვევაში მომხმარებელს აქვს **ROOT Directory**_ან წვდომა.

მეორე ბლოკის ზედა წითელი ხაზგასმა მიმდიარე ადგილმდებარეობისმანიშნებელია



სურ. 26.5 სერვერზესაქაღალდისშექმნა

ჰოსტზე დირექტორიების შექმნა შესაძლებელია მაუსის მარჯვენა კლიკის შედეგად გამოსული ფანჯარაზე **Create directory** აქტივაციის შედეგად (იხ.სურ 26.5)

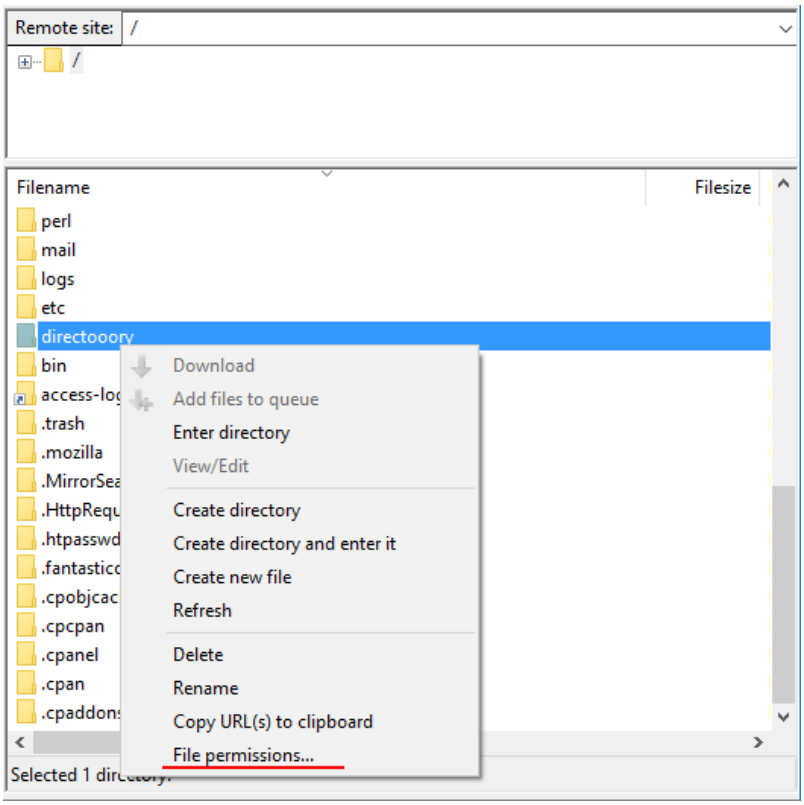
ფაილების სერვერზე განსათავსებალად საჭიროა მაუსის მარტივი მოძრაობით **პირველი** ბლოკიდან **მეორე** ბლოკში გადათრევა, რადგანაც **პირველი** ბლოკი ლოკალური კომპიუტერის ფაილების მასამართია ხოლო მეორე ბლოკი ჰოსტი. რა თქმნა უნდა ფაილების მთვარ დირექტორიაში

გადატყორცნა დამაკმაყოფილებელ შედეგად ვერ ჩაითვლება თუ ჩვენ მათი ასახვა ბრაუზერში - საიტზე გვსურს

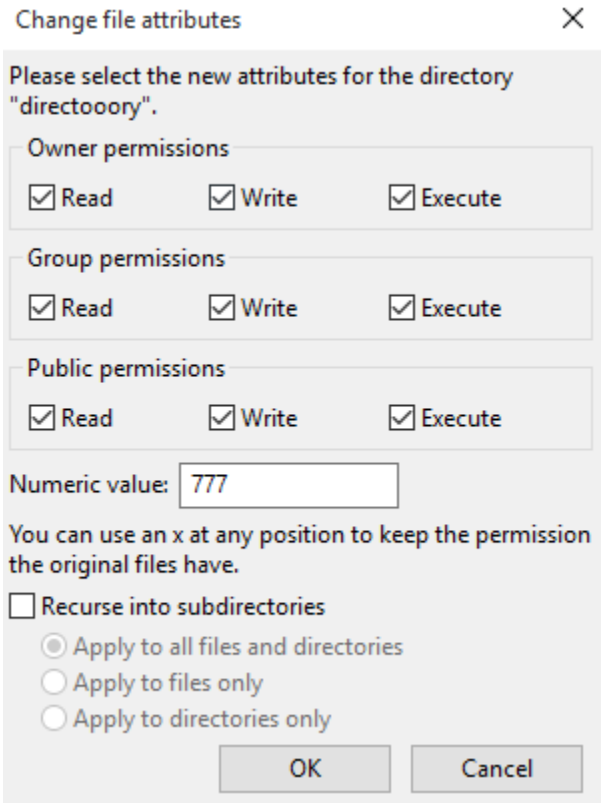
ერთ ერთ უმნიშვნელოვანეს საქაღალდეს წარმოადგენს **public_html** - საიტის ფაილების ასატვირთი საქაღალდე, ანუ საქაღალდე რომელშიც განთავსებულია საიტის ან ზოგადი ფაილები, რომლებთან წვდომაც ხორციელდება **HTTP://** _ მეშვეობით. მასში განთავსებული ფაილები გამოისახება ბრაუზერში საიტის მისმართის გამოძახების შედეგად.

Chmod

Chmod – პროგრამა, რომელიც უზრუნველყოფს ობიექტებისათვის (ფაილი, საქაღალდე) წვდომის უფლებების მართვას. FTP კლიენტის მეშვეობით შესაძლებელია ვებ სერვერზე განთავსებული ფაილების და საქაღალდების უფლებების განსაზღვრა. კონრეტულ ფაილზე (საქაღალდეზე) კურსორის მარჯვენა კლიკით გამოძახება კონტექსტური მენიუ, რომლის უკანასკნელი პუნქტით **File Permissions** ხდება არსებული ობიექტის უფლებების დემონსტრირება/ცვლილება. (იხ სურ. 27.1- 27.2)



სურ 27.1ფაილის (საქარალდის) უფლებების განსაზღვა



სურ.27.2 ფაილის უფლებების განსაზღვრა

სურათ 27.2_ზე მოცემულია ფაილის (ამ შემთხვევაში საქაღალდის) უფლებები, როგორც მოცემულობიდან ჩანს ყველა ღილაკი აქტივირებულია, ე.ი ნებისმიერი წვდომის მოხმარებელს აქვს ყველა უფლება აქტივირებული.

განვმარტოთ დაწვრილებით.

ნებისმიერ ფაილთან (საქაღალდესთან) ურთიერთობის დროს შესაძლებელია:

- **read** - კითხვა;
- **write** - ჩაწერა/ ცვლილების შეტანა;
- **execute** - გაშვება ფაილის შემთხვევაში, საქაღალდის დროს მისი შიგთავსის გაცნობა;

აქედან გამომდინარე თითოეულ ობიექტს შეიძლება შეესაბამებოდეს რამოდენიმე მოცულობა. მაგ.: **r w x** - შესაძლებელია ფაილის კითხვაც, ცვლილების დატანაც და გაშვებაც. მხოლოდ **r - -** შემთხვევაში შესაძლებელია მხოლოდ კითხვა და ა.შ

თოთოეულ სიმბოლოს რიცხვითი მნიშვნელობა შეესაბამება რომელიც გამოსახულია მოცემულ ცხრილში. შესაბამისად თითოეული ფაილისათვის შესაძლბელია, როგორც ანბანის მეშვეობით მნიშვნელობის , ასევე მათ რიცხვთა ჯამის მინიჭება. მაგალითი იხილეთ მოცემული ცხრილის უკანასკელ სვეტ **„შედეგი“**_ში

კითვხა	**ჩაწერა/ ანახლება**	**გაშვება**	**შედეგი**	
r	**w**	**x**	**rwx**	**r – x**
4	**2**	**1**	**7**	**5**

სურათ 27.2 _ზე მოემულია **Numberic value 777 ,** რომელიც თავის მხრივ წარმოადგენს სხვადასხვა მოხმარებლისთვის მინიჭებულ ფუნქციათა ერთობლიობას.

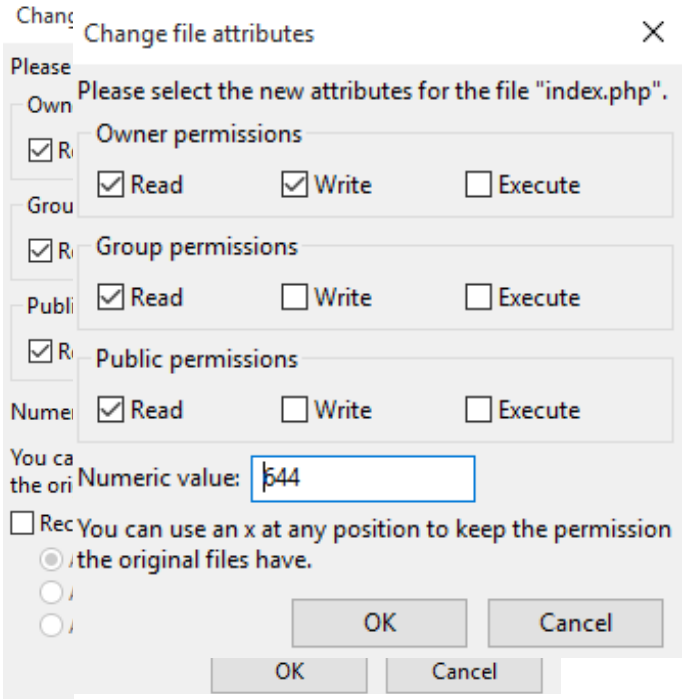
სულ არის 3 ტიპის მომხმარებელი შესაბამისად თავისი მართვის უფლებებით:

- **Owner permissions** - მეპატრონე, მფლობელი, რომელმაც შექმან ფაილი (საქაღალდე). სამივე უფლების მონიშვნით აქტივირება ნიშნავს რომ მას აქვს უფლება **r-w-x** შესაბამისად შეესაბამება **7;**
- **Group permissions** - ავტორთან დაახლოები ჯგუფი. მიმდინარე შემთხვევაშიც ყველა უფლება აქტივირებული აქვს და შესაბამისად მიიღება რიცხვი **7;**
- **Public permissions** - ზოგადი მომხარებელი რომელსაც რამენაირი შეხება მაინც შეიძლება ქონდეს ფაილთან.

შესაძლოა მოხდეს **Recurse into subdirectories** ველის აქტივაცია(საქაღალდის შემთხვევაში), რაც გამოიწვევს მინიჭებული უფლებების მასში არსებული ფაილებზე და საქაღალდეებზე მემკვიდრეობით გადაცემას.

ჰოსტი თავად განსაზღვავს მასთან კავშირისას თუ, რომელი კატეგორიის მომხარებელს წარმოადგენთ თქვენ და შესაბამისი უფლებები, რომელიც განისაზღვრა შემქმელის მიერ ავტომატურად მოგერგებათ.

ძირითად შემთხვევაში საიტზე განთავსებულ საქაღალდეებს აქვთ **755** ხოლო ფაილებს **644** უფლებელი (იხ.სურ. 27.3-27.4)



სურ. 27.4 უფლებები ფაილებზე

სურ. 27.3 უფლებელი საქაღალდეები

7.3 სერვერზე ბაზების მართვა

- მონაცემთა ბაზებთან წვდომა ინტერფეისის მეშვეობით
- მონაცემთა ბაზებში ცხრილების შექმნა
- ბაზებში ინფორმაციის იმპორტირება

მონაცემთა ბაზებთან წვდომის სხვადასხვა პროგრამული უზრუნველყოფის მეშვეობით არის შესაძლებელი. MySQL მონაცემთა ბაზასთან სამუშაოდ ფართოდ გავრცელებულ პროგრამულ უზრუნველყოფებს განეკუთვნება: **phpMyAdmin, MySQL Workbenck, Navicat.**

სამართავი პანელთა უმრავლესობაზე ინტეგრირებულია მონაცემთა ბაზებტან სამუშაოდ phpMyAdmin პროგრამული უზრუნველყოფა, რაც საკმაოდ მარტივს ხდის ბაზის სტრუქტურასა და ინფორმაციასთან მუშაობას ვიზუალურად საკამაოდ მიმზიდველი ინტერფეისის მეშვეობით. მის მთავარ ღირსებად ბაზის შემცველობის მარტივი მართვაა.

phpMyAdmin პროგრამასთან დასაკავშირებლად საჭიროა სამართავი პანელიდან (მაგალითზე მოყვანილია cPanel) მოვახდინოთ სურათ 28.1 ნაჩვენები ჩანართის აქტივაცია.

სურ.28.1 PhpMyAdmin პროგრამული უზრუნველყოფის გამოძახება

phpMyAdmin ბმული მოახდენს მომხმარებლის გადამისამართებას ინტერგეისზე, საიდანაც ხდება ბაზების ინფორმაციის მართვა (სურ 28.2)

სურ.28.2 phpMyAdmin

სურათ 28.2 მარცხენა მხარეს გამოტანილია ყველა არსებული მონაცემთა ბაზა. მონაცემთა ბაზის სახელზე მაუსის კლიკისას აქტიურდება ბაზა და შესაძლებელია მისი ცხრილური სტრუქტურის დადგენა.

ცხრილი მონაცემთა ბაზის ერთ ერთ უმთავრესი ელემენტია. სწორედ ის უზრუნველყოფს ინფორმაციის სორტირებას შესაბამისი ტიპის სვეტებში.

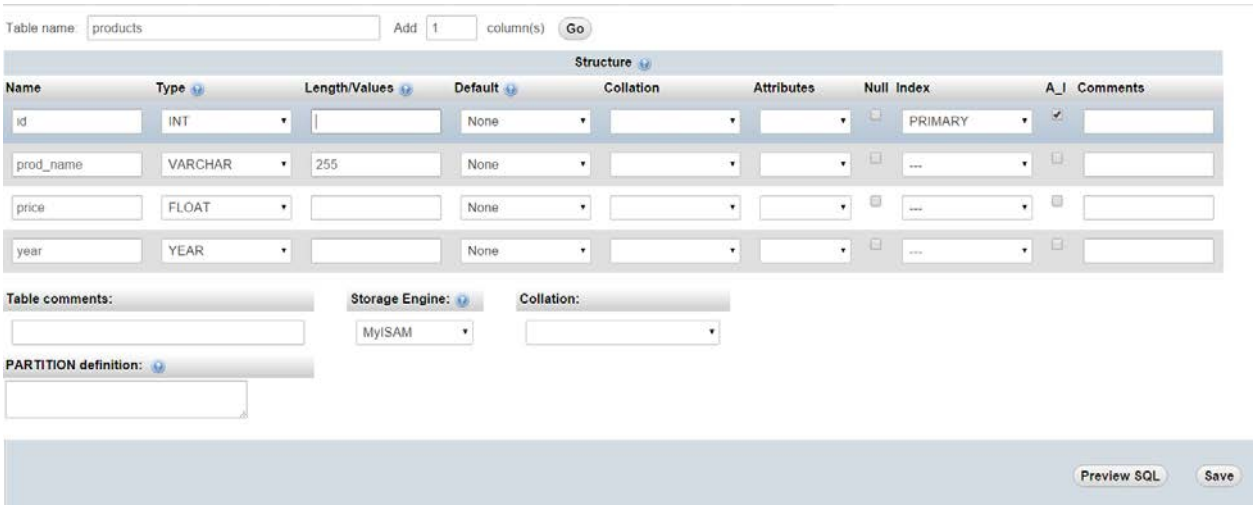
ნებისმიერი ინფორმაცია ხასიათდება საკუთარი თვისებებით. მაგ.: არსებობს **ტექსტუალური** ტიპის ინრმაცია, რომელიც აერთიანებს სიმბოლოების განსაზღვრულ რაოდენობას, არის **რიცხვითი** მნიშვნელობა , თავის მხრივ მთელი და ათწილადი რიცხვების ერთობლიობით, ცალკე ტიპადაა გამოყოფილი **დროითი** ინფორმაცია.

სურათ 28.2_ზე ცენტრალურ ნაწილში (მარჯვენა არეალში) გამოყოფილი ბლოკის მეშვეობით შესაძლებელია ახალი ცხრილის დამატება.

Name ველში მეთითება ცხრილის სახელი ხოლო **Number of columns** ველში განისაზღვრება სვეტების რაოდება. ამ ველეების ინფორმაციის ცვლილება მომავლის პერსპექტივაში არანირ პრობლემას წარმოადგენს უბრალოდ უმჯობესია საქმის გამარტივების მიზნით წინასწარ განსაზღვრული მონაცემები იდენტური იყოს შესაბამისი დავალებისა.

დავუშვათ გვაქვს მოცმეულობა: შექმენით საიტის პროდუქციის ცხრილი შემდეგი ინფორმაციით: პროდუქციის დასახელება, ფასი, გამოშვების წელი.

ძირითად შემთხვევაში ცხრილი თითოეულ ინფორმაციას სჭირდება იდენთიფიკატორი - **id,** რათა განისაზღვროს მისი უნიკალურობა. ასევე საჭიროა პროდუქციის დასახელების, ფასის და გამოშვების წელის კოლონების შექმნა. ე.ი ჯამში **Number of columns** - ველის პარამეტრი იქნება **4. Go** -ღილაკით გადავინაცვლებთ ახალი ცხრილის შექმნის პროცესისკენ. (იხ.სურ.28.3)



სურ.28.3 ახალი ცხრილის ფორმირება

როგორც ატყობთ თითოეულ სვეტს **Name ველები** მივუსადაგეთ შესაბამისი სახელწოდება. რეკომენდირებულია სახელების ლათინური ანბასინ ასოებით გამოსახვა, ლოგიკური სახელწოდების მისასდაგება, რამოდენიმე სიტყვიანი სახელის ქვედა ტირით გამოყოფა.

თითიეოულ ველს უნდა განესაზთვროს შესაბამისი ტიპი, რაც უზრუნველყოფს მიმდინარე ველში შეტანი ინფორმაციის ფილტრაციას, განსხვავებული ტიპის ინორმაციის შეტანისას დაფიქსირბდე შეცდომა.

1. **id – int** რიცხვითი ტიპი, ყოველი ახალი ჩანაწერს გაუჩნდება ახალი რიცხვითი მნიშვნელობა. მის ავტომატუ ამთვლელს უზრუნველყოფს **A_I (auto increment)** ბოქსის აქტივაცია. ის უზრუნველყოფს ასევე წაშლილი ჩანაწერის იდენთიფიკატორი აღარ გამეორდეს მომდევნო

დამატებული ჩანაწერების შემთხვევაში. ყოველი ჩანაწერი იქნება უნიკალური (მაგ.: ადამიანის პირადობის მოწმობის ნომერი არ მეორდება ის არის 1 და უნიკალური). **PRIMAY** index_ის აქტივაცია კონკრეტულ ცხრის უკეთებს მტავარ გასაღებად მიმდინარე ველს ანუ id_ს. რიცხვითი ტიპების დიაპაზონი საკმაოდ დიდ მნიშვნელობას შეიძლება შიეცავდეს, როგორც უარყოფით ასევე დადებით არეალში. მაგალითის შემთხვევაში int ტიპი დაახლოებით - 2 მილიარდიდან +2 მილიარდის დიაპაზონს მოიცავს. შესაბამისად თუ მხოლოდ დადებით დიაპაზონს გამოიყენებთ მისი ზომა 0_დან +4 მილიარდამზე იზრდება.

2. **prod_name _ varchar** - ველი ტექსტუალური ტიპია რომელიც შეიცავს მაქსიმუმ 255 სიმბოლოს. ამიტომ მისი მითითებისას აუცილებელია **Length** ველში აუცილებლად განისაზღვრის სიმბოლოთა რაოდენობა მაქსიმუმ 255, წინააღმდეგ შმეთხვევაში ცხრილის შექმნისას დაფიქსირდება შეცდომა.

3. **price_ float** - მიმდინარე ტიპი მცოცავმძიმიან რიცხვით ტიპს განეკუთვნება -ათწილადებს.

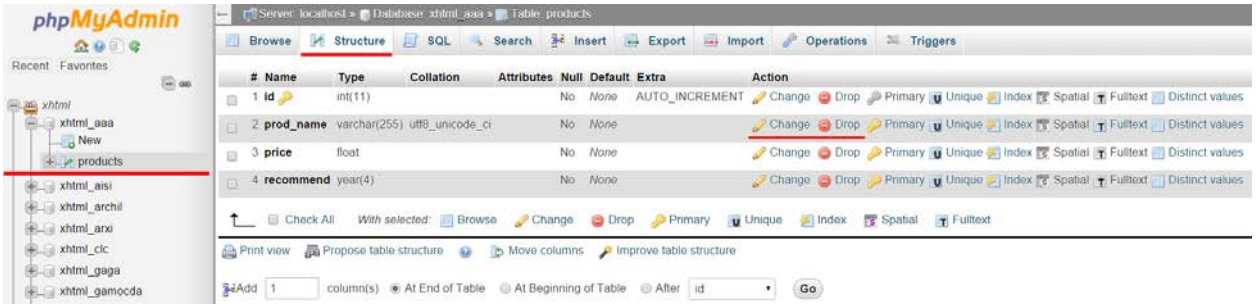
4. **recommend- year** - ტიპი რომელიც გაძლევთ თარიღის მითითების შესაძლებლობას.

თითოეული ტიპის შესახებ ინფორმაციის მოსაძიებლად შეგიძლიათ მიმართოთ MySQL _ ის ოფიციალურ საიტის შემდეგ ბმულს
http://dev.mysql.com/doc/refman/5.1/en/data-types.html .

ტიპის შერჩევისას მაუსის კურსორის მიტანისას ტიპზე გამოისახება ის დიაპაზონი თუ რამდენი სიმბოლოს დატევის შესაძლებელობა აქვს მას.

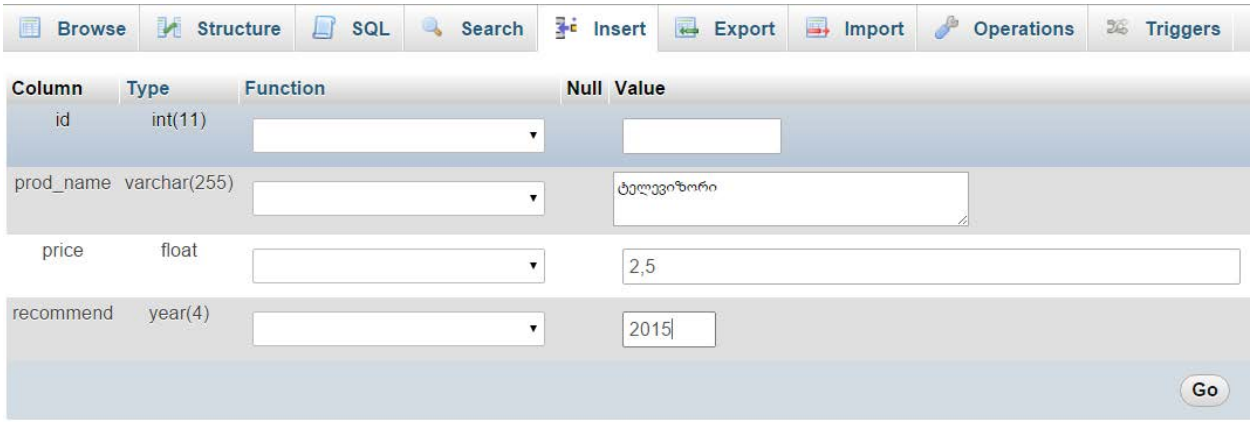
save - ღილაკის მეშვეობით დაადასტურეთ ცხრილის შემქნა.

დამატებული ცხრილი ავტომატურად აისახება ბაზაში ქვეპუნქტად. მისი სტრუქტურის სანახავად მონიშნეთ ცხრული და გააქტიურეთ **Structure** ჩანართი. წარმოდგიდგებათ ცხრილის სტუქტურა, საიდანაც შესაძლებელია კონკრეტული ველების განახლება/წაშლა/დამმატება (სურ 28.4)



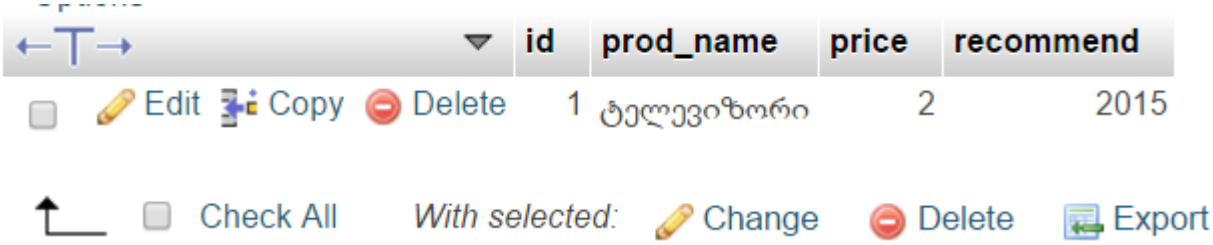
სურ 28.4 ცხრილის სტრუქტუის გაცნობა/ რედაქტირების შესაძლებლობა

ცხრილში ახალი ინფორმაციის დასამატებლად გადადით **Insert** ბმულზე და შესაბამის ველებში უზრუნველყავით შესაბამისი ტიპის ინფორმაციის დამატება.(იხ.სურ.28.5). id_იდენთიფიკატორი შევსებას არ საჭიროებს ის ავტომატურად ინომრება A_I ფუნქციის გააქტიურების გამო.



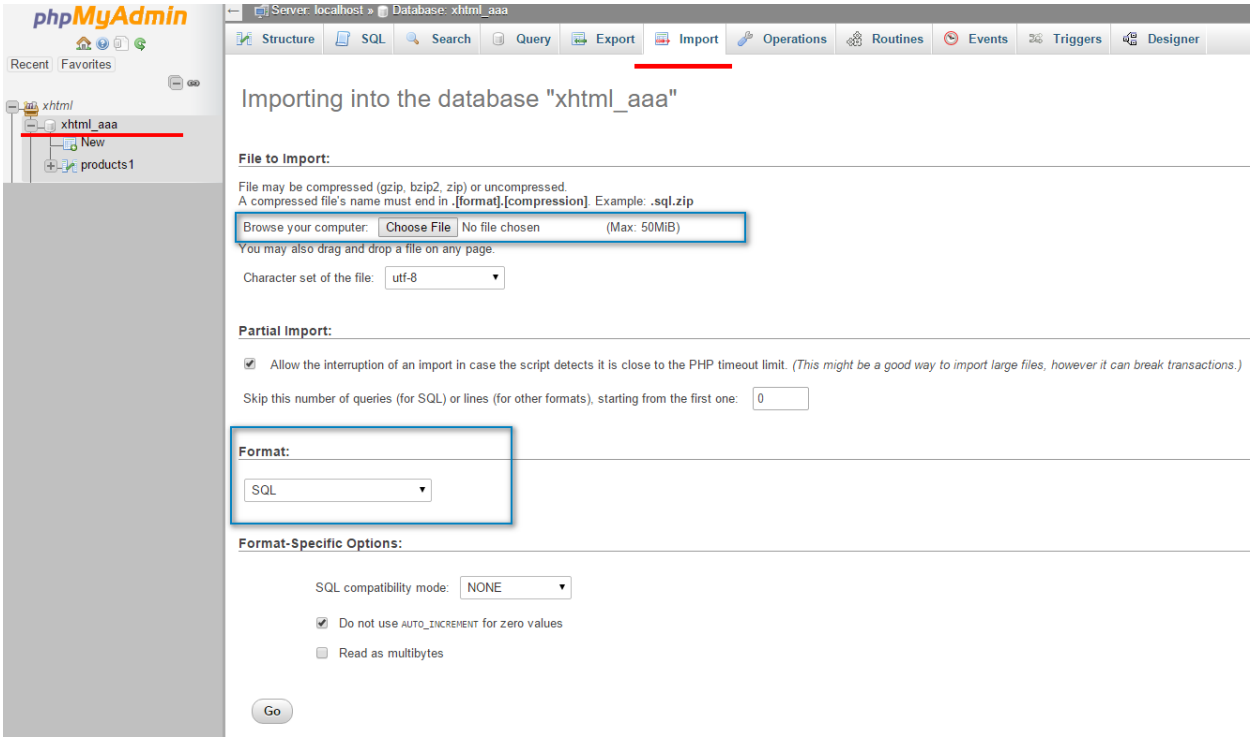
სურ. 28.5 ინფორმაციის დამატება ცხრილში

ინფორმაციით შევსებული ცხრილის შემცველობის მონახულება შესაძლებელაი **Browse** ბმულის მეშვეობით. (იხ.სურ.28.6) მიმდინარე მაგალითის შემთხვევაში არსებობს მხოლოდ 1 ჩანაწერი.



სურ.28.6 ინფორმაციის ვიზუალიზაცია

შესაძლოა ბაზაში ცხრილების შექმინს გარდა შესაძლებელია სხვა ბაზიდან ექსპორტირებბული ინფორმაციის შემოტანა. ამ პროცესის წარმატებით განსახორციელებად საჭიროა მონიშვნით გააქტიუროდ phpMyAdmin_ის მარცხენა კუთხეში არსებული მონაცემთა ბაზებიდან თქვენთვის სასურველი. შემდგომ მოახდინოთ **Import** ბმულის აქტივაცია > **Choose File** მიუთითეთ დასაიმპორტირებელი ფაილი, **Format_** ბლოკში განსაზთვრეთ ფაილის ფორმატი და GO ღილაკის მეშვეობით დაამოწმეთ პროცესი. (იხ.სურ. 28.7)



სურ.28.7 ფაილის შეტანა/იმპორტირება მონაცემთა ბაზაში

თავი 8. რასტრული გამოსახულების შექმნა და დამუშავება

8.1 დოკუმენტის საჭირო პარამეტრებით შექმნა

მიმდინარე პარაგრაფის თემატიკა

- რასტრული გამოსახულება და მისი შედარება ვექტორულ გამოსახულებასთან
- ფერთა მოდელების განხილვა
- რასტრული გამოსახულების გრაფიკული ფორმატები
- გრაფიკული რედაქტორის ინტერფეისის ტიპების გაცნობა
- ფანჯრების განლაგების რეჟიმების მიმოხილვა
- გრაფიკული რედაქტორის პარამეტრების გაცნობა
- დოკუმენტის შესაბამისი ფორმატების გაცნობა

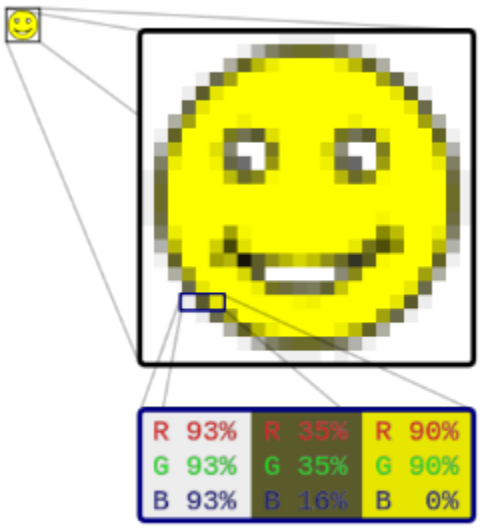
რასტრულიგამოსახულება

რასტრი(raster) - გრაფიკული გამოსახულება, რომელიც წერტილოვანი სტრუქტურის სახითაა წარმოდგენილი. როგორც წესი, წერტილს გააჩნია წრის ფორმა, თუმცა სხვა ფორმის წერტილებიც გამოიყენება: ელიფსური, რომბისებური და კვადრატული.

კომპიუტერულ გრაფიკაში რასტრული გამოსახულება იყოფა კვადრატულ ელემენტებად (ე.წ. პიქსელებად), რომელთა ერთობლიობა ქმნის გამოსახულებას ქაღალდზე, მონიტორზე ან სხვა რომელიმე ამსახველ მოწყობილობაზე. სიტყვა პიქსელი (pixel), ინგლისური Picture element – ის შემოკლებული ვარიანტია.

რასტრული გამოსახულების ძირითადი მახასიათებლებია:

- გამოსახულების ზომა პიქსელებში სიგანეზე და სიმაღლეზე (800x600px, 1024x768px და სხვა);
- გამოყენებული ფერების რაოდენობა ანუ ფერის სიღრმე;
- ფერთა მოდელები (RGB, CMYK, Lab და სხვა)
- გამოსახულების გარჩევადობა - სიდიდე, რომელიც განისაზღვრება წერტილების რაოდენობით მოცემულფართობზე,მაგალითად დუიმზე (1 დუიმი = 2.54 სმ).



სურ. 8-1. რასტრული გამოსახულების მაგალითი.

არსებობს გარჩევადობის ორი სახის საზომი ერთეული - ppi და dpi.

PPI (pixel per inch) - პიქსელების რაოდენობა დუიმზე. გამოიყენება ამსახველი მოწყობილობების გარჩევადობის შესაფასებლად: მონიტორი, ციფრული კამერის ეკრანიდა სხვა;

DPI (dot per inch) - წერტილების რაოდენობა დუიმზე. გამოიყენება საბეჭდი მოწყობილობების ან სკანერების გარჩევადობის შესაფასებლად.

საბოლო ჯამში გარჩევადობა ნიშნავს ინფორმაციას. რაც უფრო მეტია გარჩევადობა, მით მეტია ინფორმაცია.რაც უფრო მეტი პიქსელია გამოსახულებაში, მით უფრო დიდ ფორმატზე შეიძლება იყოს ის აღბეჭდილი; რაც უფრო მჭიდროა პიქსელების განლაგება გამოსახულებაში, მით უფრო დეტალური და ხარისხიანია გამოსახულება.

ძირითადად გვხვდება შემდეგი ხარისხები:

72-150pixel/inch - ამ ხარისხის გამოსახულებები ძირითადადინტერნეტისთვისან ელექტრონული პუბლიკაციებისთვის გამოიყენება;

150-250 pixel/inch - გამოიყენება ელექტრონული პუბლიკაციებში და პრინტერზე ბეჭდვისათვის;

300 pixel/inch და მეტი - გამოიყენება პოლიგრაფიაში და ბეჭდვით ტექნოლოგიებში. *შენიშვნა.*გამონაკლის შემთხვევაში, შესაძლებელია გამოყენებული იყოს მინიმუმ 250 pixel/inch ხარისხის გამოსახულება.

რასტრულ გრაფიკას გააჩნია როგორც დადებითი ისე უარყოფითი მხარეებიც.

დადებითი მხარეებია:

- რასტრულ გრაფიკაში შეიძლება ნებისმიერი სირთულის გამოსახულების შექმნა;
- პოპულარობა - რასტრული გრაფიკა პრაქტიკულად ყველგან გამოიყენება;
- რთული გამოსახულებების სწრაფად დამუშავება;

უარყოფითი მხარეები:

- მიღებული ფაილის დიდი მოცულობა;
- გამოსახულების ზომების გაზრდისას მისი ხარისხი მკვეთრად ეცემა.

ვექტორული გამოსახულება

ვექტორულ გრაფიკაში გამოსახულება წარმოდგენილია მათემატიკური ობიექტების, კონტურების სახით, რომლებშიც ჩასხმულია ფერი. თავის მხრივ, კონტური შედგება საკვანძო წერტილებისგან, რომლებიც ერთმანეთს უკავშირდებიან მრუდეებით. საკვანძო წერტილებზე მოქმედებისას ჩვენ შეგვიძლია ვცვალოთ მრუდის მოხაზულობა და საერთო ჯამში ფიგურის ან ფორმის იერსახე.

ვექტორულ გამოსახულებას ისევე, როგორც რასტრულს, თავისი დადებითი და უარყოფითი მხარეები გააჩნია.

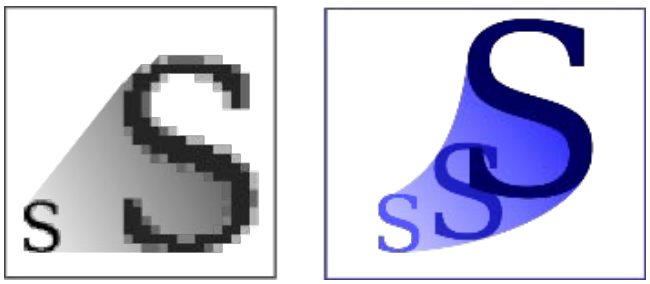
დადებითი მხარეებიდან შეგვიძლია გამოვყოთ:

- ფაილის მცირე ზომა;
- შესაძლებელია გამოსახულების გადიდება უსასრულოდ. მაგალითად თუ გავზრდით წირის მრუდეს, ის ისეთივე სუფთა დარჩება, როგორიც იყო; თუ მრუდე წარმოდგენილია ტეხილი ხაზით (რაც მცირე ზომისას არ ჩანს), გადიდებისას ის აუცილებლად გამოჩნდება;

უარყოფითი მხარეები:

- შეუძლებელია ყველა ობიექტი მარტივად გამოისახოს ვექტორული სახით; იმისათვის, რომ ვექტორული გამოსახულება დაემსგავსოს ორიგინალს, შესაძლებელია დაგჭირდეთ ძალიან დიდი რაოდენობის ვექტორები;

- ვექტორული გამოსახულების გარდაქმნა რასტრულად ძალიან მარტივია. ხოლო უკან დაბრუნება (რასტრულიდან ვექტორისკენ) საკმაოდ რთული;



სურ. 8-2. რასტრული და ვექტორული გამოსახულებების შედარება

ფერთა მოდელები

ცნობილია, რომ მზის სხივი შეიცავს ბუნებაში არსებულ ყველა ფერს. მზის სხივის უწყვეტ სპექტრში არჩევენ 130-მდე ფერის ტონალობას. რეალურ ცხოვრებაში ამ ფერების ნახვა ცისარტყელაზე შეიძლება. მზის სხივის უწყვეტ სპექტრში ნიუტონმა გამოყო 7 ძირითადი ფერი: წითელი, ნარინჯისფერი, ყვითელი, მწვანე, ცისფერი, ლურჯი და იისფერი. ყველა სხვა ფერი, სწორედ ამ 7 ფერის სხვადასხვა რაოდენობით შერევის გზით მიიღება.

მრავალი ცდისა და დაკვირვების შედეგად, საერთაშორისო კოლორიმეტრული სისტემის მთავარი ფერები გამოავლინეს: წითელი (Red), მწვანე (Green) და ლურჯი (Blue). ინგლისური დასახელების პირველი ასოების მიხედვით, ფერთა ამ სისტემას დაერქვა - RGB.

RGB ფერთა სისტემა საფუძვლად უდევს მონიტორებს, სკანერების ერთ ნაწილს და კომპიუტერული პროგრამების უმრავლესობას. კომპიუტერის ეკრანზე ყველა ფერის მიღება, სწორედ ამ სამი ფერის სხვადასხვა პროპორციის შერევით არის შესაძლებელი. RGB მოდელის გარდა, არის ფერთა სხვა მოდელებიც: HSB; Lab; CMY; CMKY; YIQ და YCC.

მათ შორის ძირითადი მოდელები გამოიყენება: RGB – კომპიუტერულ გრაფიკაში; CMYK – პოლიგრაფიაში; Lab – ტექნიკური მიზნებისთვის; Grayscale – შავ-თეთრი ფოტო-სურათების რეჟიმი; Bitmap (Monochrome) – ორფეროვანი გამოსახულების რეჟიმი.

ფერთა მოდელის გარდა ასევე უნდა განისაზღვროს ფერის სიღრმე, რომელიც იზომება ბიტებში. რაც უფრო დიდია ფერის სიღრმე ე.ი. რაც უფრო მეტ ბიტიანია გამოსახულება, მით უკეთესად გადმოსცემს პიქსელი ფერს. პიქსელი გრაფიკული გამოსახულების უმცირესი ელემენტია და მისი დანიშნულება არის შეინახოს და გადმოსცეს ინფორმაცია ფერის შესახებ.

გამოსახულებაზე, რომლის ფერის სიღრმე 1 ბიტია (2^1=2), არის მხოლოდ ორი ფერი: თეთრი და შავი. 2 ბიტიან (2^2=4) გამოსახულებაზე ოთხი ფერია - თეთრი, შავი და ორი ნაცრისფერი. შავ-თეთრი ფოტოსურათების ფერის სიღრმე, უმეტეს შემთხვევაში, 8 ბიტია (2^8=256): შავი, თეთრი და 254 ნაცრისფრის სხვადასხვა ტონალობა. ე.ი. ამ შემთხვევაში პიქსელს შეუძლია გადმოსცეს ფერის 256 შესაძლო ვარიანტიდან ერთი. ნულოვან მნიშვნელობას შეესაბამება შავი ფერი, ხოლო 255-ს – თეთრი.

ფერად ციფრულ ფოტოსურათს აქვს 3 არხი (წითელი, მწვანე, ლურჯი) და 8 ბიტიანი ფერის სიღრმე თითო არხზე. ამიტომ, ასეთ გამოსახულებას 24 ბიტიანს (2^{24}=16777216) უწოდებენ და ამ შემთხვევაში, პიქსელს ფერის გადმოცემის 16,7 მილიონი შესაძლო ვარიანტი არსებობს. ფერის ეს რაოდენობა სავსებით საკმარისია გამოსახულების ფერის რეალურად გადმოცემისათვის, თუმცა არსებობს 48 ბიტიანი გამოსახულებები (16 ბიტი თითო არხზე), რაც 281 ტრილიონი ფერის გადმოცემის შესაძლო ვარიანტს იძლევა. ადამიანის თვალი ამდენ ფერს ვერ არჩევს, მაგრამ გამოსახულების რედაქტირებისას, ბიტების დიდი რაოდენობა შედეგზე დადებითად აისახება. 16 ბიტიანი გამოსახულებით პროფესიონალი ფოტოგრაფები მუშაობენ.

დღესდღეობით ფოტოშოპს შეუძლია 8, 16 და 32 ბიტიან გამოსახულებებთან მუშაობა.

რასტრული გამოსახულების შესანახად ბევრი ფორმატი გამოიყენება. მუშაობის პროცესში თქვენ მოგიწევთ შეარჩიოთ, რომელ სასურველ ფორმატში გჭირდებათ გამოსახულების შენახვა. ეს შეიძლება ორ ეტაპად დავყოთ:

- **დაუმუშავებელი (ორიგინალი)** ფაილების შენახვა. ეს შესაძლოა იყოს ფოტოსურათები, მოძიებული ან შეძენილი ფოტომასალა ფოტობანკებიდან, სკანირებული გამოსახულებები და სხვა. ეს მასალა მომავალში შესაძლოა გამოყენებული იყოს განმეორებით მსგავსი პროექტებისთვის.
- **დამუშავებული** ფაილების შენახვა პროექტისთვის. პროექტის დანიშნულების გათვალისწინებით (ბეჭდური თუ ელექტრონული), ფაილებისთვის შესაბამისიფორმატისშერჩევა დაგჭირდებათ.

ქვემოთ ჩამოთვლილია ფაილების ფორმატები, რომელიც მუშაობის პროცესში შეგიძლიათ გამოიყენოთ:

JPEG(Join Photographic Expert Group) - ამ ფორმატის უპირატესობა არის მისი მოქნილობა. გამოსახულების რედაქტირების ყველა პროგრამა მუშაობს ამ ფორმატთან, შენახული ფაილების მოცულობა მცირეა, რადგან შენახვისას გამოიყენება *შეკუმშვა*. შეკუმშვისას გარკვეული ინფორმაცია*იკარგება*- ხდებაცალკეული პიქსელების ფერის გაშუალება, რის შედეგადაც მცირდება დეტალიზაცია და იცვლება ფერის ტონალობები. რაც უფრო მაღალია შეკუმშვის ხარისხი, მით უფრო მცირეა ფაილის მოცულობა, მაგრამ ამავდროულად გამოსახულების ხარისხი მკვეთრად ფუჭდება; და პირიქით, რაც უფრო მცირეა შეკუმშვის ხარისხი, ფაილის ზომა არ მცირდება და გამოსახულების ხარისხის მეტნაკლებად მისაღები ხდება. ერთი და იმავე ფაილზე მუშაობისას, ყოველ ახალ შენახვაზე ხდება ინფორმაციის მორიგი პორციის დაკარგვა. ამის გამო ამ ფორმატის გამოყენება ბეჭდვითი პროექტებისთვის მიუღებელია. მისი გამოყენება მისაღებია ელექტრონული პუბლიკაციებისთვის, ელფოსტით გასაგზავნად ან ინტერნეტში განსათავსებლად.

GIF (Graphics Interchange Format)- ფორმატი შექმნილია სპეციალურად რასტრული გამოსახულების გადასაცემად გლობალურ ქსელში. ორიენტირებულია კომპაქტურობაზე და იყენებს LZW (Lempel – Ziw - Welch) ინფორმაციის დაკარგვის გარეშე შეკუმშვის ალგორითმს, რომელიც უზრუნველყოფს შეკუმშვის ძალიან მაღალ ხარისხს. ამ ფორმატს შეუძლია ანიმირებული გამოსახულების შენახვა (ფაილში ინახება არა მარტო გამოსახულების კადრები, არამედ მისი დემონსტრაციის პარამეტრებიც). ფორმატის ნაკლად შეიძლება ჩაითვალოს, რომ ის მუშაობს მხოლოდ ინდექსირებულ ფერებთან.

PNG (Portable Network Graphics) - GIF ფორმატის ჩასანაცვლებლად შეიქმნა. მისგან განსხვავებით, მუშაობს ფერების დიდ რაოდენობასთან და ასვე შეუძლია შეინახოს ფენის გამჭვირვალობა. მასში გამოყენებულია ისეთი მძლავრი უდანაკარგო შეკუმშვის ალგორითმი, როგორიცაა LZW. GIF და PNG ფორმატში შენახული გამოსახულებები ძირითადად გამოიყენება ინტერნეტში განსათავსებლად და ელექტრონულ პუბლიკაციებში.

BMP (BitMap) - ფორმატი შექმნილია Microsoft-ის მიერ და ორიენტირებულია Windows-ის ოპერაციულ სისტემაში გამოსაყენებლად. იხმარება რასტრული გამოსახულების წარმოსადგენად პროგრამულ რესურსებში. იყენებს ინფორმაციის დაკარგვის გარეშე შეკუმშვის მარტივ ალგორითმს RLE (Run Length Encoding).

TIFF - ამ ფორმატს აქვს მრავალფენიანი დოკუმენტების მხარდაჭერა; გამოსახულება შეუძლია შეინახოს, როგორც შეკუმშვით (LZW ან ZIP), ისე შეკუმშვის გარეშე და მისი ხარისხი არ ფუჭდება.ამიტომ ეს ფორმატი შესაძლოა გამოდგეს თქვენი ნამუშევრების არქივში შესანახად. თუმცა მასთან მუშაობისათვის თქვენ დაგჭირდებათ პროგრამა Photoshop ან რომელი მსგავსი რედაქტორი. უნდა გახსოვდეთ, რომ შენახული ფაილის მოცულობა ბევრად აღემატება JPEG ფაილის მოცულობას.

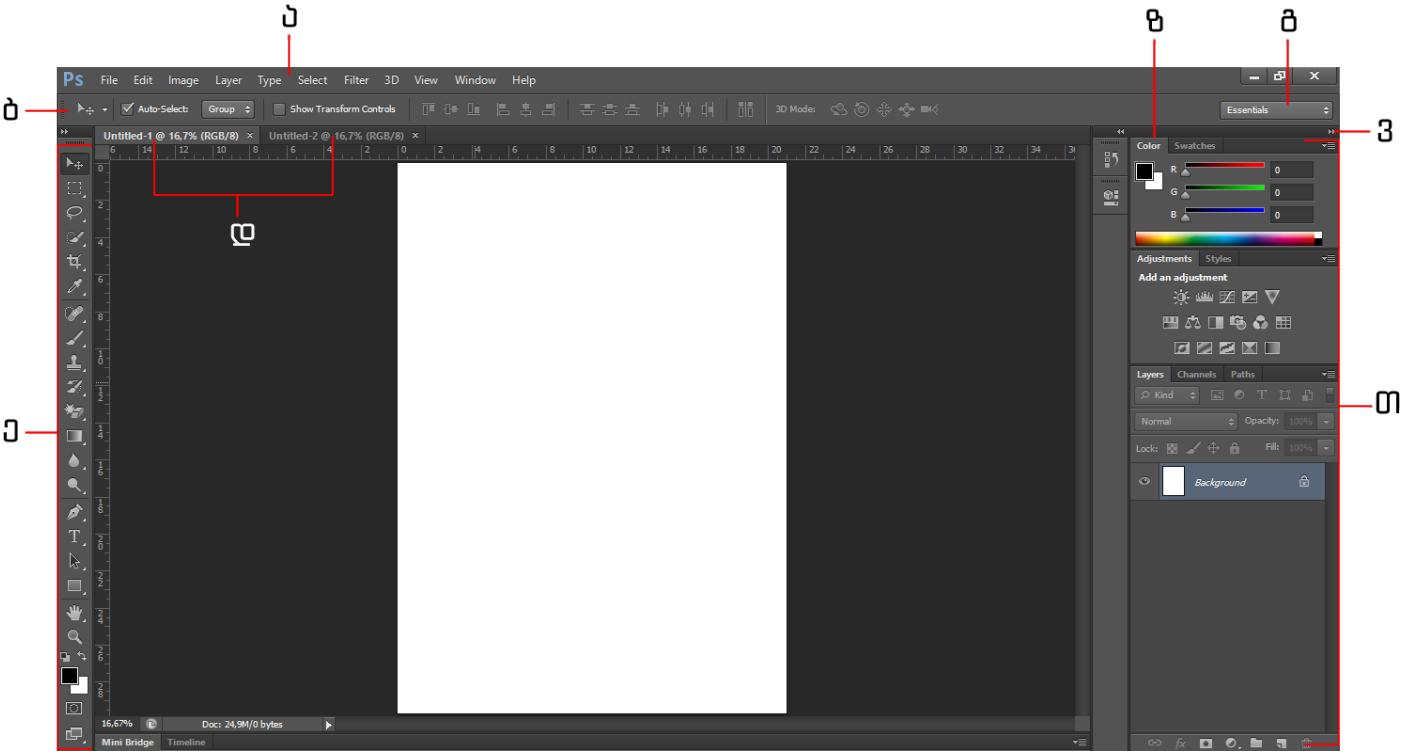
PSD (PhotoShop Document)- ფორმატი არის Adobe PhotoShop-ის საკუთარი ფორმატი. ერთადერთი ფორმატი, რომელიც ინახავს ყველაფერს ნებისმიერი სახით და პროგრამის ყველა მოთხოვნას უჭერს მხარს. შენახვის დროს ინარჩუნებს დოკუმენტის ფენოვან სტრუქტურას, რათა მარტივი იყოს გამოსახულების შემდგომი რედაქტირება. ამ ფორმატში შეიძლება შუალედური ნამუშევრის შენახვა რედაქტირების დასრულებამდე.

RAW - ეს არ არის რომელიმე კონკრეტული ფაილის ფორმატი. ის ზოგადად აღნიშნავს ე. წ. ნედლ, დაუმუშავებელმონაცემებს, რომელსაცფოტოკამერის სენსორი აფიქსირებს. ფოტოკამერების მწარმოებლებს ნედლი ინფორმაციის შესანახად საკუთარი ფორმატები აქვთ შემუშავებული. მაგალითად Canon-ს კამერები იყენებენ**CR2**გაფართოებას, ხოლო Nikon-ის კამერები **NEF**-ს. ასეთი ტიპის ფაილებთან სამუშაო შეგიძლიათ ისარგებლოთ პროგრამებით Photoshop, Bridge, Lightroom და დამუშავების შემდეგ ფაილი თქვენთვის სასურველ ფორმატში შეინახოთ (**JPEG**, **PNG**, **TIFF**, **PSD**). კომპანია Adobe-მა ასევე შეიმუშავა ფორმატი **DNG(Digital Negative)**, რომლის მიზანია მსგავსი დაუმუშავებელი მონაცემების შენახვა.

EPS (Encapsulate PostScipt) - ფაილების ფორმატი, რომელიც დაფუძნებულია PostSript[1]-ის ენაზე და განკუთვნილია პროგრამებს შორის გრაფიკული ინფორმაციის გასაცვლელად. ფორმატი ფართოდ გამოიყენება პოლიგრაფიაში და შეიძლება შეიცავდეს როგორც რასტრულ, ისე ვექტორულ გამოსახულებებს ან ორივეს ერთად. EPS ფორმატს ასევე გააჩნია Grayscale, RGB, CMYK და Lab ფერთა მოდელების მხარდაჭერა.

PDF (Portable Document Format) - ელექტრონული დოკუმენტების ფორმატი, რომელიც კომპანია Adobe System-მა შეიმუშავა. მისი პირველადი დანიშნულებაა პოლიგრაფიული პროდუქციის წარმოდგენა ელექტრონული სახით. PDF საშუალებას იძლევა ფაილში ჩაშენებული იყოს საჭირო შრიფტები, ვექტორული და რასტრული გამოსახულებები, ბმულები, ფორმები, მულტიმედია (აუდიო/ვიდეო) კონტენტი. გააჩნია Grayscale, RGB, CMYK და Lab ფერთა მოდელების მხარდაჭერა და პოლიგრაფიისათვის საჭირო საკუთარი ტექნიკური ფორმატები, მაგალითად PDF/X-1a, PDF/X-3.

[1] PostSript - გვერდების აღწერის ენა. ძირითადად გამოიყენება სამაგიდო საგამომცემლო სისტემებში.



სურ. 8-3.ა - პროგრამის მენიუ; ბ - ფორმატირების ზოლი. მისი პარამეტრები იცვლება იმის მიხედვით, თუ რომელი სამუშაო ინსტრუმენტი გაქვთ არჩეული; გ - სამუშაოო სივრცეებს შორის გადამრთველი; დ - სანიშნეები, რომლებიც მიუთითებენ გახსნილ ფაილს. მათი საშუალებით შესაძლებელია ფაილებს შორის გადასვლა; ე - ინსტრუმენტთა პანელი; ვ - სამუშაო პანელის ნიშნულის სახით ჩაკეცვის ღილაკი; ზ - სამუშაო პანელის დასახელება; თ - ვერტიკალურად ჩამაგრებული სამუშაო პანელები.

სამუშაო სივრცე

თქვენი სამუშაო პროფილის ან კონკრეტული ამოცანის სპეციფიკიდან გამომდინარე, შესაძლოა დაგჭირდეთ სხვადასხვა ტიპის სამუშაო პანელები. გარდა იმ სამუშაო პანელებისა, რომელსაც სურათზე ხედავთ (სურ. 8-3, თ), შესაძლებელია კიდევ სხვების გამოჩენაც. ამისათვის უნდა შეხვიდეთ პროგრამის მენიუში Window და ჩამოშლილი სიიდან აირჩიოთ თქვენთვის სასურველი პანელი ან დამალოთ უსარგებლო.

პანელები შეგიძლიათ დააჯგუფოთ, შეურჩიოთ მას პოზიცია და დააყენოთ, როგორ უნდა ჩანდეს ის ეკრანზე (ღილაკის სახით თუ გაშლილ მდგომარეობაში) ანუ მოაწყოთ სამუშაო სივრცე ისე, როგორც თქვენ გჭირდებათ.

ფოტოშოპს გააჩნია რამდენიმე მზა სამუშაო სივრცე. სასურველი სამუშაო სივრცის არჩევისას ხდება ინტერფეისის ნაწილობრივ შეცვლა და ამ სივრცისთვის დამახასიათებელი სამუშაო პანელების გამოტანა. გარდა ამისა ეს მოცემული სამუშაო სივრცე შეგიძლიათ გადააწყოთ თქვენი სურვილისებრ.

Essentials – ძირითადი სამუშაო სივრცე. ამ დროს გამოტანილია ძირითადი სამუშაო პანელები, როგორიცაა ფერების პალიტრა და ფერის ნიმუშები, ჩვეულებრივი და კორექტირების ფენებთან სამუშაო პანელები და სხვა.

New in CS6 - მათთვის, ვინც გადმოერთო ფოტოშოპის უფრო ძველი ვერსიიდან (CS4, CS5), სასარგებლო იქნება გაიგონ, რა სიახლეებია დამატებული პროგრამაში. გამოტანილი იქნება ინსტრუმენტთა პანელების დიდი ნაწილი, ასევე მენიუში ახალი ან განახლებული ფუნქცია ფერით იქნება მონიშნული.

3D - ამ დროს გამოტანილი იქნება ის პანელები და ინსტრუმენტები, რომლებიცსამგანზომილებიან გრაფიკასთან სამუშაოდ გამოიყენება.

Motion - სივრცე, რომელიც განკუთვნილია მათთვის ვინც ვიდეო ან ანიმაციასთან მუშაობს.

Painting - სივრცე, რომელიც გამოადგება მხატვარს ან ილუსტრატორს. აქ გამოტანილია ფერების პალიტრა, მზა ფუნჯები და ფუნჯების რეგულირების ფანჯარა

Photography - ფოტომასალის დასამუშავებლად გამზადებული სამუშაო სივრცე. გამოტანილია ის აუცილებელი ინსტრუმენტები, რომლებიც მომხმარებელს დაეხმარება ფოტოს დამუშავებაში: ფერთა კორექცია, ფენები, ჰისტოგრამა და სხვა.

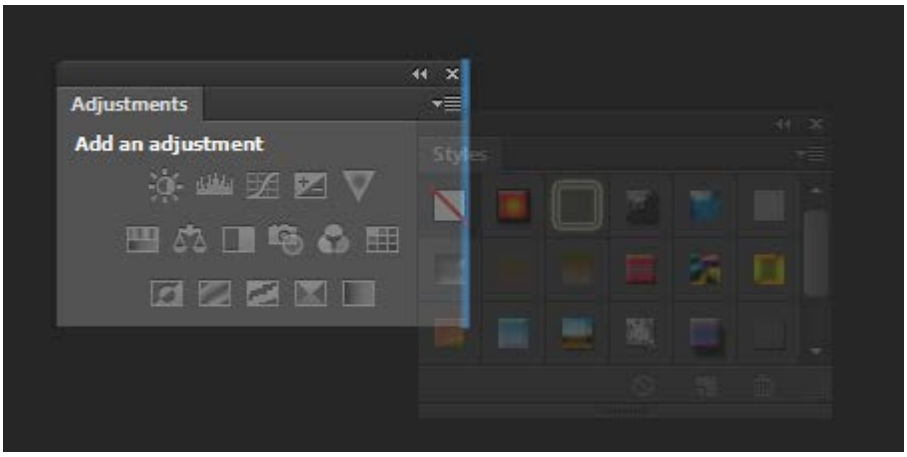
Typography - სამუშაო სივრცე მათთვის, ვინც აპირებს იმუშაოს შრიფტების გამოყენებით და შექმნას ტიპოგრაფიული ეფექტები პოსტერებისთვის, ბანერისთვის და სხვა მსგავსი სახის პროექტისთვის.

სამუშაო პანელებთან რომ გაგიადვილეთ მუშაობა, შესაძლებელია მათი დაჯგუფება. მაგალითისთვის თუ შეხედავთ გამოსახულებას (სურ. 8-3, თ), დაინახავთ რომ პანელები Layers, Channels და Path ერთად არიან დაჯგუფებული. შესაბამის დასახელებაზე დაჭერისას ხდება პანელებს შორის გადართვა და მისი ფუნქციების გამოჩენა. ასევე შეგიძლიათ უკვე არსებული ჯგუფების დაშლა და პანელების გადალაგება თქვენი სურვილის მიხედვით.

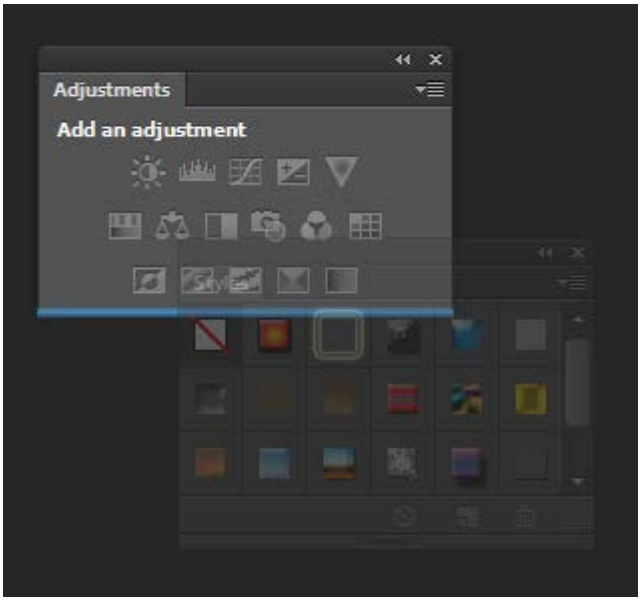
ჯგუფიდან რომ მოხსნათ პანელი, ამისათვის მაუსის კურსორით მოკიდეთ და გაიტანეთ ცალკე:

სურ. 8-4. პანელების ჯგუფის დაშლა

იმისათვის, რომ პანელებიჰორიზონტალურად ან ვერტიკალურად ჩაამაგროთ, პანელი მიიტანეთ მეორე პანელის კიდესთან (გვერდიდან ან ზემოდან/ქვემოდან). გამოჩნდება ლურჯი ზოლი, რაც კავშირის მაჩვენებელია:

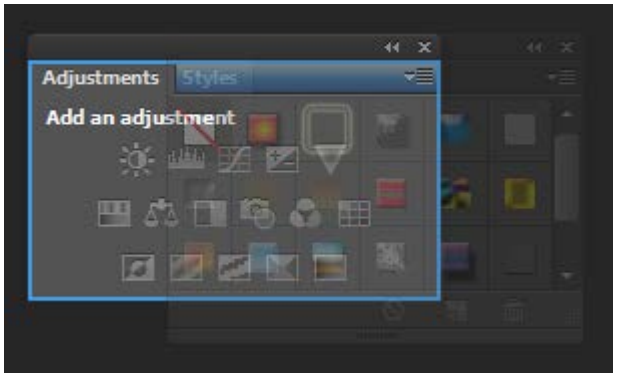


სურ. 8-5. პანელების ჰორიზონტალურად დაჯგუფება



სურ. 8-6. პანელების ვერტიკალურად დაჯგუფება

იმისათვის, რომ დააჯგუფოთ პანელები (გვერდიგვერდ), პანელი მიიტანეთ მეორე პანელის დასახელებასთან ახლოს:

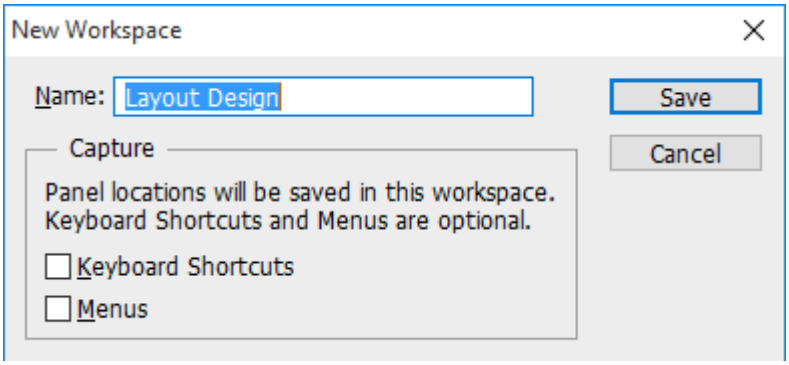


სურ. 8-7. პანელების დაჯგუფება

მას შემდეგ, რაც თქვენ დაალაგებთ საჭირო პანელებს, სასურველი იქნება ეს განლაგება შეინახოთ, რადგან სხვა სამუშაო სივრცეზე გადართვის შემდეგ მოგიწევთ ყველა პროცედურის (დაჯგუფების) ხელახლა გავლა.

ამისათვის შედით Window -> Workspaces. აქ თქვენ ნახავთ სამუშაო სივრცის განამზადებს და დამატებით სამ ფუნქციას.

იმისათვის, რომ შეინახოთ თქვენს მიერ დაყენებული სამუშაო სივრცე, აირჩიეთ Window -> Workspaces ->New Workspace (ახალი სამუშაო სივრცე).



სურ. 8-8. ახალი სამუშაო სივრცის შენახვის ფანჯარა

გამოსულ ფანჯარაში შეიყვანეთ სასურველი სახელი და დააჭირეთ OK. თქვენს მიერ შექმნილი სივრცე დაემატება სამუშაო სივრცეების სიაში.

მუშაობის პროცესში შესაძლოა დაგჭირდეთ სხვა პანელის გამოძახება, არსებულის დახურვა, ფანჯრების გადაჯგუფება და სხვა. რომ დაბრუნდეთ სამუშაო სივრცის საწყის განლაგებაზე, შედით Window -> Workspaces -> Reset Workspace (სამუშაო სივრცის გადატვირთვა). სამუშაო სივრცე აღდგება იმ სახით, რა სახითაც თქვენ შეინახეთ.

არასასურველი სამუშაო სივრცის წასაშლელად აირჩიეთ Window -> Workspaces -> Delete Workspace (სამუშაო სივრცის წაშლა). ჩამოსაშლელი მენიუდან აირჩიეთ სამუშაო სივრცე და დააწკაპუნეთ Delete.

ცალკე ყურადღება უნდა დაეთმოს ინსტრუმენტთა პანელს (სურ. 3: ე), რომელზეც ყველა საჭირო ინსტრუმენტია განთავსებული. ყოველი ინსტრუმენტი შეიცავს დამატებით ფუნქციებს, რომელთა მართვა შესაძლებელია ფორმატირების ზოლიდან (სურ. 3: ბ) ან მართვის პანელის (სურ. 3: თ) დახმარებით.

ინსტრუმენტთა პანელი

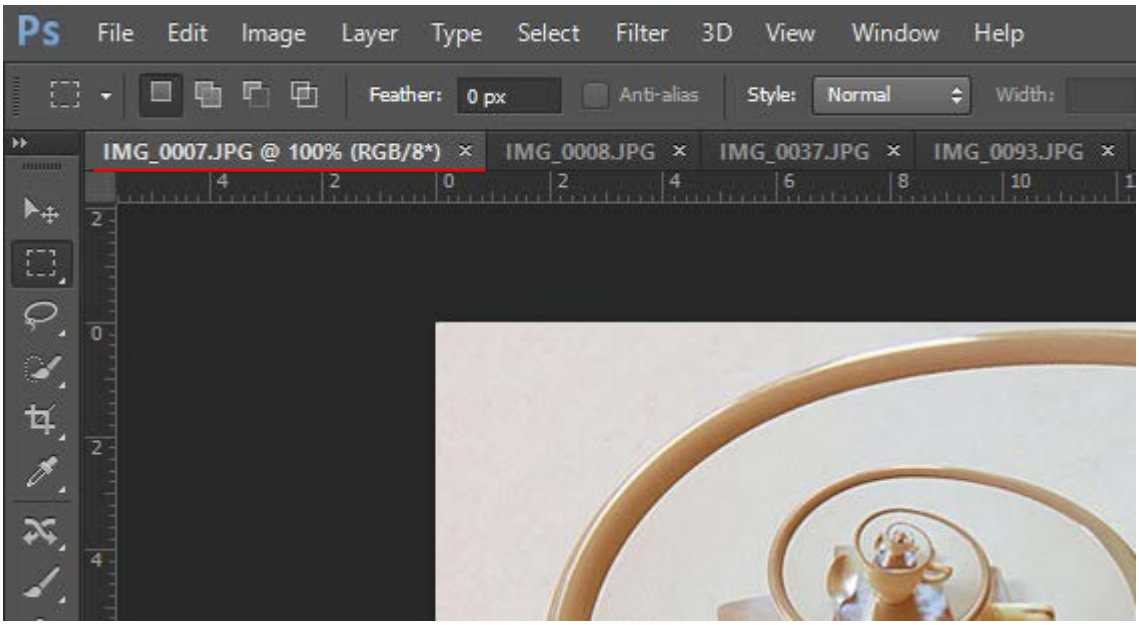
სურ. 8-9. ინსტრუმენტთა პანელი და მისი მიმოხილვა

ფანჯრების განლაგება

ფოტოშოპში შესაძლებელია ერთდროულად რამდენიმე ფაილის გახსნა და მათთან მუშაობა. ეს გაძლევთ საშუალებას ფენების, ობიექტების ან რაიმე ელემენტის ერთი ფაილიდან მეორეში მარტივად,

უბრალო გადათრევით გადაიტანოთ, მოახდინოთ გამოსახულებების შედარება ან უბრალოდ რამდენიმე მოცემული კადრიდან გსურთ აარჩიოთ საუკეთესო პროექტში გამოსაყენებლად.

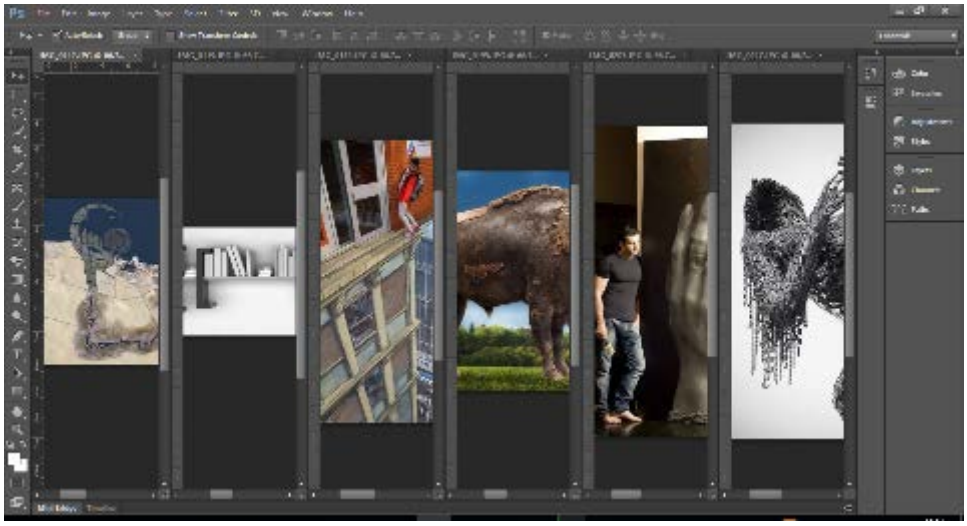
უნდა გაითვალისწინოთ, რომთითოეული ფაილი დამოუკიდებელ ფანჯარაში იხსნება.



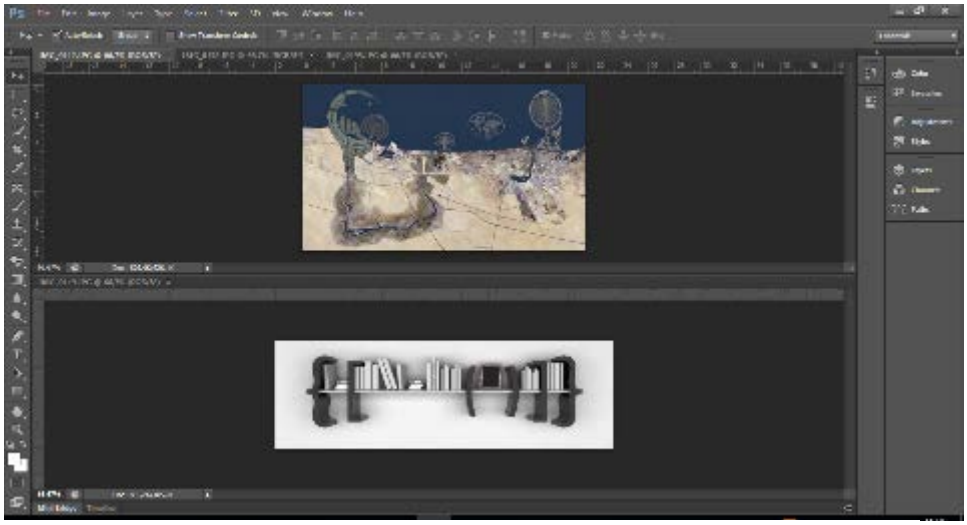
სურ. 8-10 ფანჯრის დასახელება ემთხვევა ფაილის სახელწოდებას. ეს გეხმარებათ დაინახოთ, რომელი ფაილები გაქვთ გახსნილი. ასევე აქ არის სხვა სახის ინფორმაციაც:
*IMG_0007.JPG - ფაილის სახელწოდება; @ - გამყოფი ნიშანი; 100% - გვაჩვენებს გამოსახულების სამუშაო მასშტაბს; RGB - გვაჩვენებს რა ფერთა მოდელი აღწერს გამოსახულებას; 8/16/32 - გვაჩვენებს რამდენ ბიტიანი ფერებია გამოსახულებაში; * - მიუთითებს, რომ ფაილში შეტანილია ცვლილებები.*

სტანდარტულად პროგრამაში ფანჯრების განლაგება არის ჩანართის სახით (სურ. 3: დ). თუმცა ჩვენ მისი ცვლა შეგვიძლია და გამოვაჩინოთ გახსნილი ფაილები ისე, რომ გაგვიადვილდეს მათთან მუშაობა ან მთლიანობაში ინფორმაციის აღქმა.

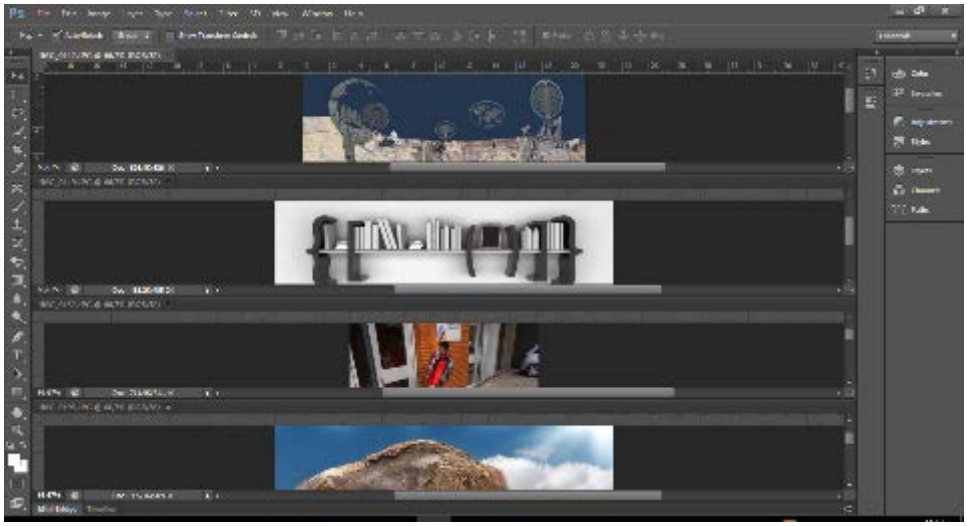
იმისათვის, რომ მართოთ ფანჯრების განლაგება, შედით Window -> Arrange. დამატებით მენიუში ჩამოთვლილია ფანჯრების განლაგების სხვადასხვა ვარიანტები:



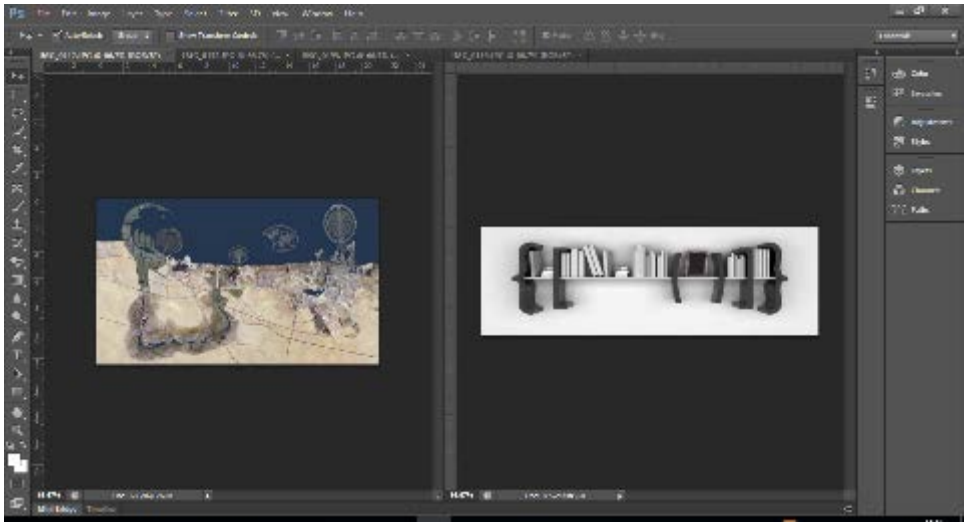
სურ. 8-11. Tile All Vertically - გვერდიგვერდ ვერტიკალურად დალაგება.



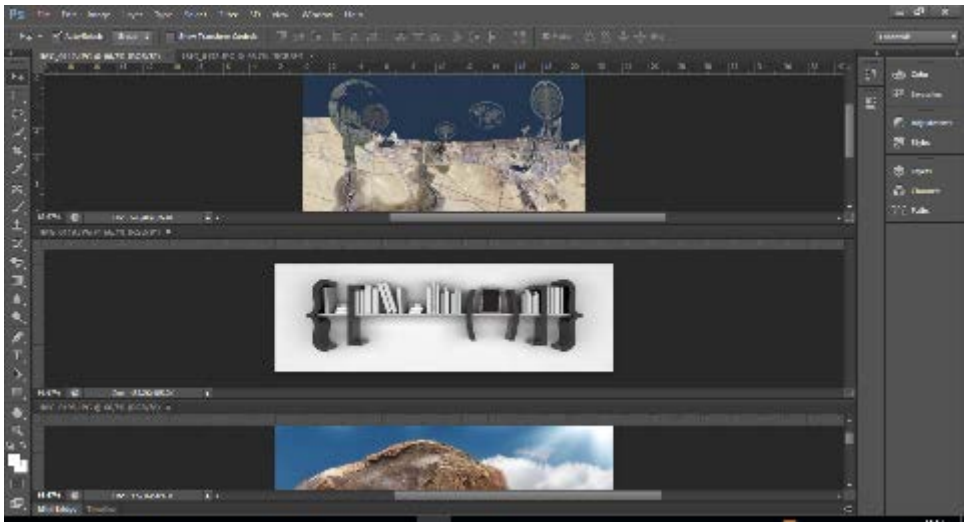
სურ. 8-13.2-up Horizontal - მხოლოდ ორი გამოსახულების დალაგება ჰორიზონტალურად.



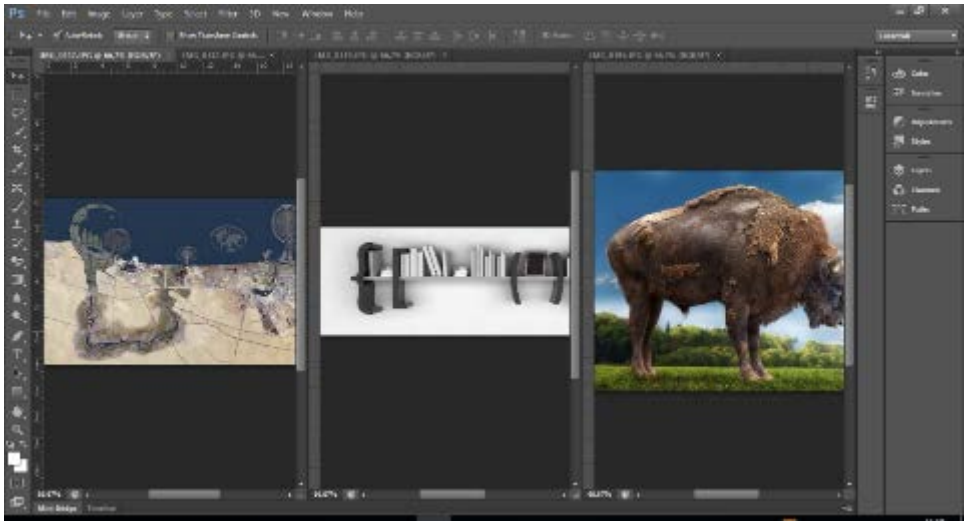
სურ. 8-14. All Tile Horizontally - გვერდიგვერდ ჰორიზონტალურად დალაგება.



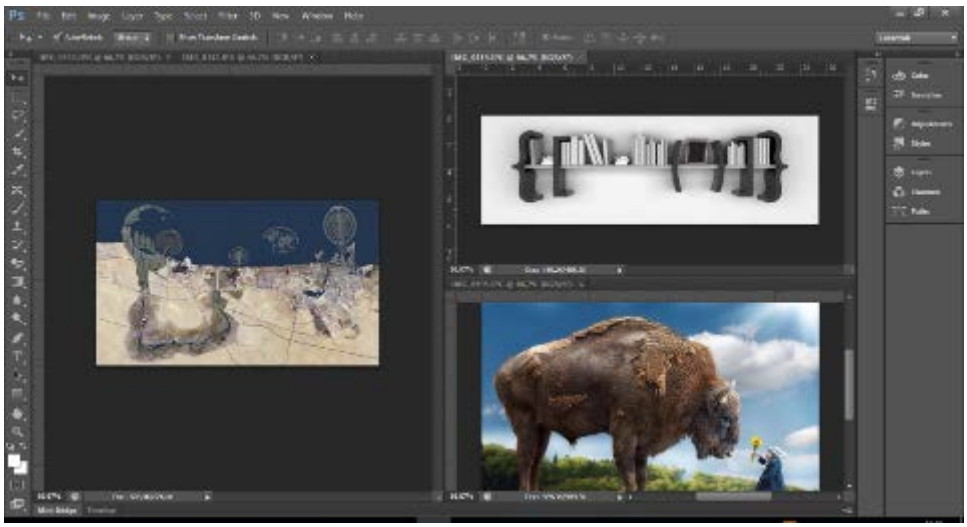
სურ. 8-12.2-up Vertical - მხოლოდ ორი გამოსახულების დალაგება ვერტიკალურად.



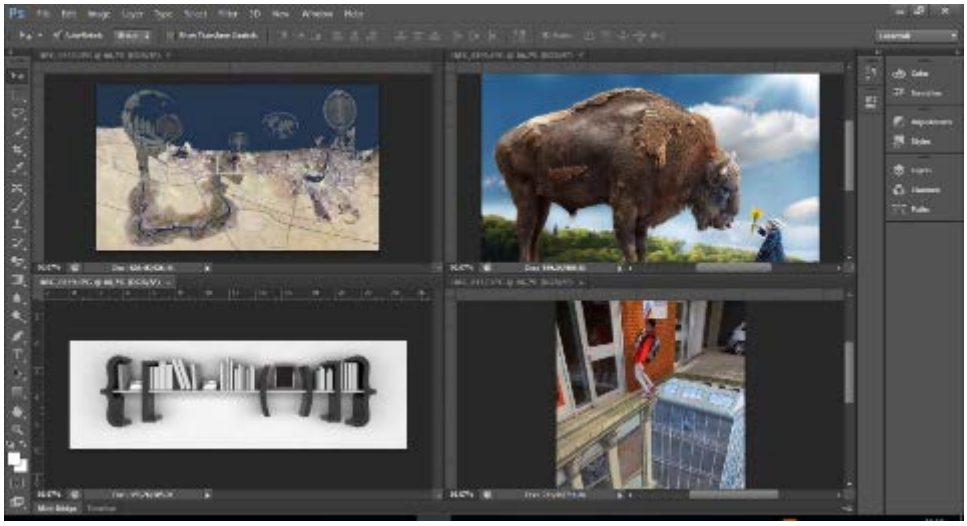
სურ. 8-16. 3-up Horizontal - მხოლოდ სამი გამოსახულების დალაგება ჰორიზონტალურად.



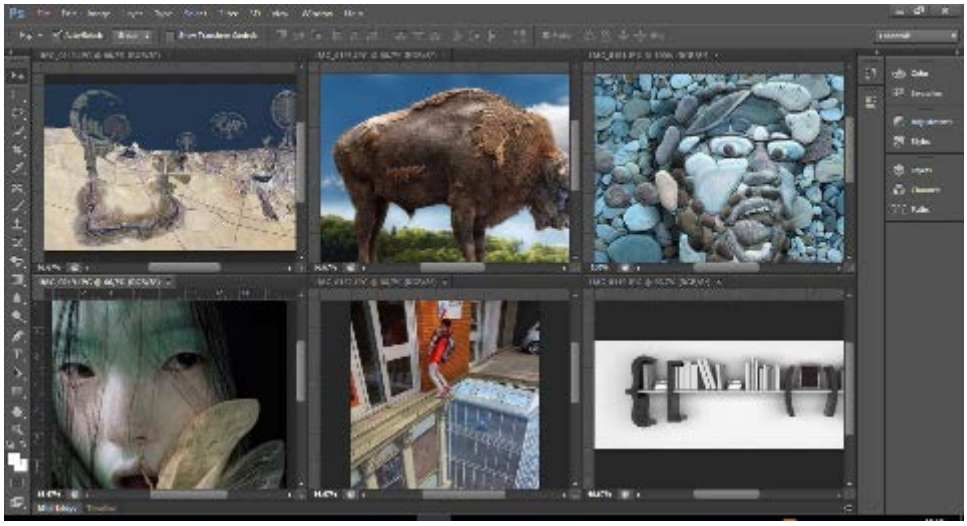
სურ. 8-15. 3-up Vertical - მხოლოდ სამი გამოსახულების დალაგება ვერტიკალურად.



სურ. 8-17. 3-up Stacked - სამად დალაგებული გამოსახულებები.

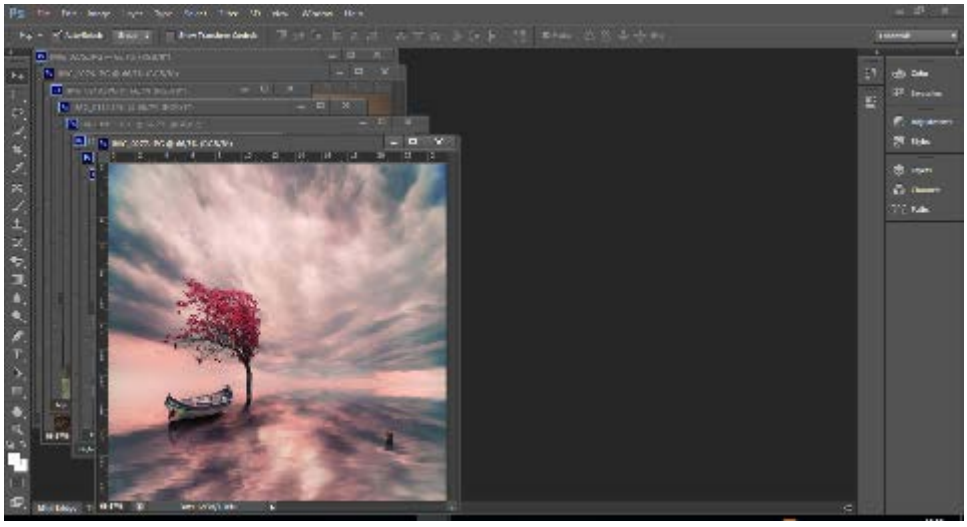


სურ. 8-19. 4-up - ოთხად დალაგებული გამოსახულებები.

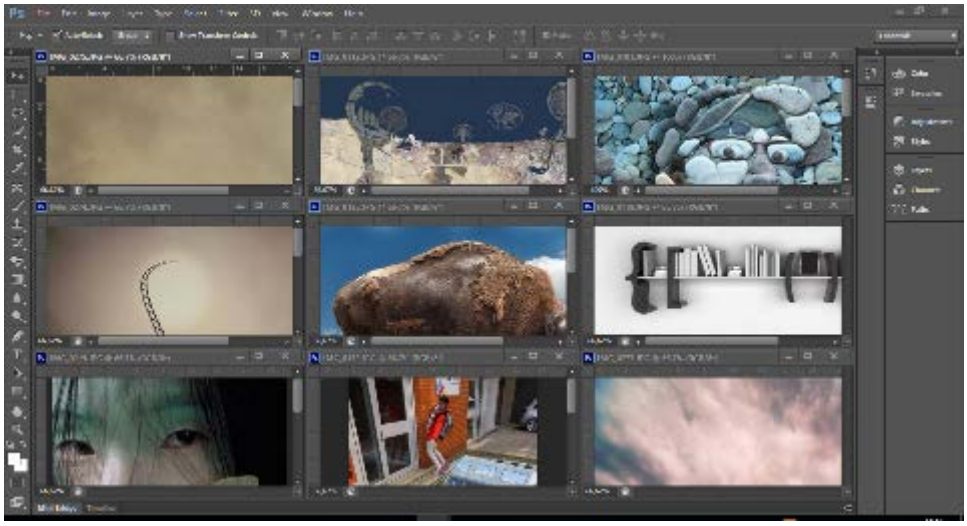


სურ. 8-18. 6-up - ექვსად დალაგებული გამოსახულებები.

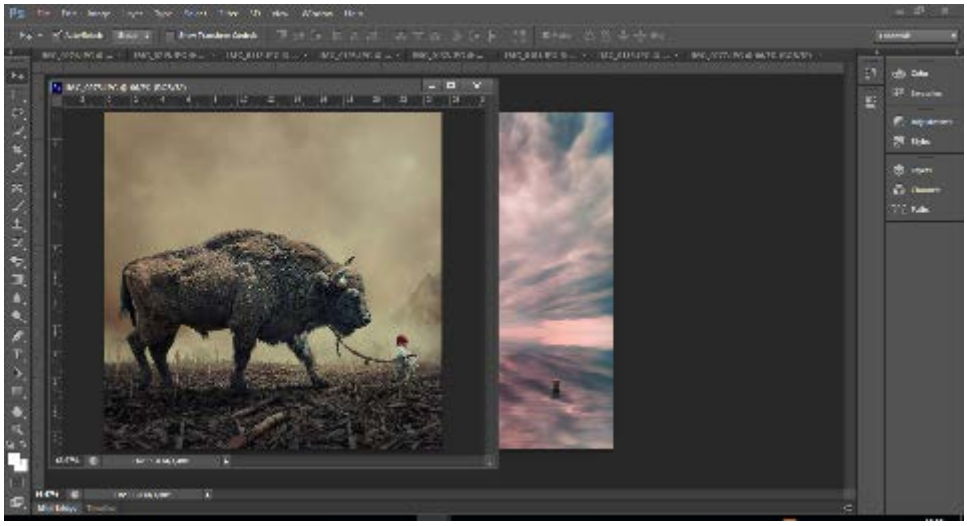
ფანჯრების ამ მეთოდებით დალაგებისას, ისინი მაინც მიმაგრებულები არიან ძირითად ფანჯარაზე ჩანართების სახით. გარდა ამისა, შეგვიძლია ფანჯრების სხვაგვარად დალაგებაც - Cascade (კასკადურად), Tile (მოზაიკურად), Float in Window(ერთი ფანჯარა თავისუფალ მდგომარეობაში), Float All in Window(ყველა ფანჯარა თავისუფალ მდგომარეობაში).



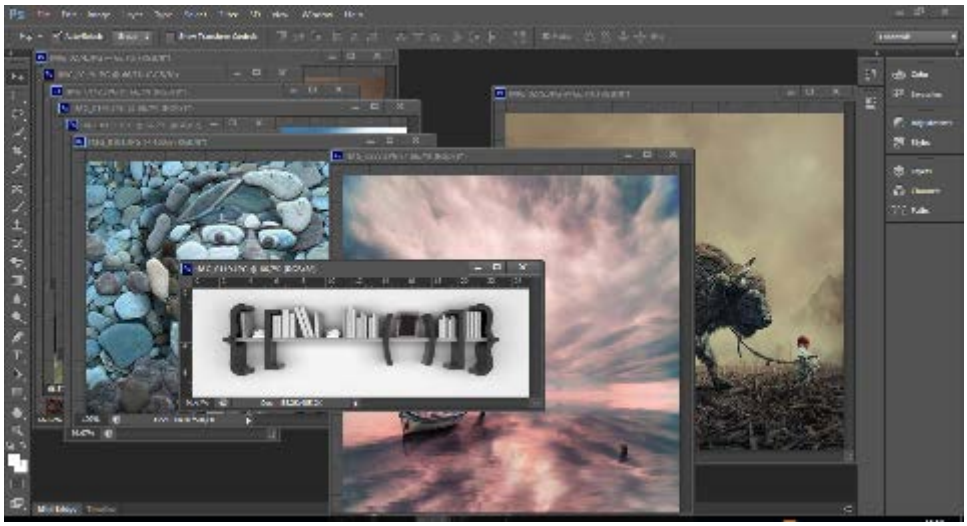
სურ. 8-20.Cascade - გამოსახულებების კასკადურად დალაგება.



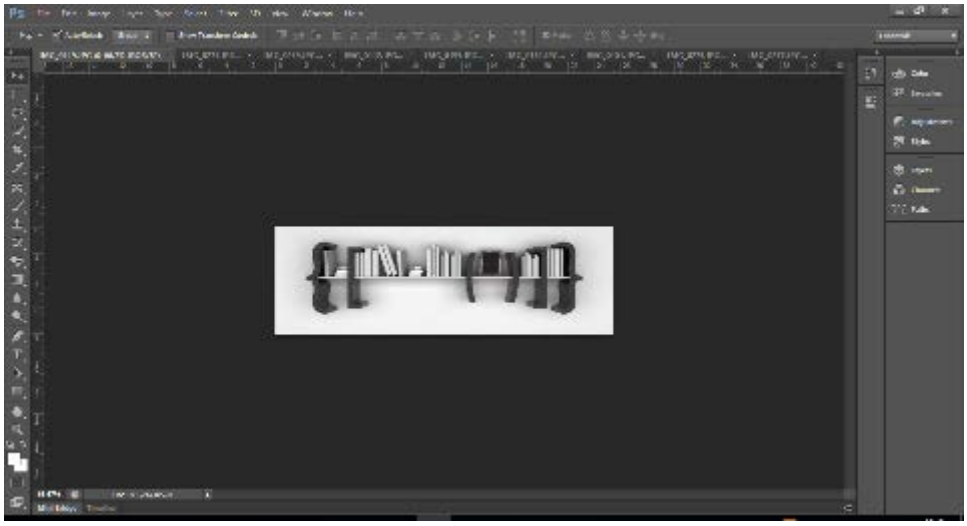
სურ. 8-22. Tile - გამოსახულებების მოზაიკურად დალაგება.



სურ. 8-21. Float in Window - ერთი გამოსახულება თავისუფალ მდგომარეობაში (ცალკე ფანჯარაში).



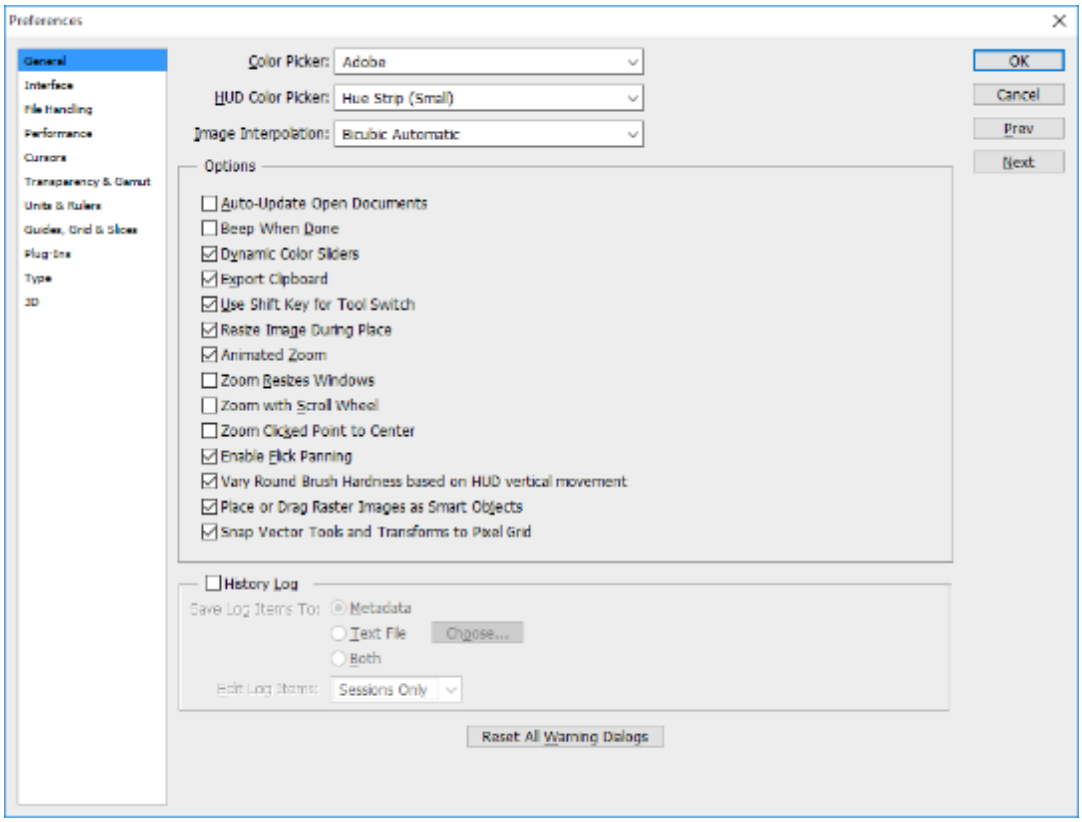
სურ. 8-23. Float All in Window - ყველა გამოსახულება თავისუფალ მდგომარეობაში (ცალკე ფანჯარაში).



სურ. 8-24. Consolidate All to Tabs - ყველა ცალკე გამოტანილ ფანჯრებს ამაგრებს ისევ სანიშნის სახით.

გრაფიკული რედაქტორის პარამეტრების განხილვა/დაყენება

სანამ მუშაობას შეუდგებით სასურველია გარკვეული ცვლილებები შეიტანოთ პროგრამის პარამეტრებში. ზოგიერთი პარამეტრის შეცვლა თქვენ დაგჭირდებათ მხოლოდ ერთხელ, ხოლო ზოგიერთი შესაძლოა პერიოდულად ცვალოთ. პარამეტრების დაყენება თვენი საქმიანობიდან და



სურ. 8-25. პროგრამა ფოტოშოპის მოწყობის ფანჯარა.

დასახული ამოცანიდან გამომდინარე შეიძლება. ფოტოგრაფი, ილუსტრატორი, მხატვარი, დიზაინერი, ტექნიკური დიზაინერი, 3D მოდელებისშემქმნელი, ანიმატორი - ყველა ირგებს პროგრამას სათავისოდ და აყენებენ მათთვის სასურველ პარამეტრებს.იმისათვის, რომ შეცვალოთ პროგრამის პარამეტრები, გამოიძახეთ Edit -> Preferences -> General (Ctrl +K). გამოსულ ფანჯარა შეიცავს ყველა იმ მახასიათებელს, რომელიც მოქმედებს სამუშაო პროცესზე ან პროგრამის წარმადობაზე:

General – პროგრამის ზოგადი მახასიათებლების დაყენება, როგორიცაა ფერების პალიტრა, მასშტაბირება ან მოქმედებების აღნუსხვა;

Interface – ინტერფეისის პარამეტრების დაყენება, როგორიცაა მაგალითად ინტერფეისის ფერის დაყენება

File Handling – პროგრამის ფაილებთან მუშაობის პარამეტრები, მაგალითად ავტომატურად შენახვის დრო, Raw ტიპის ფაილებთან მუშაობა და სხვა

Performance – წარმადობა. აქ შეგიძლიათ დააყენოთ, ოპერატიული მეხსიერების რა ნაწილი გამოიყენოს პროგრამამ. დისკი, რომელზეც პროგრამა განათავსებს ვირტუალური მეხსიერების ფაილს. მოქმედებების ისტორიის რაოდენობა და სხვა;

Cursors – აქ შეგიძლიათ დააყენოთ პროგრამაში როგორ გამოჩნდეს კურსორი სხვადასხვა ინსტრუმენტისთვის, მაგალითად ფუნჯისთვის.

Transparency & Gamut – გამჭვირვალობისა და გამა (ფერთა). აქ შეგიძლიათ დააყენოთ სამუშაო ფაილში გამჭვირვალობა როგორ გამოჩნდეს. ასევე, როდესაც მოხდება ფერთა ცდომილება, მაგალითად RGB-დან CMYK-ში გადაყვანისას, აცდენილი ფერები რა ფრად გამოისახოს ეკრანზე.

Units & Rulers – აქ შეგიძლიათ დააყენოთ თქვენი სამუშაო საზომი ერთეული. მაგალითად ჩვენ შემთხვევაში ეს იქნება სანტიმეტრი, თუმცა როდესაც იმუშავებთ გამოსახულებებზე, რომელიც ინტერნეტისთვის მზადდება, იქ შეგიძლიათ გამოიყენოთ პიქსელები. აქ ასევე შეგიძლიათ დააყენოთ ახალი დოკუმენტებისთვის ნაგულისხმევი გარჩევადობის ხარისხი. სტანდარტულად მოცემული 300 და 72 px/inch, თუმცა ეს შეგიძლიათ შეცვალოთ რასაკვირველია.

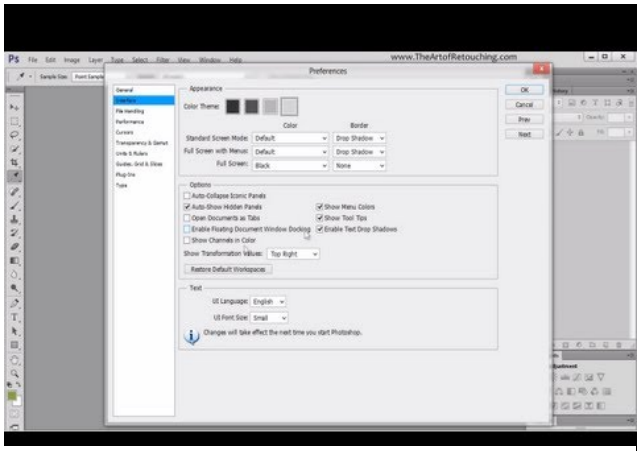
Guides, Grid & Slices – აქ შეგიძლია შეცვალოთ მიმმართველების, „ჭკვიანი“ მიმმართველების ფერები, დამხმარე ბადის დაყოფის მაჩვენებლები და სხვა.

Plug-Ins – პროგრამისთვის გამოსულია სხვადასხვა სახის დამატებები, რომელთა ადგილმდებარეობა შეგიძლიათ აქედან მიუთითოთ. ეს დამატებები გაძლევენ საშუალებას ისეთი ეფექტები შექმნათ გამოსახულებისთვის, რომელიც შეუძლებელია მიიღოთ პროგრამაში არსებული ხელსაწყოების დახმარებით.

Type – რამდენიმე პარამეტრი, რომელიც დაგეხმარებათ ტექსტთან მუშაობაში.

3D – აქ შეგიძლიათ დააყენოთ ვიდეოადაპტერის მახასიათებლები, გამოსახულების გარდაქმნის მეთოდები და სხვა.

შენიშვნა: *გახსოვდეთ, რომ ზოგიერთი პარამეტრის შეცვლა მოქმედებს კომპიუტერის წარმადობაზე, მაგალითად ოპერატიული მეხსიერების გამოყენება. ამიტომ მახასიათებლები უნდა შეირჩეს ისე, რომ არ მოხდეს მთლიანობაში კომპიუტერის წარმადობის შემცირება.*



ვიდეო 0-1. პროგრამის პარამეტრების მართვა

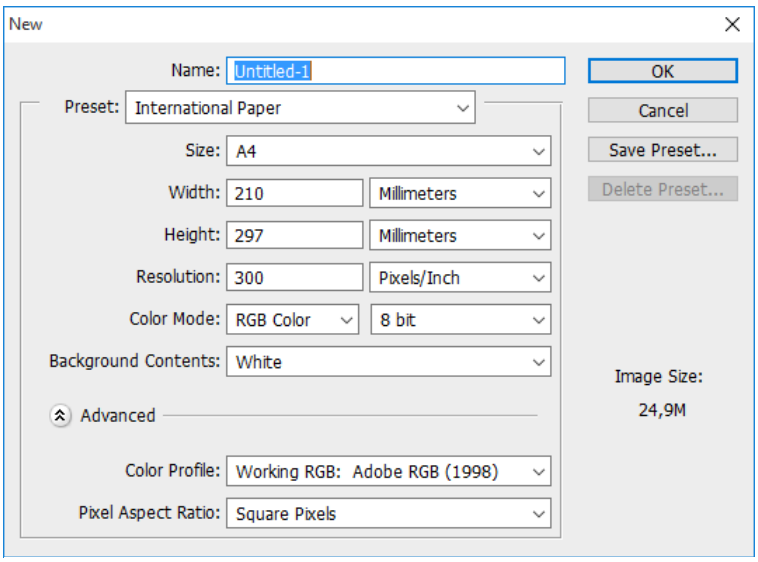
დოკუმენტის კლასიფიკაცია და საზომი ერთეულები

მას შემდეგ, რაც მოხდება სამუშაოდ ყველანაირი პარამეტრის დაყენება: პროგრამის პარამეტრები, სამუშაო სივრცე, საზომი ერთეულები და სხვა, შეგიძლიათ დაიწყოთ გამოსახულებასთან მუშაობა.

გამოსახულებასთან მუშაობა შეგიძლიათ დაიწყოთ ორი გზით: შექმნათ ახალი გამოსახულება ან/და გახსნათ უკვე არსებული.

მივყვეთ რიგს და დავიწყოთ პირველით ანუ ახალი გამოსახულების შექმნა.

ამისათვის ვიძახებთ File -> New (Ctrl + N). გამოსულ დიალოგურ ფანჯარაში ჩვენ უნდა დავაყენოთ ახალი დოკუმენტის მახასიათებლები.



სურ. 8-26. ახალი დოკუმენტის შექმნის ფანჯარა.

ამ ფანჯარაში უნდა შევიყვანოთ ისე მახასიათებლები, როგორებიცაა ფაილის დასახელება, მისი ზომა, გარჩევადობის ხარისხი და სხვა. პროგრამას გააჩნია წინასწარ გამზადებული დოკუმენტის ტიპები, რომლებიც თქვენ შეგიძლიათ ირჩიოთ ან შექმნათ თქვენი საკუთარი. და თუ მას ხშირად იყენებთ, შეინახოთ ის სიაში და მომავალში მიმართოთ მას.

გამზადებული დოკუმენტები მოცემულია ჩამოსაშლელ მენიუში Preset:

Clipboard - თუ თქვენ რომელიმე პროგრამაში ან თუნდაც ფოტოშოპში, დაკოპირებული გაქვთ რაიმე გამოსახულება, ფოტოშოპი ავტომატურად კითხულობს მის მონაცემებს (სიგანე, სიმაღლე, გარჩევადობა) და გთავაზობთ შესაბამისი მახასიათებლების დოკუმენტის შექმნას.

Default Photoshop Size - ფოტოშოპის ნაგულისხმევი ზომა,გამზადებული დოკუმენტის ერთ-ერთი ნაირსახეობა.

U.S. Paper - დოკუმენტის ნაირსახეობა, რომელსაც იყენებენ ა.შ.შ.-ში. ამ დროს აქტიურდება ჩამოსაშლელი მენიუ Size, სადაც ფურცლის სხვადასხვა ვარიანტებიც შეგიძლიათ შეარჩიოთ. გარჩევადობა 300 px/inch.

International Paper - დოკუმენტის ნაირსახეობა, სადაც მოცემულია საყოველთაოდ გამოყენებადი ფურცლის ზომები. მენიუ Size-ში თქვენ ამ ფურცლის ზომებს იხილავთ: A4, A3, B5 და სხვა. გარჩევადობა 300 px/inch.

Photo - აქ მოცემული დოკუმენტების ზომები განკუთვნილია ფოტოსურათებისთვის, როგორც ვერტიკალური ისე ჰორიზონტალური ორიენტაციისთვის. თუმცა ეს ფორმატები მორგებულია ამერიკაში მიღებული სტანდარტისთვის. გარჩევადობა 300 px/inch.

Web - აქ მოცემულია ფორმატები, რომლებიც ინტერნეტისთვისაა განკუთვნილი. საზომი ერთეულებიც შესაბამისია აღებული. გარჩევადობა 72 px/inch.

Mobile & Device - ფორმატები, რომლებიც მოწყობილობების ეკრანის ზომას ასახავს. საზომი ერთეულები აქაც პიქსელებშია მოცემული. გარჩევადობა 72 px/inch.

Film & Video - დოკუმენტის ნაირსახეობა, რომლის პარამეტრები ორიენტირებულია ვიდეოპროდუქციაზე. მენიუ Size-ში თქვენ ნახავთ უამრავ ვიდეო ფორმატს, რომელიც შეგიძლიათ აირჩიოთ თქვენი ამოცანიდან გამომდინარე. გარჩევადობა 72 px/inch.

Custom - მომხმარებლის მიერ დაყენებული პარამეტრები.

ახალი დოკუმენტი შეგიძლიათ შექმნათ უკვე არსებული მზა დოკუმენტების ბაზაზე ან შექმნათ თქვენი საკუთარი. რა ძირითად მონაცემებს ვუთითებთ, როდესაც გვინდა შევქმნათ ახალი გამოსახულება:

Width - გამოსახულების სიგანე

Height - გამოსახულების სიმაღლე

მათ გვერდით ჩამოსაშლელ მენიუში ვუთითებთ საზომ ერთეულს - დუიმი, სანტიმეტრი, მილიმეტრი ან პიქსელი. თუ ვმუშაობთ დუიმებში და საბოლოო შედეგი მილიმეტრებში უნდა დაიბეჭდოს, უნდა გავითვალისწინოთ, რომ 1 დუიმი = 2,54 სანტიმეტრს

Resolution - გარჩევადობა. აქ ვუთითებთ გამოსახულების გარჩევადობის ხარისხს. თუ გამოსახულება განკუთვნილი იქნება ინტერნეტისთვის, მაშინ ვუთითებთ 72-96 px/inch, ხოლო თუ პოლიგრაფიული მიზნებისთვის უნდა გამოვიყენოთ, მაშინ 300 px/inch ან მეტი.

Color Mode - ფერთა მოდელი, რომელიც გამოიყენება ამ შემთხვევაში. ისევ და ისევ RGB თუ გამოსახულებას ვამზადებთ ინტერნეტისთვის, ხოლო თუ პოლიგრაფიული მიზნისთვის, მაშინ ვირჩევთ CMYK ფერთა მოდელს. ხოლო მის გვერდით ვირჩევთ, რამდენ ბიტიანი იქნება ფერთა სიღრმე. 8 ბიტი სავსებით საკმარისია ორივე მიზნისთვის, თუ ჩვენ წინაშე არ დგას რაიმე კონკრეტული ამოცანა, რომელიც უფრო მაღალბიტიან ფერთა სიღრმეს მოითხოვს. დანარჩენი ამ ეტაპზე დავტოვოთ უცვლელად და დავაწვეთ OK. იმ შემთხვევაში თუ თქვენ საკმაოდ ხშირად გიწევთ ამ ზომებთან მუშაობა, მაშინ ღილაკი Save Preset-ის დაგეხმარებათ შეინახოთ დოკუმენტი შაბლონის სახით და შემდეგში თქვენ მას აირჩევთ დოკუმენტების სიიდან და აღარ დაგჭირდებათ მონაცემების შეყვანა ხელით.

შემდეგ ეტაპზე ჩვენ განვიხილავთ, თუ როგორ უნდა ვიმუშაოთ უშუალოდ ფაილში და გამოვიყენებთ შესაბამის ხელსაწყოებს, რომლებიც დაგვეხმარებიან გამოსახულების შექმნასა თუ რედაქტირებაში.

პრაქტიკული დავალებები/კითხვები თვითშეფასებისათვის

1. დაასახელეთ სხვაობა რასტრულ და ვექტორულ გამოსახულებებს შორის;
2. რა არის გარჩევადობა? ჩამოთვალეთ ყველაზე გავრცელებული.
3. ჩამოთვალეთ ფაილის ფორმატები;
4. ჩამოთვალეთ სამუშაო სივრცის ტიპები;
5. გამოაჩინეთ სასურველი სამუშაო პანელები;მოაწყეთ სამუშაო სივრცე, როგორც თქვენ ჩათვლით საჭიროდ და შეინახეთ.
6. მოიძიეთ ინფორმაცია ფბ-ს მთავარი სურათის (cover) შესახებ;გაეცანით ფბ-ს რეკომენდაციების გამოსახულების მახასიათებლებთან დაკავშირებით;შექმენით ახალი ფაილი, რომელიც დააკმაყოფილებს ამ მახასიათებლებს (ზომა, გარჩევადობა, ფერის მოდელი, ფაილის ფორმატი) და შეინახეთ ის შაბლონის სახით;
7. შექმენით ფაილი, რომელიც მომავალში გამოყენებული იქნება წიგნის ყდის გასაფორმებლად. ზომა 130 x 195 მმ. ფაილი შექმენით ბეჭდვისთვის საჭირო პირობების გათვალისწინებით (ფერთა მოდელი, გარჩევადობა, ფაილის ფორმატი).

მიმდინარე პარაგრაფის თემატიკა

- მონიშვნის ხელსაწყოების გაცნობა
- მოქმედებები მონიშნულ არეზე
- დაფარვის მეთოდების მიმოხილვა
- ფუნჯი და მისი ნაირსახეობები
- საშლელი და მისი ნაირსახეობები

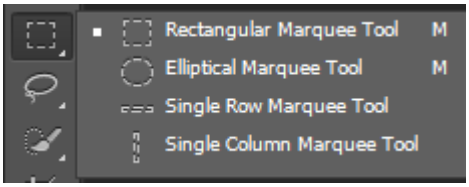
მონიშვნის ხელსაწყოები და მონიშვნა

მონიშვნის ინსტრუმენტები ერთ-ერთი უმნიშვნელოვანესია ფოტოშოპში. მოქმედების პრინციპი მარტივია: როდესაც ჩვენ ვნიშნავთ რაიმე არეს, ჩვენი შემდეგი მოქმედებები შეეხება მხოლოდ არის შიგნით მყოფ გამოსახულების ფრაგმენტს.ინსტრუმენტების დახმარებით ჩვენ შეგვიძლია მოვნიშნოთ როგორც ერთი პიქსელი, ისე პიქსელთა დიდი ჯგუფი და მათზეგარკვეული მოქმედებები მოვახდინოთ: ფერთა კორექცია, შეფერვა, კოპირება, ტრანსფორმირება და სხვა.

ფოტოშოპში მონიშვნის სხვადასხვა ხელსაწყო არსებობს. თვისებებიც განსხვავდება ერთმანეთისგან - შეგვძლია გამოსახულებაზე მოვნიშნოთ მართკუთხა, ოვალური ან თავისუფალიფორმის ფრაგმენტი, ერთფეროვანი პიქსელები, მოვნიშნოთ გამოსახულების ორი ან მეტი ფრაგმენტი და სხვა. ინსტრუმენტთა პანელზე არსებობს მონიშვნის ხელსაწყოთა მთელი ჯგუფი (სურ. 10: ა). პროგრამის მენიუში მას ეთმობა ცალკე მენიუ Selection - აქ მოთავსებული ფუნქციები გვაძლევენ საშუალებას მოვახდინოთ ჩვენი მონიშვნა გავხადოთ უფრო ზუსტი, მოვახდინოთ მასზე სხვადასხვა მოქმედებები, რომ შემდეგში მიღებული შედეგი ჩვენს მოთხოვნებს შეესაბამებოდეს.

მონიშვნის ინსტრუმენტები (Marque Tool)

აქ გაერთიანებული ოთხი ინსტრუმენტი: Rectangular, Oval, Single Row და Single Column



სურ. 8-27. Rectangular Marque Tool - მართკუთხა მონიშვნის ინსტრუმენტი.

Single Row და Single Column შემთხვევაში ხდება 1 პიქსელის სიგანის არის მონიშვნა მთელი გამოსახულების გასწვრივ შესაბამისად ჰორიზონტალურად და ვერტიკალურად.

შენიშვნა: Shift კლავიშთან ერთად მონიშვნის შემთხვევაში ხდება სიმეტრიული არეების (კვადრატი და წრე) მონიშვნა. ხოლო Shift და Alt კლავიშების კომბინაციით მონიშვნა სრულდება ცენტრიდან და მიიღება კვადრატი და წრე.

ნებისმიერი ინსტრუმენტის არჩევის შემდეგ, ფორმატირების ზოლზე (სურ. 3: ბ) ჩნდება ინსტრუმენტისთვის დამახასიათებელი პარამეტრები. ამ შემთხვევაშიც აქ გამოჩნდება ის პარამეტრები, რომელიც ამ მონიშვნის ინსტრუმენტს ახასიათებს:

სურ. 8-28. ფორმატირების ზოლი. მასზე გამოტანილია მონიშვნის ინსტრუმენტის დამატებითი პარამეტრები.

ა. მონიშვნის მეთოდები (მარცხნიდან მარჯვნივ):

ახალი მონიშვნა (New Selection) - ახალი მონიშნული არე. ამ დროს თუ გაქვთ რამე მონიშნული, ის გაუქმდება და შეიქმნება ახალი;

მონიშვნის დამატება (Add to Selection) - უკვე მონიშნულ არეს დაემატება თქვენს მიერ მონიშნული ახალი არე;

მონიშვნის გამოკლება (Subtract from Selection) - უკვე მონიშნულ არეს შეგიძლიათ გამოაკლოთ ფრაგმენტი;

მონიშნული არეების გადაკვეთა (Intersect with Selection) – ორი (ან მეტი) მონიშნული არის გადაკვეთიდან შეგიძლიათ გამოყოთ ერთი სასურველი არე.

დეტალური ინფორმაცია, თუ როგორ უნდა გამოიყენოთ ეს ოთხი ინსტრუმენტი, შეგიძლიათ იხილოთ სტატიაში Unlock The Full Power Of Basic Selections In Photoshop.

ბ. დარბილება (Feather)

მისი საშუალებით ხდება მონიშნული არის საზღვრების დარბილება. უთითებთ რამდენ პიქსელიანი იყოს საზღვრის დარბილება. რაც მეტია მაჩვენებელი, მით უფრო რბილია საზღვარი.

სურ. 8-29. ამ გამოსახულებაზე მოხდა მონიშნული ფრაგმენტის წაშლა. მარცხნივ დარბილება 0 პქს, მარჯვნივ დარბილება 5 პქს.

გ. მონიშვნის სტილი (Style)

ჩვეულებრივი მონიშვნის დროს ჩვენ ვხაზავთ ნებისმიერი პროპორციისა თუ ზომის არეს. თუმცა, პროპორციების დაცვა და წინასწარ განსაზღვრული ზომის მონიშნული არის შექმნა შესაძლებელია. ამისათვის ჩამოსაშლელი მენიუ Style-შიარის სამი რეჟიმი:

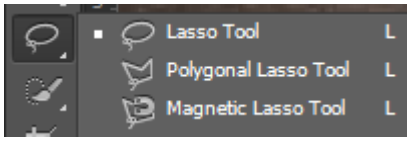
Normal (ჩვეულებრივი) მეთოდი ავტომატურად არის დაყენებული და მონიშვნის პროპორციებსა და ზომებს მომხმარებელი არეგულირებს.

Fixed Ratio (ფიქსირებული პროპორცია). ამ შემთხვევაში, აქტიურდება სიგრძისა (Width) და სიმაღლის (Height) ველები, რომლებშიც უთითებთ მონიშვნის სიგანისა და სიმაღლის პროპორციებს: 3:4, 16:9 და ა.შ.;

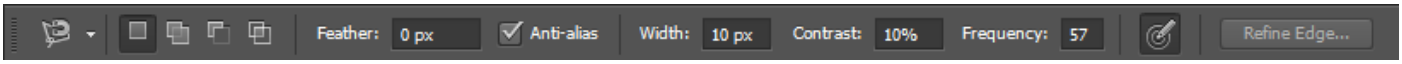
Fixed Size (ფიქსირებული ზომა).ამ დროს შესაძლებელია სიგრძისა (Width) და სიმაღლის (Height) ველებში სასურველი ზომების მითითება. შემდეგ საკმარისია გამოსახულებაზე მაუსის დაწკაპუნება და ავტომატურად შეიქმნება მითითებული ზომების შესაბამისი მონიშვნის არე.

ინსტრუმენტები ლასო (Lasso Tool)

ამ ინსტრუმენტების დახმარებით ჩვენ ნებისმიერი ფორმის ობიექტი შეგვიძლია მოვნიშნოთ. შედის სამი ინსტრუმენტი: Lasso Tool, Polygonal Lassoდა Magnetic Lasso.

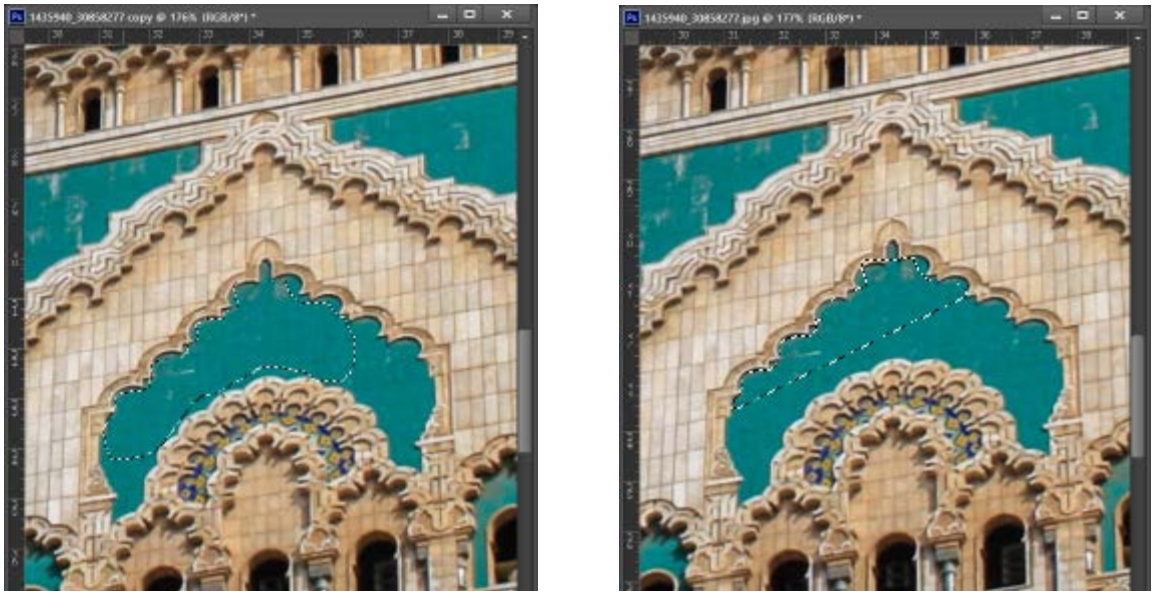


ფორმატირების ზოლზე თქვენ შეგიძლიათ დააყენოთ სხვადასხვა პარამეტრი. აქ თქვენ შეხვდებით ისეთივე ღილაკებს, როგორებიცაა მონიშნული არის დამატება, გამოკლება, დარბილება. ასევე მაგნიტური ლასოს შემთხვევაში ჩნდება კიდევ რამდენიმე დამატებითი ფუნქცია:



ა. ლასო (Lasso Tool)

მისი დახმარებით ჩვენ შეგვიძლია არასწორი ფორმის ობიექტების მონიშვნა. მუშაობის პრინციპი ესეთია: ხელაუღებლად უნდა შემოვატაროთ მოსანიშნი ობიექტის კონტურზე და მივიყვანოთ კურსორი იმ წერტილში, საიდანაც დავიწყეთ. იმ შემთხვევაში თუ გავუშვებთ ხელს, მაშინ ბოლო და თავი შეერთდებიან უმოკლესი წერტილით.



სურ. 8-30. მარცხენა სურათზე თქვენ ხედავთ, რომ მონიშნულია გარკვეული არე, ხოლო მარჯვენა სურათზე პირდაპირ შეერთდა ბოლო და საწყისი წერილი.

ლასოთი მუშაობა მოითხოვს გარკვეულ სიზუსტეს, იმისათვის, რომ ფრაგმენტი მოინიშნოს ისე, როგორც საჭიროა. დიდი და უსწორმასწორო ფრაგმენტების მონიშვნა მოუხერხებელია ამ ინსტრუმენტით. ასეთი მეთოდისთვის ძალიან კარგია მეორე ინსტრუმენტი - **სწორხაზოვანი ლასო** (Polygonal Lasso).

გ. სწორხაზოვანი ლასო (Polygonal Lasso)

სწორხაზოვანი ლასოს გამოყენების დროს თქვენ კი არ ხაზავთ ერთ სრულ ფორმას, არამედ წერტილ-წერტილ მიყვებით ობიექტის კონტურს და როდესაც შემოავლებთ მთლიანად, ბრუნდებით საწყის წერტილში და ბოლო წერტილს აერთებთ საწყისთან. თუ რომელიმე ადგილას ორჯერ დააწკაპებთ სწრაფად, ისევე როგორც ჩვეულებრივი**ლასოს** შემთხვევაში, ბოლო წერტილი დაუკავშირდება საწყისს უმოკლესი გზით.

სწორხაზოვანი ლასო კარგია გამოიყენოთ ისეთი ობიექტების მოსანიშნად, რომელსაც არასწორი გეომეტრიულიფორმა აქვს.

სურ. 8-31. სწორხაზოვანი ლასოს გამოყენების მაგალითი.

შენიშვნა: დააწექით Backspace კლავიშს, რომ წაშალოთ დასმული წერტილი; თუ მონიშვნისას შეგხვდათ გამრუდებული კონტური, დააწექით Alt კლავიშს და ინსტრუმენტი დროებით გადაიქცევა ჩვეულებრივ ლასოთ.

გ. მაგნიტური ლასო (Magnetic Lasso)

მაგნიტური ლასო ძალიან მოსახერხებელია გამოიყენოთ იქ, სადაც ჭარბობს ფერი და მისი ტონალობა. როდესაც მოსანიშნი ობიექტის ტონალობა ძალიან მიახლოებულია ფონის ფერთან, ეს ინსტრუმენტი შეუცვლელია. ფორმატირების ზოლზე სტანდარტული მახასიათებლების გარდა (მოსანიშნი არის დამატება, გამოკლება და სხვა),არის კიდევ სამი პარამეტრი:

სიგანე (Width) - ვუთითებთ, ცენტრალური წერტილიდან რა მანძილზე უნდა განისაზღვროს ობიექტის საზღვრები. მერყეობს 1-დან 256 პიქსელამდე;

კონტრასტი (Contrast) – ინსტრუმენტისთვის აქ ვუთითებთ, რამდენად კონტრასტულია ობიექტი ფონთან შედარებით. თუ ობიექტი კარგად გამოიკვეთება უკანა ფონზე, მიუთითეთ 30% და მეტი. თუ სუსტად გამოიკვეთება, მაშინ ნაკლები;

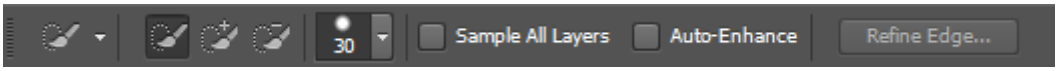
სიხშირე (Frequency) - განსაზღვრავს, რამდენად ხშირად უნდა დასვას ინსტრუმენტმა საყრდენი წერტილები. რაც უფრო მეტია სიხშირე, მით მეტ საყრდენ წერტილს დასვამს.

სწრაფი მონიშვნის ინსტრუმენტები

სწრაფი მონიშვნის ინსტრუმენტებს მიეკუთვნებიან **სწრაფი მონიშვნა** (Quick Selection), **ჯადოსნური ჯოხი** (Magic Wand). მათი მოქმედება ფერისა და მის ტონალობაზეა დამყარებული.

ა. სწრაფი მონიშვნა (Quick Selection)

ინსტრუმენტის მოქმედება დაფუძნებულია მოსაზღვრე პიქსელების მსგავსებაზე. მონიშვნის პროცედურა ფუნჯით ხატვის პროცესს გავს. **ჯადოსნური ჯოხისგან** (Magic Wand) განსხვავებით არ გააჩნია პარამეტრი **შეღწევადობა** (Tolerance) და გაძლევთ საშუალებას თქვენ თვითონ განსაზღვროთ მონიშვნის არე (მოსაზღვრე პიქსელები) გამოსახულების შემოხატვით.



მისი პარამეტრებია:

ახალი მონიშნული არე (New Selection) - გამოსახულებაზე ახალი მონიშნული არის შექმნა;

მონიშნული არის დამატება (Add to Selection) - ახალი მონიშნული არის დამატება გამოსახულებაზე უკვე არსებულ მონიშნულ არესთან;

მონიშნული არის გამოკლება (Subtract from Selection) - გამოსახულებაზე უკვე არსებულ მონიშნულ არეს ნაწილის გამოკლება;

ფუნჯი და მისი მახასიათებლები:

ზომა (Size) - ფუნჯის ზომა. მინიმალური სივრცე, რომელიც გამოსახულებაზე იქნება მონიშნული.

სიმკვეთრე (Hardness) - განსაზღვრავს მონიშნული არის კიდეების სიმკვეთრეს.

ინტერვალი (Spacing) - მონიშნულ არეებს შორის მონიმალური მანძილი, რომელიც გამოსახულებაზე ხატვისას წარმოიქმნება.

კუთხე (Angle) - გამოსახულებაზე ხატვისას ფუნჯის დახრის კუთხე.

ფორმა (Roundness) - მონიშნული არის ფორმა, რომელიც გამოსახულებაზე ხატვისას წარმოიქმნება.

ყველა ფენის ნიმუში (Sample all Layers) - თუ ეს პარამეტრი ჩართულია მონიშვნისას გათვალისწინებული იქნება ყველა არსებულ ფენაზე მყოფი პიქსელების მსგავსება;

ავტომატური გაძლიერება (Auto-Enhance) - თუ ეს ფუნქცია ჩართულია, შესაძლებელია კიდეების გასწორება მოსანიშნი ობიექტის კონტურების მიხედვით.

სურ. 8-32. სწრაფი მონიშვნის ინსტრუმენტის გამოყენების მაგალითი, ფოტო: გიორგი კევლიშვილი

ბ. ჯადოსნური ჯოხი (Magic Wand)

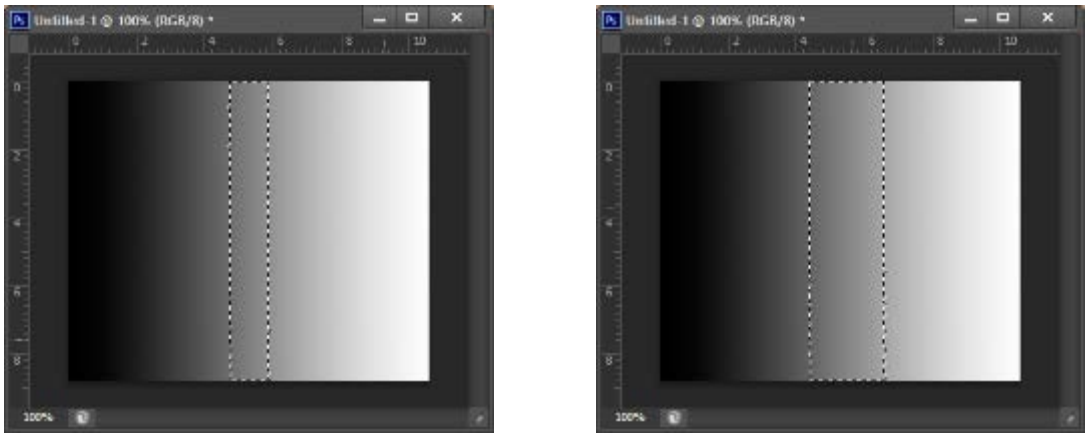
მისი მოქმედება დამოკიდებულია 4 მახასიათებელზე: **ნიმუშის ზომა** (Sample Size), **შეღწევადობა** (Tolerance),**მომიჯნავე პიქსელები** (Contiguous) და **ყველა ფენის ნიმუში** (Sample All Layers).

შეღწევადობა (Tolerance) ძირითადი პარამეტრია, რომელიც განსაზღვრავს, სიკაშკაშის რამდენ მახასიათებელს მოიცავს ინსტრუმენტი მონიშნულ არეში. მისი მნიშვნელობა მერყეობს 0-დან 255-მდე. აქ 0 არის სიკაშკაშის მხოლოდ ერთი მაჩვენებელი, 32 (რაც ნაგულისხმევად გამოიყენება) მოიცავს სიკაშკაშის 32 მაჩვენებელს, 255 კი ყველ მაჩვენებელს (ფაქტობრივად მთლიან გამოსახულებას). ინსტრუმენტი მაშინვე წყვეტს მონიშვნას, როდესაც აღმოაჩენს, რომ ფერის ტონალობა განსხვავდება **შეღწევადობაში** მითითებული მაჩვენებლისგან.

მიაქციეთ ყურადღება, რომ აქ საუბარია სიკაშკაშის დონეზე და არა ფერზე, რადგან ინსტრუმენტი ფერს საერთოდ არ ითვალისწინებს. ამის ნაცვლად ინსტრუმენტი აანალიზებს გამოსახულების RGB (CMYK)არხებს, რომლებიც წარმოადგენენ ერთმანეთზე დადებულ სამ (ოთხ) გამოსახულებას ნაცრისფერ ტონალობაში.

ქვემოთ მოცემულ მაგალითზე ნათლად ჩანს, როგორ მუშაობს **შეღწევდობა**.

მარცხენა სურათზე **შეღწევადობა** დაყენებულია 16. ეს იმას ნიშნავს, რომ დაწკაპუნების წერტილიდან ინსტრუმენტმა მონიშნა 16 პიქსელი მარცხნივ და 16 პიქსელი მარჯვნივ. მეორე სურათზე შეღწევადობის მაჩვენებელი არის 32, რაც ნიშნავს, რომ ყოველი მხრიდან მან მონიშნა 32 ტონალობა.



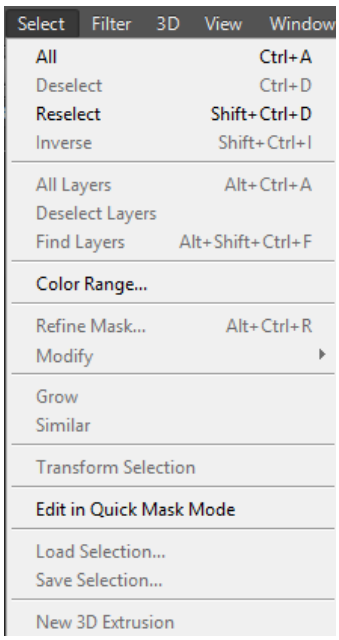
სურ. 8-33. მარცხენა სურათზე შეღწევადობის მაჩვენებელია 16, ხოლო მარჯვენაზე შეღწევადობის მაჩვენებელია 32

მომიჯნავე პიქსელები (Contiguous) - ეს პარამეტრი ავტომატურად ჩართულია და მიუთითებს ინსტრუმენტს, რომ უნდა მონიშნოს ის პიქსელები, რომლებიც ესაზღვრებიან იმ ადგილს, სადაც ინსტრუმენტით დააწკაპუნეთ. იმ შემთხვევაში თუ გსურთ მსგავსი ტონალობის პიქსელების მონიშვნა მთელი გამოსახულების მასშტაბით, მაშინ ეს პარამეტრი გამორთეთ.

ყველა ფენის ნიმუში (Sample All Layers) - ჩართეთ ეს პარამეტრი იმ შემთხვევაში, თუ გსურთ რომ ინსტრუმენტის მოქმედება სხვა ფენებზეც გავრცელდეს.

ნიმუშის ზომა (Sample Size) - ამ პარამეტრებს ინსტრუმენტი იყენებს ტონალობის შესაფასებლად. **წერტილოვანი ნიმუში** (Point Sample) - ითვალისწინებს ზუსტად იმ წერტილის მახასიათებელს, სადაც თქვენ დააწკაპუნეთ ინსტრუმენტი; **3x3 საშუალო** (3x3 Avarage) - ითვალისწინებს 9 პიქსელის მახასიათებელს; 5x5 საშუალო (5x5 Avarage) - ითვალისწინებს 25 პიქსელის მახასიათებელს და ა.შ.

გამოსახულებაზე არეების, პიქსელების, ცალკეული ფერის ტონალობების მონიშვნას ძალიან დიდი მნიშვნელობა ენიჭება. პროგრამაში ცალკეა გამოტანილი **მონიშვნის** მენიუ (Select), რომლის დახმარებითაც თქვენ შეგიძლიათ როგორც მონიშვნა, ისე სხვადასხვა მოქმედების შესრულება.



სურ. 8-34. მონიშვნის მენიუ (Select)

ყველაფრის მონიშვნა (All– Ctrl+A)- გამოსახულებას მთლიანად მონიშვნა.

მონიშვნის გაუქმება (Deselect – Ctrl+D).

ხელახლა მონიშვნა (Reselect – Shift+Ctrl+D) - ბოლოს გაუქმებული მონიშვნის აღდგენა.

მონიშვნის შებრუნება (Inverse– Shift+Ctrl+I)- მოინიშნება გამოსახულების ის ნაწილი რაც არ არის მონიშნული.

ყველა ფენა (All Layers – Alt + Ctrl + A) - ფენების პანელში მოხდება გამოსახულების ყველა ფენის მონიშვნა.

ფენების მონიშვნის გაუქმება(Deselect Layers).

მსგავსი ფენების მონიშვნა (Similar Layers).

ფერთა დიაპაზონი (Color Range) - წარმოადგენს **ჯადოსნური ჯოხის** (Magic Wand) ინტელექტუალურ ვერსიას. საშუალებასიძლევა ფერისა და მისი ტონალობის საფუძველზე შექმნას მონიშვნა. ეს იმდენად მძლავრი ფუნქციაა, რომ ცალკე განხილვის იმსახურებს.

საზღვრის სრულყოფა (Refine Edge) - გამოიყენება მონიშვნის საზღვრის რედაქტირებისთვის. მისი დახმარებით შესაძლებელია ჩვენი მონიშნული არე უფრო ზუსტი გავხადოთ, რაც შემდგომ ეტაპზე მუშაობას გაცილებით გაგვიადვილებს.

საზღვრების ცვლილება (Modify), რომლის ქვემენიუში 5 ბრძანებაა მოთავსებული:**საზღვარი** (Border), **გასწორება** (Smooth), **გაფართოება** (Expand), **შეკუმშვა** (Contract), **დარბილება** (Feather).

ა. **საზღვარი** (Border) - აქ მითითებული მნიშვნელობის შემდეგ, პროგრამა ხელახლა გადახატავს მონიშნულ არეს ორმაგი საზღვრით, რომლის სიგანე იქნება თქვენს მიერ მითითებული მნიშვნელობის ტოლი. დიაპაზონი 1-დან 200 პიქსელამდე. მცირე მნიშვნელობების მითითება სასარგებლოა, როდესაც თქვენ გინდათ აკურატულად გააბუნდოვნოთ, ჩაამუქოთ ან სხვანაირად დაამუშაოთ მონიშნული ობიექტის საზღვრები.

ბ. **გასწორება** (Smooth) - გამოიყენება იმ შემთხვევაში, როდესაც თქვენს მიერ შექმნილ მონიშნულ არეს გააჩნია ბევრი კუთხოვანი ადგილები. ამ ბრძანებით თქვენ ახდენთ კუთხოვანი ადგილების გასწორებას. მისათითებელი დიაპაზონი მერყეობს 1-დან 100 პიქსელამდე.

გ. **გაფართოება** (Expand) - მონიშნულ არეს ზრდის მითითებული მნიშვნელობით. მისათითებელი დიაპაზონი მერყეობს 1-დან 100 პიქსელამდე.

დ. **შეკუმშვა** (Contract) - მონიშნულ არეს ამცირებს მითითებული მნიშვნელობით. მისათითებელი დიაპაზონი მერყეობს 1-დან 100 პიქსელამდე.

ე. **დარბილება** (Feather) - არბილებს მონიშნული არის საზღვრებს. მისათითებელი დიაპაზონი მერყეობს 0.2-დან 250 პიქსელამდე. (იხ. სურ. 8-30)

ბრძანებები **მომიჯნავე** (Grow) და **მსგავსი** (Similar), მონიშვნას ქმნიან წინასწარ განსაზღვრული ფერის ეტალონის საფუძველზე. ორივე ბრძანების მოქმედებას განსაზღვრავს**ჯადოსნური ჯოხის**(Magic Wand) ინსტრუმენტის**შეღწევადობის** (Tolerance) სიდიდე (იხ. გვ.309). გამოსახულებაზე უნდა შევქმნათ მონიშვნა იმ ფერებზე, რომლის გამოყოფაც გვინდა და ამოვირჩიოთ ორი ბრძანებიდან ერთ-ერთი:

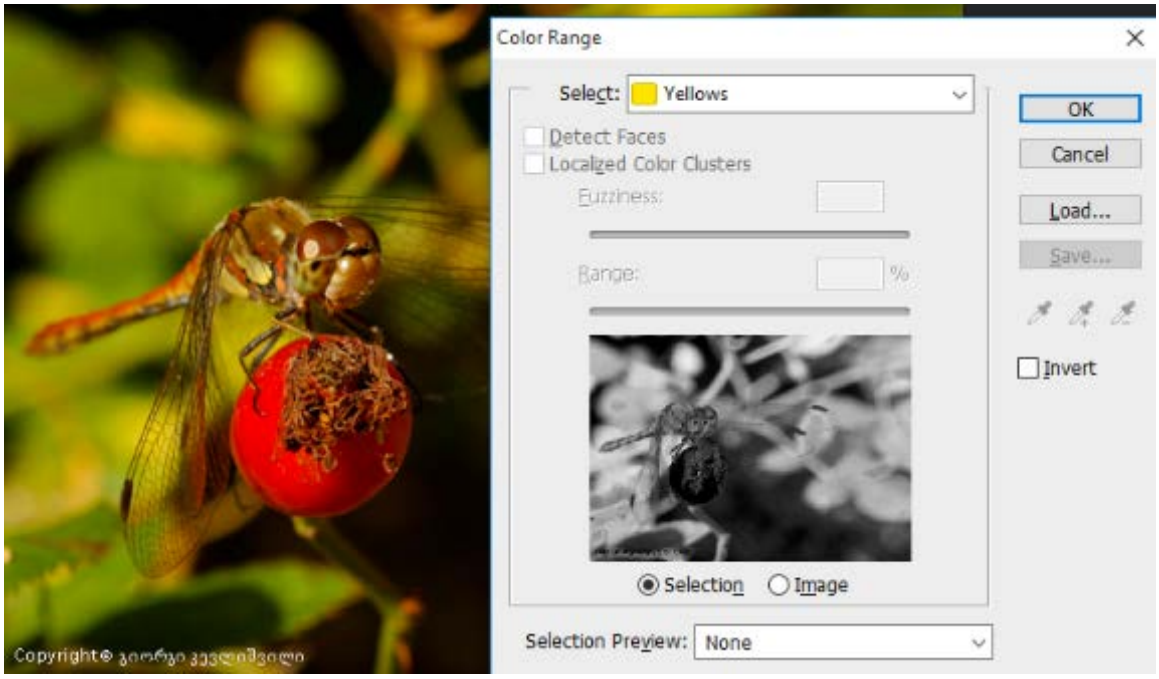
მომიჯნავე(Grow) მონიშნავს მხოლოდ იმ პიქსელებს, რომლებიც საწყის მონიშნულ არეს ემიჯნებიან.

მსგავსი (Similar)ახდენს მთლიანი გამოსახულების ანალიზს და ყველა იმ პიქსელს მონიშნავს, რომელიც **შეღწევადობაში** (Tolerance) მითითებული პარამეტრის კრიტერიუმებს აკმაყოფილებს.

ცალკე ყურადღებას იმსახურებს ორი მნიშვნელოვანი ფუნქცია: მონიშვნა **ფერის დიაპაზონით** (Color Range) და მონიშნული არის **საზღვრის სრულყოფა** (Refine Edge).

ფერის დიაპაზონი (Color Range)

ეს ფუნქცია წარმოადგენს **ჯადოსნური ჯოხის** (Magic Wand) „ინტელექტუალურ“ ვერსიას, რადგან ის ახდენს მონიშვნას ფერისა და ფერთა ტონალობის მიხედვით, გაძლევთ წინასწარ ხედს და შესაბამისად მეტი კონტროლის საშუალებას. **ჯადოსნური ჯოხი**(Magic Wand) ან ნიშნავს პიქსელს ან არა. **ფერთა დიაპაზონი**(Color Range) კი საშუალებას იძლევა პიქსელი ნაწილობრივ მონიშნოთ.

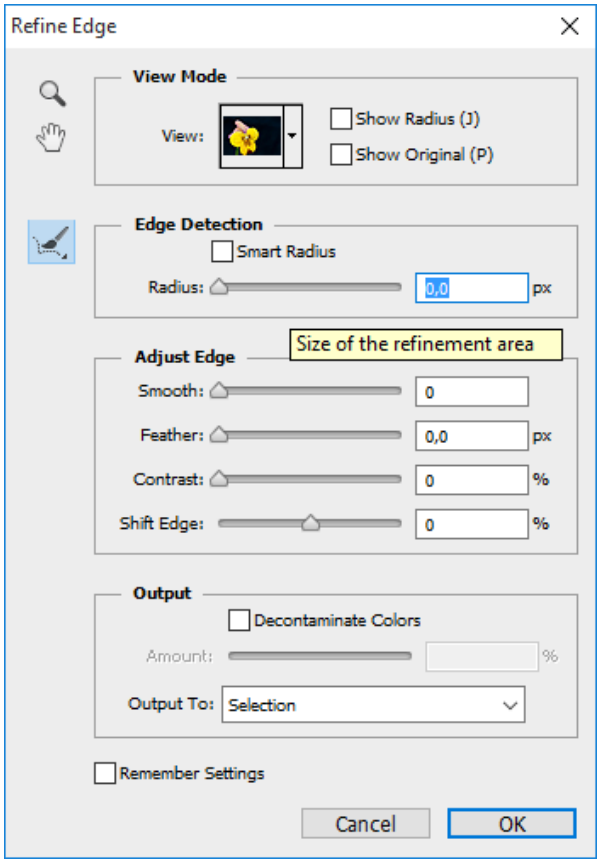


სურ. 8-35. ფერთა დიაპაზონის (Color Range) დიალოგური ფანჯარა. ამ შემთხვევაში მითითებულია, რომ გამოსახულებაზე მოინიშნოს ყვითელი ფერი.ფოტო: გიორგი კევლიშვილი

დიალოგური ფანჯრის დახმარებით შესაძლებელია გამოსახულებაზე რომელიმე კონკრეტული ფერის მონიშვნა ან ფერის მონიშვნა მოხდეს ჩვენს მიერ მითითებული ნიმუშის მიხედვით. ნიმუშის მითითებისას ვიყენებთ პიპეტის ინსტრუმენტს, რომელის დახმარებითაც ჩვენ შეგვიძლია შევარჩიოთ, დავამატოთ ან გამოვაკლოთ საჭირო ფერი ან მისი ტონალობა.

საზღვრის სრულყოფა (Refine Edge)

ზემოთ განხილული მონიშვნის ყველა ინსტრუმენტს ფორმატირების პანელზე და მონიშვნის (Select) მენიუში**საზღვრებისსრულყოფის** (Refine Edge) ფუნქცია.



სურ. 8-36. საზღვრის დაზუსტების (Refine Edge)დიალოგური ფანჯარა.

მისი დახმარებით ჩვენ შეგვიძლია მონიშნული არე მაქსიმალურად დახვეწილი და სამუშაოდ გამოყენებადი გავხადოთ. ერთ ფანჯარაში არის მოთავსებული ისეთი ინსტრუმენტები, როგორებიცაა **გასწორება** (Smooth), **შერბილება** (Feather), **სიმკვეთრე** (Contrast) და სხვა. ყველაფერი ეს საშუალებას გაძლევთ თქვენი მონიშნული არე მაქსიმალურად მორგებული გახადოთ სამუშაოდ.

მონიშვნის ინსტრუმენტების და ბრძანებების კარგად ფლობა ძალიან მნიშვნელოვანია შემდეგი ეტაპისთვის, იქნება ეს ფერთა კორექცია თუ ობიექტის განცალკევება ფონისგან, მისი წაშლა თუ სხვა ფენაზე ან ფაილში გადატანა. ზემოთ ჩამოთვლილი ინსტრუმენტების დახმარებით თქვენ შეგიძლიათ შექმნათ მონიშნული არე, დააზუსტოთ მისი საზღვრები და გადახვიდეთ გამოსახულებასთან მუშაობის შემდეგ ეტაპზე:

მოქმედები მონიშნულ არეზე

როდესაც გამოსახულებაზე გვაქვს მონიშნული არე, მასთან მიმართებით შეგვიძლია სხვადასხვა მოქმედებები შევასრულოთ:

გადატანა -**გადაადგილების** (Move) ინსტრუმენტის დახმარებით შეგვიძლია გამოსახულებაზე მონიშნულფრაგმენტს ადგილმდებარეობა შეუცვალოთ;

კოპირება - Edit -> Copy (Ctrl + C) მონიშნული ფრაგმენტის ასლის შექმნა;

ამოჭრა -> Edit -> Cut (Ctrl + X) მონიშნული ფრაგმენტის ამოჭრა;

ჩასმა -> Edit -> Paste (Crl + V) კოპირებული ან ამოჭრილი ფრაგმენტის ჩასმა (სხვა ფენაზე, სხვა ფაილში).

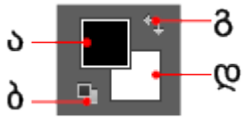
დაფერვის მეთოდები

გამოსახულებასთან მუშაობა რასაკვირველია არ შემოიფარგლება მხოლოდ მისი მონიშვნით და სტანდარტული მოქმედებების შესრულებით (ამოჭრა, კოპირება და სხვა). ჩვენ ასევე შეგვიძლია მოვახდინოთ გამოსახულების დაფერვა. ილუსტრატორები საერთოდ სუფთა ტილოდან იწყებენ მუშაობას და საოცარ შედეგებს აღწევენ. ბუნებრივია აქ მხოლოდ პროგრამის კარგად ცოდნა არ არის საკმარისი. გარდა ამისა, დაფერვა შეგვიძლია გამოვიყენოთ როგორც მთლიანად გამოსახულებასთან, ისე მონიშნულ არესთან.

პროგრამაში დაფერვის უამრავი ხერხი არსებობს. სანამ ამ მეთოდებს განვიხილავთ, მინდა დავიწყო ჯერ ფერების განსაზღვრით.

ფერების განსაზღვრა პროგრამაში

ფოტოშოპში მუშაობისას გამოიყენება ორი ფერი: მთავარი ანუ წინა ფონის ფერი (Foreground Color) და უკანა ფონის (Background Color) ფერი. ინსტრუმენტების პანელის ქვედა ნაწილში მდებარეობს ფერის მართვის ელემენტები (ორი ერთმანეთზე დადებული კვადრატი):

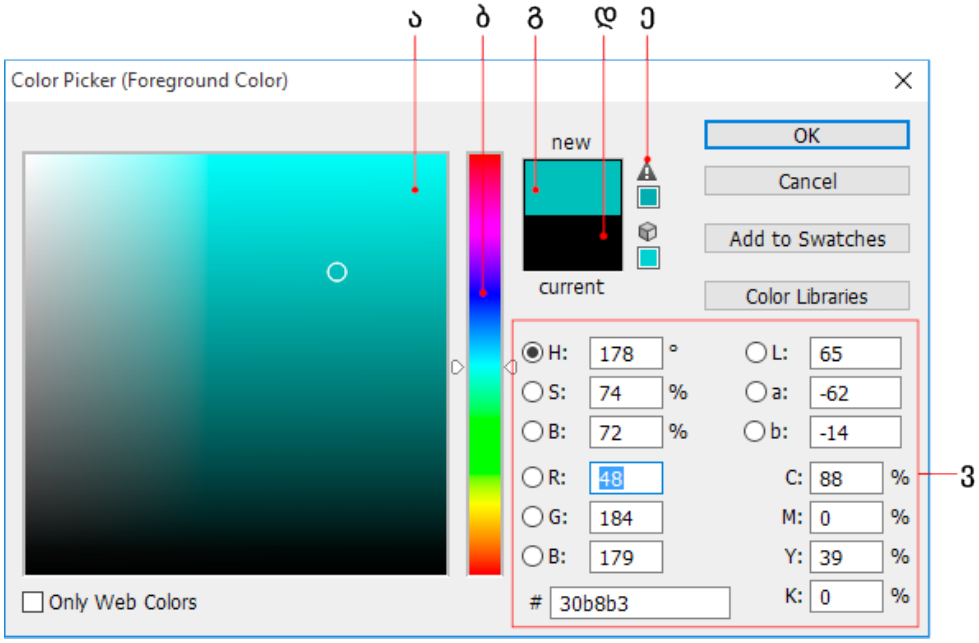


სურ. 8-37. ა - წინა ფონის ფერი (Foreground Color); ბ - საწყის მდგომარეობაში დაბრუნება (წინა ფონი შავი, უკანა - თეთრი); გ - წინა და უკანა ფონების ადგილების გადანაცვლება; დ - უკანა ფონის ფერი (Background Color).

რას ნიშნავს წინა და უკანა ფერები?

ინსტრუმენტები, როგორებიცაა ფუნჯი, ფანქარი, ფერის ჩასხმა და სხვა მსგავსი ინსტრუმენტები იყენებენ წინა ფონის ფერს. როდესაც ხდება სურათზე მონაკვეთის წაშლა, ამოჭრა და სხვა მსგავსი მოქმედება, ამ მონაკვეთების ჩანაცვლება ხდება უკანა ფონის ფერით.

რომელიმე კვადრატზე დაჭერისას იხსნება ფერების პალიტრა, საიდანაც ვირჩევთ სამუშაო ფერს.

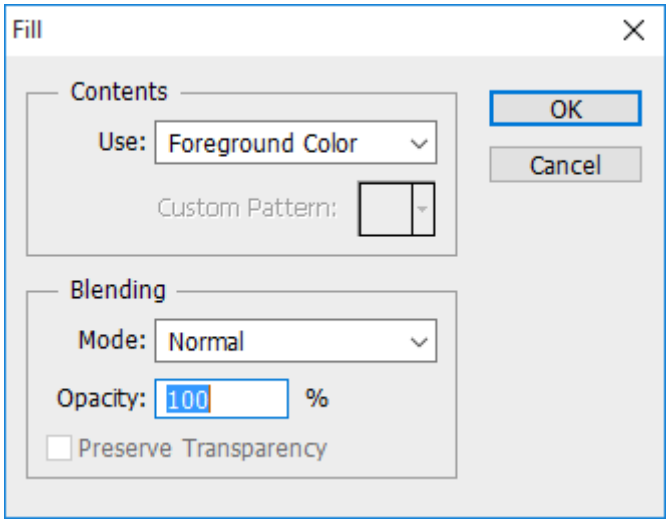


სურ. 8-38. ა - ფერთა ველი, რომელზეც პატარა წრით მონიშნულია შერჩეული ფერი; ბ - ფერთა შკალა, სადაც ვირჩევთ ფერს მოცემული სპექტრიდან; გ - ახალი შერჩეული ფერი; დ - მიმდინარე ფერი; ე - ნიშანი მიუთითებს, რომ ეს ფერი ვერ თავსდება CMYK ფერთა მოდელის სპექტრში; ვ - ფერთა მნიშვნელობები (ფერის შერჩევისას აისახება მისი რიცხობრივი მნიშვნელობები HSB, Lab, RGB, CMYK ფერთა მოდელებისთვის. ასევე მისი მნიშვნელობა თექვსმეტობით ფორმატში).

შევსება (Fill)

შევსება შეიძლება გამოყენებული იყოს ცარიელი ადგილების შესავსებად ან გამოსახულებიდან არასასურველი ელემენტების მოსაცილებლად.

ამისათვის Fill ბრძანება უნდაგამოიძახოთ: Edit -> Fill (Shift + F5).



სურ. 8-39. დიალოგური ფანჯარა Fill

გამოყენების (Use) ჩამოსაშლელი მენიუდან ირჩევთ შევსების მეთოდს:

Foreground Color, Background Color, Color... - ვიყენებთ მაშინ, როდესაც არე გვინდა ფერით შევავსოთ;

Content Aware, Patern, History - ვიყენებთ მაშინ, როდესაც გვინდა არე გამოსახულებით ან რაიმე ტექსტურით შევავსოთ. ცალკე ყურადღებას იმსახურებს Content Aware.

ვიდეო 0-1. Content Aware-ის გამოყენება Photoshop CS6-ში

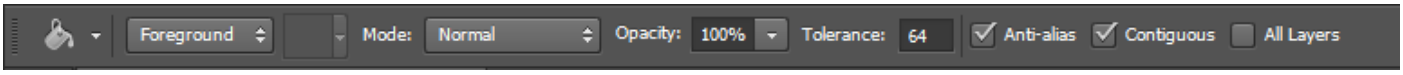
შენიშვნა: *Content Aware აქტიურია მხოლოდ მაშინ, როდესაც გამოსახულებაზე რამე ფრაგმენტია მონიშნული.*

Black, 50% Gray, White - სტანდარტული ფერებით შევსება.

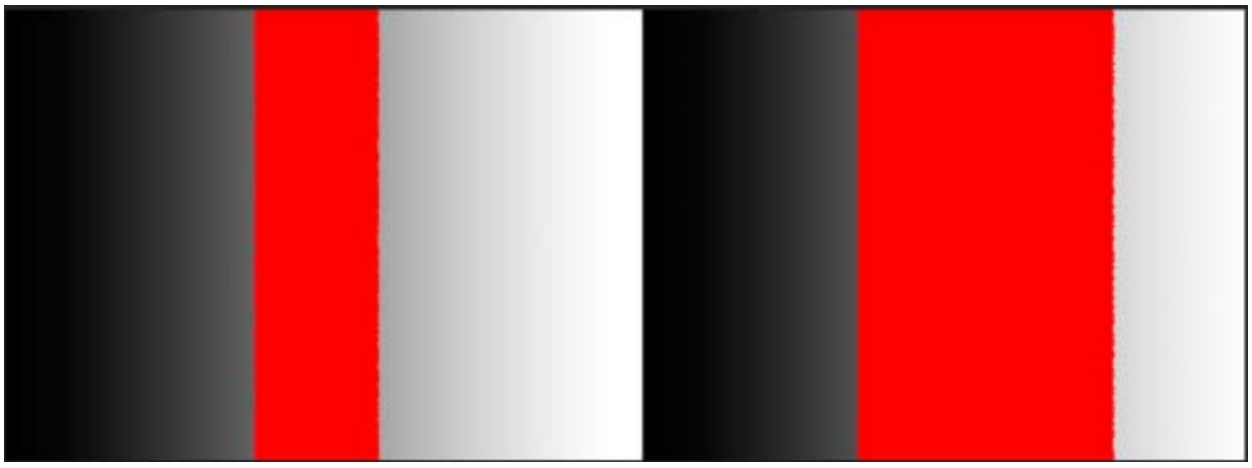
შერევისა (Blending) და **გამჭვირვალობის** (Opacity) დახმარებით შეგიძლიათ მართოთ **შევსების** პროცესი.

ფერის ჩასხმის ინსტრუმენტი (Paint Backet Tool)

მოქმედების პრინციპით ეს ინსტრუმენტი ძალიან გავს **ჯადოსნურ ჯოხს** (Magic Wand).



ფორმატირების ზოლზე თქვენ შეხვდებით ისეთივე ფუნქციებს როგორიცაა **შეღწევადობა** (Tolerance), **შერბილება** (Anti-alias), **მომიჯნავე პიქსელები** (Contiguous) და **ყველა ფენა** (All Layers). დანარჩენი ფუნქციები კი დამახასიათებელია ამ ინსტრუმენტისთვის: შეგიძლიათ აირჩიოთ ინსტრუმენტმა **ფერი** გამოიყენოს თუ **ტექსტურა** (ჩამოსაშლელი მენიუ - Foreground / Patern), აირჩიოთ **შერევის** (Mode) მეთოდები და **გამჭვირვალობა** (Opacity).



სურ. 8-40. სურათზე ნაჩვენებია ინსტრუმენტის მოქმედება შეღწევადობის (Tolerance) სხვადასხვა მაჩვენებლებით. მარცხენა სურათზე მაჩვენებელი მითითებულია 32, ხოლო მარჯვენაზე 64.

გრადიენტის ინსტრუმენტი (Gradient Tool)

გრადიენტის შექმნაში მონაწილეობს მინიმუმ ორი ფერი. ამ დროს პირველი ფერი ნელ-ნელა გადადის მეორეში.

გრადიენტის ინსტრუმენტის არჩევისას ფორმატირების პანელზე მისთვის დამახასიათებელი ფუნქციები ჩნდება.

აქ შეგიძლიათ აირჩიოთ გრადიენტის შეფერილობა და მისი ნაირსახეობები, დააყენოთ შერევის რეჟიმი, გამჭვირვალობა და სხვა.

გრადიენტი არის ხუთი სახის:

სურ. 8-41. მარცხნიდან მარჯვნივ: ხაზოვანი (Linear), წრიული (Radial), კუთხოვანი (Angular), არეკლილი (Reflected) და რომბისებრი (Dimond).

პროგრამაში მოცემული გამზადებული მაგალითების გარდა თქვენ თვითონაც შეგიძლიათ შექმნათ სასურველი მიმართულების გრადიენტი ფერების სასურველი რაოდენობით.

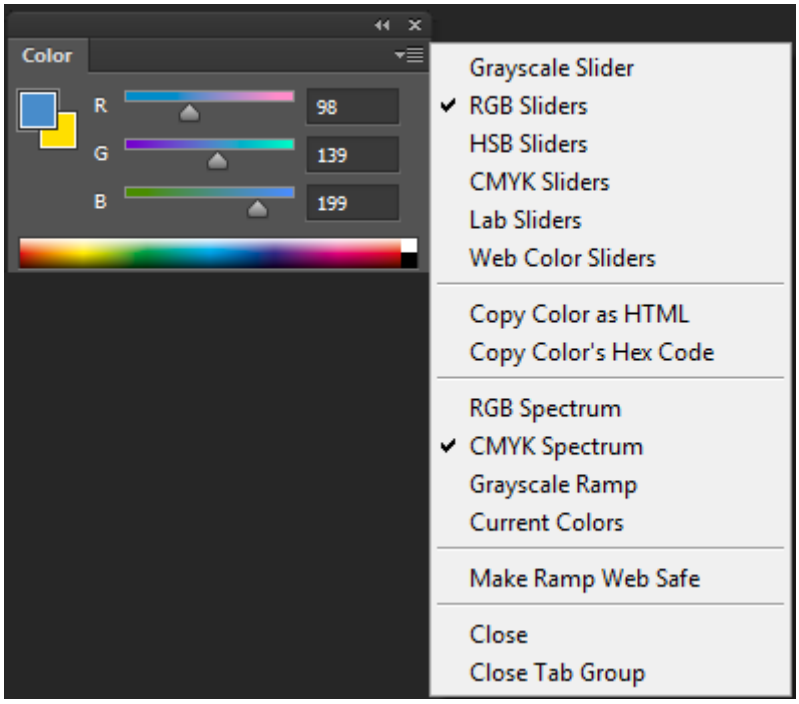
სურ. 8-42. გრადიენტის გამოყენების ერთ-ერთი მაგალითი

ფერთა ნიმუშები (Swatches)

მისი გამოძახება ხდება Windows -> Swatches. აქ მოცემული ძირითადი სამუშაო ფერები. გარდა ამისა მუშაობის პროცესში თქვენ შესაძლოა შექმნათ საკუთარი ფერების პალიტრა. რომ არ იზეპიროთ მათი რიცხვითი მნიშვნელობები, შეგიძლიათ შეინახოთ ნიმუშად და შემდეგში გამოიყენოთ.

ფერი დასამატებლად რამდენიმე გზა არსებობს. ყველაზე მარტივია გამოიძახოთ ფერთა პალიტრა (იხ. სურ. 8-39) და ფერის შერჩევის შემდეგ დააწვეთ ღილაკს **ნიმუშებში დამატება** (Add to Swathes).

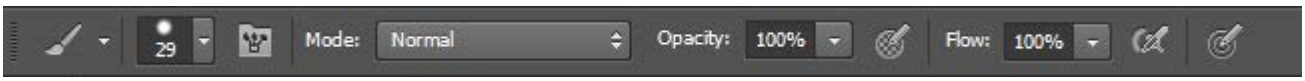
როდესაც **ფერთა ნიმუშები პანელიდან** აირჩევთ ფერს ის ჯდება როგორც წინა ფონის ფერი (Foreground Color). გარდა ამისა ის ასევე გამოისახება **ფერების** (Color) პანელზე. ამ პანელის (ისევე როგორც ფერთა პალიტრის) დანიშნულებაა შექმნათ ახალი ფერი. პანელის დამატებით მენიუში შეგიძლიათ აირჩიოთ, რომელი ფერთა მოდელისთვის ქმნით ფერს და შემდეგ მცოცავების გადაადგილებით ან გრაფებში მნიშვნელობების მითითებით, ადგენთ ახალ ფერს.



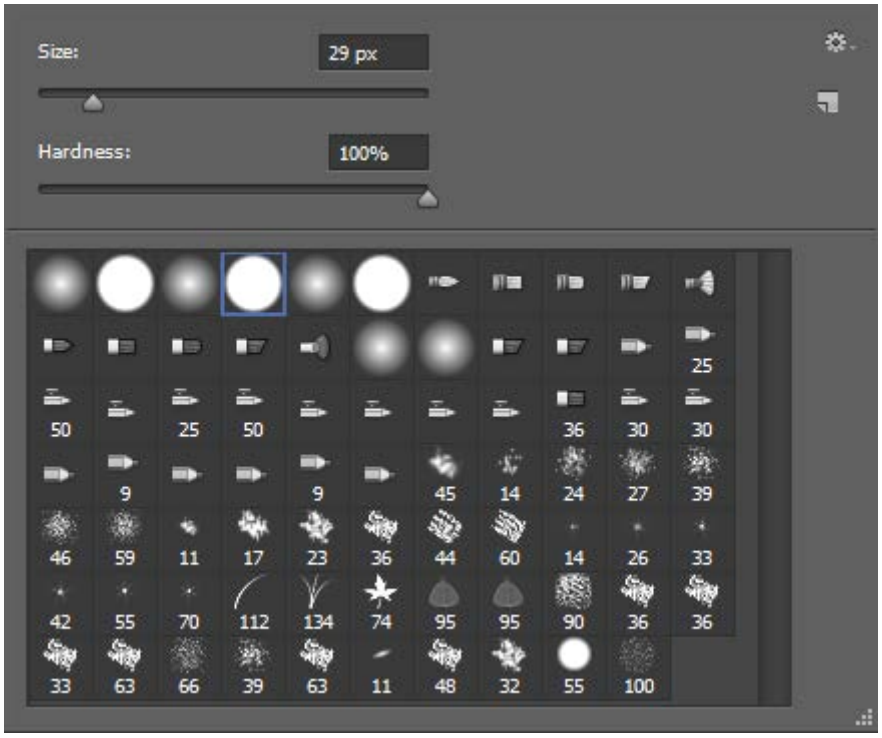
სურ. 8-43. პანელი ***ფერი*** *(Color)და მისი დამატებითი მენიუ. ფერის შექმნა შეგიძლიათ როგორც წინა ფონისთვის (Foreground), ისე უკან ფონისთვის (Background)*

ფუნჯი (Brush)

ფუნჯი საკმაოდ ხშირად გამოყენებადი ინსტრუმენტია. შეიძლება ითქვას უნივერსალურიც კი: შეგიძლიათ უცვალოთ ფორმა, ფერი, ზომა, შექმნათ საკუთარი ფუნჯი. გარდა ამისა ის შეიძლება გამოიყენოთ ობიექტისთვის ნიღბისა (სწრაფი ნიღბისრეჟიმში) და კონტურის შესაქმნელად.



ფორმატირების ზოლიდან თქვენ შეგიძლიათ დააყენოთ მისი ზომა, აირჩიოთ ფუნჯის ტიპი, დააყენოთ ფერის შერევის რეჟიმები, გამჭვირვალობა და სხვა.



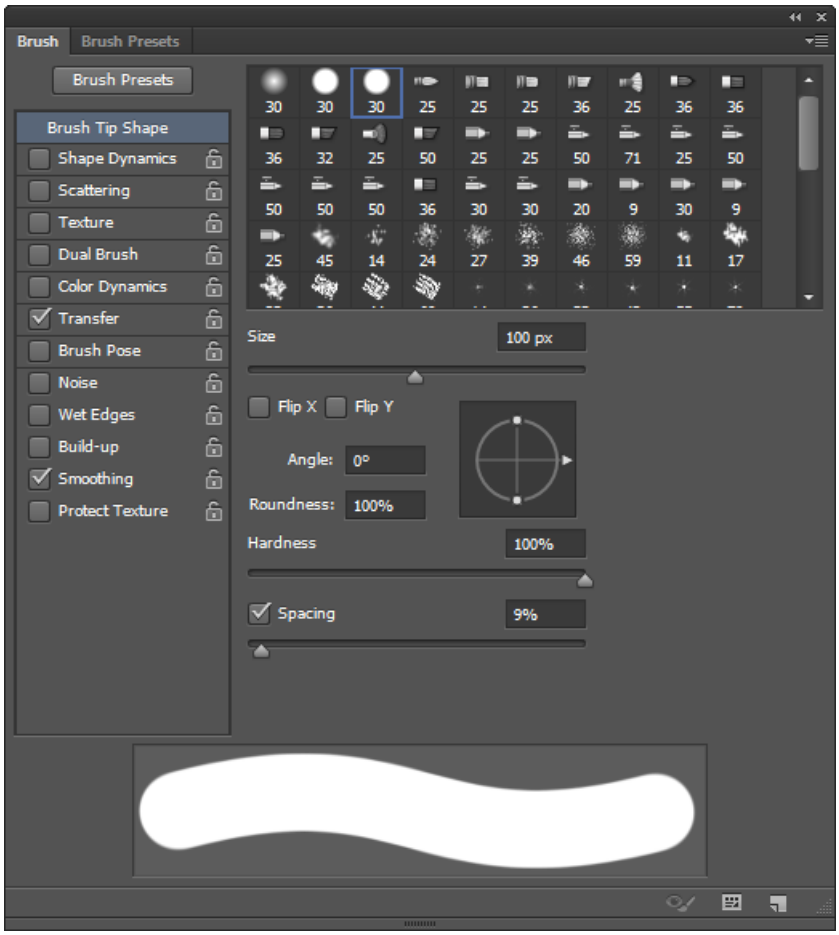
სურ. 8-44. აქ შეგიძლიათ აირჩიოთ ფუნჯის ტიპი. ასევე დააყენოთ მისი ზომა და სიმკვეთრე

შერჩეული ფუნჯისთვის შეგიძლიათ ზომა (px) და კიდეების სიმკვეთრე/სიმსუბუქე განსაზღვროთ: 100% - მკვეთრი, 0% - მსუბუქი.

სურ. 8-45. ერთნაირი დიამეტრი, მაგრამ სხვადასხვა სიმკვეთრის კიდეებით. პირველი ხაზი - 100%, მეორე ხაზი - 50%, მესამე ხაზი - 0%.

შენიშვნა: *ზოგი ფუნჯი ეფექტურია გამოიყენოთ სახატავი დაფითა და ფუნჯით სარგებლობისას.*

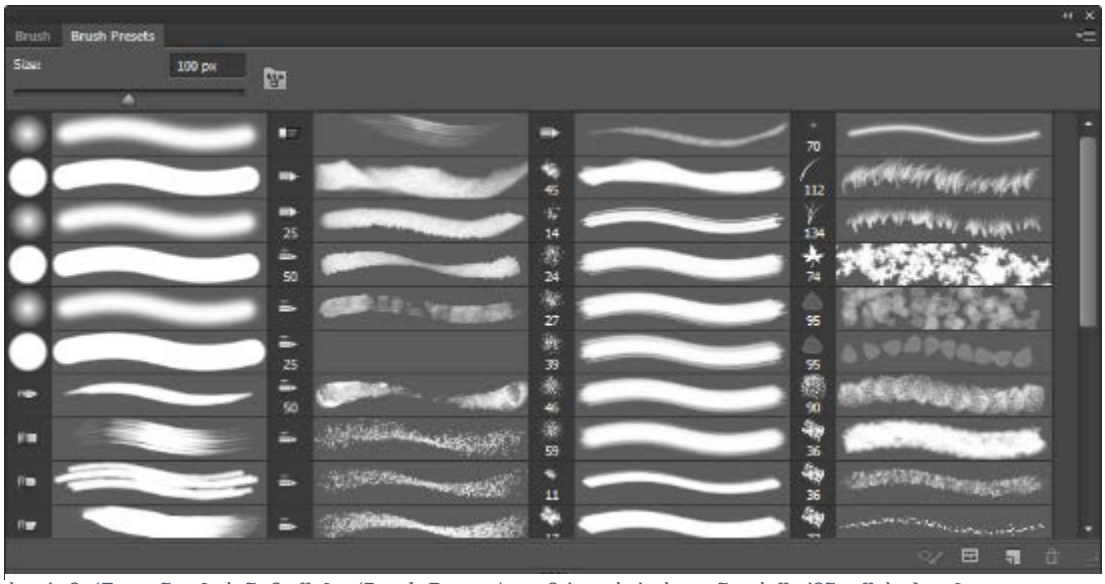
ძირითადი მახასიათებლების გარდა ასევე შეგიძლიათ ფუნჯის სხვა პარამეტრებიც ცვალოთ. ამისათვის გამოიძახეთ Window -> Brushes (F5).



სურ. 8-46. ფანჯარაში Brush შეგიძლიათ როგორც ფუნჯის არჩევა, ისე მისი მახასიათებლების დაყენება: სიმკვეთრე, დახრის კუთხე, ინტენსივობა და სხვა.

არჩეული ფუნჯისთვის უამრავი მახასიათებლის დაყენება შეგიძლიათ, ხოლო ფანჯრის ქვემოთ მოცემულია წინასწარი ხედის ფანჯარა, სადაც დაინახავთ, როგორ იცვლება ფუნჯი. სამომავლოდ ახალი ფუნჯის სახით შეგიძლიათ შეინახოთ.

ამავე ფანჯარაზე არის ჩანართი Brush Presets სადაც მოცემულია ფუნჯის მზა მაგალითები. ამავე ფანჯრის გამოძახება შეგიძლიათ Windows -> Brush Presets



სურ. 8-47. ფუნჯების ნიმუშები (Brush Presets). უამრავი სახის ფუნჯის შექმნაა შესაძლებელი.

გარდა არსებული ფუნჯებისა, რაც პროგრამას ახლავს თან, თქვენ შეგიძლიათ ჩატვირთოთ სხვა ფუნჯებიც, რომლებიც ინტერნეტში სხვადასხვა გრაფიკულ რესურსებზე მოიპოვება. ამისათვის ფორმატირების ზოლზე, ფუნჯების (Brushes) ნაირსახეობების ფანჯარაში (სურ. 8-45), დამატებით მენიუში აირჩიოთ ფუნჯების ჩატვირთვა (Load Brushes…) და გამოსულ ფანჯარაში მიუთითეთ ფუნჯის ადგილმდებარეობა (ფუნჯის ფაილის ფორმატია *.ABR).

თუ დაგიგროვდათ ძალიან ბევრი ფუნჯი და უკვე რთულია მათ შორის ორიენტირება, იმავე მენიუში მიუთითეთ ფუნჯების ჩამოყრა(Reset Brushes…) და ფუნჯების რაოდენობა და ტიპები დაუბრუნდება საწყის მდგომარეობას.

თქვენს მიერ შექმნილი ფუნჯების ნაკრები თუ გინდათ, რომ გაუგზავნოთ მეგობარს ან გადაიტანოთ სხვა კომპიუტერში, მენიუში აირჩიეთ ფუნჯების შენახვა (Save Brushes…). მიუთითეთ მისამართი, ფაილის დასახელება და მიღებული ფაილი (რომელშიც მოთავსებული ფუნჯები), შეგიძლიათ გაუზიაროთ სხვას.

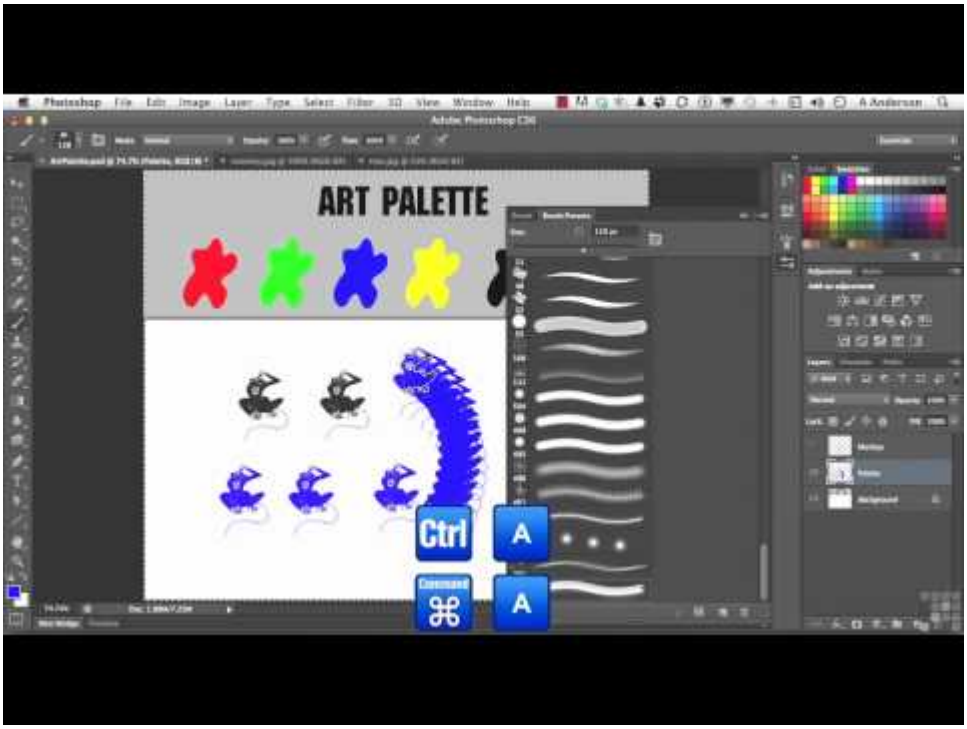
ფუნჯის შექმნა ნებისმიერი გამოსახულებიდან შეიძლება, თუ ის აკმაყოფილებს ორ პირობას: გამოსახულების ზომა არ უნდა აღემატებოდეს 2500 X 2500 პიქსელს და უნდა ჰქონდეს გამჭვირვალე ან თეთრი ფონი.

ფუნჯისათვის გამოსახულების მომზადების შემდეგ, Edit მენიუდან უნდა გავააქტიუროთ Define Brush Preset ბრძანება. ეკრანზე გამოვა სადიალოგო ფანჯარა Brush Name, რომელშიც იწერება შექმნილი ფუნჯის სახელი და ვადასტურებთ OK საბრძანებო ღილაკით.

ჩვენს მიერ შექმნილი ფუნჯი დაიკავებს ადგილს ფუნჯთა მენიუში, ფუნჯთა ნაკრების ბოლოს.

ფუნჯის მსგავსად სხვა ინსტრუმენტებიც იყენებენ ისეთ პარამეტრებს, როგორებიცაა ფუნჯის ზომა, სიმკვეთრე, დახრილობა და სხვა. ესენია: რეტუშირებისა და ხატვის ინსტრუმენტები (სურ. 8-10).

ვიდეო 0-2. ფუნჯების გამოყენება Photoshop CS6-ში



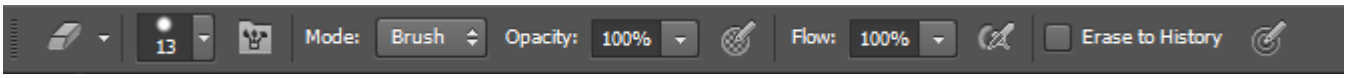
ვიდეო 0-3. ფუნჯის შექმნა პროგრამა Photoshop CS6-ში

საშლელი (Eraser Tools)

გამოსახულების რედაქტირებისას დაფერვის მსგავსად ჩვენ დაგვჭირდება რაიმე ფრაგმენტის წაშლა. ინსტრუმენტთა პანელზე არის სამი სახის საშლელი: ჩვეულებრივი საშლელი (Eraser Tool), ფონის საშლელი (Background Eraser Tool) და ჯადოსნური საშლელი (Magic Eraser Tool).

სამივე ინსტრუმენტს ფორმატირების ზოლზე თავისთვის დამახასიათებელი ფუნქციები გააჩნია:

ჩვეულებრივი საშლელი (Eraser Tool)



ფორმატირების ზოლიდან თქვენ შეგიძლიათ სხვადასხვა მახასიათებლების დაყენება: ზომა, საშლელის ტიპი, გამჭვირვალობა, დაჭერის ინტენსივობა და სხვა. ისევე, როგორც ფუნჯის შემთხვევაში, შეგიძლიათ დააყენოთ ფუნჯის ზომა და კიდეების სიმკვეთრე.

საშლელის მოქმედება დამოკიდებულია თუ რომელ ფენაზე (Layer – სამუშაო ფენა; Background - ფონის ფენა) ხდება მისი გამოყენება. გამოსახულების უკანა ფონზე (Background) მუშაობისას, წაშლილ პიქსელებს ჩაანაცვლებს უკანა ფონის ფერის (Background Color) პიქსელებით. თუ ფონის ფენას სამუშაო ფენად (Layer) გადავაქცევთ (Layer + New + Layer From Background), საშლელი დაიწყებს პიქსელების წაშლას სრულ გამჭვირვალობამდე (დარჩება ჭადრაკისებრი ფონი). აღნიშნულ ინსტრუმენტს ფერის შეგრძნება არ გააჩნია და შლის ყველაფერს, რაც კურსორის ქვეშ მოხვდება.

საშლელი სამ რეჟიმში მუშაობს: Brush, Pencil და Block, რომელთა არჩევა ფორმატირებისზოლზე Mode ჩამოსაშლელი მენიუდან შეიძლება. Brush ან Pencil რეჟიმების არჩევის შემთხვევაში, საშლელისთვის ხელმისაწვდომი ხდება აღნიშნული ინსტრუმენტის პარამერტები. Block რეჟიმში ხელსაწყო კვადრატის ფორმას ღებულობს, მისი ზომა გამოსახულების ზომაზე არის დამოკიდებული და პარამეტრების პანელზე ყველა პარამეტრი დაბლოკილია.

ფონის საშლელი (Background Eraser)

ფონის საშლელის პარამეტრების პანელი ძალიან ჰგავს ფერის ჩანაცვლების პარამეტრების პანელს და მოხმარების წესიც ანალოგიურია. განსხვავებული მხოლოდ შედეგია: ფერის ჩანაცვლების ინსტრუმენტი ფერს უცვლის პიქსელებს, ხოლო ფონის საშლელი შლის პიქსელებს.

ფუნქცია **წინა ფონის ფერის დაცვა**(Protect Foreground Color) ჩამრთველი იცავს წაშლისაგან გამოსახულების იმ ფერს, რომელიც წინა ფონის ფრად (Foreground Color) არის დაფიქსირებული.

სურ. 8-48. ფონის საშლელის (Background Eraser Tool) გამოყენების

ჯადოსნური საშლელი (Magic Eraser Tool)

ჯადოსნური საშლელი გამოიყენება გამოსახულებაზე ერთფეროვანი არეების მარტივად და სწრაფად წასაშლელად. მისი მოქმედების პრინციპი ისეთივეა, როგორიც ჯადოსნურ ჯოხს (Magic Wand) გააჩნია. და ფორმატირების ზოლზე მსგავსი ფუნქციები აქვს. თუმცა მონიშნული არის შექმნის ნაცვლად, საშლელი შლის გამოსახულების ფრაგმენტებს.

სურ. 8-49. ჯადოსნური საშლელის (Magic Eraser) გამოყენების მაგალითი.

მიმდინარე პარაგრაფის თემატიკა

- ტრანსფორმაციის ინსტრუმენტები და ბრძანებები
- გამოსახულების ზომის რედაქტირება - გამოსახულებისა და ტილოს ზომა, შემოჭრის ინსტრუმენტი
- ფენების შექმნა და მათი მართვა. ფენების ნაირსახეობები
- ფენების სტილების მიმოხილვა

ტრანსფორმაციის ინსტრუმენტები

მუშაობის პროცესში აუცილებლად დაგჭირდებათ ისეთი მოქმედებების შესრულება, როგორიცაა ობიექტის შემოტრიალება, ზომებისა თუ ფორმის შეცვლა. ამისათვის გამოიყენება ტრანსფორმაციის ფუნქცია.

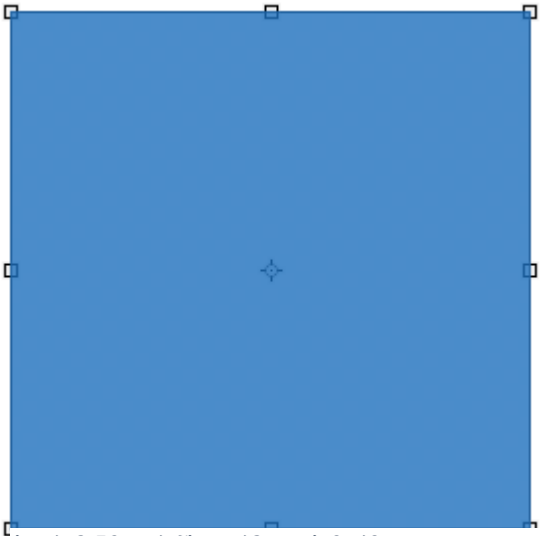
თავისუფალი ტრანსფორმაცია

თუ ობიექტი ძირითად ფონზეა განთავსებული (Background) ის უნდა მონიშნოთ (იხ. მონიშვნის მეთოდები, გვ.304). ასევე ტრანსფორმაცია შეგიძლიათ გაუკეთოთ ცალკე ფენაზე მოთავსებულ ობიექტს, ობიექტთა ჯგუფს და ა.შ.

თავისუფალი ტრანსფორმაციის გამოძახება ხდება Edit -> Free Transform (Ctrl +T).

ფორმატირების ზოლზე რიცხობრივი მნიშვნელობების მითითებით შეგიძლიათ ცვალოთ ობიექტის მდებარეობა, მისი ზომები, მასშტაბურობა, დახრილობის კუთხე.

მონიშნული ობიექტი (რა ფორმისაც არ უნდა იყოს) თავსდება მართკუთხედში ე.წ. **ტრანსფორმაციის ჩარჩოში**, რომელსაც ნაპირებზე და კუთხეებში გააჩნია წერტილები. მათი საშუალებით ხდება ობიექტზე ზემოქმედება. სულ ცხრა ასეთი წერტილია. ცენტრში (ნაგულისხმევად) არის კიდევ ერთი წერტილი. მას **საკონტროლო წერტილს** უწოდებენ. ის განსაზღვრავს თუ რის მიმართ უნდა მოხდეს ტრანსფორმაცია, მაგალითად ობიექტის დახრა, სარკისებურად შებრუნება ან შემოტრიალება. სხვა წერტილებისგან განსხვავებით მისი გადაადგილება თავისუფლად შეგიძლიათ როგორც ჩარჩოს შიგნით, ისე მის გარეთ.



სურ. 8-50. ტრანსფორმაციის ჩარჩო.

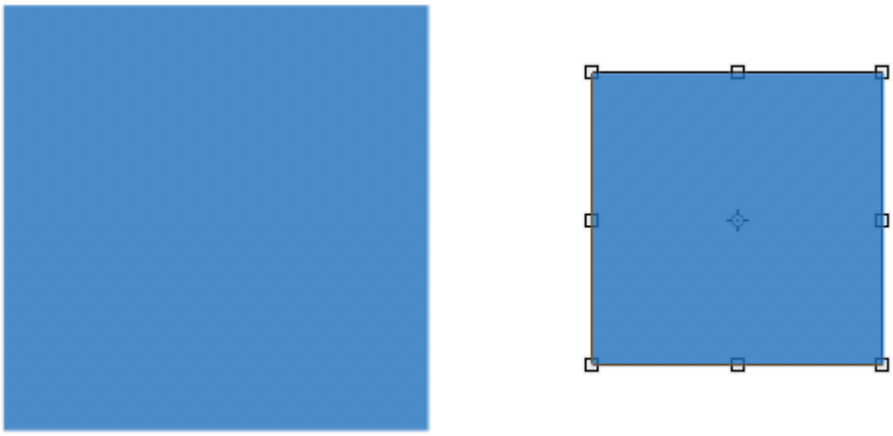
ობიექტზე ზემოქმედებისთვის კურსორი უნდა მიიტანოთ ერთ-ერთ წერტილთან ან კიდესთან. ის შეიცვლის ფორმას და ეს ფორმა მიგითითებთ მოქმედებაზე (ზომის, ფორმის შეცვლა ან შემოტრიალება).

თუ გამოიყენებთ Ctrl კლავიშს, მაშინ წერტილზე მოქმედებთ ინდივიდუალურად. Shift კლავიშის დახმარებით კი, ხდება ობიექტის ზომის ცვლილება პროპორციულად.

გარდა **თავისუფალი ტრანსფორმაციისა**, პროგრამაში ტრანსფორმაციის შესაძლებლობები ცალ-ცალკეამოცემული. ისინი შეგიძლიათ იხილოთ Edit -> Transform მენიუში; ან, როდესაც ობიექტი მოთავსებულია **ტრანსფორმაციის ჩარჩოში**, ნებისმიერ ადგილას მაუსის მარჯვენა ღილაკზე დაჭერისას გამოსულ მენიუში:

მასშტაბირება (Scale):

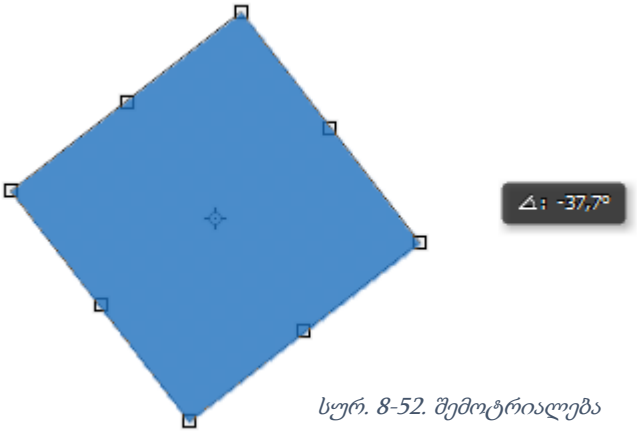
ობიექტის მასშტაბირება საკონტროლო წერილის მიმართ. მასშტაბირება შეიძლება როგორც ჰორიზონტალურად, ისე ვერტიკალურად ან ორივე მიმართულებით ერთდროულად;



სურ. 8-51. მასშტაბირება

შემოტრიალება (Rotate):

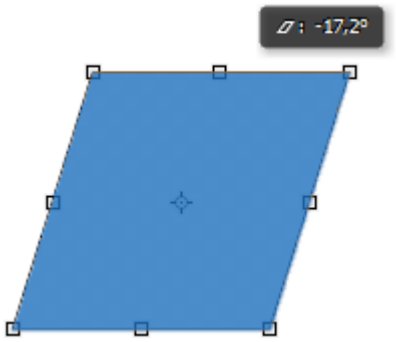
ობიექტის შემოტრიალება საკონტროლო წერტილის გარშემო. ნაგულისხმევად ეს წერტილი ცენტრშია მოთავსებული (სურ. 8-51);



სურ. 8-52. შემოტრიალება

დახრა (Skew):

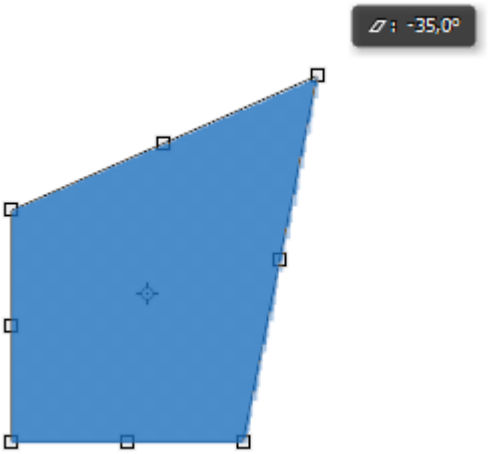
ობიექტის დახრა ჰორიზონტალურად ან ვერტიკალურად;



სურ. 8-53. დახრა

დამახინჯება (Distort):

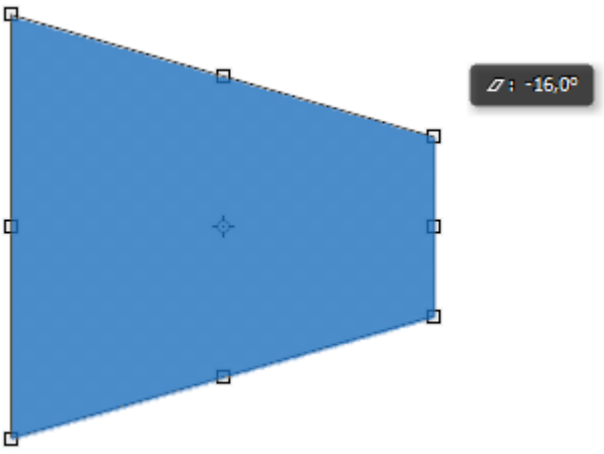
ობიექტის გაწელვა ყველა მიმართულებით;



სურ. 8-54. დამახინჯება

პერსპექტივა (Perspective):

ობიექტთან მიმართებაში გამოიყენება ერთ წერტილში შეკრებისპერსპექტივა (ინფორმაცია პერსპექტივის შესახებ);

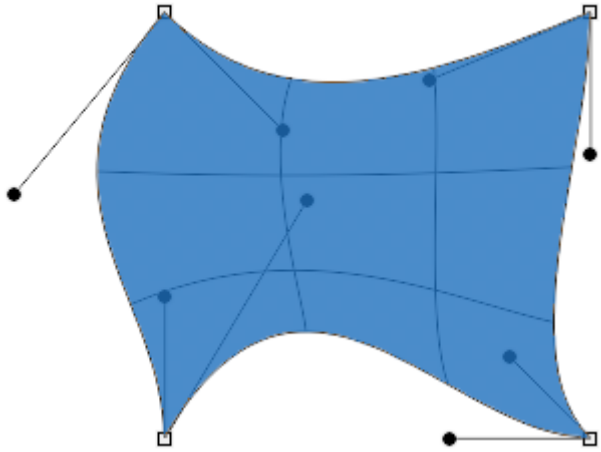


სურ. 8-55. პერსპექტივა

დეფორმაცია (Warp):

ობიექტის ფორმის შეცვლა; შეიცავს დამატებით მზა ფორმებს, როგორებიცაა თაღი, თევზი, დროშა და სხვა. ფორმატირების ზოლზე **დეფორმაცია** (Warp) ჩამოსაშლელი მენიუდან აირჩიეთ გამზადებული ფორმა. ამ დროს გამოჩენილ ბადეს გააჩნია ერთი საკვანძო წერტილი, რომლის დახმარებითაც თქვენ შეგიძლიათ ცვალოთ მისი ფორმა.

თუ მენიუდან აირჩევთ ფუნქციას **მორგებული** (Custom),გამოსახულებაზე დადებულ ბადეს ოთხი საკვანძო წერილი ექნება (კუთხეებში), რომელთა დახმარებითაც შეგიძლიათ თავისუფლად ცვალოთ ობიექტის ფორმა და მიანიჭოთ მას ნებისმიერი ფორმა.



სურ. 8-56. დეფორმაცია

შემოტრიალება 180°, 90° საათის ისრის ან მის საწინააღმდეგო მიმართულებით:

ობიექტი შემობრუნდება მოცემული კუთხით საათის ისრის ან მის საწინააღმდეგო მიმართულებით;

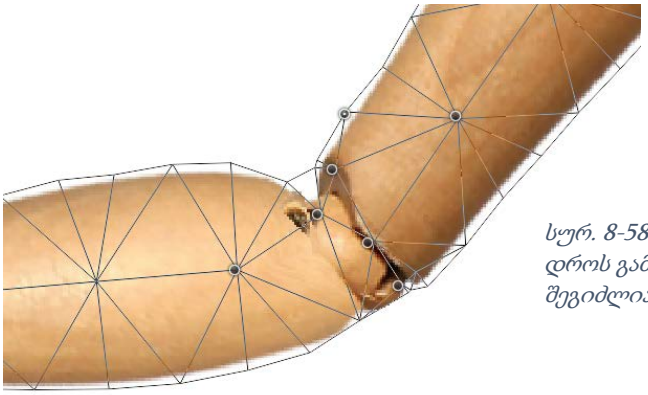
ანარეკლი *(Flip Horizontal / Vertical):*

ობიექტის ანარეკლი ჰორიზონტალურ ან ვერტიკალურ სიბრტყეში.

სურ. 8-57. ანარეკლი (ვერტიკალურად)

მარიონეტული დეფორმაცია (Puppet Warp)

ეს ობიექტის ტრანსფორმირების ერთ-ერთი ინსტრუმენტია. მისი გამოძახება ხდება Edit -> Puppet Warp.გამოსახულება ამ დროს იფარება დეტალური ბადით. ხაზების გადაკვეთის წერტილში კი შეგიძლიათ დეფორმაციის წერტილების დამატება (). წერტილების დახმარებით კი შეგიძლიათ ცვალოთ ბადის სტრუქტურა (წანაცვლება, შემობრუნება), რაც შესაბამისად გამოიწვევ ობიექტის ფორმის შეცვლას.

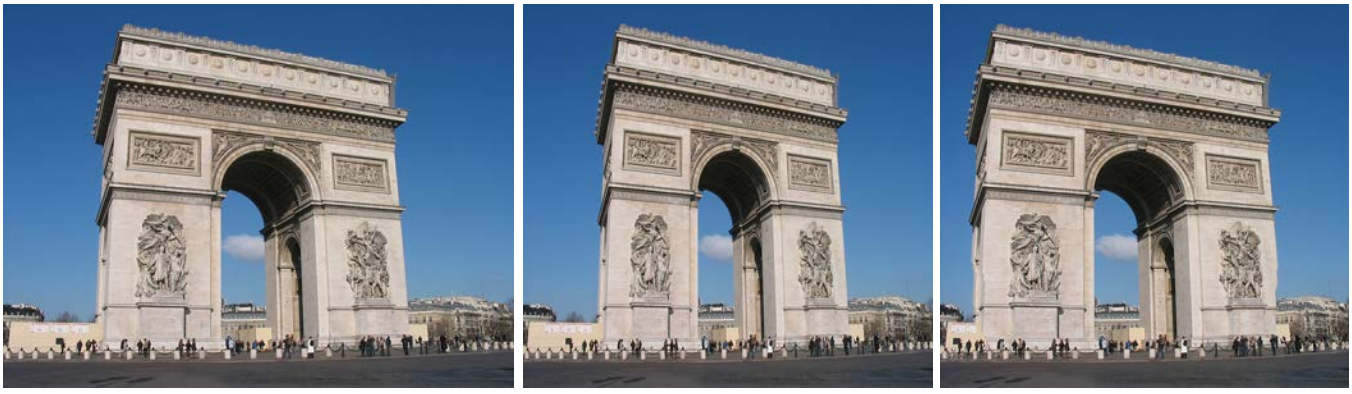


სურ. 8-58. მარიონეტული დეფორმაციის (Puppet Warp) დროს გამოსახულება იფარება ბადით, რომელზეც შეგიძლიათ დეფორმაციის წერტილების დამატება.

სურ. 8-59. ობიექტი ტრანსფორმაციამდე და ტრანსფორმაციის შემდეგ

მასშტაბირება შიგთავსის გათვალისწინებით (Content-Aware Scale)

პროგრამა გამოსახულებას აანალიზებს და მისი ზომისა და/ან პროპორციების შეცვლასამ ინფორმაციის საფუძველზე ახდენს. თუ ჩვეულებრივი ტრანსფორმაციის დროს ხდება ერთბაშად ყველა პიქსელების შემჭიდროვება, ასეთ შემთხვევაში ხდება მხოლოდ იმ პიქსელების შემჭიდროვება სადაც მინიმალური ან მსგავსიინფორმაცია ინახება.

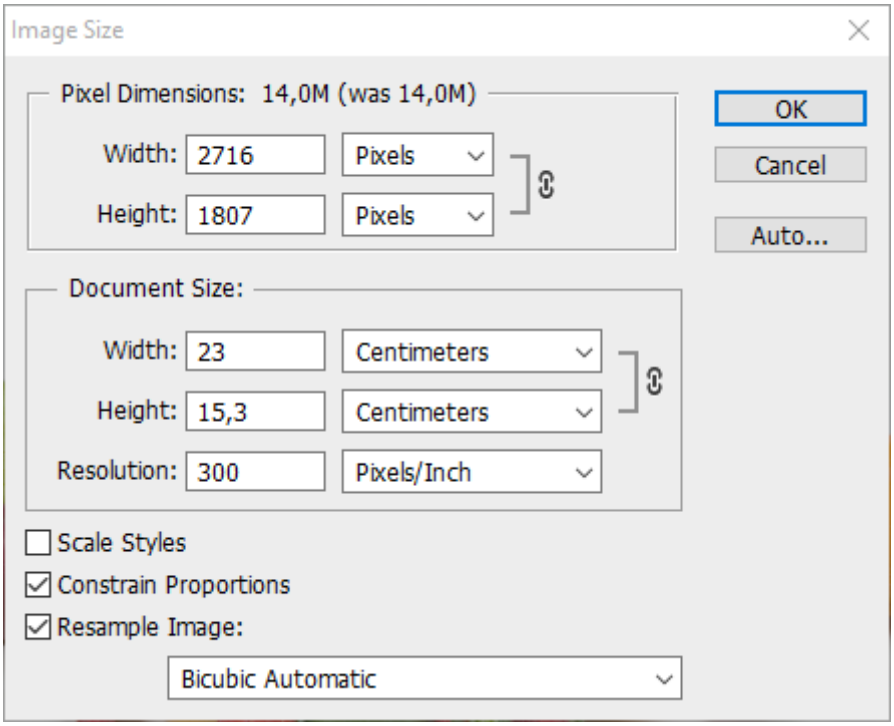


სურ. 8-60. პირველი საწყისი გამოსახულებაა. მეორე გამოსახულება შემჭიდროვებულია 20%-ით თავისუფალი ტრანსფორმაციის (Free Transform) გამოყენებით. მესამე გამოსახულება შემჭიდროვებულია 20%-ით მასშტაბირებისშიგთავსის გათვალისწინების მეთოდით (Content-Aware Scale).

გამოსახულების ზომა (Image Size)

გამოსახულების ელემენტებისა და ცალკეული ფრაგმენტების რედაქტირების გარდა, ჩვენ ასევე შეგვიძლია მისი ზომებისა და გარჩევადობის შეცვლა/რედაქტირება.

ნებისმიერ ფაილს, რომელსაც ჩვენ შექმნით თუ გავხსნით პროგრამაში გააჩნია ზომები და გარჩევადობის ხარისხი. მუშაობის პროცესში რომ ვაკონტროლოთ ზომები, ობიექტების განლაგება და სხვა მსგავსი პარამეტრები, შეგვიძლია ვისარგებლოთ სახაზავით. მისი გამოძახება/გათიშვა ხდება მენიუდან View -> Rulers (Ctrl + R). მაგრამ ეს არ არის საკმარისი იმისათის, რომ გავიგოთ დეტალური ინფორმაცია გამოძახებული ფაილის ზომების შესახებ. ამისათვის კი ვიძახებთ Image -> Image Size (Alt + Ctrl + I).

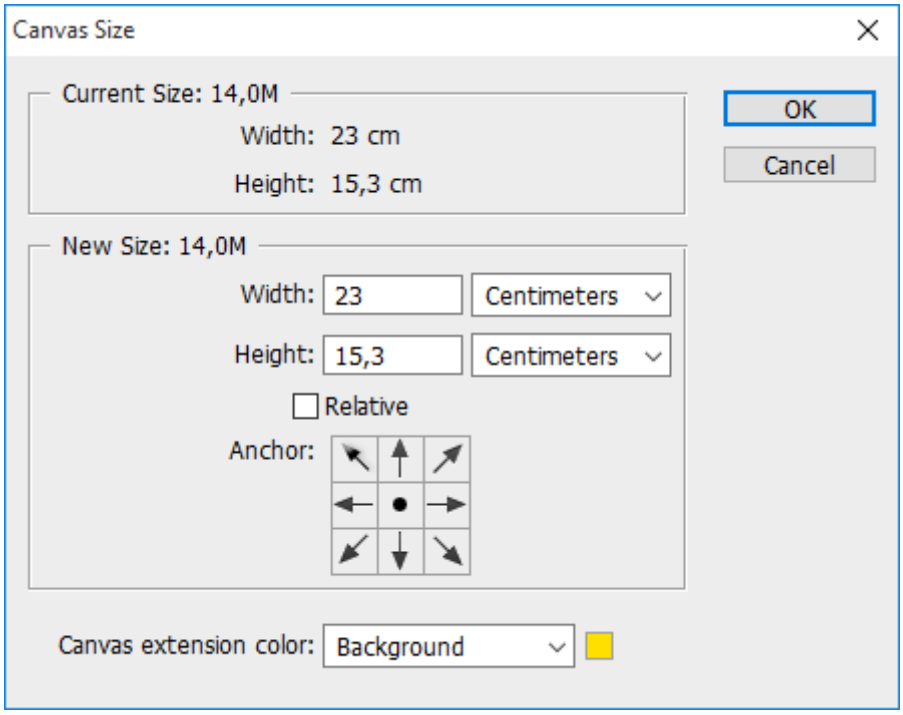


სურ. 8-61. გამოსახულების ზომა

აქ ჩვენ შეგვიძლია უკვე დეტალური ინფორმაცია ვიხილოთ ობიექტის შესახებ: მისი ზომა პიქსელებში და რომელიმე სხვა საზომ ერთეულში (სმ, მმ, პქს და სხვა), გარჩევადობის ხარისხი. ამავე ფანჯრიდან შეგვიძლია ვცვალოთ სიგანე, სიმაღლე ან ორივე მახასიათებელი ერთად (ასეთ შემთხვევაში უნდა იყოს მონიშნული **პროპორციების დაცვა** (Constrain Proportion).

ტილოს ზომა (Canvas Size)

მუშაობის პროცესში ასევე დაგვჭირდება არამხოლოდ ფაილის ზომების ცვლილება, არამედ სამუშაო ტილოს გაზრდა. ამისათვის ვიძახებთ Image -> Canvas Size (Alt + Ctrl + C):



სურ. 8-62. ტილოს ზომის ცვლილება (Canvas Size)

ამ ფანჯრის დახმარებით ჩვენ ვცვლით ტილოს ზომას - ვუმატებთ ან ვაკლებთ სამუშაო სივრცეს სიმაღლეში, სიგანეში ან ორივე მიმართულებით.

შემოჭრის ინსტრუმენტი (Crop Tool)

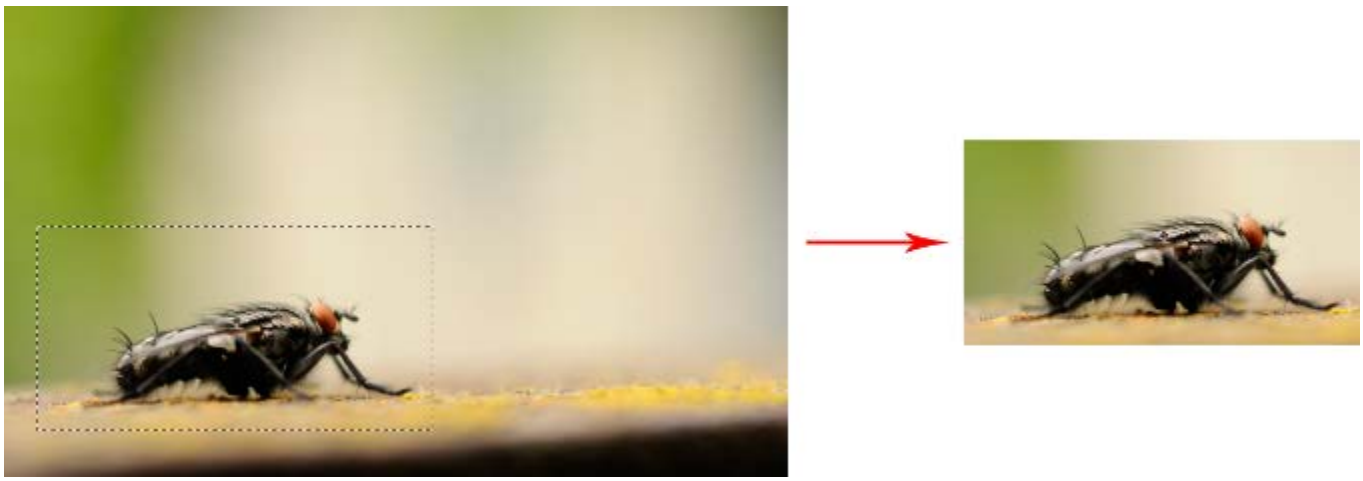
შემოჭრის ინსტრუმენტი გვეხმარება მოვაჭრათ გამოსახულებას ზედმეტი, არასასურველი არეები. როდესაც ამ ინსტრუმენტს ავირჩევთ, გამოსახულება მოთავსდება ჩარჩოში, რომელიც წააგავს ტრანსფორმაციისას. კიდეებზე ან კუთხეებზე თუ მოვკიდებთ კურსორს, შეგვიძლია მისი გადაადგილება (როგორც წესი ცენტრისკენ). ჩარჩოს გარეთ დარჩენილი გამოსახულება ჩამოიჭრება.

სურ. 8-63. გამოსახულების შემოჭრა. ჩარჩოს გარეთ მოხვედრილი სივრცე მოიჭრება, ხოლო ჩარჩოს შიგნით კი დარჩება.

გარდა ამისა, ფორმატირების ზოლზე მოცემული პარამეტრების დახმარებით ჩვენ შეგვიძლია ვმართოთ ჩამოჭრა: ზომები, დახრილობა და სხვა. საინტერესოა ჩამრთველი **ჩამოჭრილი პიქსელების წაშლა** (Delete Cropped Pixels). თუ ეს ფუნქცია ჩართულია, მაშინ ჩარჩოს გარეთ დარჩენილი პიქსელები წაიშლება, ხოლო თუ გამორთულია, მაშინ ისინი უბრალოდ დაიმალებიან. ხოლო მათი გამოჩენა შესაძლებელი იქნება თუ გამოიძახებთ ფუნქციას Image ->Reveal All.

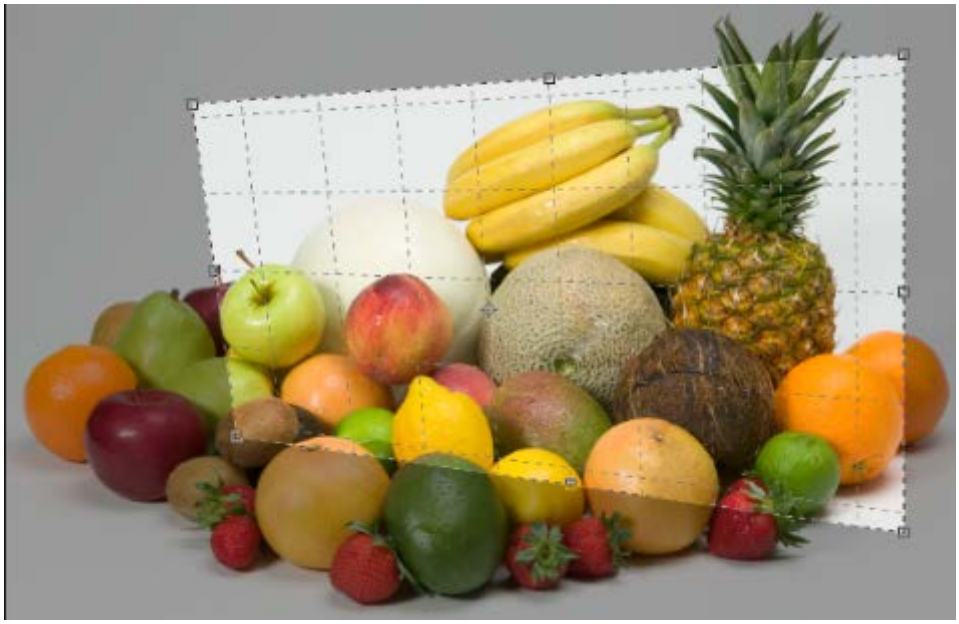
შენიშვნა:*გაითვალისწინეთ, რომ თუ პიქსელები დაიფარება და არა წაიშლება, ფაილის მოცულობა (დისკზე) არ შეიცვლება.*

შემოჭრის ინსტრუმენტის გარდა, ზედმეტი არეების მოსაჭრელად ასევე შეგიძლიათ გამოიყენოთ მართკუთხა მონიშვნის ინსტრუმენტი. მონიშნეთ სასურველი არე (რაც უნდა დარჩეს), შემდეგ გამოიძახეთ Image – Crop.



სურ. 8-64. მართკითხა მონიშვნის ინსტრუმენტის (Rectangular Marque Tool) გამოყენება გამოსახულების შემოსაჭრელად. ფოტო: გიორგი კევლიშვილი

პროგრამაში არსებობს ასევე მეორე სახის შემოჭრის ინსტრუმენტი, რომელსაც **პერსპექტივაში შემოჭრა** ეწოდება. გამოიყენება დეფორმირებული სურათების შემოსაჭრელად. მოქმედებით ის ძალიან წააგავს ჩვეულებრივ შემოჭრის ინსტრუმენტს. განსხვავება კი ისაა, რომ აქ შეგვიძლია ცალკეული (საკონტროლო) წერტილების გადაადგილება და შემოჭრის არისთვის ფორმის მიცემა



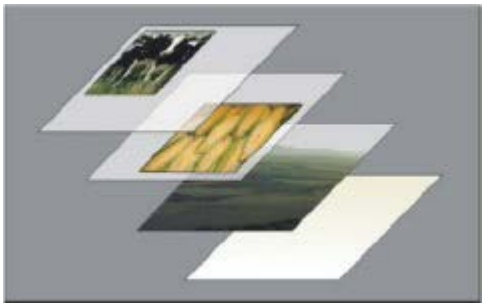
სურ. 8-65. პერსპექტივაში შემოჭრის ინსტრუმენტი (Perspective Crop Tool)

გარდა ამ მოქმედებისა, ჩვენ ასევე შეგვიძლია გამოსახულება დავატრიალოთ სხვადასვა კუთხით ან სულაც სარკულად შევაბრუნოთ.ამისათვის შედით Image -> Rotation. თქვენ შეგიძლიათ შემოაბრუნოთ გამოსახულება 180°, 90° საათის ისრის ან მის საწინააღმდეგო მიმართულებით, შეგიძლიათ შეაბრუნოთ გამოსახულება სარკულად ჰორიზონტალურ ან ვერტიკალურ სიბრტყეში, ხოლო ფუნქცია Arbitrary დაგეხმარებათ მოაბრუნოთ გამოსახულება ნებისმიერი კუთხით - უბრალოდ მიუთითეთ კუთხის რიცხვითი მნიშვნელობა და მიმართულება (საათის ისრის ან მის საწინააღმდეგო მიმართულებით).

ფუნქცია Image -> Dublicate… გაძლევთ საშუალებას შექმნათ გამოსახულების ზუსტი ასლი, როგორც თავიდანვე, ისე მუშაობის პროცესში. კარგია გამოიყენოთ მაშინ, როდესაც არ გინდათ, რომ დაზიანდეს გამოსახულების ორიგინალი; ან მუშაობის რაღაც ეტაპზე გინდათ მოსინჯოთ რაიმე ფილტრი ან ფუნქცია, მაგრამ შედეგში არ ხართ დარწმუნებული - შექმენით ასლი და გააგრძელეთ მუშაობა.

ფენები და მათთან მუშაობა

ფენები გამჭვირვალე ქაღალდების დასტას გავს. ზემოთ მოთავსებული ფენების გამჭვირვალე არეების წყალობით ჩვენ შეგვიძლია დავინახოთ ქვედა ფენები. ჩვენ შეგვიძლია ვცვალოთ ფენების მიმდევრობა ისევე, როგორც დასტაში ფურცლები.



სურ. 8-66. ფენების ვიზუალური წარმოდგენა

ფენები შეგვიძლია გამოვიყენოთ ტექსტის, ვექტორული ობიექტის ან სხვა გამოსახულების დადებისას ძირითად გამოსახულებაზე. შესაძლებელია ამ ფენებზე სპეციალური ეფექტებიც გამოვიყენოთ, როგორიცაა ჩრდილი, განათება, ფერის შეცვლა და სხვა.

ფენებთან სამუშაოდ ვიყენებთ მენიუ Layer (ფენა) და ფენების პანელს - Window -> Layers (F7).

მენიუდან ჩვენ შეგვიძლია შევქმნათ სხვადასხვა სახის ფენები, დავაჯგუფოთ ან დავაკავშიროთ ისინი ერთმანეთთან, გამოვიყენოთ ეფექტები და ა.შ. ფენების პანელის დახმარებით კი ჩვენ ვიზუალურად ვხედავთ ფენებს, შეგვიძლია მათ შევუცვალოთ მიმდევრობა, გამოვიყენოთ ეფექტები, წავშალოთ, გავთიშოთ/გამოვაჩინოთ და ა.შ. ფუნქციების ნაწილი შესაძლებელია გამოიყენოთ როგორც ერთი ფენისთვის, ისე ფენათა ჯგუფისთვის.

შენიშვნა:*გაითვალისწინეთ, რომ ზოგი ფუნქცია, რაც არის ხელმისაწვდომი მენიუში, არ არის ფენების (Layers) პანელზე და პირიქით.*

ფენების ნაირსახეობები

პროგრამაში რამდენიმე ნაირსახეობის ფენა არსებობს. ყველა ფენას თავისი დანიშნულება აქვს.

ფენების ჯგუფი (Layer Group): ეს ერთგვაროვანი საქაღალდეა, რომელში შეგიძლიათ მოათავსოთ ან ამოიღოთ ფენები (იხ. ფენების დაჯგუფება/დაშლა);

ტექსტური ფენა (Type Layer): მახასიათებლებით გაქვს გამოსახულების ფენას, მაგრამ მასზე შესაძლებელია ტექსტის რედაქტირება (მაგ. ფერი, ზომა, შრიფტი) (იხ.8.4 ტექსტი და მისი რედაქტირება);

მაკორექტირებელი ფენა (Adjustment Layer): ამ ფენის დახმარებით შესაძლებელი ქვემდებარე ფენების ფერის ან/და ტონალობის შეცვლა;

ფენის ნიღაბი (Layer Mask): ნიღაბზე ფუნჯით (იხ. ფუნჯი (Brush)) ან საშლელით (იხ. საშლელი (Eraser Tools)) მოქმედების დროს თქვენ შეგიძლიათ გამოაჩინოთ ან პირიქით დამალოთ გამოსახულების ცალკეული ნაწილები;

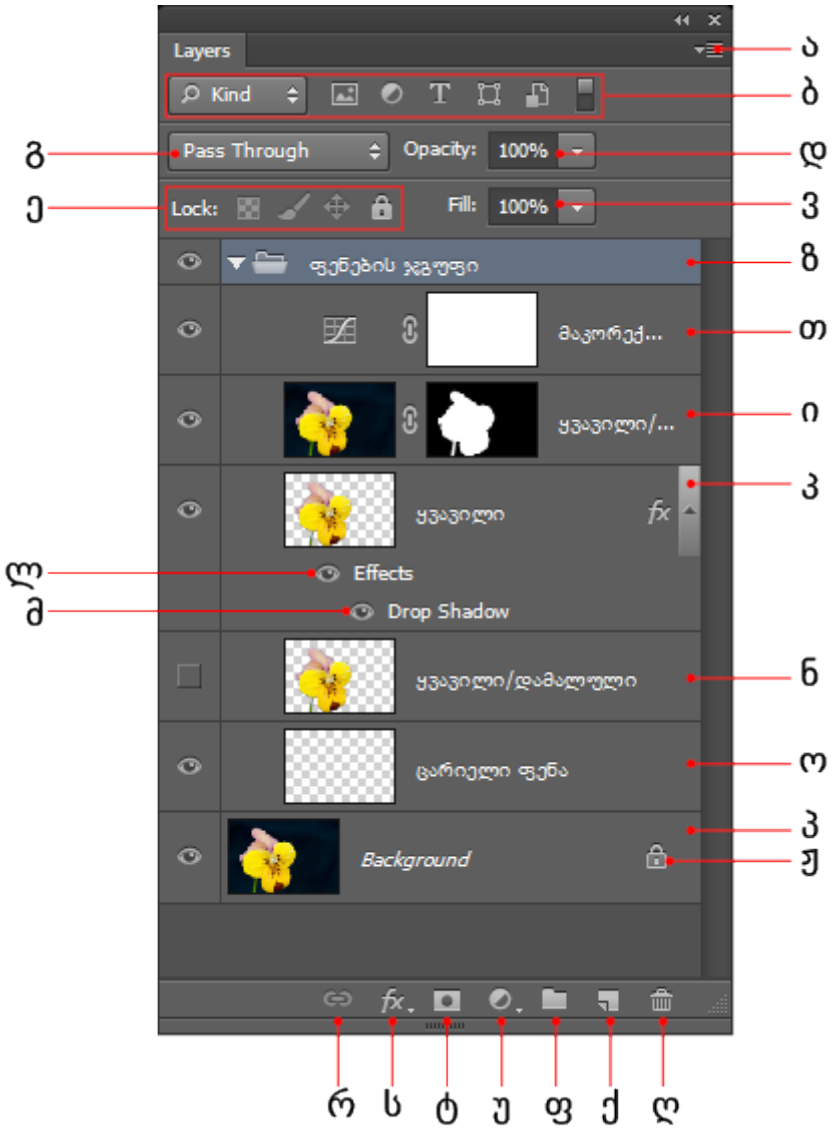
სმარტ-ობიექტი (Smart Object): ეს ფენა ერთგვარ კონტეინერს წარმოადგენს. კონტეინერში შეიძლება განვათავსოთ როგორც ერთი, ისე რამდენიმე ობიექტი (ისინი თავის მხრივ შეიძლება ფენებზე იყვნენ განლაგებულნი): ვექტორული ობიექტები, RAW ფაილები, ვიდეო, 3D და კიდევ უამრავი სხვა ტიპის ობიექტი;

მხატვრული ფენა (Art Layer): ძირითადი სამუშაო ფენა;

ძირითადი ფენა (Background): ფენა ყოველთვის ჩაკეტილი, მასზე განთავსებული გამოსახულების გადაადგილება შეუძლებელი და მის მიმართ ვერ გამოვიყენებთ ფენების სტილებს (იხ. ფენების სტილები).

ვიდეოს ფენა (Video Layer): ამ ფენაზე ვიდეო გამოსახულება თავსდება. შეგიძლიათ მისი ფერების კორექცია, გამოიყენოთ ფილტრები, დაჭრათ, მოახდინოთ მისი მარტივი მონტაჟი.

3D-ს ფენა (3D Layer): პროგრამაშიშესაძლებელიაროგორც 3D გამოსახულების იმპორტირება, ისე მათი შექმნა და მუშაობა. ბოლო ვერსიებში გაუმჯობესებულია 3D ბეჭდვის ტექნოლოგია.



სურ. 8-67. ფენების პანელი (Layers).
ა - პანელის მენიუ;
ბ - ფილტრაციის ინსტრუმენტები;
გ - ფენების შერევის რეჟიმები;
დ - ფენის გამჭვირვალობა;
ე - ფენის ჩაკეტვის ინსტრუმენტები;
ვ - შევსების გამჭვირვალობა;
ზ - ფენათა ჯგუფი;
თ - მაკორექტირებელი ფენა;
ი - შენიღბული ფენა;
კ - ეფექტების დამალვა/გამოჩენა;
ლ - ყველა ეფექტის დროებით გათიშვა;
მ - კონკრეტული ეფექტის დროებით გათიშვა;
ნ - დამალული ფენა;
ო - ცარიელი ფენა;
პ - ძირითადი ფენა;
ჟ - ფენის ჩაკეტვის ნიშნული
რ - ფენებს შორის ბმა/ბმის დაშლა;
ს - ფენაზე ეფექტების გამოყენება;
ტ - ფენაზე ნიღბის დამატება;
უ - მაკორექტირებელი ფენის შექმნა;
ფ - ახალი ჯგუფის შექმნა (ფენების);
ქ - ახალი ფენის შექმნა;
ღ - ფენის წაშლა.

შერევის (ზედდების) რეჟიმები

შერევის რეჟიმები დაყოფილია მცირე ჯგუფებად. სულ ექვსი ესეთი ჯგუფია.

საბაზო რეჟიმები:

Normal - ნორმალურ რეჟიმში ზედა და ქვედა ფენის პიქსელების შერევა არ ხდება. ზედა ფენის პიქსელები ავტომატურად ჩაანაცვლებენ ქვედა ფენის პიქსელებს, თუ ზედა ფენის გამჭვირვალობა 100%-ის ტოლია. პიქსელების ფერების შერევის ეფექტი, ფენის გამჭვირვალობის შემცირებით მიიღწევა, ისიც იმ შემთხვევაში, თუ ფენებზე სხვადასხვა სახის გამოსახულება არის მოთავსებული. ავტომატურ მდგომარეობაში, ყველა ფენა Normal რეჟიმში იმყოფება.

Dissolve - გახსნის რეჟიმში პიქსელების ფერი შემთხვევითი შერჩევის გზით მიიღება და მიღებულ გამოსახულებას აქვს მარცვლოვანი, ფოროვანი სახე. ეფექტი შესამჩნევია თუ ზედა ფენის გამჭვირვალობა 100%-ზე ნაკლებია.

დამუქების რეჟიმები:

Darken - მუქი ფერით ჩანაცვლება. ზედა და ქვედა ფენის პიქსელების ფერების შედარების შემდეგ, საბაზო ფენის იმ პიქსელების ფერის ჩანაცვლება ხდება ზედა ფენის პიქსელებით, რომლებიც უფრო ღია ფერისაა, ვიდრე მის ზემოთ მდებარე პიქსელი ფერი, ხოლო პიქსელი, რომელიც უფრო მუქი ფერისაა ვიდრე ზედა ფენის პიქსელი, უცვლელი რჩება.

Multiply - გამრავლება. ხდება ზედა და ქვედა ფენების ფერების გადამრავლება და შედეგად მიღებული გამოსახულების ფერი უფრო მუქია, ვიდრე ცალკე აღებული თითოეული ფენის გამოსახულების ფერი. პროცესში მონაწილეობს ყველა ფერის პიქსელი, თეთრი ფერის გარდა.

Color Burn - ფუძის დამუქება. ხდება მუქი ფერის პიქსელების დამუქება, ხოლო ღია ფერის პიქსელები უცვლელი რჩება.

Linear Burn - ხაზოვანი დამუქება. ისევე მოქმედებს, როგორც Color Burn რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Darker Color - შედარებით მუქი ფერი. ისევე მოქმედებს, როგორც Darken რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

გაღიავების რეჟიმები:

Lighten - ღია ფერით ჩანაცვლება. Darken რეჟიმის საპირისპირო რეჟიმია. ზედა და ქვედა ფენის პიქსელების ფერების შედარების შემდეგ, საბაზო ფენის იმ პიქსელების ფერის ჩანაცვლება ხდება ზედა ფენის პიქსელებით, რომლებიც უფრო მუქი ფერისაა, ვიდრე მის ზემოთ მდებარე პიქსელის ფერი, ხოლო პიქსელი, რომელიც უფრო ღია ფერისაა ვიდრე ზედა ფენის პიქსელი, უცვლელი რჩება.

Screen - განათება. Multiply რეჟიმის საპირისპირო რეჟიმია. ხდება გამოსახულების ყველა ფერის პიქსელის გაღიავება, შავი ფერის გარდა.

Color Dodge - ფუძის განათება. Color Burn რეჟიმის საპირისპირო რეჟიმია. აღიავებს გამოსახულების ღია ფერის პიქსელებს და უცვლეს ტოვებს მუქი ფერის პიქსელებს.

Linear Dodge - ხაზოვანი განათება. იგივენაირად მოქმედებს, როგორც Color Dodge რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Lighter Color - ღია ფერი. ორივე ფენის პიქსელების ფერის შედარების შემდეგ, რჩება უფრო ღია ფერის პიქსელი.

კონტრასტის რეჟიმები:

Overlay - გადაფარვა. ეს არის Multiply და Screen რჟიმების კომბინირებული რეჟიმი. ხდება მუქი პიქსელების დამუქება და ღია პიქსელების გაღიავება. ეფექტი არ მოქმედებს შავ და თეთრ ფერებზე. ამ რეჟიმში 50%-იანი ნაცრისფერი გაუმჭვირვალე ხდება.

Soft Light - რბილი განათება. იგივენაირად მოქმედებს, როგორც Overlay რეჟიმი, მაგრამ ეფექტი უფრო სუსტია.

Hard Light - ძლიერი განათება. იგივენაირად მოქმედებს, როგორც Overlay რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Vivid Light - მკვეთრი განათება. იგივენაირად მოქმედებს, როგორც Overlay და Hard Light რეჟიმები, მაგრამ ეფექტი უფრო ძლიერია.

Linear Light - ხაზოვანი განათება. Vivid Light რეჟიმივით მოქმედებს, მაგრამ ეფექტი უფრო ძლიერია.

Pin Light - წერტილოვანი განათება. ღია ფერის პიქსელების შერევის დროს იყენებს Lighten რეჟიმს, ხოლო მუქი ფერის პიქსელების შერევისას, Darken რეჟიმს.

Hard Mix - მკვეთრი შერევა. RGB არხების მიხედვით ხდება ზედა და ქვედა პიქსელების სიმკვეთრის მაჩვენებლების გადათვლა. დაჯამების შემდეგ პიქსელი იღებს ზღვრულ მნიშვნელობას: თუ ჯამური მაჩვენებელი 255-ზე ნაკლებია, არხის რიცხვითი მაჩვენებელი 0-ს უტოლდება, ხოლო თუ ჯამური მაჩვენებელი 255 -ია ან 255-ზე მეტია - 255-ს. ამის შედეგად, გამოსახულებაზე 8 ძირითადი ფერის პიქსელი მიიღება ანუ ხდება გამოსახულების პოსტერიზაცია.

შედარების რეჟიმები:

Difference - განსხვავება. ხდება პიქსელების სიმკვეთრის მაჩვენებლების შედარება, დიდი მაჩვენებლებიდან პატერა მაჩვენებლების გამოკლება და სხვაობისშებრუნება. ერთნაირი მაჩვენებლების პიქსელები შავი ფერით აისახება.

Exclusion - გამორიცხვა. ისევე მოქმედებს, როგორც Difference რეჟიმი, მაგრამ ეფექტი უფრო სუსტია.

გამოსახულების კომპონენტების რეჟიმები:

Hue - ტონი. მიიღება გამოსახულება ზედა ფენის ტონით და ქვედა ფენის გაჯერებულობით და სიმკვეთრით.

Saturation - გაჯერებულობა. მიიღება გამოსახულება ზედა ფენის გაჯერებულობით და ქვედა ფენის ტონით და სიმკვეთრით.

Color - ფერი. მიიღება გამოსახულება ზედა ფენის ტონით და გაჯერებულობით და ქვედა ფენის სიმკვეთრით.

Luminosity - სიმკვეთრე. მიიღება გამოსახულება ზედა ფენის სიმკვეთრით და ქვედა ფენის ტონით და გაჯერებულო- ბით.

ახალი ფენის შექმნა

Layer -> New -> New Layer (Ctrl + Shift + N) - ახალი ფენი შექმნა;

Layer -> New -> New Layer from Background - ახალი ფენის შექმნა ძირითადი ფენისგან;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> New Layer (Ctrl + Shift + N);

მართვის პანელზე ახალი ფენის შექმნის ღილაკი (სურ. 8-68, ქ).

ფენის ასლის/დუბლიკატის შექმნა

Layer -> New -> Layer via Copy (Ctrl + J) - შექმნის მიმდინარე ფენის ასლს. თუ გამოსახულებაზე მონიშნული გაქვთ არე, მაშინ მოხდება მისი ასლის შექმნა და მისი განთავსება ახალ ფენაზე;

ფენების პანელის მენიუ (სურ. 8-68, ა) ->Dublicate Layer…;

ფენების პანელზე ფენის გადათრევით ახალი ფენის შექმნის ღილაკზე (სურ. 8-68, ქ).

ფენის დაბლოკვა/გააქტიურება

ფენების პანელზე ჩამკეტი ღილაკების (სურ. 8-68, ე) დახმარებით. ჩაკეტვის ოთხნაირი საშუალება არსებობს - მხოლოდ გამჭვირვალე პიქსელები, გამოსახულების პიქსელები, პოზიციის დაბლოკვა, ყველაფრის დაბლოკვა. მაგალითად ძირითადი ფენა (სურ. 8-68, პ) ნაგულისხმევად სრულად არის ჩაკეტილი და გვერდით შესაბამისი ნიშნული აქვს (სურ. 8-68, ჟ).

დასაბლოკად, მონიშნეთ სასურველი ფენა და დააწექით შესაბამის ღილაკს (სურ. 8-68, ე).

ბლოკის მოსახსნელად ხელმეორედ დააწექით იმავე ღილაკს.

Layer -> Lock All Layers in Group - ჯგუფში (სურ. 8-68, ზ) არსებულ ყველა ფენის დაბლოკვა

ფენის დამალვა/გამოჩენა

ფენა ნაგულისხმევად (არსებული თუ ახალი შექმნილი) ყოველთვის ჩანს. ამას მიუთითებს ყოველი ფენის გვერდით (მარცხნიდან) არსებული თვალის ნიშნული. ფენის დასამალად დააწექით ამ ნიშნულს - ის გაქრება, რაც მიუთითებს, რომ ფენა დამალულია ანუ მასზე არსებული გამოსახულება არ გამოჩნდება სამუშაო ტილოზე (სურ. 8-68, ნ). გამოსაჩენად დააწექით ცარიელ კვადრატს.

ასევე შეგიძლიათ ისარგებლოთ ფუნქციით Layer -> Hide/Show Layers.

ფენის მონიშვნა (არჩევა)

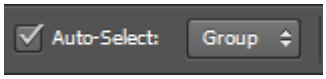
მუშაობის პროცესში გარკვეული მოქმედებები შესაძლოა არა ერთ ფენაზე, არამედ რამდენიმე ფენაზე მოგიწიოთ (დამალვა, ეფექტების გამოჩენა, ფენაზე განთავსებული ობიექტების ერთმანეთთან გასწორება და სხვა).

ფენა რომ მონიშნოთ, დააწექით შესაბამის ფენას. ფენათა ჯგუფის მოსანიშნად გამოიყენება კლავიშები Shift და Ctrl.

Shift კლავიშისგამოყენება: როდესაც ფენები გინდათ მონიშნოთ მიმდევრობით, მონიშნეთ ერთი ფენა და შემდეგ კლავიშის დახმარებით მონიშნეთ სხვა ფენა. მოინიშნება ეს ორივე ფენა და მათ შორის მდებარე ყველა ფენა.

Ctrl კლავიშის გამოყენება: მონიშნეთ ერთი ფენა და კლავიშის გამოყენებით (უნდა გეჭიროთ) შეგიძლიათ მონიშნოთ ფენები არჩევითად.

ფენების მონიშვნა შესაძლებელია ასევე გადაადგილების (Move) ინსტრუმენტის დახმარებით. ფორმატირების ზოლზე უნდა ჩართოთ ავტომატური მონიშვნა (Auto-Select), ხოლო მენიუდან აირჩიოთ რისი მონიშვნა გინდათ:ჯგუფის (სურ. 8-68, ზ) თუ ცალკეული ფენის.



სურ. 8-68. გადაადგილების (Move)ინსტრუმენტისფუნქცია ავტომატური მონიშვნა (Auto-Select) ფორმატირების ზოლზე

შემდეგში, მუშაობის პროცესში, გამოსახულებაზე ცალკეულ ელემენტზე დაჭერისას მოინიშნება ის ფენა, რომელზეც ეს ელემენტი მდებარეობს.

იმ შემთხვევაში, თუ ფენას გააჩნია ბმა (იხ. ფენებს შორის ბმა) სხვა ფენებთან, მაშინ გამოიძახეთ ფუნქცია Layer -> Select Linked Layers. ამ დროს მოინიშნება ყველა ფენა, რომელსაც ამ ფენასთან ბმა გააჩნია.

ფენების დაჯგუფება/დაშლა

Layers -> Group Layers (Ctrl + G) - ფენების დაჯგუფება;

Layers ->Ungroup Layers (Shift + Ctrl + G) - ფენების დაჯგუფება;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> New Group - შეიქმნება ცარიელი ჯგუფი, სადაც შემდეგში გადაიტანთ სასურველ ფენებს;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> New Group from Layers - ჯგუფის შექმნა მონიშნული ფენებისგან;

ფენების პანელზე ახალი ჯგუფის შექმნის ღილაკი - ქმნის ახალ ცარიელ ჯგუფს (სურ. 8-68, ფ).

ფენებს შორის ბმა

გამოიყენება იმ დროს, როდესაც გსურთ რაიმე საერთო მოქმედების შესრულება (გადაადგილება, ტრანსფორმაცია და სხვა), მაგრამ არ არის საჭირო მათი დაჯგუფება. მონიშნული უნდა იყოს ორი ან მეტი ფენა. ბმის შექმნის შემდეგ ფენაზე (დასახელების გვერდით) ჩნდება ჯაჭვის ნიშანი.

Layer - > Link/Unlink Layers - ფენებს შორის ბმა/ბმის წყვეტა;

ფენების პანელის მენიუ (სურ. 8-68, ა) - > Link/Unlink Layers;

ფენების პანელზე ბმა/ბმის წყვეტის ღილაკი (სურ. 8-68, რ).

ფენების სწორება ერთმანეთის მიმართ

ხანდახან საჭიროა და აუცილებელიც კი ხდება ფენებზე განთავსებული ობიექტები ერთმანეთს გაუსწოროთ. სწორება შესაძლებელია მოხდეს კიდეზე, ცენტრის მიმართ ან განისაზღვროს თანაბარი მანძილი ობიექტებს შორის. ამისათვის უნდა მონიშნოთ ის ფენები, რომლებიც უნდა გაუსწოროთ ერთმანეთს, შემდეგ გამოიძახეთLayer -> Align. ამ მენიუში მოთავსებული ფუნქციების დახმარებით თქვენ შეგიძლიათ ობიექტები ექვსი მახასიათებლის მიხედვით გაასწოროთ.

იმისათვის, რომ მოახდინოთ ობიექტების თანაბრად განაწილება, უნდა მონიშნოთ ფენები (მინიმუმ სამი) და გამოიძახოთ Layer -> Distribute. აქაც ექვსი მახასიათებლის მიხედვით შეგიძლიათ გაანაწილოთ ობიექტები.

თუ არჩეული გაქვთ გადაადგილების (Move) ინსტრუმენტი, მაშინ შეგიძლიათ ისარგებლოთ ფორმატირების ზოლზე განლაგებული სწორებისა და განაწილების შესაბამისი ღილაკებით.

სურ. 8-69. სწორებისა და განაწილები ღილაკები ფორმატირების ზოლზე.

ფენების გადაადგილება

ფენების გადაადგილება გადათრევით - მონიშნეთ ფენა, მოკიდეთ კურსორით და გადაათრიეთ იმ ფენებს შორის, სადაც გინდათ, რომ იყოს ეს ფენა განთავსებული;

ფენების გადასაადგილებლად ზემოთ ან ქვემოთ:

Layers -> Arrange -> Bring to Front (Shift + Ctrl +))

Layers -> Arrange ->Bring Forward (Ctrl +))

Layers -> Arrange ->Bring Backward (Ctrl + ()

Layers -> Arrange ->Bring to Back (Shift + Ctrl + ()

შენიშვნა:*ძირითადი ფენა (Background) ყველაზე ქვედა ფენაა. მასზე ქვემოთ ფენის გადაადგილება შეუძლებელი მანამ, სანამ ის არ გადაიქცევა სამუშაო ფენად (იხ.*

შერევის (ზედდების) რეჟიმები

შერევის რეჟიმები დაყოფილია მცირე ჯგუფებად. სულ ექვსი ესეთი ჯგუფია.

საბაზო რეჟიმები:

Normal - ნორმალურ რეჟიმში ზედა და ქვედა ფენის პიქსელების შერევა არ ხდება. ზედა ფენის პიქსელები ავტომატურად ჩაანაცვლებენ ქვედა ფენის პიქსელებს, თუ ზედა ფენის გამჭვირვალობა 100%-ის ტოლია. პიქსელების ფერების შერევის ეფექტი, ფენის გამჭვირვალობის შემცირებით მიიღწევა, ისიც იმ შემთხვევაში, თუ ფენებზე სხვადასხვა სახის გამოსახულება არის მოთავსებული. ავტომატურ მდგომარეობაში, ყველა ფენა Normal რეჟიმში იმყოფება.

Dissolve - გახსნის რეჟიმში პიქსელების ფერი შემთხვევითი შერჩევის გზით მიიღება და მიღებულ გამოსახულებას აქვს მარცვლოვანი, ფოროვანი სახე. ეფექტი შესამჩნევია თუ ზედა ფენის გამჭვირვალობა 100%-ზე ნაკლებია.

დამუქების რეჟიმები:

Darken - მუქი ფერით ჩანაცვლება. ზედა და ქვედა ფენის პიქსელების ფერების შედარების შემდეგ, საბაზო ფენის იმ პიქსელების ფერის ჩანაცვლება ხდება ზედა ფენის პიქსელებით, რომლებიც უფრო ღია ფერისაა, ვიდრე მის ზემოთ მდებარე პიქსელი ფერი, ხოლო პიქსელი, რომელიც უფრო მუქი ფერისაა ვიდრე ზედა ფენის პიქსელი, უცვლელი რჩება.

Multiply - გამრავლება. ხდება ზედა და ქვედა ფენების ფერების გადამრავლება და შედეგად მიღებული გამოსახულების ფერი უფრო მუქია, ვიდრე ცალკე აღებული თითოეული ფენის გამოსახულების ფერი. პროცესში მონაწილეობს ყველა ფერის პიქსელი, თეთრი ფერის გარდა.

Color Burn - ფუძის დამუქება. ხდება მუქი ფერის პიქსელების დამუქება, ხოლო ღია ფერის პიქსელები უცვლელი რჩება.

Linear Burn - ხაზოვანი დამუქება. ისევე მოქმედებს, როგორც Color Burn რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Darker Color - შედარებით მუქი ფერი. ისევე მოქმედებს, როგორც Darken რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

გაღიავების რეჟიმები:

Lighten - ღია ფერით ჩანაცვლება. Darken რეჟიმის საპირისპირო რეჟიმია. ზედა და ქვედა ფენის პიქსელების ფერების შედარების შემდეგ, საბაზო ფენის იმ პიქსელების ფერის ჩანაცვლება ხდება ზედა ფენის პიქსელებით, რომლებიც უფრო მუქი ფერისაა, ვიდრე მის ზემოთ მდებარე პიქსელის ფერი, ხოლო პიქსელი, რომელიც უფრო ღია ფერისაა ვიდრე ზედა ფენის პიქსელი, უცვლელი რჩება.

Screen - განათება. Multiply რეჟიმის საპირისპირო რეჟიმია. ხდება გამოსახულების ყველა ფერის პიქსელის გაღიავება, შავი ფერის გარდა.

Color Dodge - ფუძის განათება. Color Burn რეჟიმის საპირისპირო რეჟიმია. აღიავებს გამოსახულების ღია ფერის პიქსელებს და უცვლეს ტოვებს მუქი ფერის პიქსელებს.

Linear Dodge - ხაზოვანი განათება. იგივენაირად მოქმედებს, როგორც Color Dodge რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Lighter Color - ღია ფერი. ორივე ფენის პიქსელების ფერის შედარების შემდეგ, რჩება უფრო ღია ფერის პიქსელი.

კონტრასტის რეჟიმები:

Overlay - გადაფარვა. ეს არის Multiply და Screen რჟიმების კომბინირებული რეჟიმი. ხდება მუქი პიქსელების დამუქება და ღია პიქსელების გაღიავება. ეფექტი არ მოქმედებს შავ და თეთრ ფერებზე. ამ რეჟიმში 50%-იანი ნაცრისფერი გაუმჭვირვალე ხდება.

Soft Light - რბილი განათება. იგივენაირად მოქმედებს, როგორც Overlay რეჟიმი, მაგრამ ეფექტი უფრო სუსტია.

Hard Light - ძლიერი განათება. იგივენაირად მოქმედებს, როგორც Overlay რეჟიმი, მაგრამ ეფექტი უფრო ძლიერია.

Vivid Light - მკვეთრი განათება. იგივენაირად მოქმედებს, როგორც Overlay და Hard Light რეჟიმები, მაგრამ ეფექტი უფრო ძლიერია.

Linear Light - ხაზოვანი განათება. Vivid Light რეჟიმივით მოქმედებს, მაგრამ ეფექტი უფრო ძლიერია.

Pin Light - წერტილოვანი განათება. ღია ფერის პიქსელების შერევის დროს იყენებს Lighten რეჟიმს, ხოლო მუქი ფერის პიქსელების შერევისას, Darken რეჟიმს.

Hard Mix - მკვეთრი შერევა. RGB არხების მიხედვით ხდება ზედა და ქვედა პიქსელების სიმკვეთრის მაჩვენებლების გადათვლა. დაჯამების შემდეგ პიქსელი იღებს ზღვრულ მნიშვნელობას: თუ ჯამური მაჩვენებელი 255-ზე ნაკლებია, არხის რიცხვითი მაჩვენებელი 0-ს უტოლდება, ხოლო თუ ჯამური მაჩვენებელი 255 -ია ან 255-ზე მეტია - 255-ს. ამის შედეგად, გამოსახულებაზე 8 ძირითადი ფერის პიქსელი მიიღება ანუ ხდება გამოსახულების პოსტერიზაცია.

შედარების რეჟიმები:

Difference - განსხვავება. ხდება პიქსელების სიმკვეთრის მაჩვენებლების შედარება, დიდი მაჩვენებლებიდან პატერა მაჩვენებლების გამოკლება და სხვაობისშებრუნება. ერთნაირი მაჩვენებლების პიქსელები შავი ფერით აისახება.

Exclusion - გამორიცხვა. ისევე მოქმედებს, როგორც Difference რეჟიმი, მაგრამ ეფექტი უფრო სუსტია.

გამოსახულების კომპონენტების რეჟიმები:

Hue - ტონი. მიიღება გამოსახულება ზედა ფენის ტონით და ქვედა ფენის გაჯერებულობით და სიმკვეთრით.

Saturation - გაჯერებულობა. მიიღება გამოსახულება ზედა ფენის გაჯერებულობით და ქვედა ფენის ტონით და სიმკვეთრით.

Color - ფერი. მიიღება გამოსახულება ზედა ფენის ტონით და გაჯერებულობით და ქვედა ფენის სიმკვეთრით.

Luminosity - სიმკვეთრე. მიიღება გამოსახულება ზედა ფენის სიმკვეთრით და ქვედა ფენის ტონით და გაჯერებულო- ბით.

ახალი ფენის შექმნა *ძირითადი ფენისგან).*

ფენების მიმდევრობა რომ შეაბრუნოთ, აირჩიეთ Layers ->Arrange -> Reverse.

ფენების წაშლა

შეგიძლიათ წაშალოთ როგორც ფენა, ისე ფენათა ჯგუფი.

Layer -> Delete -> Layer - ფენ(ებ)ის წაშლა;

Layer -> Delete -> Hidden Layers - ყველა დამალული ფენის (იხ. ფენის დამალვა/გამოჩენა) წაშლა;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> Delete Layer;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> Delete Hidden Layers;

ფენების პანელზე ფენის წაშლის ღილაკი (სურ. 8-68, ღ)

ფენების შერწყმა/გაერთიანება

Layer -> Merge Down (Ctrl + E) - ქვედა ფენასთან შერწყმა. აქტიურია ერთი მონიშნული ფენის დროს;

Layer -> Merge Layers (Ctrl + E) - ფენების შერწყმა. აქტიურია ერთზე მეტი მონიშნული ფენის დროს. მონიშვნის მიმდევრობას მნიშვნელობა არ აქვს;

Layer -> Merge Visible (Shift + Ctrl + E) - ხილული ფენების შერწყმა. დამალული ფენები ხელუხლებელი დარჩება;

Layer -> Flatten Image - ყველა ფენის შერწყმა. ამის შემდეგ დარჩება მხოლოდ ერთი ფენა, რომელიც იქნება ძირითადი (Background);

ფენების პანელის მენიუ (სურ. 8-68, ა) ->Merge Down/Layers;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> Merge Visible;

ფენების პანელის მენიუ (სურ. 8-68, ა) -> Flatten Image;

შენიშვნა: *დამატებითი კლავიშების კომბინაცია Shift + Ctrl + Alt + E აერთიანებს ყველა ფენას, მაგრამ ამ დროს ყველა ფენა რჩება როგორც იყო, ხოლო წარმოქმნილი ფენა თავსდება ყველაზე ზემოთ.*

სმარტ-ობიექტი

სმარტ-ობიექტის შექმნა ორი გზით შეიძლება (წინასწარ ფენა უნდა მონიშნოთ):

Layer -> Smart Objects -> Convert to Smart Object;

ფენების პანელის მენიუ -> Convert to Smart Object;

სხვა ფაილის შემოტანისას მიმდინარე ფაილში - File -> Place.

ამ ფენაზე შესაძლებელია ჩვეულებრივი მოქმედებების შესრულება, როგორიცაა მისი ტრანსფორმაცია, ფენების სტილის გამოყენება ან ფენების ჯგუფში მოთავსება.

სმარტ-ობიექტის პირდაპირი რედაქტირება შეუძლებელია (მოჭრა, წაშლა და სხვა). ამისათვის უნდა გამოიძახოთ:

Layer -> Smart Objects -> Edit Content;

ფენების პანელის მენიუ -> Edit Content;

გამოსახულება გაიხსნება სხვა ფაილში, სადაც თქვენ მოახდენთ მის რედაქტირებას (წაშლა, ხატვა და ა.შ.), შეინახავთ, როგორც ჩვეულებრივ ფაილს და სამუშაო ფაილში დაბრუნებისთანავე, თქვენ დაგხვდებათ გამოსახულება რედაქტირებული.

სმარტ-ობიექტზე ფილტრის გამოყენებისას, ის იმუშავებს უკვე როგორც სმარტ-ფილტრი. ჩვეულებრივისგან ეს ფილტრი იმით განსხვავდება, რომ მისი მოქმედება შესაძლებელია შეცვლილი ან გაუქმებული იყოს მუშაობის ნებისმიერ ეტაპზე.

ჩვეულებრივ ფენაზე სმარტ-ფილტრი არ მუშაობს - ის ჯერ გარდაიქმნება სმარტ-ობიექტად და ამის შემდეგ მასზე შესაძლებელია სმარტ ფილტრით მოქმედება.

ფენების სტილები

იმისათვის, რომ ფენას რაიმე მხატვრული ან თუნდაც რეალისტური ეფექტი შევძინოთ, ამისათვის შეგვიძლია სხვადასხვა სახის ეფექტები გამოვიყენოთ. ეფექტები შესაძლებელი გამოვიყენოთ როგორც ერთ ფენაზე, ისე ფენათა ჯგუფზე. ასევე ერთდროულად შესაძლებელია გამოვიყენოთ რამდენიმე ეფექტი ერთდროულად.

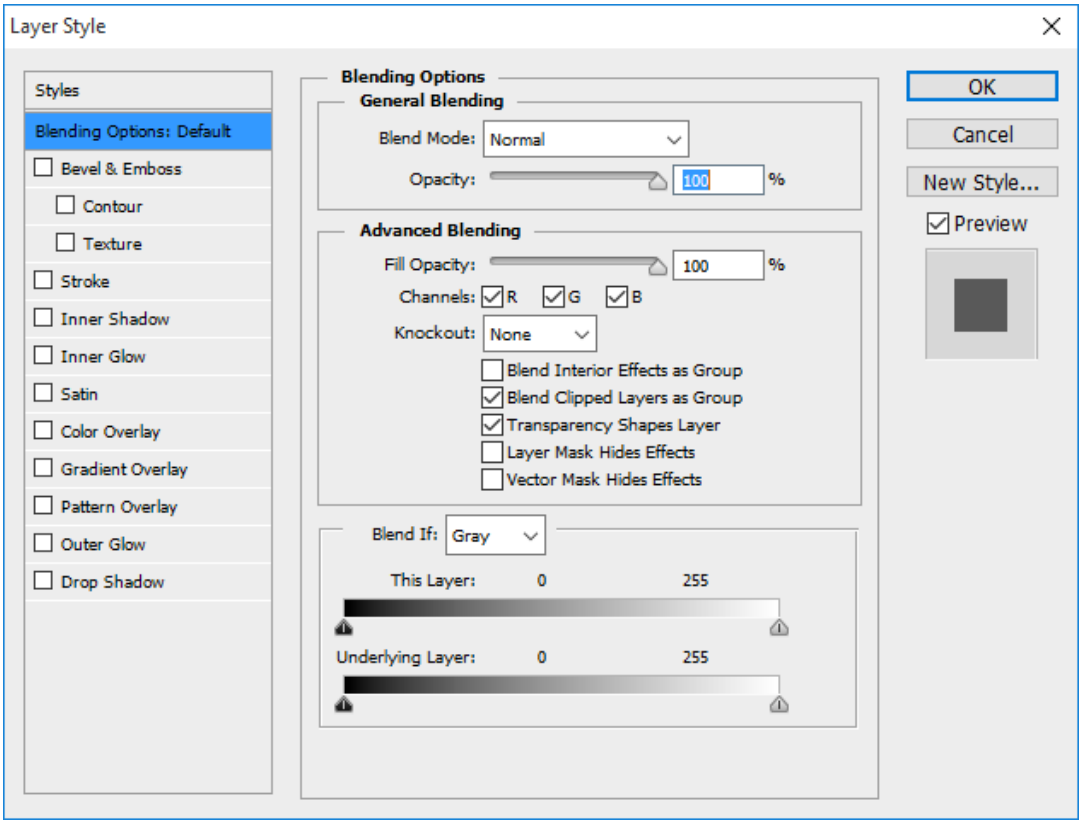
სტილი წარმოადგენს სხვადასხვა ეფექტის ერთობლიობას (ჩრდილი, ფაქტურა და ა.შ.)

სტილის მოქმედება ვრცელდება ფენაზე განთავსებულ ობიექტებზე მთლიანად. ამიტომ თუ ამავე ფენაზე დაემატება ახალი ელემენტი, მასზეც გავრცელდება ის სტილი, რაც ფენაზეა გამოყენებული.

თქვენ შეგიძლიათ გამოიყენოთ გამზადებული სტილები (Window -> Styles) ან შექმნათ საკუთარი. საკუთარი სტილები რომ შექმნათ, წინასწარ უნდა მონიშნოთ სასურველი ფენა და გამოიძახოთ:

Layer -> Layer Styles. შემდეგ მენიუდან აირჩიოთ თქვენთვის სასურველი სტილი.

ფენების მართვის პანელზე დააწვეთ ეფექტების ღილაკს (სურ. 8-68, ს) და გამოსული მენიუდან აირჩიეთ სასურველი სტილი.

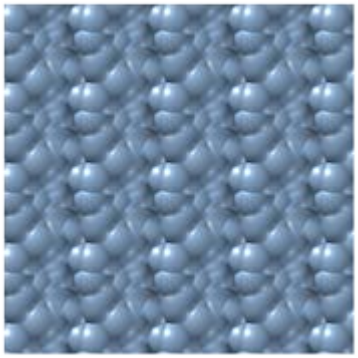


სურ. 8-70. ფენის სტილის (Layer Style) დიალოგური ფანჯარა. მისი დახმარებით თქვენ შეგიძლიათ ფენის გამჭვირვალობის, შერევის მეთოდებისა და ეფექტების მართვა/გამოყენება

შენიშვნა: *წინა ვერსიებისგან განსხვავებით, ამ ვერსიაში (და მომდევნოებში) ეფექტებისდალაგების მიმდევრობა შეცვლილია.პირველად გამოსახულებაზე გამოიყენება* ***დაცემული ჩრდილი*** *(Drop Shadow), ამიტომ ის ქვემოთ არისგანთავსებული, ხოლო ეფექტი* ***დაცერება & რელიეფი*** *(Bevel & Emboss) გამოიყენება ბოლოს, ამიტომ ის ზემოთ არის განთავსებული.*

სურ. 8-71. დაცემული ჩრდილი (Drop Shadow)

სურ. 8-72. გარე მნათობი კონტური (Outer Glow)



სურ. 8-73. ფაქტურის გადაკვრა (Pattern Overlay)

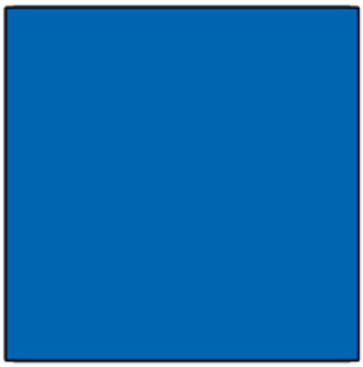
სურ. 8-74. გრადიენტის გადაკვრა (Gradient Overlay)

სურ. 8-75. ფერის გადაკვრა (Color Overlay)

სურ. 8-76. ატლასის გადაკვრა

სურ. 8-77. შიდა მნათობი კონტური (Inner Glow)

სურ. 8-78. შიდა ჩრდილი (Inner Shadow)



სურ. 8-79. კანტი (Stroke)

სურ. 8-80. დაცერება და რელიეფი (Bevel & Emboss)

1. მოიძიეთ სხვადასხვა სურათი, რომელზეც არეების მოსანიშნად გამოიყენებთ ინსტრუმენტებს: **ლასო**, **სწორხაზოვანი** ან **მაგნიტური ლასო**, **სწრაფი მონიშვნა**, მონიშვნა **გეომეტრიული ფორმების** მიხედვით, **ფერის დიაპაზონის** მიხედვით; ინსტრუმენტი სწორად შეარჩიეთ ამოცანის მიხედვით;მოახდინეთ მონიშნული არის **საზღვრების სრულყოფა** (Refine Edge).
2. ჩამოთვალეთ შერევის რეჟიმები;
3. დაფერეთ მონიშნული არე ახალი ფენის გამოყენებით;გამოიყენეთ **შერევის რეჟიმი** ფერის დასადებად გამოსახულებაზე.
4. შეარჩიეთ გამოსახულება, რომლის რედაქტირებასაც მოახდენთ შევსებით **შიგთავსის გათვალისწინებით** (Content Aware).
5. შექმენით ახალი **ფუნჯები**. შეინახეთ ისინი თქვენი საკუთარი ფუნჯების ნაკრების სახით.
6. მოიძიეთ კონტურული გამოსახულებები. მოახდინეთ მათი დაფერვა **ფუნჯის** დახმარებით და შესაბამისი **შერევის რეჟიმების** გამოყენებით;საჭიროების შემთხვევაში შექმენით ახალი ფუნჯი.
7. შეარჩიეთ რამდენიმე გამოსახულება, რომელზეც ფონს წაშლით **ჯადოსნური საშლელის** ან **ფონის საშლელის** დახმარებით.
8. ჩამოთვალეთ **ფენების სტილები**. შექმენით და შეინახეთ ახალი სტილი.
9. ჩამოთვალეთ ტრანსფორმაციის მეთოდები.
10. მოახდინეთ სხვადასხვა გამოსახულების კომბინირება. გამოიყენეთ ტრანსფორმაციის სხვადასხვა ინსტრუმენტები ფრაგმენტების მოსარგებად;გამოსახულების კომბინირებისას გამოიყენეთ **შერევის რეჟიმები;** საჭიროების შემთხვევაში გამოიყენეთ ფენების ეფექტები.
11. გამოსახულებას შეუცვალეთ ფორმა **დეფორმაციისა** და **მარიონეტული ტრანსფორმაციის** დახმარებით;
12. რა არის სმარტ-ობიექტი. მოახდინეთ გამოსახულებების კომბინირება ისე, რომ გამოიყენოთ მხოლოდ სმარტ-ობიექტები.
13. მოახდინეთ გამოსახულების ზომების შეცვლა შიგთავსის გათვალისწინებით (Conten Aware Scale).
14. შერჩეულ გამოსახულებებს შეუცვალეთ ზომები;შეცვალეთ **ტილოს** ზომა;**შემოჭრის** ინსტრუმენტის დახმარებით მოახდინეთ გამოსახულების შემოჭრა. ჩამოთვალეთ შემოჭრის მეთოდები.
15. შეარჩიეთ სურათი (გარჩევადობა 300 dpi) და მოახდინეთ მისი შემოჭრა ფბ-ს ან ტვიტერის მთავარი სურათის მახასიათებლების მიხედვით. შეინახეთ შესაბამის ფორმატში.

მიმდინარე პარაგრაფის თემატიკა

- ტექსტისთვის შრიფტის არჩევა, ზომების დაყენება, სიმბოლოებს შორის ინტერვალის რეგულირება, სტრიქონებს შორის მანძილის დაყენება, აბზაცების სწორების მართვა
- ტექსტისთვის სტილების შექმნა: აბზაცის სტილი (Paragraph Style)და სიმბოლოს სტილი (Character Style)
- ტექსტის გარდაქმნა ვექტორულ ან/და რასტრულ ობიექტად
- ტექსტის ტრანსფორმირების ინსტრუმენტები
- ტექსტური ფენისთვის ფენების სტილების (Layer Style)გამოყენება.

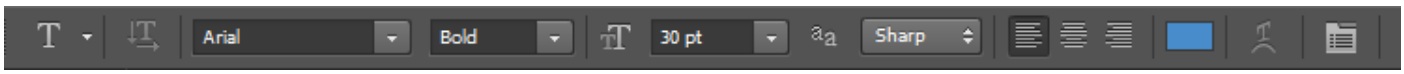
ტექსტთან სამუშაოდ გამოიყენება**ტექსტისინსტრუმენტი** (სურ. 8-10, ვ).

ორი მიმართულებით შეგვიძლია ტექსტის ბეჭდვა - ჰორიზონტალურად და ვერტიკალურად.

ასევე შეგვიძია 2 ნაირსახეობის ტექსტის შექმნა - ჩვეულებრივი ვექტორული სახის ტექსტი და მონიშნული არის სახით.

პირველის შემთხვევაში ჩვენ გვაქვს ტექსტის ვიზუალური მხარე, ხოლო მეორე შემთხვევაში ჩვენ გვაქვს მხოლოდ მონიშნული არე და ამ შემთხვევაში შესაბამისი მოქმედებების შესრულება შეგვიძლია (იხ. მონიშვნის ხელსაწყოები და მონიშვნა).

მაგრამ, როგორი სახითაც ჩვენ არ უნდა ვქმნიდეთ ტექსტს - ჩვეულებრივი თუ მონიშნული არის - ორივე შემთხვევაში ჩვენ შეგვიძლია შევასრულოთ მსგავსი მოქმედებები: შრიფტის არჩევა, მისი ზომა, სწორება და სხვა. ტექსტის ინსტრუმენტის არჩევის შემდეგ, ფორმატირების ზოლზე ჩვენ შესაბამის ფუნქციებს ვნახავთ, რომელიც დაგვეხმარება ტექსტის შექმნაში და შემდეგ ფორმატირებაში.



სურ. 8-81. ტექსტის ფორმატირების ზოლი.

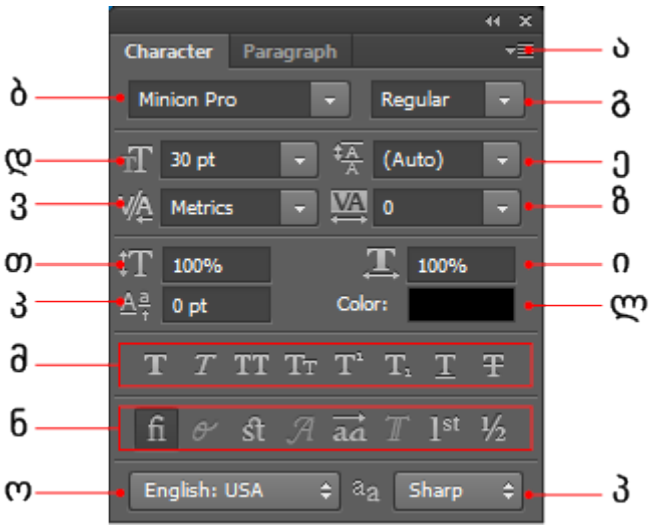
ფორმატირების ზოლიდან ჩვენ შეგვიძლია ავირჩიოთშრიფტი, მისი ზომა, სწორება, ფერი და სხვა.

შენიშვნა:*ტექსტის ფერი ნაგულისხმევად არის წინა ფონის ფერი.*

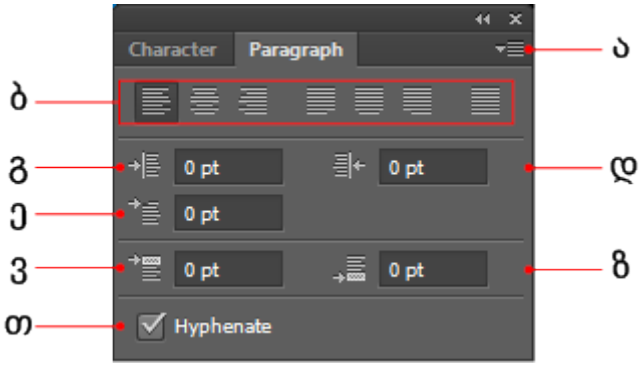
გარდა ფორმატირების ზოლისა, რომელზეც მხოლოდ ძირითადი ფუნქციებია გამოტანილი, არის კიდევ ორი პანელი, რომელიც ტექსტის ფორმატირებისთვის გამოიყენება:

Window -> Character - სიმბოლოს ფორმატირების პანელი;

Window -> Paragraph - აბზაცის ფორმატირების პანელი.



სურ. 8-82. ა - სიმბოლოს ფორმატირების პანელის მენიუ; ბ - შრიფტის არჩევა; გ - შრიფტის სტილის დაყენება (მუქი, დახრილი და ა.შ.); დ - შრიფტის ზომის დაყენება; ე - სტრიქონებს შორის მანძილი; ვ - კერნინგი; ზ - ინტერვალი; თ - სიმბოლოს მასშტაბირება (ვერტიკალურად); ი - თ - სიმბოლოს მასშტაბირება (ჰორიზონტალურად); კ - საბაზისო ხაზის მართვა; ლ - ტექსტის ფერი; მ - ცალკეული სიმბოლოების ფორმატირების ინსტრუმენტები; ნ - Open Type ტიპის შრიფტების ფორმატირების ინსტრუმენტები; ო - ენის არჩევა; პ - შრიფტის მოხაზულობის გასწორების მეთოდები



სურ. 8-83. ა - პარაგრაფის პანელის მენიუ; ბ - აბზაცის სწორების მეთოდები (მარჯვნივ, მარცხნივ, ცენტრზე და ა.შ.; გ - დაცილება მარცხენა კიდიდან; დ - დაცილება მარჯვენა კიდიდან; ე - პირველი სტრიქონის დაცილება მარცხნიდან; ვ - დაცილება აბზაცამდე; ზ - დაცილება აბზაცის შემდეგ; თ - გადატანები ტექსტისთვის.

ზემოთ ჩამოთვლილი პანელების დახმარებით თქვენ შეგიძლიათ მაქსიმალურად კარგად მოახდინოთ ნებისმიერი ზომის ტექსტის ფორმატირება.

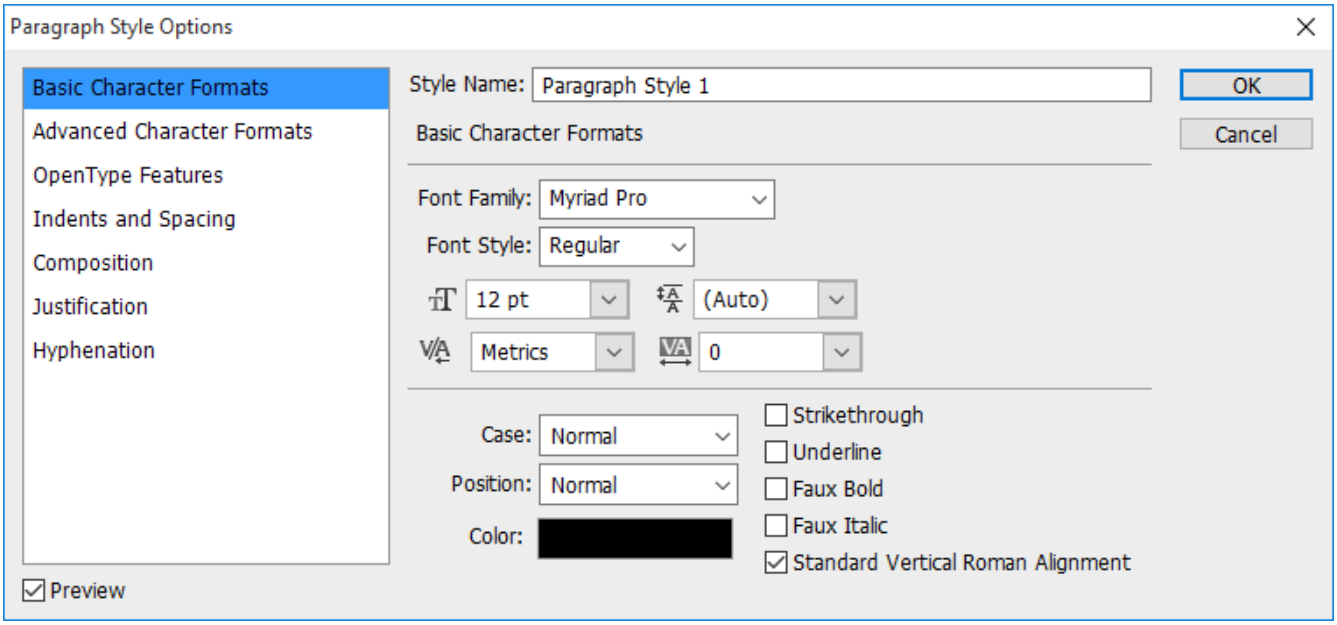
მუშაობის პროცესში არის ისეთი მომენტები, როდესაც თქვენ ხშირად გჭირდებათ ერთი და იმავე ფორმატირების გამოყენება ტექსტისთვის, მაგალითად ბანერზე ტექსტური წარწერები ან სათაურები, ცალკეული სიმბოლოს ფორმატი და სხვა.

ამისათვის თქვენ შეგიძლიათ ისარგებლოთ აბზაცისა და სიმბოლოს სტილებით. სტილები, ისევე როგორც ინდიზაინში, გაძლევენ საშუალებას სწრაფად მოახდინოთ ტექსტის ფორმატირება. მასში უკვე ჩადებულია ფორმატირების ის მახასიათებლები, რომლებიც თქვენ გამოიყენეთ აბზაცისთვის თუ ცალკეული სიმბოლოსთვის.

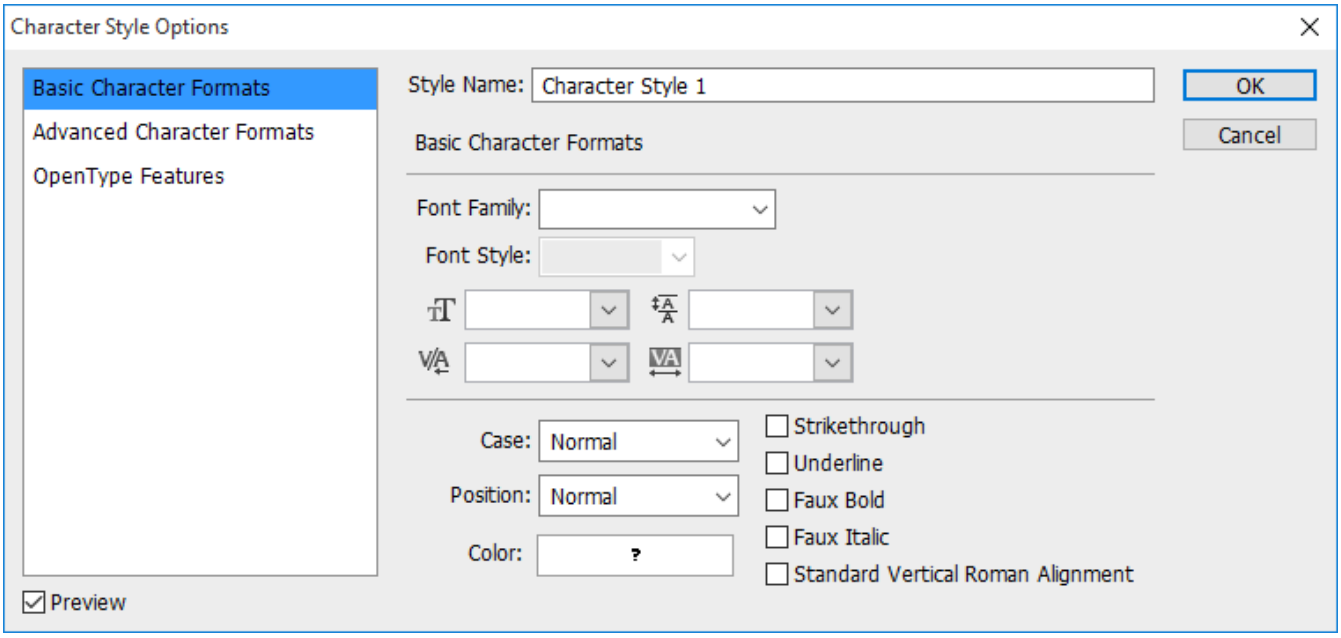
Window -> Paragraph Style - აბზაცის სტილების პანელი;

Window -> Character Style - სიმბოლოების სტილების პანელი.

უნდა გაითვალისწინოთ, რომ აბზაცის სტილი მოქმედებს მთლიანად აბზაცზე, ხოლო სიმბოლოს სტილი ვრცელდება მხოლოდ ერთ კონკრეტულ სიმბოლოზე (ის წინასწარ უნდა მონიშნოთ).



სურ. 8-84. ახალი აბზაცის სტილის შექმნის ფანჯარა. აქ შეგიძლიათ სტილს მიანიჭოთ დასახელება, დააყენოთ შრიფტი, ზომა, ფერი და ყველა ის მახასიათებელი რაც სიმბოლოსა და აბზაცის ფორმატირების პანელებზე იყო მოცემული. შემდეგ საკმარისია მონიშნოთ ტექსტი აირჩიოთ სტილი და ის ავტომატურად მიიღებს შესაბამის სახეს.



სურ. 8-85. სიმბოლოს სტილის შექმნა. აქ თქვენ გამოიყენებთ მხოლოდ სიმბოლოს ფორმატირების პანელის მახასიათებლებს. შეგიძლიათ სტილს დაარქვათ სახელი, აირჩიოთ შრიფტი, დააყენოთ ზომა, ფერი და ა,შ. შემდეგში ეს სტილი გავრცელდება მხოლოდ იმ სიმბოლოზე ან სიტყვაზე, რომელსაც მონიშნავთ. მაგალითად, როდესაც გინდათ, რომ მთელი აბზაცის მასშტაბით ერთი კონკრეტული სიტყვა იყოს ყოველთვის წითელი, ასეთ შემთხვევაში სიმბოლოების სტილი შეუცვლელია.

ტექსტი ავტომატურად თავსდება ახალ ფენაზე და თავისი მახასიათებლებით ის ვექტორული ობიექტია. საჭიროების შემთხვევაში თქვენ შეგიძლიათ მისი რედაქტირება, შრიფტისა და ზომის შეცვლა. მაგრამ ზოგიერთი მოქმედება (მაგ. ფილტრების მოქმედება, მისი ფუნჯით ან სხვა მსგავსი ინსტრუმენტით დამუშავება) შეუძლებელია და საჭირო ხდება მისი გარდაქმნა რასტრულ გამოსახულებად:

Layer -> Rasterize -> Type;

Type -> Rasterize Type Layer;

გარდა ამისა შეგიძლიათ ტექსტი ვექტორულ ფორმად გარდაქმნათ. ამის შემდეგაც ტექსტის რედაქტირება შეუძლებელი იქნება, თუმცა ის ვექტორულ ობიექტად დარჩება: Type -> Convert to Shape

ტექსტის ტრანსფორმაცია

ტექსტის აკრეფისას ფორმატირების ზოლზე არის ღილაკი  . მისი დახმარებით შეგიძლიათ ტექსტის ტრანსფორმირება და მისთვის სხვადასხვა ფორმის მინიჭება. ეს ფუნქცია ანალოგიურია ფუნქციის**დეფორმაცია**(Warp), რომელიც ობიექტების ტრანსფორმაციისას გამოიყენება. განსხვავება მხოლოდ ისაა, რომ აქ მხოლოდ გამზადებული შაბლონების მიხედვით ახდენთ მასზე მოქმედებას.

ზოგადად ტექსტის (როგორც ობიექტის) ტრანსფორმირება შეგიძლიათ ჩვეულებრივი ობიექტის მსგავსად - შეცვალოთ მისი ზომა, ფორმა, შეატრიალოთ სარკულად ჰორიზონტალურად თუ ვერტიკალურად და ა.შ.

ტრანსფორმირების ფუნქციები იხ.ტრანსფორმაციის ინსტრუმენტები.

ფენების სტილები:

რადგან ტექსტი ცალკე ფენაზე თავსდება, მასზე შესაძლებელია ფენების სტილების გამოყენება ისევე, როგორც სხვა ობიექტებზე. ფენების სტილების შესახებ იხ. ფენების სტილები.

პრაქტიკული დავალებები/კითხვები თვითშეფასებისათვის

1. ჩამოთვალეთ ტექსტისა და აბზაცის რედაქტირების პარამეტრები;
2. შექმენით აბზაცისა და სიმბოლოების სტილები.
3. შექმენით გამოსახულებისთვის წარწერა. მოახდინეთ მისი ფორმატირება. ვიზუალური გაფორმებისთვის გამოიყენეთ **ტექსტის ტრანსფორმაცია** და/ან **ფენის სტილები**.
4. შექმენით ტექსტი. მოახდინეთ მისი გადაყვანა რასტრში და გამოიყენეთ სხვადასხვა ინსტრუმენტები და ბრძანებები მის ვიზუალურად გასაფორმებლად.
5. გამოსახულებაზე შექმენით მონიშნული არე ტექსტის სახით. მოახდინეთ მისი **დაფერვა** ან სხვადასხვა ეფექტების გამოყენებით ვიზუალურად გააფორმეთ.
6. შეარჩიეთ სასურველი წიგნის დასახელება და შექმენით ყდის წარწერა (ქართულად). შექმნისას გამოიყენეთ შესაბამისი ინსტრუმენტები.

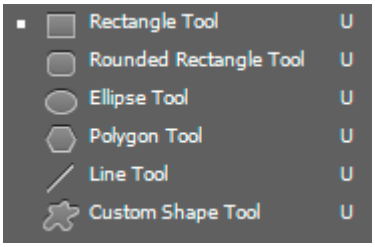
მიმდინარე პარაგრაფის თემატიკა

- მარტივი ვექტორული ობიექტების შექმნა და მათთან მუშაობა.
- ვექტორული ობიექტების შექმნა კალმის (Pen) გამოყენებით. თავისუფალი კალმის (Freeform Pen)გამოყენება, საკვანძო წერტილების დამატება წაშლა. მოქმედებები საკვანძო წერტილებზე.
- ვექტორული ობიექტის მთლიანი მონიშვნა. ცალკეული წერტილების მონიშვნა.

ვექტორული ობიექტი პროგრამაში ორი გზით შეიძლება შეიქმნას - გეომეტრიული ფორმებისა (Shape) და კალმის (Pen)გამოყენებით. ორივე ინსტრუმენტთა პანელზეა განთავსებული და მიეკუთვნება მოხაზულობის ინსტრუმენტებს (იხ. სურ. 8-10, ვ).

გეომეტრიული ფიგურები

გეომეტრიული ფორმების შესაქმნელად მივმართავთ **მართკუთხედის ინსტრუმენტს** (Rectangle Tool) (U). ჯგუფში შედის კიდევ დამატებითი ინსტრუმენტები:



სურ. 8-86. ვექტორული ფორმები: მართკუთხედი (Rectangle), მართკუთხედი მომრგვალებული კუთხეებით (Rounded Rectangle), ელიფსი (Ellipse), მრავალკუთხედი (Polygon), ხაზი (Line) და მორგებული ფორმა (Custom Shape)

ინსტრუმენტის არჩევის შემდეგ, ფორმატირების ზოლზე გამოტანილი პარამეტრების დახმარებით ჩვენ შეგვიძლია განვსაზღვროთ ფორმის შევსებისა და კონტურის ფერები, მივუთითოთ კონტურის სისქე და ნაირსახეობა, ფორმის სიგანე ან/და სიმაღლე.

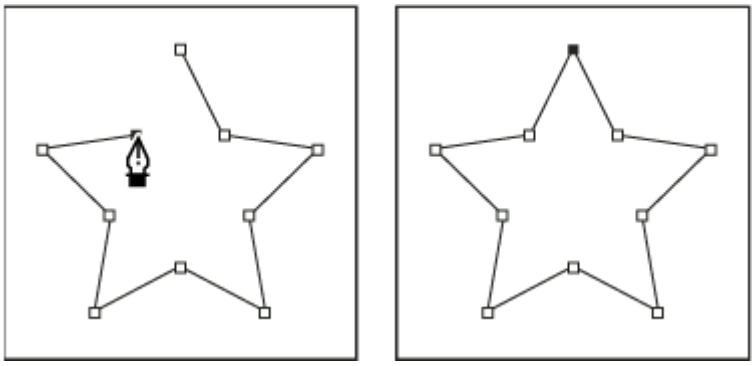
შევსებისთვის და კონტურისთვის შეგვიძლია სხვადასხვა გავაკეთოთ:

- შეგვიძლია შევსება ან კონტური საერთოდ მოვხსნათ;
- გამოვიყენოთ მხოლოდ ფერი (იხსნება ფერთა პალიტრა);
- გამოვიყენოთ გრადიენტი (იხსნება გრადიენტის პალიტრა. იხ. გრადიენტის ინსტრუმენტი (Gradient Tool));
- გამოვიყენოთ ფაქტურა (იხსნება ფაქტურების პალიტრა);
- ან ფერთა პალიტრიდან ავარჩიოთ ფერი, რომელის ფერთა ნიმუშებში არ არის მოცემული.

შექმნილი ობიექტი თავსდება ახალ ფენაზე.შეგიძლიათგამოიყენოთტრანსფორმაციის ბრძანებები (იხ. ტრანსფორმაციის ინსტრუმენტები) ან გამოვიყენოთ სტილები (იხ. ფენების სტილები).

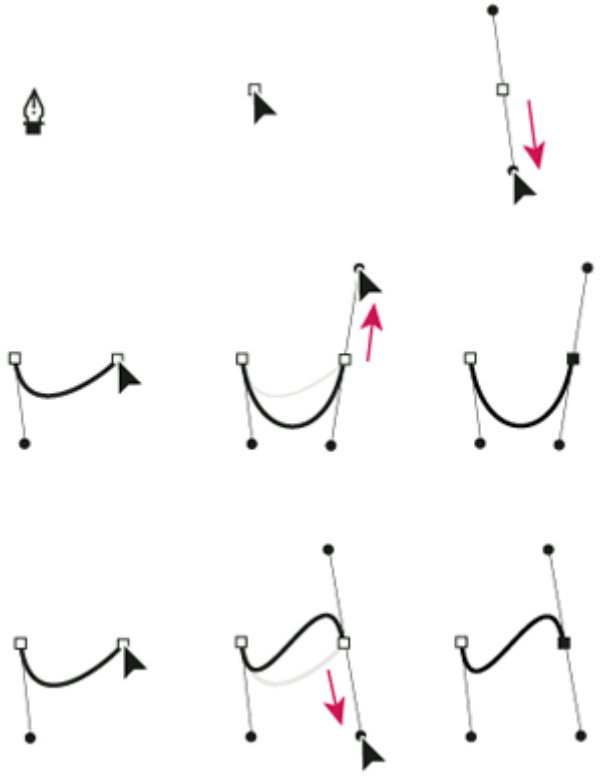
სასურველი ფორმების შექმნა (Pen Tool და Freeform Pen)

კალმის (Pen) ინსტრუმენტით შესაძლებელია როგორ უბრალოდ ფორმების შექმნა, ის გამოსახულების კონტურებზე შემოტარება.ინსტრუმენტს იყენებთ ისევე, როგორც სწორხაზოვან ლასოს (Polygonal Lasso) - უნდა დასვათ წერტილები და ბოლო წერტილი შეაერთოთ პირველ წერტილთან (სურ. 8-88).



სურ. 8-87. კალმის (Pen) ინსტრუმენტის გამოყენება.

ინსტრუმენტის გამოყენება არ შემოიფარგლება მხოლოდ სწორი წირების გავლებით.შეგიძლიათ მრუდი წირების გავლებაც. ამისათვის მომდევნო წერტილის დასმისას ნუ აუშვებთ მაუსის ღილაკს, გეჭიროთ და ისე ამოძრავებთ. დაინახავთ, რომ საკვანძო წერტილს გაუჩნდება ორი, ერთმანეთის საპირისპირო ხაზები. ამ ხაზებზე მოქმედებით თქვენ მოქმედებთ საკვანძო წერტილზე და ცვლით წირის მოხაზულობას. წირს შეიძლება ნებისმიერი ფორმა მიანიჭოთ.



სურ. 8-88. მრუდი წირის შექმნა.

ინსტრუმენტი თავისუფალი კალამი (Freeform Pen) მუშაობს, როგორც ჩვეულებრივი ლასო. უნდა შემოავლოთ ის გამოსახულებაზე და ინსტრუმენტი თვითონ დასვამს საკვანძო წერტილებს.

ინსტრუმენტი **საკვანძო წერტილის დამატება** (Add Anchor Point) გაძლევთ საშუალებას დაამატოთ წირზე (სწორხაზოვანზე ან მრუდზე) საკვანძო წერტილები ნებისმიერი რაოდენობის.

ინსტრუმენტი **საკვანძო წერტილების წაშლა** (Delete Anchor Point) გაძლევთ საშუალებას წაშალოთ წირზე (სწორხაზოვანზე ან მრუდზე) არსებული საკვანძო წერტილები.

დაიმახსოვრეთ:*რაც უფრო ნაკლები წერტილისგან შედგება წირი, მით ნაკლებად ტეხილი გამოვა.*

წირის ხაზვისას თქვენ შეგიძლიათ შექმნათ მრუდი, მაგრამ როდესაც საჭირო გახდება უკვე არსებული სწორხაზოვანი მონაკვეთის გამრუდება, ამაში დაგეხმარებათ კუთხის ინსტრუმენტი (Convert Point). ის

კალმის ინსტრუმენტთან ერთ ჯგუფში იმყოფება. მისი დახმარებით თქვენ მოქმედებთ საკვანძო წერტილზე, რის შედეგადაც თქვენ მოქმედებთ მთლიანად წირზე - ამრუდებთ, უცვლით ფორმას და ა.შ.

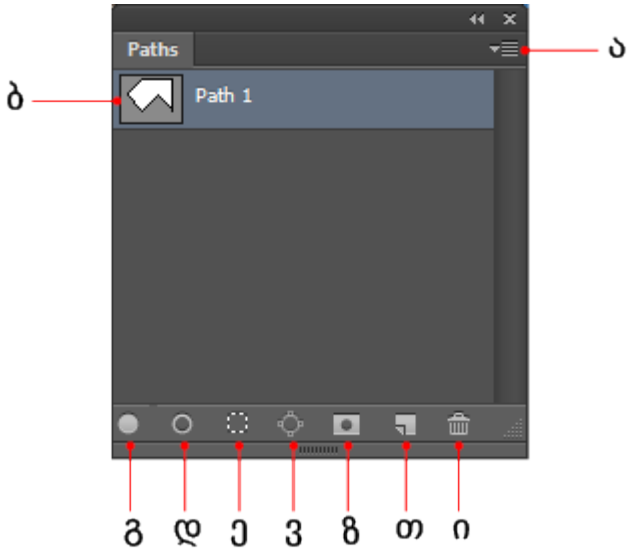
უკვე არსებულ წირის (თავისუფალი ფორმის წირი იქნება თუ გეომეტრიული ფორმა) მონიშვნა ჩვეულებრივი ინსტრუმენტებით ვერ ხერხდება. ამისათვის ინსტრუმენტთა პანელზე მოცემულია ორი ინსტრუმენტი: **კონტურის მონიშვნა** (Path Selection) და **ისარი** (Direct Selection).

პირველი ახდენს მთლიანი კონტურის მონიშვნას. მისი დახმარებით ასევე შესაძლებელი კონტურის გადაადგილება ტილოს ფარგლებში.

მეორე ინსტრუმენტის დახმარებით თქვენ ნიშნავთ საკვანძო წერტილს, რომელზეც გინდათ შემდეგ დამატებითი მოქმედებების შესრულება.

კონტურთან მუშაობისას ეს ორი ინსტრუმენტი შეუცვლელია.

კონტურთან სამუშაოდ შესაბამისი პანელის გამოძახება ხდება Window -> Path. მისი დახმარებით ჩვენ შეგვიძლია დამატებითი მოქმედებები შევასრულოთ.



სურ. 8-89. კონტურის პანელი (Path). ა - პანელის მენიუ; ბ - გამოსახულებაზე შექმნილი კონტური. რამდენი დამოუკიდებელო კონტური შეიქმნება, იმდენი აღმნიშვნელი იქნება აქ. კონტურებს შეგიძლიათ სახელებიც შეუცვალოთ; გ - კონტურის ფერით შევსება; დ - კონტურის მიხედვით კანტის შექმნა; ე - კონტურის გარდაქმნა მონიშნულ არედ; ვ - მონიშნული არის გარდაქმნა კონტურად; ზ - კონტურისგან გამოსახულების ნიღაბის მიღება; თ - ახალი სივრცის შექმნა, რომელზეც კონტური განთავსდება; ი - კონტურის წაშლა.

პრაქტიკული დავალებები/კითხვები თვითშეფასებისათვის

1. შექმენით სხვადასხვა სახის გეომეტრიული ფიგურები. გამოიყენეთ მათთვის ტრანსფორმაციის ინსტრუმენტები და ფენის სტილები.
2. კალმის გამოყენებით გამოსახულების რომელიმე ობიექტის მიხედვით შექმენით ვექტორული ფორმა. შემდეგ ვექტორული ფორმა აქციეთ მონიშნულ არედ და მოახდინეთ მისი რედაქტირება.
3. ვექტორის შექმნისა და რედაქტირების ინსტრუმენტების დახმარებით, მარტივი გეომეტრიული ფიგურა აქციეთ რთულ ფორმად.

გამოყენებული ლიტერატურა

1. http://www.w3schools.com/
2. Robin Nixon - HTML5 20 Lessons to Successful Web Development (ENG): https://goo.gl/e9mhwn
3. OReilly - CSS3 The Missing Manual, 3rd Edition (ENG):https://goo.gl/rmzYCo
4. OReilly - CSS3 The Missing Manual, 3rd Edition (RUS):https://goo.gl/mIgh4G
5. OReilly - Bootstrap Responsive Web Development (ENG):https://goo.gl/lU4wru
6. OReilly - Speaking JavaScript (ENG): https://goo.gl/MkQ0hP
7. Николас Закас - JavaScript для профессиональных веб-разработчиков, 3-е издание (RUS): https://goo.gl/FEfPY7
8. OReilly - JavaScript and jQuery The Missing Manual 3rd Edition (ENG):https://goo.gl/1bEHK8
9. McGraw-Hill - JavaScript A Beginner's Guide (ENG):https://goo.gl/6PY6jx
10. McGraw-Hill - jQuery A Beginner's Guide (ENG): https://goo.gl/iYNSJd
11. Andy Harris - JavaScript & AJAX For Dummies (ENG):https://goo.gl/XmItxb
12. Бенкен Е., Самков Г. - AJAX. Программирование для Интернета (RUS): https://goo.gl/sfbbb5
13. Ben Smith - Beginning JSON (ENG): https://goo.gl/2KqSjF