

თეიმურაზ სტურუა, თეა თოდუა,
ბესიკ ტაბატაძე

ვებტექნოლოგიების საფუძვლები

HTML



&

CSS



(მეორე გამოცემა)



საგამომცემლო სახლი
„ტექნიკური უნივერსიტეტი“

საქართველოს ტექნიკური უნივერსიტეტი

თეიმურაზ სტურუა, თეა თოდუა,
ბესიკ ტაბატაძე

ვებტექნოლოგიების საფუძვლები (HTML5 & CSS3)

(მეორე გამოცემა)



დამტკიცებულია სახელმძღვანელოდ
საქართველოს ტექნიკური უნივერსიტეტის
სარედაქციო-საგამომცემლო საბჭოს
მიერ. 07.07.2021, ოქმი №1

თბილისი
2021

უაკ 681.3

განხილულია ენა HTML-ის შესაძლებლობები, HTML5 ვერსიაში შემოღებული სიახლეებისა და დამატებების გათვალისწინებით, HTML-ის ელემენტარული საფუძვლებიდან დაწყებული რთული პრაქტიკული საკითხებით დამთავრებული. მოცემულია CSS3 ვებსაიტების დამუშავების უახლესი სტანდარტები, რომლებიც ვებდაპროგრამების ენების ფუნქციურ შესაძლებლობებს მნიშვნელოვნად აუმჯობესებს და ინტერნეტ-პროექტებისთვის ორიგინალური ვიზუალური გადაწყვეტის საშუალებას იძლევა.

განკუთვნილია ინფორმატიკის საბაკალავრო პროგრამის სტუდენტებისა და HTML და CSS ენების გამოყენებით ვებგვერდების შექმნით დაინტერესებული სხვა პირებისთვის.

რეცენზენტები: საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მარტის სისტემების ფაკულტეტის პროფესორი გელა ღვინეფაძე,

საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მარტის სისტემების ფაკულტეტის პროფესორი რომან სამხარაძე

© საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 2021

ISBN 978-9941-0-28-756-5

<http://www.gtu.ge/publishinghouse/>

ყველა უფლება დაცულია. ამ წიგნის არც ერთი ნაწილის (იქნება ეს ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) გამოყენება არანაირი ფორმით და საშუალებით (იქნება ეს ელექტრონული თუ მექანიკური), არ შეიძლება გამომცემლის წერილობითი ნებართვის გარეშე.

საავტორო უფლებების დარღვევა ისჯება კანონით.

წიგნში მოყვანილი ფაქტების სიზუსტეზე პასუხისმგებელია ავტორი/ავტორები.

ავტორის/ავტორთა პოზიციას შეიძლება არ ემთხვეოდეს საგამომცემლო სახლის პოზიცია



Verba volant,
scripta manent

შ ი ნ ა ა რ ს ი

შესავალი.....	11
თავი 1	13
HTML დოკუმენტის სტრუქტურა, აბზაცები და სათაურები..	13
HTML დოკუმენტის შექმნა	14
HTML დოკუმენტის სტრუქტურა.....	15
lang ატრიბუტი	17
<Body> ტეგის ატრიბუტები.....	18
კომენტარები	20
ფერები	20
სათაურის დონეები	22
ფონური სურათის ჩასმა <Body> ტეგისთვის	27
აბზაცი და სტრიქონების წყვეტა	30
თავი 2	43
შრიფტის დაფორმატება; ჰორიზონტალური ხაზები. სიები...	43
შრიფტის დაფორმატების ტეგები.....	44
ტექსტის დაფორმატება.....	46
მონოსიგანის შრიფტების გამოყენება.....	48
აკრონიმი.....	52
HTML სიმბოლოები	53
ჰარების დასმა	54
ჰორიზონტალური ხაზები	57

სიები.....	59
დანომრილი სიები.....	59
მარკირებული სიები.....	62
განსაზღვრებათა სია.....	65
თავი 3	71
გრაფიკული გამოსახულება HTML დოკუმენტში.....	71
გრაფიკული გამოსახულების ჩასმა Html დოკუმენტში	72
გამოსახულების მასშტაბირება.....	78
ტეგი <picture>	80
გამოსახულების რუქა (გამოსახულების დაშლა არეებად).....	82
თავი 4	91
ცხრილები.....	91
ცხრილების შექმნის პრინციპები	92
ცხრილის სტრიქონებისა და სვეტების შექმნა.....	93
მარტივი და რთული ცხრილების შექმნა	95
თავი 5	109
ბმულები.....	109
ბმულები	110
ბმულების დოკუმენტში ჩასმა.....	110
შიგა ბმულები	111
გარე ბმულები.....	112
დოკუმენტიდან დოკუმენტზე გადასვლა	112
დოკუმენტის შიგნით გადასვლა	113

დოკუმენტიდან არსებულ Web-გვერდზე გადასვლა ...	120
ტეგი <frames>.....	123
თავი 6	127
ფორმები.....	127
ფორმების შექმნის პრინციპები	128
მართვის ელემენტების ფორმაზე განთავსება.....	132
Input ტეგის ახალი ატრიბუტები	142
ლინკის შექმნა	150
მულტიმედია.....	154
ფაილების ფორმატი და კოდირების ფორმატები.....	154
MIME ტიპები.....	157
აუდიო-რგოლის ჩასმა.....	159
ვიდეო-რგოლის ჩასმა	161
თავი 7	165
CSS3	165
CSS კოდის HTML დოკუმენტთან დაკავშირება	166
style ატრიბუტი	166
<style> ტეგი	170
სტილების გარე ცხრილის გამოყენება	175
HTML ფერები	179
ფერთა სახელები	179
ფერთა მნიშვნელობები.....	181

ფერის RGB მნიშვნელობა	182
ფერის HEX მნიშვნელობა	184
ფერის HSL მნიშვნელობა	186
Saturation გაჯერებულობა	187
Lightness სიკაშკაშე	188
ფერის RGBA მნიშვნელობა	190
ფერის HSLA მნიშვნელობა	191
თავი 8	193
ფონური გამოსახულება, ტექსტის დაფორმატება და შრიფტი CSS3-ში.....	193
ფონური გამოსახულება CSS-ში.....	194
ტექსტის დაფორმატება CSS-ში.....	207
შრიფტები	219
თავი 9	227
ჩარჩო CSS3-ში	227
ჩარჩოს ტიპები (border-style).....	228
ჩარჩოს სისქე (border-width).....	231
ჩარჩოს ფერი (border-color).....	231
CSS3-ის border-radius თვისება	231
ელემენტის საზღვრის ველების (margin) განსაზღვრა.....	240
ჩარჩოსა და ელემენტს შორის დაშორების განსაზღვრა....	241
ელემენტების სიგანე და სიმაღლე	243
თავი 10	247

გრადიენტი CSS3-ში.....	247
ფერთა გადასვლა (გრადიენტი) CSS3-ში	248
ხაზოვანი გრადიენტი	248
გრადიენტის მიმართულების კონტროლი კუთხეების გამოყენებით	250
გამჭვირვალობის გამოყენება გრადიენტებში.....	253
რადიალური გრადიენტი ფერების თანაბარი განაწილებით	256
რადიალური გრადიენტი ფერების არათანაბარი განაწილებით	258
თავი 11.....	263
ჩრდილის ეფექტები და გადასვლები CSS3-ში	263
ჩრდილის ეფექტები CSS3-ში	264
ჩრდილის ეფექტები ტექსტისთვის	264
ჩრდილის რამდენიმე ეფექტის ერთდროული გამოყენება	267
ჩრდილის ეფექტი სხვადასხვა ელემენტისთვის.....	268
გადასვლები CSS3-ში	271
ობიექტის რამდენიმე თვისების ერთდროული ცვლილება	273
გადასვლის სიჩქარე.....	273
გადასვლის ეფექტი დაყოვნებით.....	274
გადასვლის ეფექტი და გარდაქმნა.....	275

თავი 12	281
გამჭვირვალობა, ელემენტების პოზიციონირება და ფენები	
CSS3.....	281
CSS-ში სურათის გამჭვირვალობა	282
ელემენტების სასურველ ადგილას განთავსება	286
ელემენტების პოზიციონირება.....	288
ფარდობითი პოზიციონირება.....	291
ფენების შექმნა Z-ინდექსის საშუალებით.....	299
თავი 13	307
ანიმაცია CSS3-ში.....	307
ანიმაცია CSS3-ში	308
დაყოვნება ანიმაციაში.....	311
ანიმაციის შესრულების მიმართულება.....	313
ანიმაციის შესრულების სიჩქარე.....	316
ანიმაციის შესრულების შემოკლებული ჩანაწერი.....	318
თავი 14	321
CSS ტრანსფორმაციები.....	321
CSS ტრანსფორმაციები.....	322
2D ტრანსფორმაციები	322
3D ტრანსფორმაციები	333
თავი 15	343
განლაგებები Flexbox.....	343
Flexbox კონტეინერი	344

Flexbox კონტეინერში მოთავსებული ელემენტები	351
თავი 16.....	359
მედიამოთხოვნები.....	359
CSS3 Media Queries (მედიამოთხოვნები)	360
მედიამოთხოვნების ზოგიერთი მაგალითი.....	362
ადაპტირებული Flexbox	371
HTML-ის სემანტიკური ტეგები.....	377
სემანტიკური ტეგები HTML-ში	378
ტეგი <section>	379
ტეგი <article>	380
ტეგი <header>.....	384
ტეგი <footer>	386
ტეგი <nav>	387
ტეგი <aside>.....	388
ტეგები <figure> და <figcaption>	391
დანართი 1. ძირითადი მათემატიკური სიმბოლოების	
დასახელება, მათი ათობითი და თექვსმეტობითი კოდები.....	395
დანართი 2. HTML-ში სემანტიკური ტეგების სია.....	398
ლიტერატურა:.....	400

შესავალი

HTML (Hyper Text Markup Language) ვებგვერდების აგებისა და მისი სტრუქტურის შემუშავების ტექნოლოგიაა.

ტიმ ბერნერს ლიმ 1989 წელს შეიმუშავა ვებგვერდების შექმნის სტრუქტურა და საფუძველი დაუდო HTML ჰიპერტექსტური მარკირების ენას. ამჟამად ტიმ ბერნერს ლი World Wide Web Consortium-ის (W3C) ხელმძღვანელია. ეს ორგანიზაცია ვებტექნოლოგიების სტანდარტების შემუშავებასა და განვითარებას უზრუნველყოფს.

HTML არ არის დაპროგრამების ენა, მისი მეშვეობით ვებგვერდის სტრუქტურის აწყობა და შემუშავება ხდება. HTML მარტივი და ასათვისებლად ადვილი ტექნოლოგიაა. HTML-ის შესასწავლად ერთ-ერთ პოპულარულ ტექსტურ რედაქტორ Notepad++-ს გამოვიყენებთ.

HTML ფაილი ტეგების სახით წარმოდგენილი ტექსტური ინფორმაციაა. HTML ფაილს html ან htm გაფართოება აქვს. ამ ფაილის შექმნა მარტივი ტექსტური რედაქტორებით არის შესაძლებელი. უკვე დამუშავებული და ჩამოყალიბებული HTML ფაილის ნახვა ნებისმიერი Web ბრაუზერით ხდება. არსებობს შემთხვევები, როდესაც ზოგიერთი ფაილი ბრაუზერის კონკრეტულ ვერსიას მოითხოვს.

მიუხედავად მრავალი სიახლისა, თანამედროვე ვებსაიტების აგება მხოლოდ HTML ტექნოლოგიის საშუალებით ვერ ხერხდება, რადგანაც HTML-ს ინსტრუმენტების საკმაოდ შეზღუდული ნაკრები აქვს, რომლის საშუალებითაც ვებდიზაინერის ზოგიერთი ჩანაფიქრის განხორციელება შეუძლებელია.

CSS (Cascading Style Sheets) სტილების კასკადური ცხრილები HTML-ის დამატებაა და მის შესაძლებლობებს მნიშვნელოვნად აფართოებს. CSS ვებდიზაინერს საშუალებას აძლევს ვებგვერდი უფრო ეფექტური და, დიზაინის თვალსაზრისით, ადვილად მართვადი გახადოს.

CSS3 სტილების კასკადური ცხრილების ევოლუციური გადაწყვეტაა და მისი წინა ვერსიების შესაძლებლობების გაფართოებისკენაა მიმართული.

მოცემული სახელმძღვანელო 16 თავისგან შედგება.

ყოველი თავის დასაწყისში მოცემულია თემის დასახელება და მასში შემავალი საკითხები. ყოველ თავში განხილულია როგორც თეორიული საკითხები, ისე პრაქტიკული მაგალითები. თითოეული ამოცანის შესრულების თანამიმდევრობა დეტალურადაა აღწერილი. თვითშემოწმების მიზნით ყოველი თავის ბოლოს განხილული თემის მიხედვით მოცემულია დავალება.

თავი 1

HTML დოკუმენტის სტრუქტურა, აბზაცები და სათაურები

- HTML დოკუმენტის შექმნა, შენახვა და გახსნა;
- HTML დოკუმენტის სტრუქტურა;
- ტეგები და მათი ატრიბუტები;
- ფერები;
- სათაურები HTML დოკუმენტში;
- აბზაცები და სტრიქონების წყვეტა.

HTML დოკუმენტის შექმნა

უპირველეს ყოვლისა, ცალკე საქალაღდე შევქმნათ მომავალი ვებგვერდისთვის. შემდეგ გავხსნათ Notepad++ და File ⇒ New ბრძანებით მასში გავხსნათ ახალი ფურცელი. ვიდრე მუშაობას შევუდგებოდეთ, შეგვიძლია დოკუმენტის შენახვაც. ამისათვის შევასრულოთ File ⇒ Save ან File ⇒ Save As ბრძანება.

ვებსაიტი ერთმანეთთან დაკავშირებული რამდენიმე ფაილისგან შედგება, ეს ფაილები შეიძლება საკმაოდ ბევრი იყოს. მათი ინტერნეტ სივრცეში განსათავსებლად აუცილებელია გვეჩონდეს ჰოსტინგი ანუ მეხსიერების ნაწილი, რომელშიც ჩაიწერება ჩვენს ვებგვერდზე განსათავსებელი ინფორმაცია. ვებჰოსტინგის ადმინისტრატორთა უმეტესობა გვთავაზობს ფაილების ატვირთვის მარტივ და მოსახერხებელ მეთოდებს, ჩვეულებრივ, ფაილების გადატანის გზით. ისმის კითხვა: როგორ უნდა აღიქვას ბრაუზერმა, რომელი გვერდი ჩამოტვირთოს პირველად? ყველა ვებბრაუზერისათვის მიღებულია, რომ საწყისი გვერდის სახელწოდება იყოს index. ამის გამო, ფაილის დამახსოვრებისას File name ველში ჩავწერთ index.html. ფაილის სახელი გაფართოებასთან ერთად უნდა დაიწეროს, წინააღმდეგ შემთხვევაში ის სისტემის მიერ ნაგულისხმევი წესით შეინახება, როგორც ჩვეულებრივი .txt გაფართოების ტექსტური ფაილი. როგორც კი ფაილი .htm (.html) გაფართოებით ჩაიწერება, მისი ნიშნაკი დაუყოვნებლივ შესაბამისი ბრაუზერის ნიშნაკით შეიცვლება.

ფაილის სახელის შერჩევის შემდეგ ყურადღება უნდა მივაქციოთ კოდირების სისტემას. თუ HTML დოკუმენტში ვებგვერდზე გამოსატანად ქართული ტექსტია გამოყენებული,

მაშინ Encoding ჩამოშლად მენიუში Encode in UTF-8 უნდა მოვნიშნოთ და ისე შევინახოთ.

თუ კომპიუტერში არ გვაქვს Notepad++ გადმოწერილი, ის შეგვიძლია გადმოვწეროთ ინტერნეტიდან (<https://notepad-plus-plus.org/download>) ან გამოვიყენოთ Notepad. მისი გამოძახება შემდეგი ბრძანებით შეიძლება: Start ⇒ All Programs ⇒ Accessories ⇒ Notepad.

HTML დოკუმენტის სტრუქტურა

HTML კოდირების პროცესში გამოიყენება სპეციალური კოდირების სისტემა, რაც გულისხმობს, რომ ნებისმიერი ინფორმაცია, რომელიც HTML-ის საშუალებით ჩანს ბრაუზერში, იწერება ტეგებში.

ტეგები სპეციალური საკვანძო სიტყვებისგან შემდგარი კოდირების სისტემაა. ყველა HTML ტეგი მარცხენა კუთხური ფრჩხილით (<) იწყება და მარჯვენა კუთხური ფრჩხილით (>) მთავრდება. თითქმის ყველა ტეგი წყვილ-წყვილად არსებობს, ყოველ გახსნის ტეგს დახურვის ტეგი შეესაბამება. ასეთ ტეგებს კონტეინერები ეწოდება. ისინი შეიძლება შეიცავდეს სხვა ტეგებს და ტექსტს. ზოგიერთი ტეგი, მაგალითად, ტეგი
, არ მოითხოვს დახურვის ტეგს.

<html> და </html> ტეგები აღნიშნავს, რომ მათ შორის მდებარე სტრიქონები ერთიან HTML დოკუმენტს ქმნის.

HTML დოკუმენტი სტრუქტურულად ორ ნაწილად იყოფა: სათაური და ძირითადი ნაწილი ანუ დოკუმენტის ტანი.

სათაურის ნაწილი <head> და </head> ტეგებს შორის, ხოლო ძირითადი ნაწილი <body> და </body> ტეგებს შორისაა მოთავსებული.

<head>...</head> ტეგი დოკუმენტის სათაურის ნაწილის დასაწყისსა და დასასრულზე მიუთითებს. ყველაფერი, რაც მოთავსებულია <title> და </title>-ს შორის, არის დოკუმენტის სახელწოდება, რომელიც ბრაუზერის ფანჯრის სათაურის ზოლში ჩნდება. <title> ... </title> ტეგი <head>...</head> ტეგებს შორის თავსდება.

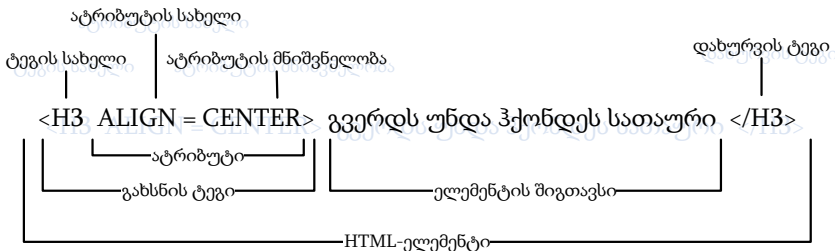
<body>... </body> HTML დოკუმენტის ტანის (ძირითადი ნაწილის) დასაწყისსა და დასასრულზე მიუთითებს. ამ ტეგში იწერება ყველაფერი ის, რის განთავსებასაც ვებგვერდზე ვაპირებთ (ტექსტი, სურათები, ცხრილები და ა.შ.).

ყურადღება მიაქციეთ ტეგების გახსნისა და დახურვის მიმდევრობას:

<ტეგი1> <ტეგი2> <ტეგი3>... </ტეგი3> </ტეგი2> </ტეგი1>
სხვა მიმდევრობით ტეგების განლაგებამ შეიძლება შეცდომა გამოიწვიოს. ტეგები შეგიძლიათ დაწეროთ როგორც დიდი, ასევე პატარა ასოებით, ბრაუზერისთვის ამას მნიშვნელობა არა აქვს.

HTML ტეგების უმეტესობას თავისი ატრიბუტი აქვს. ატრიბუტი ტეგის დამახასიათებელი თვისებაა და მხოლოდ გამხსნელ ტეგშია მოთავსებული. ატრიბუტებს შეიძლება მივანიჭოთ შესაბამისი მნიშვნელობები, რომლებიც ორმაგ ბრჭყალებში („ “) ან აპოსტროფებშია (‘ ’) მოთავსებული. რეკომენდებულია ორმაგი ბრჭყალების გამოყენება.

HTML-ელემენტის სტრუქტურა ასეთია:



ატრიბუტები დამატებითი მმართველი სიტყვებია, რომელიც ტეგისა და ერთმანეთისაგან ინტერვალითაა დაშორებული. შეიძლება ითქვას, რომ ატრიბუტები ტეგის თვისებების სახელებია, რომლებსაც გარკვეული მნიშვნელობის მიღება შეუძლია. ატრიბუტები მხოლოდ გახსნის ტეგებს აქვს, ხოლო დახურვის ტეგებს არა. ატრიბუტები ბრაუზერის მიერ ტეგების ინტერპრეტაციაზე მოქმედებს.

ზოგიერთ ატრიბუტს კონკრეტული მნიშვნელობა აქვს, რომელიც ტოლობის ნიშნის შემდეგ უნდა მიეთითოს. ადრე ატრიბუტის მნიშვნელობა აუცილებლად უნდა ჩასმულიყო ბრჭყალებში, მაგრამ ახლა, უმეტეს შემთხვევაში, ეს ბრჭყალები შესაძლებელია გამოტოვებული იყოს, რაც შედეგზე არ იმოქმედებს.

lang ატრიბუტი

HTML დოკუმენტში ენა <html> ტეგში lang ატრიბუტის საშუალებით შეიძლება გამოცხადდეს.

ენის გამოცხადება პროგრამაზე წვდომის და საძიებო სისტემებისთვის მნიშვნელოვანია. ენის გამოცხადება შეიძლება ასე განხორციელდეს:

```
<!DOCTYPE html>

<html lang = "en-US">

<body>

...

</body>

</html>
```

ატრიბუტის მნიშვნელობაში პირველი ორი სიმბოლო (en) განსაზღვრავს ენას. თუ ამ ენაში არსებობს დიალექტი, მაშინ უნდა დაამატოთ კიდევ ორი სიმბოლო - (US).

<Body> ტეგის ატრიბუტები

<body> ტეგი ყველა იმ ინფორმაციას შეიცავს, რომლისგანაც არსებული დოკუმენტი რეალურად შედგება. <body> ტეგს შეიძლება რამდენიმე ატრიბუტი ჰქონდეს, მაგალითად:

```
<body bgcolor="#808080" text="yellow" leftmargin=0
topmargin=40 rightmargin=0 bottommargin=40 link="#000099"
vlink="#000099" alink="#cc0000">
```

• bgcolor ატრიბუტი დოკუმენტში ფონის ფერს, ფერის ინტენსიურობის მოდელის RGB (Red, Green, Blue - წითელი, მწვანე, ლურჯი) თექვსმეტობითი მნიშვნელობით ან შესაბამისი ფერის სახელით განსაზღვრავს, მისი სინტაქსია:

```
<body bgcolor="#ff0000">
<body bgcolor="red">
```

• background ატრიბუტი გრაფიკულ გამოსახულებას

განსაზღვრავს, რომელიც დოკუმენტის ფონს მოზაიკურად შეავსებს. მისი სინტაქსი ასეთია:

<body background="(url)|(გზა) ფაილის სახელი">

თუ გამოსახულება იმავე საქაღალდეშია ჩაწერილი, რომელშიც დოკუმენტია, მაშინ url მისამართისა და გზის მითითება არ დაგვჭირდება.

- **bgproperties** ატრიბუტი ფონური გამოსახულების თვისებებს განსაზღვრავს. მას მხოლოდ ერთადერთი **fixed** მნიშვნელობა აქვს. თუ ეს ატრიბუტი მითითებულია, მაშინ რბიას გადაადგილების დროს ტექსტი გადაადგილდება, ხოლო ვებგვერდის ფონად გამოყენებული გამოსახულება უძრავად განთავსდება;

- **text** ატრიბუტი გამოყენებული ტექსტის ფერს განსაზღვრავს. სისტემაში ნაგულისხმევი წესის თანახმად, ეს შავი ტექსტია;

- **topmargin/bottommargin/leftmargin/rightmargin** ატრიბუტები გვერდის ზედა/ქვედა/მარცხენა და მარჯვენა მინდვრის საზღვრებს განსაზღვრავს პიქსელებში;

- **link** ატრიბუტი ბმულის ფერს განსაზღვრავს, სისტემაში ნაგულისხმევი წესის თანახმად, უმეტეს ბრაუზერებში, ის მუქი ლურჯი ფერისაა;

- **alink** ატრიბუტი აქტიური ბმულის ფერს განსაზღვრავს, რომელიც მასზე მაუსის დაწკაპუნების მომენტში იცვლება. სასურველია ის **link** ატრიბუტით მოცემული ბმულის ფერისგან განსხვავდებოდეს;

- **vlink** ატრიბუტი უკვე ნანახი ბმულის ფერს განსაზღვრავს. სასურველია, იგი ბმულის ფერისა (**link** ატრიბუტითაა მოცემული) და აქტიური ბმულის ფერისაგან

(alink ატრიბუტითაა მოცემული) განსხვავდებოდეს.

კომენტარები

კომენტარი ეწოდება კოდის იმ ფრაგმენტს, რომლის გამოტანა/აღქმა ბრაუზერის ეკრანზე არ ხდება. როგორც ნებისმიერ ენაში, HTML-შიც, კომენტარი დოკუმენტის ტანის ნებისმიერ ადგილში და ნებისმიერი რაოდენობით შეიძლება განთავსდეს. მისი სინტაქსია:

```
<!-- ერთსტრიქონიანი კომენტარი -->  
<!-- მრავალსტრიქონიანი კომენტარი -->
```

ფერები

HTML-ში ფერები თექვსმეტობით კოდში განისაზღვრება. ფერების გამა სამ ძირითად ფერს ეყრდნობა: წითელს, მწვანესა და ლურჯს და RGB-თი აღინიშნება. ფერი შეიძლება აღიწეროს როგორც მისი დასახელებით, ისე მნიშვნელობით RGB (Red, Green, Blue) პალიტრაში. თითოეული ფერისათვის 00-დან ff-მდე თექვსმეტობითი მნიშვნელობა მიეთითება, რასაც ათობით სისტემაში 0-255 დიაპაზონი შეესაბამება. შემდეგ ეს მნიშვნელობები ერთ რიცხვში ერთიანდება, რომელთა წინ # სიმბოლო იწერება. მაგალითად, რიცხვი #000000 მნიშვნელობა შავ ფერს შეესაბამება, #ffffff თეთრ ფერს, #800080 იისფერს და ა. შ.

მაგალითად:

```
<body bgcolor="#ffffff" text="#000000" link="#9690cc">
```

მოცემული სტრიქონი დოკუმენტის ფონის თეთრ ფერს, შავ ტექსტს და ვერცხლისფერ ბმულს განსაზღვრავს.

ქვემოთ მოცემულია ზოგიერთი ფერი თავისი შესაბამისი

თექვსმეტობითი კოდით:

ფერი - კოდი

შავი - Black - #000000

შინდისფერი - Maroon - #800000

მწვანე - Green - #008000

ზეთისხილისფერი - Olive - #808000

მუქი ლურჯი - Navy - #000080

იისფერი - Purple - #800080

ფირუზისფერი - Teal - #008080

ნაცრისფერი - Gray - #808080

ვერცხლისფერი - Silver - #c0c0c0

წითელი - Red - #ff0000

ღია მწვანე - Lime - #00ff00

ყვითელი - Yellow - #ffff00

ლურჯი - Blue - #0000ff

იასამნისფერი - Violet - #EE82EE

ჟოლოსფერი - Fuchsia - #ff00ff

ცისფერი - Aqua - #00ffff

თეთრი - White - #ffffff

მწვანე ვაშლის ფერი - Apple Green - #8DB600

ზღვისფერი - Aquamarine - #7FFFD4

ღია ცისფერი - Baby Blue - #89CFF0

ბეგონიის ფერი - Begonia - #FA6E79

ნარინჯისფერი - orange - #FFA500

ინდიგო - Indigo - #4B0082

ყავისფერი - Brown - #993300

ღია ყავისფერი - Light Brown - #B5651D

კრემისფერი - Cream - #FFFDD0

მუქი მწვანე - Dark Green - #013220

მუქი ნაცრისფერი - Dark Gray - #A9A9A9

მუქი იასამნისფერი - Dark Violet - #9400D3

ოქროსფერი - Gold - #FFD700

ლიმონისფერი - Lemon - #FFF700

ვარდისფერი - Pink - #FFC0CB

ფერების შესახებ ინფორმაციას დაწვრილებით შეგიძლიათ გაეცნოთ მე-7 თავში.

სათაურის დონეები

დიდი მოცულობის ტექსტები ფორმალურად, სათაურის სხვადასხვა დონის საშუალებით, შეიძლება ცალკეულ თავებად დავყოთ. სათაურის პირველი, ყველაზე მაღალი დონე ციფრი 1-ით აღინიშნება, შემდეგი დონე 2-ით და ა.შ. ნაკლები ნომრის მქონე ტეგი უფრო მაღალი დონის სათაურს აღნიშნავს. ბრაუზერების უმეტესობას სათაურების ტეგების ექვსი დონის მხარდაჭერა აქვს. სათაურის ტეგის სინტაქსი შემდეგია:

```
<hx> სათაურის x დონე </hx>
```

სადაც x სათაურის დონის განმსაზღვრელი ციფრია 1-დან 6-მდე.

align ატრიბუტით ჰორიზონტალის მიმართ ტექსტის სწორება იმართება და left (მარცხენა კიდის მიმართ), center (ცენტრის მიმართ) და right (მარჯვენა კიდის მიმართ) მნიშვნელობები შეიძლება მიიღოს.

მაგალითი 1.1:

```
<!DOCTYPE html>  
<html>  
<head><title> Head </title></head>
```



```
<body>
<h1> Head 1 – GEORGIA </h1>
<h2> Head 2 – GEORGIA </h2>
<h3> Head 3 – GEORGIA </h3>
<h4> Head 4 – GEORGIA </h4>
<h5> Head 5 – GEORGIA </h5>
<h6> Head 6 – GEORGIA </h6>
</body>
</html>
```

სურათზე სხვადასხვა დონის სათაურის მაგალითია მოცემული.

Head 1 – GEORGIA

Head 2 – GEORGIA

Head 3 – GEORGIA

Head 4 – GEORGIA

Head 5 – GEORGIA

Head 6 – GEORGIA

მაგალითი 1.2:

```
<!DOCTYPE html>
<html>
<head>
<title> ჩემი პირველი პროგრამა</title>
</head>
<body bgcolor=ffff05 >
<h2> საქართველო </h2>
```

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

საქართველოში წარმომადგენლობითი დემოკრატიაა, ორგანიზებული როგორც უნიტარული, ნახევრად საპრეზიდენტო რესპუბლიკა. ქვეყანა ამჟამად რამდენიმე საერთაშორისო ორგანიზაციის წევრია, მათ შორის გაეროს, ევროპის საბჭოს, სუამის, მსოფლიო სავაჭრო ორგანიზაციისა და შავი ზღვის ეკონომიკური თანამშრომლობის. საქართველო ასევე მისწრაფვის ევროპის კავშირში ინტეგრაციასა და ჩრდილო-ატლანტიკურ ალიანსში გაწევრიანებას.

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

საქართველო

საქართველო - სახელმწიფო ევრაზიაში, კავკასიაში, შავი ზღვის აღმოსავლეთ სანაპიროზე. ესაზღვრება ჩრდილოეთიდან რუსეთი, სამხრეთიდან - თურქეთი და სომხეთი, სამხრეთ-აღმოსავლეთიდან - აზერბაიჯანი. ტრანსკონტინენტური ქვეყანა სამხრეთ-აღმოსავლეთ ევროპისა და დასავლეთ აზიის გასაყარზე მდებარეობს, თუმცა სოციალურ-პოლიტიკურად და კულტურულად ევროპის ნაწილია. საქართველოში წარმომადგენლობითი დემოკრატიაა, ორგანიზებული როგორც უნიტარული, ნახევრად საპრეზიდენტო რესპუბლიკა. ქვეყანა ამჟამად რამდენიმე საერთაშორისო ორგანიზაციის წევრია, მათ შორის გაეროს, ევროპის საბჭოს, სუამის, მსოფლიო სავაჭრო ორგანიზაციისა და შავი ზღვის ეკონომიკური თანამშრომლობის. საქართველო ასევე მიისწრაფვის ევროპის კავშირში ინტეგრაციასა და ჩრდილო-ატლანტიკურ ალიანსში გაწევრიანებას.

შემდეგ სურათზე მოცემულია ვებგვერდი, რომლის-თვისაც ყველა მინდორი 40 პიქსელის ზომისაა. ჩვენ ვხედავთ, რომ ტექსტი ზედა, მარცხენა და მარჯვენა კიდეებიდან 40 პიქსელითაა დაცილებული, რისთვისაც <body> ტეგში შემდეგი ცვლილებაა შეტანილი:

```
<body bgcolor=ffff05 topmargin=40 bottommargin=40  
leftmargin=40 rightmargin=40>
```

საქართველო

საქართველო - სახელმწიფო ევრაზიაში, კავკასიაში, შავი ზღვის აღმოსავლეთ სანაპიროზე. ესაზღვრება ჩრდილოეთიდან რუსეთი, სამხრეთიდან - თურქეთი და სომხეთი, სამხრეთ-აღმოსავლეთიდან - აზერბაიჯანი. ტრანსკონტინენტური ქვეყანა სამხრეთ-აღმოსავლეთ ევროპისა და დასავლეთ აზიის გასაყარზე მდებარეობს, თუმცა სოციალურ-პოლიტიკურად და კულტურულად ევროპის ნაწილია. საქართველოში წარმომადგენლობითი დემოკრატიაა, ორგანიზებული როგორც უნიტარული, ნახევრად საპრეზიდენტო რესპუბლიკა. ქვეყანა ამჟამად რამდენიმე საერთაშორისო ორგანიზაციის წევრია, მათ შორის გაეროს, ევროპის საბჭოს, სუამის, მსოფლიო სავაჭრო ორგანიზაციისა და შავი ზღვის ეკონომიკური თანამშრომლობის. საქართველო ასევე მიისწრაფვის ევროპის კავშირში ინტეგრაციასა და ჩრდილო-ატლანტიკურ ალიანსში გაწევრიანებას.

მაგალითი 1.3. მუქი ფერის ფონზე ღია ფერის ტექსტის განთავსება:

```
<!DOCTYPE html>
<html>
<head>
<title>ფერადი ფონი</title>
</head>
<body bgcolor=123456 text=ffffff topmargin=50 leftmargin=50>
ლურჯი ფონი და თეთრი ფერის ტექსტი
</body>
</html>
```

ლურჯი ფონი და თეთრი ფერის ტექსტი

ფონური სურათის ჩასმა <Body> ტეგისთვის

თანამედროვე ბრაუზერები დოკუმენტში ფონური სურათების განთავსების საშუალებას იძლევა. უმეტეს შემთხვევაში, ფონის სახით გამოყენებული სურათი საკმაოდ ეფექტურია. ფონური სურათების აღწერა <BODY> ტეგში ხდება და ზოგადად შემდეგი სახით გამოიყურება: <body background = "picture.gif">

მაგალითი 1.4:

```
<!DOCTYPE html>
<html>
<head>
<title>ფონური გამოსახულება</title> </head>
<body topmargin="100" leftmargin="50" background="rose.jpg"
text="white">
<h1>My First Heading</h1>
<p>My first paragraph.</p>
საქართველო <br>
საქართველოს ტექნიკური უნივერსიტეტი <br>
```

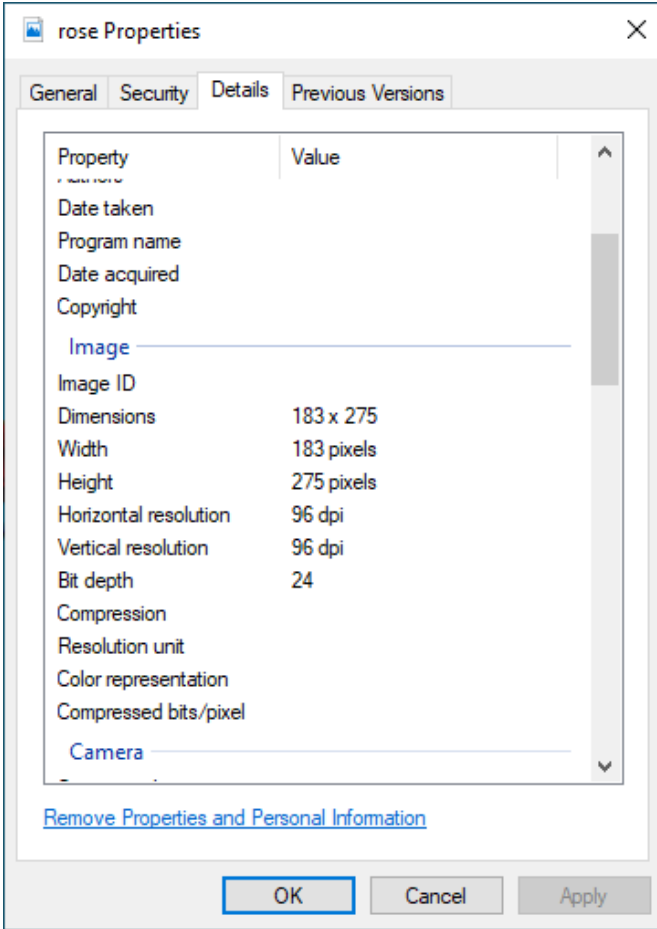
</body>

</html>

ჩვენ მიერ შექმნილი მარტივი ვებგვერდი ასე გამოიყურება:



დოკუმენტის ფონად არჩეულია 183×275 ზომის სურათი. მიუხედავად სურათის მცირე ზომისა, მან მთლიანად დაფარა გვერდის ფონი, მოხდა ფოტოს განმეორება. ქვემოთ სურათზე ნაჩვენებია ფონად დადებული გამოსახულების ზომები.



ამ მაგალითში `<body topmargin="100" leftmargin="50" background="rose.jpg" text="white">` გვიჩვენებს, რომ ტექსტი ვებგვერდის ზედა კიდიდან 100 პიქსელით, ხოლო მარცხენა კიდიდან 50 პიქსელითაა დაშორებული, ამასთან ფონად გამოყენებულია სურათი, ხოლო ტექსტის ფერი თეთრია.

აბზაცი და სტრიქონების წყვეტა

HTML დოკუმენტში ტექსტის აბზაცებად დაყოფა Enter კლავიშის გამოყენებით შეუძლებელია. შემდეგ სტრიქონზე გადასასვლელად
 ტეგით, ხოლო ტექსტის აბზაცით დასაწყებად <p> და <div> ტეგით უნდა ვისარგებლოთ. თუ ამ ტეგებს არ გამოვიყენებთ, მაშინ დოკუმენტი მთლიანად ერთი აბზაცი იქნება.

<p> ტეგს აქვს align ტექსტის სწორების ატრიბუტი. ამ ატრიბუტს შემდეგი მნიშვნელობების მიღება შეუძლია:

- left - ტექსტის სწორება მარცხენა კიდის მიმართ;
- right - ტექსტის სწორება მარჯვენა კიდის მიმართ;
- center - ტექსტის სწორება ცენტრის მიმართ;
- justify - ტექსტის სწორება ორივე კიდის მიმართ.

<p> ტეგში შეიძლება გამოყენებული იყოს title ატრიბუტი. ამ ატრიბუტის გამოყენების დროს ბრაუზერის ფანჯარაში გამოჩნდება <p> ტეგებს შორის მოთავსებული ინფორმაცია, ხოლო ამ ინფორმაციაზე მაუსის მაჩვენებლის მიყვანის შემდეგ მასზე გამოჩნდება ატრიბუტში მითითებული ტექსტი. მაგალითად,

```
<p title="Georgian Technical University">  
საქართველოს ტექნიკური უნივერსიტეტი  
</p>
```

გამოტანილ ტექსტზე მაუსის მაჩვენებლის მიყვანით მივიღებთ:

საქართველოს ტექნიკური უნივერსიტეტი

Georgian Technical University

 ტეგი ბრაუზერს ახალ სტრიქონზე გადასვლას ატყობინებს.

<div> (ინგლ. *division* - განყოფილება) ტეგი საშუალებას გვაძლევს დოკუმენტის სტრუქტურაში რამდენიმე ნაწილი გამოვყოთ. ის კონტენერული ტეგია და <p> ტეგის მსგავსად ფუნქციონირებს, შეიძლება ჰქონდეს align ატრიბუტი, რომელსაც left, center ან right მნიშვნელობების მიღება შეუძლია. მისი სინტაქსია:

```
<div align=სწორება> ტექსტი </div>
```

<div> ტეგი CSS საშუალებებთან ერთობლიობაში გამოიყენება. იგი დოკუმენტის სტრუქტურის აგების მძლავრი საშუალებაა.

ელემენტების სწორება ცენტრის მიმართ სხვადასხვა მეთოდითაც შეიძლება. მაგალითად, ეს შეიძლება აბზაცის ტეგით

```
<p align=center>
```

ან

```
<center> ... </center>
```

ტეგით განხორციელდეს.

მაგალითი 1.5:

```
<!DOCTYPE html>
<html >
<head>
```

<title>html documents formatting </title>

</head>

<body bgcolor=ffff05 topmargin=20 bottommargin=20
leftmargin=20 rightmargin=20>

<h1>ვარდი და ია</h1>

<h2 align = "center"> ვარდი </h2>

<p align = "justify"> ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10 - 12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენჭფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2 - 15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუჭაა, რომელიც ხორციან ცრუნაყოფშია.

ვარდის გვარი იყოფა 4 ქვეგვარად, რომელთაგან ყველაზე დიდია Rosa, იგი მოიცავს 192 სახეობას. აქედან განვითარდა ბალის ვარდები. ბალის ვარდების ბევრი ჯიში წარმოიშვა კვირტოვანი მუტაციისა და მათგან შერჩეული ძვირფასი სამეურნეო ფორმებისაგან. ვარდის ბევრი სახეობა ბუნებრივ პირობებში ადვილად ეჯვარება ერთმანეთს და წარმოქმნის სახეობათაშორის ჰიბრიდებს. ვარდების დიდი მრავალფეროვნებით გამოირჩევა აზია (ჩინეთი, შუა აზია,

ამიერკავკასია), უფრო ნაკლებია ევროპასა და ჩრდილოეთ ამერიკაში. ცნობილია ვარდის 25 ათასი ჯიში და ფორმა, რომლებიც 3 ჯგუფად იყოფა, ესენია: მარადმწვანე ბუჩქები, ფოთოლმცვივანი ბუჩქები და ლიანები.

ვარდი მრავლდება დაკალმებით, ამონაყარით, ბუჩქის დაყოფითა და გადაწვენით; ბალისა და საპარკე ვარდების უმეტესობა - კვირტით მცნობით და კალმით. საქართველოში მცნობა ტარდება ივნისიდან ოქტომბრამდე, ხოლო დარგვა - გაზაფხულსა და შემოდგომაზე. მის მოსაშენებლად საუკეთესოა დაწრეტილი (დრენირებული) თიხნარი, ქვიშნარი, ქვიშიანი და თიხიანი ნიადაგები.</p>

<h2 align="center">ია </h2>

<p align="justify">ია (ლათ. Viola) - მცენარეთა გვარი იისებრთა ოჯახისა. მრავალწლოვანი, იშვიათად ერთწლოვანი ბალახები, ზოგჯერ ნახევრად ბუჩქები. აქვს მორიგეობით გაწყობილი ან ფესვთან როზეტად შეკრებილი ფოთლები, იისფერი, ყვითელი და თეთრი მარტოული ყვავილები. გვირგვინის ქვედა ფურცელი წარზიდულია დეზად, რომელშიც სანექტრეა მოქცეული. ზოგიერთს უვითარდება ვავილები. ნაყოფი სამსაგდულიანი კოლოფია. ზოგ სახეობას ახასიათებს ამფიკარპია. ცნობილია 500-მდე სახეობა. უმრავლესობა გავრცელებულია ჩრდილოეთ ნახევარსფეროს ზომიერ სარტყელში. საქართველოში გავრცელებულია 26 სახეობა, მათგან 3 კავკასიისა და 2 საქართველოს ენდემია. საქართველოში ია იზრდება ტყეებსა და ბუჩქნარებში, სუბალპურ და ალპურ მდელოებზე, კლდეების ნაპრალებში, ჩამონახვავებზე, ზოგიერთი სახეობა (Viola alba) სარეველაც არის, ასევე დეკორატიულიც (იაჟუჟუნა - Viola tricolor და სხვა),

ზოგს (სურნელოვანი ია - Viola odorata) პარფიუმერიაში

იყენებენ.

</body>

</html>

ვარდი და ია

ვარდი

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მრავალწლოვან ან ფოთლოვან ბუჩქსა, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთი სახეობის ერთეული (10 - 12 მ-მდე) ხვრია ტოტემი აქვს, ღერო დაფარულია ცვლებით. ფოთლოვანი კუნთვრთისებრია, ფეროზე მორიგებით განლაგებული. ყვავილი მარტივია ან ხუთხეობა (2 - 15 სმ დიამეტრის), პროზაული ან უსუნო; ფერი - თეთლი, თეთრი, ყვითლი, წარწივიწყვრი ან ანაწივიწყვრი. შტერეულია ქოლესისტერ-საგვლა ყვავილობად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკულუსა, რომელიც ხორციან ცრუნაყოფშია. ვარდის გვარი იყოფა 4 ქვეგვარად, რომელთაგან ყველაზე დიდია Rosa. იგი მოიცავს 192 სახეობას. აქედან განეორად ბაღის ვარდები. ბაღის ვარდების ბერი ჯიში წარმოიშვა კვრთვანი მუტაციისა და მათგან შერწყული ჭირფასი სამეურნეო ფორმებისაგან. ვარდის ბერი სახეობა ბუნებრივ პირობებში ადვილად ეუვარება ერთმანეთს და წარმოქმნის სახეობათაშორის ჰიბრიდებს. ვარდების დიდი მრავალფეროვნებით გამოირჩევა აზია (ჩინეთი, შუა აზია, ამიერკავკასია), უფრო ნაკლებია ევროპასა და ჩრდილოეთ ამერიკაში. ცნობილია ვარდის 25 ათასი ჯიში და ფორმა, რომლებიც 3 კულუვად იყოფა. ესენია: მრავალწლოვანი ბუჩქები, ფოთლოვანი ბუჩქები და ლიანები.

ვარდი მრავლდება დაკალმებით, ამონაყრით, ბუჩქის დაყოფითა და გადამწვებით; ბაღისა და საპარკ ვარდების უმეტესობა - კვრითი მცნობით და კალმით. საქართველოში მცნობა ტრდება იენისიდან ოქტომბრამდე, ხოლო დარგვა - გაზაფხულსა და შემოდგომაზე. მის მოსამწებლად საუკეთესოა დარწეტილი (დრწინრებული) თიხნარი, ქვიშარი, ქვიშიანი და თიხიანი ნიადაგები.

ია

ია (ლათ. Viola) - მცენარეთა გვარი იისებრთა ოჯახისა. მრავალწლოვანი, იშვიათად ერთწლოვანი ბალახები, ზოგჯერ ნახევრად ბუჩქები. აქვს მორიგებით განწყობილი ან ფესვიან როზეტად მკვრებილი ფოთლები, იისფერი, ყვითელი და თეთრი მარტული ყვავილები. გვირგვინის ქვედა ფურცელი წარსიღულია დეზად, რომელზეც სანქტრე მოქეული. ზოგიერთს უფითარდება ვავილები. ნაყოფი სამსაგვლიანი კოლოფი. ზოგ სახეობას ახასიათებს ამფიკარპია. ცნობილია 500-მდე სახეობა. უმრავლესობა გავრცელებულია ჩრდილოეთ ნახევარსფეროს ზომიერ სარტელებში. საქართველოში გავრცელებულია 26 სახეობა, მათგან 3 კავკასიისა და 2 საქართველოს ენდემია. საქართველოში ია ოზრდება ტყეებსა და ბუჩქნარებში. სუბალპურ და ალპურ მდელოებზე, კლდეების ნარარლებში, ჩამონახავებზე. ზოგიერთი სახეობა (Viola alba) სარეულად არის, ასევე დეკორატიული (იყუფუნა - Viola tricolor და სხვა). ზოგს (სურნელოვანი ია - Viola odorata) პარფიუმერიაში იყენებენ.

ქვემოთ მოცემულ HTML კოდში ჩასმულია ფონური სურათი, მისი ატრიბუტი არის bgcolor=fixed.

```
<body bgcolor=rose.jpg bgcolor=fixed text=yellow topmargin=20 bottommargin=20 leftmargin=20 rightmargin=20 >
```



ქვემოთ მოცემულია <div> ტეგის გამოყენების მაგალითი:

მაგალითი 1.6:

```

<!DOCTYPE html>
<html >
<head>
<title>html documents formatting </title>
</head>
<body bgcolor=#00ff05 topmargin=10 bottommargin=40
leftmargin=40 rightmargin=40 >
<h1>ვარდი და ია</h1>
<h2 align = "center"> ვარდი </h2>
<div align = "left"> ვარდი (ლათ. Rosa) - მცენარის გვარი
ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური
სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ
ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი

```

მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10 - 12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენტფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2 - 15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუჭაა, რომელიც ხორციან ცრუნაყოფშია. </div>

<div align = "right>ვარდის გვარი იყოფა 4 ქვეგვარად, რომელთაგან ყველაზე დიდია Rosa, იგი მოიცავს 192 სახეობას. აქედან განვითარდა ბალის ვარდები. ბალის ვარდების ბევრი ჯიში წარმოიშვა კვირტოვანი მუტაციისა და მათგან შერჩეული ძვირფასი სამეურნეო ფორმებისაგან. ვარდის ბევრი სახეობა ბუნებრივ პირობებში ადვილად ეჯვარება ერთმანეთს და წარმოქმნის სახეობათაშორის ჰიბრიდებს. ვარდების დიდი მრავალფეროვნებით გამოირჩევა აზია (ჩინეთი, შუა აზია, ამიერკავკასია), უფრო ნაკლებია ევროპასა და ჩრდილოეთ ამერიკაში. ცნობილია ვარდის 25 ათასი ჯიში და ფორმა, რომლებიც 3 ჯგუფად იყოფა, ესენია: მარადმწვანე ბუჩქები, ფოთოლმცვივანი ბუჩქები და ლიანები.</div>

<p align="justify"> ვარდი მრავლდება დაკალმებით, ამონაყარით, ბუჩქის დაყოფითა და გადაწვევით; ბალისა და საპარკე ვარდების უმეტესობა - კვირტით მცნობით და კალმით. საქართველოში მცნობა ტარდება ივნისიდან ოქტომბრამდე, ხოლო დარგვა - გაზაფხულსა და შემოდგომაზე. მის მოსაშენებლად საუკეთესოა დაწრეტილი (დრენირებული)

თიხნარი, ქვიშნარი, ქვიშიანი და თიხიანი ნიადაგები.</p>

<h2 align = "center">ია </h2>

<div align="center"> ია (ლათ. Viola) - მცენარეთა გვარი იისებრთა ოჯახისა. მრავალწლოვანი, იშვიათად ერთწლოვანი ბალახები, ზოგჯერ ნახევრად ბუჩქები. აქვს მორიგეობით გაწყობილი ან ფესვთან როზეტად შეკრებილი ფოთლები, იისფერი, ყვითელი და თეთრი მარტოული ყვავილები. გვირგვინის ქვედა ფურცელი წარზიდულია დეზად, რომელშიც სანექტრეა მოქცეული. ზოგიერთს უვითარდება ვავილები. ნაყოფი სამსაგდულიანი კოლოფია. ზოგ სახეობას ახასიათებს ამფიკარპია. ცნობილია 500-მდე სახეობა. უმრავლესობა გავრცელებულია ჩრდილოეთ ნახევარსფეროს ზომიერ სარტყელში. საქართველოში გავრცელებულია 26 სახეობა, მათგან 3 კავკასიისა და 2 საქართველოს ენდემია. საქართველოში ია იზრდება ტყეებსა და ბუჩქნარებში, სუბალპურ და ალპურ მდელოებზე, კლდეების ნაპრალებში, ჩამონაზვავებზე, ზოგიერთი სახეობა (Viola alba) სარეველაც არის, ასევე დეკორატიულიც (იაჟუჟუნა - Viola tricolor და სხვა), ზოგს (სურნელოვანი ია - Viola odorata) პარფიუმერიაში იყენებენ.</div>

</body>

</html>

ვარდი და ია

ვარდი

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთლომცვეიანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10 - 12 მ-მდე) ხვიათ ტიპები აქვს. ღერო დაფარულია ველუბით, ფოთოლი კენჭურთისებრია, ღეროზე მორიგეობით განლაგებული ყვავილი მარტივია ან ზუთხეზა (2 - 15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ითელი, თეთრი, ყვითლი, წარინჯისფერი ან იასამინისფერი. შეკრებილია ქოლვისებრ-საგველა ყვავილეტებად ან ერთფეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკულუტა, რომელიც ხორციან ცრუნაყოფემა.

ვარდი მრავლდება დაკალმებით, ამინაყარით, ბუჩქის დაყოფითა და გადაწვევით; ზალისა და საპარკე ვარდების უმეტესობა - კვირტი მჭობით და კალმით. საქართველოში მჭობა ტარდება ივნისიდან ოქტომბრამდე, ხოლო დარგვა - გაზაფხულსა და შემოდგომაზე. მის მოსამუშებლად საუკეთესოა დაწრტილი (დრენირებული) თიხნარი, ქვიშარი, ქვიშიანი და თიხიანი ნიადაგები.

ია

ია (ლათ. Viola) - მცენარეთა გვარი იისებრთა ოჯახისა. მრავალწლოვანი, იშვიათად ერთწლოვანი ზალახეობი, ზოგჯერ ნახევრად ბუჩქები. აქვს მორიგეობით განწყობილი ან ფესვთან როზეტად შეკრებილი ფოთლები, იისფერი, ყვითელი და თეთრი მარტოული ყვავილები. გვირგვინის ქვედა ფურცელი წარზიდულია ღეზად, რომელშიც სანქტურა მოქცეული ზოგიერთს უფითარდება ვაგილები. ნაყოფი სამსაგდელიანი კოლოფია. ზოგ სახეობას ახასიათებს ამფიკარპია. ცნობილია 200-მდე სახეობა. უმრავლესობა გავრცელებულია ჩრდილოეთ ნახევარსფეროს ზომიერ სარტყელში. საქართველოში გავრცელებულია 26 სახეობა, მათგან 3 კავკასიისა და 2 საქართველოს ენდემია. საქართველოში ია იზრდება ტყეებსა და ბუჩქნარებში, სუბალპურ და ალპურ მდელოებზე, კლდეების ნაპრალებში, ჩამონახევებზე, ზოგიერთი სახეობა (Viola alba) სარეველაც არის, ასევე დეკორატიული (იყუფუნა - Viola tricolor და სხვა), ზოგს (სურმელოვანი ია - Viola odorata) პარფუმერიაში იყენებენ.

დავალება 1. შექმენი ვებგვერდი ქვემოთ მოცემული ტექსტის გამოყენებით, რომლის ფონი იქნება მწვანე; ტექსტი - ვებგვერდის კიდეებიდან დაცილებული 25 პიქსელით. სათაური - ყველაზე დიდი ზომის, სწორება მარცხნივ, ქვესათაურები - მეორე დონის განთავსებული ცენტრში; პირველი აბზაცი - ყვითელი, ტექსტის სწორება ორივე მხრიდან, მეორე აბზაცი - წითელი, ტექსტის სწორება ცენტრში. მესამე აბზაცი - თეთრი, ტექსტის სწორება მარჯვნივ. მეოთხე აბზაცი - ლურჯი და ტექსტის სწორება მარცხნივ.

საქართველო

კახეთი

კახეთი, მხარე აღმოსავლეთ საქართველოში. მოიცავს მდინარე ივრის შიდა და ქვემო დინების და მდინარე ალაზნის აუზს. კახეთის ცალკეულ მხარეებს ეწოდებოდა გარეკახეთი (მდინარე ივრის შუა წელი), ქიზიყი (მდინარე ივრის ქვემო წელი), შიგნიკახეთი (მდინარე ალაზნის მარჯვენა სანაპირო) და გაღმა მხარე (მდინარე ალაზნის მარცხენა სანაპირო). უძველეს დროს კახეთი გაცილებით მცირე ტერიტორიას მოიცავდა (მდინარე ივრის ზემო წელი თიანეთსა და უჯარმას შორის). ცენტრი ჩელები (ქალები). IV საუკუნიდან - უჯარმა. კახეთზე გადიოდა მნიშვნელოვანი გზები. ხელსაყრელი სტრატეგიული მდებარეობის გამო, დროთა ვითარებაში თავდაპირველი კახეთის გარშემო გაერთიანდა მომიჯნავე პოლიტიკური ერთეულები. ადრინდელ ფეოდალურ ხანაში კახეთი უკვე ერწო-თიანეთისა და თუშ-ხევსურეთის ტერიტორიასაც მოიცავდა. VIII საუკუნის ბოლოს კახეთის შედგენილობაში შევიდა აგრეთვე კუხეთი და ჰერეთის ტერიტორიის ნაწილი (XV საუკუნიდან კი მთელი ჰერეთი). XI საუკუნიდან კახეთის პოლიტიკური ცენტრი

თელავი გახდა, XV საუკუნიდან - გრემი.

საქართველოს ადმინისტრაციულ-ტერიტორიული დაყოფით კახეთი მოიცავს ახმეტის, გურჯაანის, დედოფლისწყაროს, თელავის, ლაგოდეხის, საგარეჯოს, სიღნაღის, ყვარლის რაიონების ტერიტორიასა და ქართველ მთიელთა ეთნოგრაფიულ კუთხეს - თუშეთს.

იმერეთი

იმერეთი დასავლეთ საქართველოს ერთ-ერთი ისტორიულ-გეოგრაფიულ მხარე, ამჟამად იმერეთის მხარის ნაწილი. ვრცელი მნიშვნელობით იგი დასავლეთ საქართველოს ისტორიული სახელწოდებაა, ისევე როგორც ძველი კოლხეთი, ეგრისი, აფხაზეთი. საკუთრივ იმერეთი შემოსაზღვრულია აღმოსავლეთით ლიხის ქედით, დასავლეთით - ცხენისწყლით, ჩრდილოეთით - კავკასიონის ქედით და სამხრეთით - ფერსათის ანუ მესხეთის მთებით. სახელწოდება დაკავშირებულია ამ მხარის მდებარეობასთან, იმერეთი ანუ ლიხსიქითა მხარე.

იმერეთი იყოფა ორ ნაწილად: ზემო და ქვემო იმერეთად. იმერეთის ტერიტორიაზე აღმოჩენილი არქეოლოგიური ძეგლები ადასტურებს, რომ ამ მხარეში ადამიანს ცხოვრება ჯერ კიდევ ქვედა პალეოლითის ხანაში დაუწყია. ამის დასტურია საკაჟიის და ჭახათის (მდ. წყალწითელას ნაპირზე), დევისხვრელის (მდ. ჩხერიმელას ნაპირზე) გამოქვაბულები, სათაფლიის მიდამოები და სხვა. საქალაქო ცხოვრების უძველესი პერიოდის არქეოლოგიური ძეგლები ნაპოვნია ქუთაისში, ვანში, ვარციხეში (როდოპოლისი), შორაპანში და სხვა. მხარის ხელსაყრელი გეოგრაფიული მდებარეობის გამო, ამ ქალაქებს ოდითგანვე დიდი სტრატეგიული, ეკონომიკური და პოლიტიკური მნიშვნელობა ჰქონდა.

საქართველო

კახეთი

კახეთი, მხარე აღმოსავლეთ საქართველოში. მოიცავს მდინარე ივრის შიდა და ქვემო დინებისა და მდინარე ალაზნის აუზს. კახეთის ცალკეულ მხარეებს ეწოდებოდა გარეკახეთი (მდინარე ივრის შუა წელი), ქიზიყი (მდინარე ივრის ქვემო წელი), შიგნიკახეთი (მდინარე ალაზნის მარჯვენა სანაპირო) და გაღმამხარი (მდინარე ალაზნის მარცხენა სანაპირო). უძველეს დროს კახეთი გაცილებით მცირე ტერიტორიას მოიცავდა (მდინარე ივრის ზემო წელი თიანეთსა და უჯარმას შორის). ცენტრი ჩელითი (ყალბითი). IV საუკუნიდან - უჯარმა. კახეთზე გადიოდა მნიშვნელოვანი გზები. ხელსაყრელი სტრატეგიული მდებარეობის გამო დროთა ვითარებაში თავდაპირველ კახეთის გარშემო გაერთიანდა მომიჯნავე პოლიტიკური ერთეულები ადრინდელ ფეოდალურ ხანაში კახეთი უკვე ერწო-თიანეთისა და თუმ-ხევსურეთის ტერიტორიასაც მოიცავდა. VIII საუკუნის ბოლოს კახეთის შემადგენლობაში შევიდა აგრეთვე კუხეთი და ჰერეთის ტერიტორიის ნაწილი (XV საუკუნიდან კი მთელი ჰერეთი). XI საუკუნიდან კახეთის პოლიტიკური ცენტრი თელავი, XV საუკუნიდან - გრემი.

საქართველოს ადმინისტრაციულ-ტერიტორიული დაყოფით კახეთი მოიცავს შემდეგს: გურჯაანის, დედოფლისწყაროს, თელავის, ლიგოჯინის, საგარეჯოს, სიღნაღის, ყვარლის რაიონების ტერიტორიასა და ქართველ მთელითა ეთნოგრაფიულ კუთხეს - თელავს.

იმერეთი

იმერეთი - დასავლეთ საქართველოს ერთ-ერთი ისტორიულ-გეოგრაფიულ მხარე, ამჟამად იმერეთის მხარის ნაწილი. ვრცელი მიწმეულობით იგი დასავლეთ საქართველოს ისტორიული სახელწოდებაა, ისევე როგორც მგელი კოლხეთი, ეგრისი, აფხაზეთი. საკუთრივ იმერეთი შემოსაზღვრულია აღმოსავლეთით ლიხის ქედით, დასავლეთით - ცხენისწყლით, ჩრდილოეთით - კავკასიონის ქედით და სამხრეთით - ფერსათის, ანუ მესხეთის მთებით. სახელწოდება დაკავშირებულია ამ მხარის მდებარეობასთან, იმერეთი, ანუ ლიხისქეთა მხარე.

იმერეთი ადრე-იან ნაწილად ზემო და ქვემო იმერეთად იმერეთის ტერიტორიაზე ადმინისტრაციულ-გეოგრაფიულად მრავლად დასატერმინ. რომ ამ მხარეს ადრინდეს ეძვოდა, ვერ ვიცით. ზედა საბოლოოაღის ხანაში დიდუბის ამის დასტურია საკვიოს და ჯახათის ორ შედარებითი ნაპირზე. დედოფლისწყაროს მხ. მხარეთმცოდნეობის ინსტიტუტის მკვლევარმა ბათუმის მდინარეზე და მთა სატალას ქობორქის ფეხებზე უფროდს. არქეოლოგიური ძეგლები ნაპირის ქუთაისში უკნო ვარდობით ჩრდილოეთით შიდაში და სხვა. მართს შედარებით დეტალური მდებარეობის ცნობ. ამ კარტეში იდენიფიკაცია დიდა სტრატეგიული, ეკონომიკური და პოლიტიკური მნიშვნელობის ექნება.

დავალება 2. იმავე ტექსტის გამოყენებით შექმენი ვებგვერდი, რომლის ფონი იქნება საქართველოს რუკა; ტექსტი - ვებგვერდის კიდეებიდან დაცილებული 25 პიქსელით. სათაური - ყველაზე დიდი ზომის, სწორება მარცხნივ, ქვესათაურები - მეორე დონის განთავსებული ცენტრში. ტექსტი - შავი ფერის, ტექსტის სწორება ორივე მხრიდან.

საქართველო

კახეთი

კახეთი მხარე აღმოსავლეთ საქართველოში, მოიცავს მდინარე იფრის შიდა და ქვემო დინებას და მდინარე ალაზნის აუზს. კახეთის ცალკეულ მხარეებს წოდებოდა გარეკახეთი (მდინარე იფრის შიდა წელი), ქიზიყი (მდინარე იფრის ქვემო წელი), შიგნკახეთი (მდინარე ალაზნის მარჯვენა სანაპირო) და გუბაშხარი (მდინარე ალაზნის მარცხენა სანაპირო). კახეთს დროს კახეთი გავილობთ მცირე ტერიტორიას მოიცავდა (მდინარე იფრის ზემო წელი თბილისთან და კლდარის შორის), ცენტრ ჩელითი (ერევანი). IV საუკუნიდან უკან. კახეთზე წავიდა მონღოლები გუბა, ხელსაყრელ სტრატეგიული მდებარეობის გამო დროთა განმავლობაში თავდაპირველად გახდეს გარეკახეთი გაერთიანდა მთელს კავკასიურ ციხეებს. აღინიშნა ფეოდალურ ხანაში კახეთი უკვე ერთ-ერთიანობა და თურ-ხელსაყრელ ტერიტორიასაც მოიცავდა. VIII საუკუნის ბოლოს კახეთის შემადგენლობაში შევიდა აგრეთვე კახეთი და ჰერეთის ტერიტორიის ნაწილი (XV საუკუნიდან კი მთელი ჰერეთი). XI საუკუნიდან კახეთის პოლიტიკური ცენტრი თელავი. XV საუკუნიდან - ერეკლე.

საქართველოს ადმინისტრაციულ-ტერიტორიული დაყოფით კახეთი მოიცავს ახმეტის, გურჯაანის, დედოფლისწყაროს, თელავის, ლალოცხის, საგარეოს, სიღნაღის, ყვარლის რაიონების ტერიტორიას და ქართველ მთელთა ეთნოგრაფიულ ჯგუფებს: ათაშენის.

იმერეთი

იმერეთი - დასავლეთ საქართველოს ერთ-ერთი ისტორიულ-გეოგრაფიულ მხარე, აშკამად იმერეთის მხარის ნაწილი. ვრცელი მნიშვნელობით იგი დასავლეთ საქართველოს ისტორიული სახელწოდებაა, ისევე როგორც მცელი კოლხეთი, ერისი, აფხაზეთი. საკუთრივ იმერეთი შემოსაზღვრულია აღმოსავლეთით ლიხის ქედით, დასავლეთით - ცხენისწყლით, ჩრდილოეთით - კავკასიონის ქედით და სამხრეთით - ფორსისის, ანუ მტხებითი მთებით. სახელწოდება დაკავშირებულია ამ მხარის მდებარეობასთან, იმერეთი, ანუ ლიხისქეთა მხარე.

იმერეთი იყოფა ორ ნაწილად: ზემო და ქვემო იმერეთად. იმერეთის ტერიტორიაზე ადმინისტრაციული არქეოლოგიური ძეგლები აღსკურებს, რომ ამ მხარეში ადამიანს ცხოვრობდა, აგრ კიდევ ქვედა პალეოლითის ხანაში დაუწყია ამის დასტურება საკვირის და ქახაიის (მდ. წყალწითელას ნაპირზე), დევისხერის (მდ. მხერომელს ნაპირზე) გამოქვაბულები, სითფლისის მთაძოხში და სხვა. სახალკეთი ცხოვრობის უძველესი პერიოდის არქეოლოგიური ძეგლები ნაპოვნია ქუთაისში, განაში, ვარციხეში (როდომოლის), შორაპანში და სხვა. მხარის ხელსაყრელ გეოგრაფიული მდებარეობის გამო ამ აქლავებს დიდი სტრატეგიული, ეკონომიკური და პოლიტიკური მნიშვნელობა ჰქონდა.

თავი 2

შრიფტის დაფორმატება, ჰორიზონტალური ხაზები, სიები

- შრიფტების დაფორმატება;
- ჰორიზონტალური ხაზი;
- სიები;
- დანომრილი სიები;
- მარკირებული სიები;
- სია-განსახვრება.

შრიფტის დაფორმატების ტეგები

 ტეგისა და მისი ატრიბუტების საშუალებით ხდება შრიფტის ზომებისა და ფერის მართვა. ეს ატრიბუტებია:

- size ატრიბუტი შრიფტის ზომას მართავს. შრიფტის ზომები 1-დან 7-მდე იცვლება. შრიფტის ზომა შეიძლება მიეთითოს როგორც ციფრით (ნაგულისხმევი წესის თანახმად, ის 3-ის ტოლია), ისე მისი საბაზო მნიშვნელობის მიმართ, უარყოფითი ან დადებითი მიმართულებით, წანაცვლებით. ქვემოთ მოყვანილ ცხრილში შრიფტის ზომებსა და ტიპურ სიდიდეებს შორის დამოკიდებულებაა მოცემული:

შრიფტის ზომა	ტიპური სიდიდე პუნქტებში
1	8
2	10
3	12
4	14
5	18
6	24
7	36

შრიფტის ზომა შემდეგი ბრძანებითაც შეიძლება შეიცვალოს:

- color ატრიბუტი შრიფტის ფერის შესაცვლელად გამოიყენება. მისი სინტაქსია:

ფერი მისი დასახელებით ან RGB მოდელით უნდა მივუთითოთ, რასაც ფერის თექვსმეტობითი ფორმატი შეესაბამება.

 ტეგს ამ ატრიბუტების გარდა შეიძლება ჰქონდეს face ატრიბუტი, რომელშიც შეიძლება მიეთითოს ერთმანეთისაგან მძიმით გამოყოფილი რამდენიმე შრიფტის

პროორიტეტული სია.

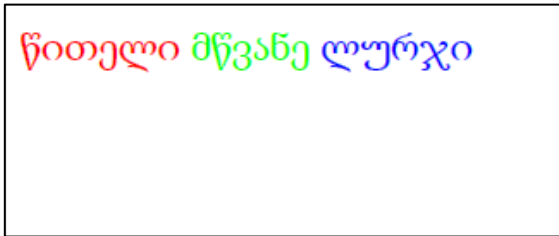
მაგალითი 2.1:

```
<!DOCTYPE html>
<html>
<head>
<title> შრიფტის ზომა </title>
</head>
<body>
<font face = "Sylfaen", "Verdana", "Arial", "Calibri">
<font size = 1> font size 1 </font><br>
<font size = 2> font size 2 </font><br>
<font size = 3> font size 3 </font><br>
<font size = 4> font size 4 </font><br>
<font size = 5> font size 5 </font><br>
<font size = 6> font size 6 </font><br>
<font size = 7> font size 7 </font><br>
</body>
</html>
```

```
*
font size 1
font size 2
font size 3
font size 4
font size 5
font size 6
font size 7
```

მაგალითი 2.2:

```
<!DOCTYPE html>
<html>
<head> <title>შრიფტის ფერი </title></head>
<body>
<font size=5 color="#ff0000"> წითელი </font>
<font size=5 color="#00ff00"> მწვანე </font>
<font size=5 color="#0000ff"> ლურჯი </font>
</body>
</html>
```



ამჟამად შრიფტის ზომის შესაცვლელად CSS საშუალებების ან `` ტეგის გამოყენებაა სასურველი.

ტექსტის დაფორმატება

დოკუმენტში ტექსტური ინფორმაციის გამოსაყოფად შრიფტების დაფორმატების სხვადასხვა საშუალება გამოიყენება, ეს ტეგებია:

- `` და `` - მუქი ტექსტი;
- `<i>` და `</i>` - დახრილი ტექსტი;
- `<ins>` და `</ins>` - ხაზგასმული ტექსტი;
- `` და `` - ხაზგადასმული ტექსტი;

- <small> და </small> - ტექსტი პატარა შრიფტით;
- და - მნიშვნელოვანი ტექსტი;
- და - აქცენტირებული ტექსტი;
- <mark> და </mark> - მარკირებული ან მონიშნული ტექსტი;
- <code> და </code> - ტექსტს კომპიუტერული კოდის სახით აფორმებს;
- _{და} - ქვედა ინდექსი;
- ^{და} - ზედა ინდექსი.

მაგალითი 2.3:

```

<body>
<b> ეს ტექსტი მუქია </b><br>
<i> ეს ტექსტი დახრილია </i><br>
<ins> ეს ტექსტი ხაზგასმულია </ins><br>
ეს ტექსტი <del> ხაზგადასმულია </del><br>
ეს ტექსტი <small> პატარა შრიფტითაა </small><br>
ეს ტექსტი <strong> მნიშვნელოვანი ტექსტია </strong><br>
ამ ტექსტზე <em>აქცენტი უნდა გააკეთოთ </em><br>
ეს ტექსტი <mark> მარკირებულია </mark><br>
ეს ტექსტი კომპიუტერული კოდის სახით გამოვა <code> if
(i==5) echo "i = 5"; <code><br>
ეს ტექსტი <sub> ინდექსად ქვემოთაა </sub><br>
ეს ტექსტი <sup> ინდექსად ზემოთაა </sup><br>
</body>

```

მართალია, ზემოთ ჩამოთვლილი ტეგები მოძველებული არ არის და ამდენად ტექსტის გასაფორმებლად CSS საშუალებები შეიძლება წარმატებით იქნეს გამოყენებული.

<p>ეს ტექსტი მუქია <i>ეს ტექსტი დახრილია</i> <u>ეს ტექსტი ხაზგასმულია</u> ეს ტექსტი ხაზგადასმულია ეს ტექსტი პატარა შრიფტითაა ეს ტექსტი მნიშვნელოვანი ტექსტია ამ ტექსტზე <i>აქცენტი უნდა გააკეთოთ</i> ეს ტექსტი მარკირებულია ეს ტექსტი კომპიუტერული კოდის სახით გამოვა if (i==5) echo "i = 5"; ეს ტექსტი ინდექსად ქვემოთაა ეს ტექსტი ინდექსად ზემოთაა</p>
--

მონოსიგანის შრიფტების გამოყენება

მონოსიგანის შრიფტით (საბეჭდი მანქანის შრიფტის ეფექტით) წარწერების შესრულება <code> და <code> ტეგების საშუალებით ხდება. ამ ტეგების გამოყენების შედეგები ჩვეულებრივ ერთნაირია.

<code> ტეგი გამოიყენება მომხმარებლის მიერ შესატანი ინფორმაციის ან ხმოვანი ბრძანების, ან კლავიშთა კომბინაციით შესასრულებელი ბრძანების გამოსატანად.

მისი სინტაქსია:

<code> კლავიატურით შესატანი ინფორმაცია </code>.

მაგალითი 2.4:

```
<code>  
დოკუმენტის შესანახად გამოიყენეთ კლავიშთა კომბინაცია
```

```
<kbd> <Ctrl>+<S> </kbd>
</body>
```

დოკუმენტის შესანახად გამოიყენეთ კლავიშთა კომბინაცია Ctrl+S

<samp> ტეგი მომხმარებლის კომპიუტერთან დიალოგის გამოყოფის მიზნით, ძირითადად, აუცილებლად პირდაპირი მნიშვნელობით გასაგები ტექსტის გამოსაყოფად გამოიყენება. მისი სინტაქსია:

<samp> კომპიუტერის სისტემური შეტყობინება</samp>

მაგალითი 2.5:

```
<body><font size=5>
```

<p>კომპიუტერში არასწორი ინფორმაციის შეტანის შემთხვევაში ეკრანზე გამოვა შეტყობინება -

```
<samp> ERROR!!! </samp></ p>
```

```
</font>
```

```
</body>
```

კომპიუტერში არასწორი ინფორმაციის შეტანის შემთხვევაში ეკრანზე გამოვა შეტყობინება - ERROR!!!

<var> ტეგი გამოიყენება მათემატიკური ან პროგრამული ფორმულებისა და ცვლადების გამოსატანად.

მაგალითი 2.6:

```
<body><font size=5>  
<p>ენერჯის გამოსათვლელი ფორმულაა -  
<var>E</var> = <var>mc</var><sup>2</sup></ p>  
</font>  
</body>
```

ენერჯის გამოსათვლელი ფორმულაა - $E = mc^2$

 ტეგი ტექსტის ფრაგმენტის მისი შემდგომი დაფორმატების მიზნით გამოყოფის საშუალებას იძლევა, მაგრამ <div> ტეგისაგან განსხვავებით, ახალ აზვაცს არ იწყებს. ტეგი მომხმარებლის მიერ განსაზღვრულ ტექსტური დონის სტრუქტურას ქმნის.

 ტეგი CSS სტილებთან ერთობლიობაში გამოიყენება.

<nobr> ტეგი (ინგლ. *no break* - წყვეტის გარეშე) ბრაუზერს ბრძანებას აძლევს მთელი ტექსტი წყვეტის გარეშე ერთ სტრიქონში განათავსოს. ამ ტეგით ის ტექსტი უნდა მოინიშნოს, რომელიც აუცილებლად ერთ სტრიქონში უნდა განთავსდეს. ამგვარად, თავისი ფუნქციებით <nobr> ტეგი,
 ტეგის ფუნქციების საწინააღმდეგოა.

თუ ტექსტის სტრიქონს <nobr> და </nobr> ტეგებს შორის

მოვათავსებთ, მაშინ, მიუხედავად იმისა, ეს ტექსტი ეკრანის საზღვრებს გადასცდა თუ არა, ბრაუზერი ახალ სტრიქონზე აღარ გადავა. ამის სანაცვლოდ, გამოტანილ ფანჯარაში ჰორიზონტალური გადაფურცვლის რბია გაჩნდება.

<wbr> ტეგს დამხურავი ტეგი არ გააჩნია, იგი მხოლოდ <nobr> </nobr> ტეგის ტანში გამოიყენება და მისი ფუნქცია <nobr> </nobr> ტეგში ჩაწერილი ტექსტის ახალ სტრიქონზე გადატანაა.

მაგალითი 2.7:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>ტექსტი ერთ სტრიქონად </title></head>
```

```
<body bgcolor=#eaaaac>
```

```
<h1>საქართველო </h1>
```

<nobr> საქართველო - ევროპისა და აზიის გასაყარზე, კერძოდ კავკასიაში მდებარეობს. დასავლეთიდან მას შავი ზღვა, ჩრდილოეთით - რუსეთის ფედერაცია, სამხრეთ-აღმოსავლეთით - აზერბაიჯანი, სამხრეთით - სომხეთი და თურქეთი ესაზღვრება. <wbr>

საქართველო იმ უძველეს სატრანსპორტო გზაჯვარედინზე მდებარეობს, რომელიც ჩრდილოეთისა და სამხრეთის, დასავლეთისა და აღმოსავლეთის ქვეყნებს აკავშირებდა და აკავშირებს. სწორედ მასზე გადიოდა ევროპა -აზიის დამაკავშირებელი სატრანსპორტო მაგისტრალი - ძველი აბრემუმის გზა. <wbr>

საქართველო შავი ზღვით - შავი ზღვისპირეთის ქვეყნებს,

ხოლო ბოსფორისა და დარდანელის სრუტეებით-ხმელთაშუა ზღვის აუზის ქვეყნებს უკავშირდება; გიბრალტარის სრუტით - მთელს მსოფლიოსთან, ხოლო მდინარე დუნაის მეშვეობით კი - აღმოსავლეთ და ცენტრალურ ევროპის ქვეყნებთანაა შესაძლებელი კავშირი. </nobr>

</body>

</html>

საქართველო

საქართველო - ევროპისა და აზიის გასაყარზე, კერძოდ კავკასიაში მდებარეობს. საქართველო იმ უძველეს სატრანსპორტო გზაჯვარედინზე მდებარეობს, რომელიც საქართველო შავი ზღვით - შავი ზღვისპირეთის ქვეყნებს, ხოლო ბოსფორისა და

აკრონიმი

<abbr> ტეგი ტექსტის ფარგლებში აკრონიმის ანუ აბრევიატურის გამოყოფის საშუალებას იძლევა. <abbr> ტეგის შიგთავსის გამოტანა "მცურავი" მოკარნახის სახით ხდება, რომლებიც მომხმარებელს აკრონიმის გაშიფვრის საშუალებას აძლევს.

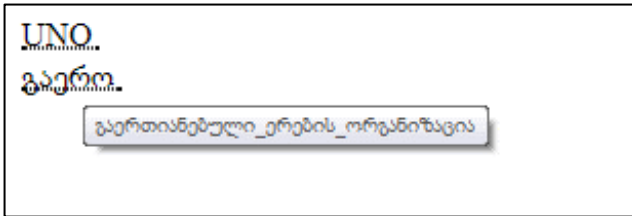
მაგალითი 2.8:

<body>

<!-- აკრონიმი ანუ აბრევიატურა ცნობილი დასახელებების შემოკლება -->

<abbr title=United_Nations_Organization> UNO </abbr>


```
<abbr title=გაერთიანებული_ერების_ორგანიზაცია> გაერო  
</abbr><br>  
</body>
```



არ უნდა დაგვავიწყდეს, რომ კოდში ტოლობის (=) ნიშნის გარშემო ინტერვალი არ უნდა გამოვიყენოთ.

HTML სიმბოლოები

მრავალი მათემატიკური, ტექნიკური და ვალუტის სიმბოლო ჩვეულებრივ კლავიატურაზე არ გვხვდება. ასეთი სიმბოლოების HTML კოდებში დასამატებლად შეგიძლიათ გამოიყენოთ ამ სიმბოლოების HTML-ისათვის განკუთვნილი სპეციალური დასახელება, ხოლო თუ ასეთი სახელი არ არსებობს, მაშინ შეგიძლიათ გამოიყენოთ მისი ათობითი ან თექვსმეტობითი კოდი.

მაგალითად, ჯამის აღმნიშვნელი სიმბოლოს გამოსატანად შეიძლება გამოვიყენოთ შემდეგი სამი ჩანაწერიდან ერთ-ერთი:

მაგალითი 2.9:

```
<body>  
<p>I will display &sum;</p>  
<p>I will display &#8721;</p>  
<p>I will display &#x2211;</p>
```

</body>

სამივე შემთხვევაში შედეგი ერთი და იგივე იქნება:

```
I will display Σ  
I will display Σ  
I will display Σ
```

UTF-8 მათემატიკური სიმბოლოებისათვის ათობითი კოდისათვის გამოყოფილია 8704-8959 დიაპაზონი, ხოლო თექვსმეტობითი კოდისათვის 2200-22FF დიაპაზონი.

თუ გსურთ რომელიმე ამ სიმბოლო HTML კოდში გამოყენება, მაშინ მათი HTML დასახელება ან მათი ათობითი, ან თექვსმეტობითი კოდი შეგიძლიათ მოიძიოთ დანართში მოყვანილ ცხრილში (დანართი 1).

ჰარების დასმა

თუ ტექსტურ რედაქტორში აკრეფთ ტექსტს და მასში ბევრ ჰარს გამოვიყენებთ, ბრაუზერი ამ ჰარების იგნორირებას მოახდენს და მათ მინიმუმამდე შეამცირებს, ხოლო თუ <listing> ტეგს გამოვიყენებთ, მაშინ ეს ტეგი ტექსტს განსხვავებული შრიფტით და ყველა იმ ჰარით გამოიტანს, რომლითაც იგი აიკრიფა.

მაგალითი 2.10:

```
<body bgcolor=#bbcfff>
```

რეკლამა - საქონელზე, მომსახურებასა და სამუშაოზე (შემდგომში - საქონელზე), ფიზიკურ და იურიდიულ პირებზე, იდეასა და წამოწყებაზე ნებისმიერი საშუალებითა და ფორმით გავრცელებული ინფორმაცია, რომელიც პირთა განუსაზღვრელი წრისათვისაა გამიზნული და ფიზიკური და იურიდიული პირების, საქონლის, იდეისა და წამოწყებისადმი ინტერესის ფორმირებასა და შენარჩუნებას, აგრეთვე საქონლის, იდეისა და წამოწყების რეალიზაციის ხელშეწყობას ემსახურება.

```
<listing> არსებობს                      შემდეგი                      სახის
```

რეკლამა:

```
სასაქონლო,                      კორპორატიული,                      სოციალური,
```

პოლიტიკური.

```
</listing>
```

```
</body>
```

რეკლამა - საქონელზე, მომსახურებასა და სამუშაოზე (შემდგომში - საქონელზე), ფიზიკურ და იურიდიულ პირებზე, იდეასა და წამოწყებაზე ნებისმიერი საშუალებითა და ფორმით გავრცელებული ინფორმაცია, რომელიც პირთა განუსაზღვრელი წრისათვისაა გამიზნული და ფიზიკური და იურიდიული პირების, საქონლის, იდეისა და წამოწყებისადმი ინტერესის ფორმირებასა და შენარჩუნებას, აგრეთვე საქონლის, იდეისა და წამოწყების რეალიზაციის ხელშეწყობას ემსახურება.

არსებობს
სასაქონლო,

შემდეგი
კორპორატიული,

სახის
სოციალური,

რეკლამა:
პოლიტიკური.

მსგავსი ფუნქცია აკისრია HTML-ის <pre> ტეგს.

HTML-ის <pre> ტეგს შიგნით განთავსებული ტექსტის გამოტანა ეკრანზე ხდება ფიქსირებული სიგანის შრიფტით და

ინარჩუნებს ყველა იმ ინტერვალს და ხაზს, როგორც არის ჩაწერილი HTML კოდში.

მაგალითი 2.11: ქვემოთ მოყვანილია ორი მაგალითი, სადაც ერთში გამოყენებულია ტეგი <pre>, ხოლო მეორეში - არა.

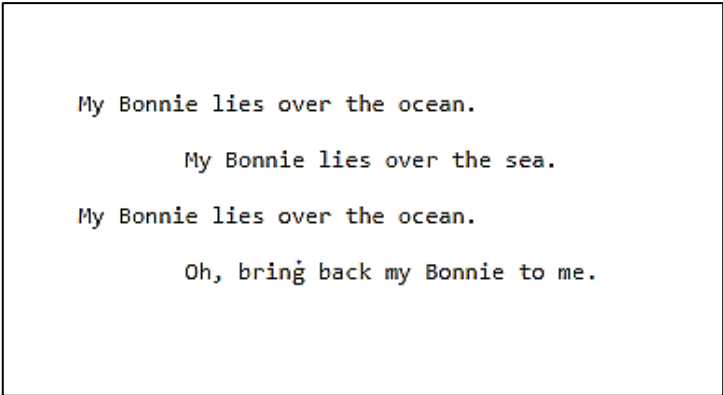
```
<body topmargin="100" leftmargin="50" >
<pre>
    My Bonnie lies over the ocean.

        My Bonnie lies over the sea.

            My Bonnie lies over the ocean.

                Oh, bring back my Bonnie to me.
</pre>
</body>
```

შედეგად, ამ შემთხვევაში შენარჩუნებულია ყველა ის ჰარი და სტრიქონიდან სტრიქონზე გადასვლა, როგორც ეს კოდში იყო მითითებული.



```
<body topmargin="100" leftmargin="50" >
```

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

```
</body>
```

მეორე შემთხვევაში ყველა ჰარი და სტრიქონიდან სტრიქონზე გადასვლა შეცვლილია ერთი ჰარით:

```
My Bonnie lies over the ocean. My Bonnie lies over the sea.  
My Bonnie lies over the ocean. Oh, bring back my Bonnie to  
me.
```

ჰორიზონტალური ხაზები

დოკუმენტში ჰორიზონტალური ხაზების ჩასასმელად `<hr>` ტეგი გამოიყენება. ამ ტეგს დახურვის ტეგი არ გააჩნია. სტანდარტულად ხაზი არის მოცულობითი და ჩრდილით.

ჰორიზონტალურ ხაზებს დიდი მოცულობის ტექსტი ცალკეულ ნაწილებად შეუძლია დაეოს. მისი სინტაქსია:

```
<hr size=რიცხვი width=რიცხვი|პროცენტი  
align=left|right|center noshade color="#xxxxxx">
```

<hr> ტეგის ატრიბუტებია:

- size - ხაზის სისქე პიქსელებში;
- width - ხაზის სიგრძე პიქსელებში ან ბრაუზერის ფანჯრის სიგანის პროცენტი;
- align - ეკრანზე მდებარეობა (მარცხნივ, მარჯვნივ, ცენტრში);
- color - ხაზის ფერი;
- noshade - ხაზი ერთფერი მუქი წრფის სახით იქნება მოცემული.

მაგალითი 2.12:

```
<body bgcolor="#ff8c00" leftmargin=60>
<h2 align=center>ლამაზი სქელი და წვრილი ხაზები </h2>
<hr>
<center>
Horizontal rule - ჰორიზონტალური ხაზი.
<h3 align=center> size = და width = მნიშვნელობა</h3>
align=center size = 40 width = &quot; 20 % &quot; noshade;
<hr align=center width="20%" noshade size=40>
align=center size = 30 width = &quot; 30 % &quot;
<hr align=center width="30%" color=#ff0000 size=30>
align=center size = 20 width = &quot; 40 % &quot;
<hr align=center width="40%" color=#00ff00 size=20>
align=left size = 10 width = &quot; 50 % &quot;
<hr align=left width="50%" color=#0000ff size=10>
align=right size = 5 width = &quot; 60 % &quot; noshade;
<hr align=right width="60%" noshade size=5>
</body>
```

მიღებული შედეგი:

ლამაზი სქელი და წვრილი ხაზები

Horizontal rule - ჰორიზონტალური ხაზი.

size = და width = მნიშვნელობა

align=center size = 40 width = " 20 % " noshade;



align=center size = 30 width = " 30 % "



align=center size = 20 width = " 40 % "



align=left size = 10 width = " 50 % "



align=right size = 5 width = " 60 % " noshade;



სიები

HTML დოკუმენტში ძირითადად სამი სახის სია არსებობს:

1. დანომრილი სიები;
2. მარკირებული სიები;
3. განსაზღვრებათა სია.

HTML-ში შესაძლებელია ჩადგმული სიების შექმნა. ამისათვის, უბრალოდ ერთი წყვილი ტეგი მეორე წყვილის შიგნით უნდა განვათავსოთ.

დანომრილი სიები

ბრაუზერი დანომრილ სიებში ელემენტთა ნომრებს მიმდევრობით, ავტომატურად სვამს. ეს ნიშნავს, რომ თუ სიიდან

დანომრილი სიის ერთ ან რამდენიმე ელემენტს ამოვიღებთ, დანარჩენი ნომრები ავტომატურად შეიცვლება.

დანომრილი სიები წყვილი (ინგლ. *Ordered List - დანომრილი სია*) ტეგის საშუალებით იქმნება. სიის თითოეული ელემენტი (ინგლ. *List Item - სიის ელემენტი*) ტეგით იწყება.

მაგალითი 2.13:

```
<!DOCTYPE html>
<html>
<head>
<title>დანომრილი სიები</title>
</head>
<body bgcolor = "floralwhite">
<h3> კომპიუტერების კლასიფიკაცია კონსტრუქციული
შესრულების თვალსაზრისით </h3 >
კონსტრუქციული შესრულების თვალსაზრისით
განასხვავებენ
<ol type=I>
<li>სამაგიდო (Desktop) </li>
<li>პორტატულ (Notebook) </li>
<li>ჯიბის (PDA) </li>
</ol>
კომპიუტერებს
</body>
</html>
```

მიღებული შედეგი:

კომპიუტერების კლასიფიკაცია კონსტრუქციული შესრულების თვალსაზრისით

კონსტრუქციული შესრულების თვალსაზრისით განასხვავებენ

- I. სამაგიდო (Desktop)
- II. პორტატულ (Notebook)
- III. ჯიბის (PDA)

კომპიუტერებს

 ტეგს შეიძლება type და start ატრიბუტი ჰქონდეს. მისი სინტაქსია: <ol type=A|a|I|i|1 start=n>

სადაც type სიის სიმბოლოებია:

A - ლათინური ანბანის დიდი ასოები (A, B, C . . .);

a - ლათინური ანბანის პატარა ასოები (a, b, c . . .);

I - რომაული დიდი ციფრები (I, II, III . . .);

i - რომაული პატარა ციფრები (i, ii, iii . . .);

1 - არაბული ციფრები (1, 2, 3 . . .);

start=n - სიის საწყისი მნიშვნელობა.

ქვემოთ მოყვანილია მაგალითი, სადაც ლათინური ანბანის დიდი ასოებით გადანომრილი სია ანბანის მე-11 ასოთი იწყება.

მაგალითი 2.14:

```
<body bgcolor=moccasin>  
ვებგვერდების მარკირების ენებია:<br>  
<pre> _____ </pre>  
<br>  
<ol type=A start=11>  
<li> HTML (Hyper Text Markup Language)</li>
```

```
<li> SGML (Standard Generalized Markup Language)</li>
<li> XML</li>
<li> XHTML</li>
<li> Dynamic HTML</li>
</ol>
<pre> _____ </pre>
</body>
```

ვებგვერდების მარკირების ენებია:

-
- K. HTML (Hyper Text Markup Language)
 - L. SGML (Standard Generalized Markup Language)
 - M. XML
 - N. XHTML
 - O. Dynamic HTML
-

მარკირებული სიები

მარკირებული სიები (ინგლ. *Unordered List*) ტეგის საშუალებით იქმნება. სიის თითოეული ელემენტი, როგორც დანომრილი სიების შემთხვევაში, აქაც (ინგლ. *List Item* - სიის ელემენტი) ტეგით იწყება. ტეგს შეიძლება type ატრიბუტი ჰქონდეს: <ul type=disc|circle|square>

type ატრიბუტი მარკერის სახეს განსაზღვრავს. თუ იგი მითითებული არ არის, მაშინ მარკირების ნიშანი ავტომატურად

disc იქნება. მისი მისაღები მნიშვნელობებია:

disc - გაფერადებული წრიული მარკერი;

circle - წრიული მარკერები;

square - კვადრატული მარკერები.

მაგალითი 2.15:

```
<body bgcolor=snow>  
<h2>სამაგიდო კომპიუტერები, თავის მხრივ, შემდეგ  
ჯგუფებად იყოფა:</h2>  
<br>  
<ul>  
<li>უნვერსალური ტიპის კომპიუტერები; </li>  
<li>სპეციალური დანიშნულების კომპიუტერები; </li>  
<li>სამაგიდო მინიკომპიუტერები; </li>  
<li>სერვერები. </li>  
</ul>  
</body>
```

სამაგიდო კომპიუტერები, თავის მხრივ, შემდეგ ჯგუფებად იყოფა:

- უნვერსალური ტიპის კომპიუტერები;
- სპეციალური დანიშნულების კომპიუტერები;
- სამაგიდო მინიკომპიუტერები;
- სერვერები.

ერთმანეთში ჩადგმული სიების მაგალითი ნაჩვენებია ქვემოთ.

მაგალითი 2.16:

```
<body bgcolor = "floralwhite">
<h3> კომპიუტერების კლასიფიკაცია კონსტრუქციული
შესრულების თვალსაზრისით </h3 >
<ol>
<li>სამაგიდო (Desktop):</li>
<ul type=square>
<li> უნივერსალური ტიპის კომპიუტერები; </li>
<li> სპეციალური დანიშნულების კომპიუტერები; </li>
<li> სამაგიდო მინიკომპიუტერები; </li>
<li> სერვერები. </li>
</ul>
<li>პორტატული (Notebook):
<ul type=circle>
<li> დიდი ზომის პორტატული პერსონალური
კომპიუტერები (Notebook); </li>
<li> მცირე ზომის პორტატული პერსონალური
კომპიუტერები (Netbook); </li>
<li> პლანშეტური პერსონალური კომპიუტერები (Tablet
PC). </li>
</ul>
<li>ჯიბის (PDA). </li>
</ol>
</body>
```

კომპიუტერების კლასიფიკაცია კონსტრუქციული შესრულების თვალსაზრისით

1. სამაგიდო (Desktop):
 - უნივერსალური ტიპის კომპიუტერები;
 - სპეციალური დანიშნულების კომპიუტერები;
 - სამაგიდო მინიკომპიუტერები;
 - სერვერები.
2. პორტატული (Notebook):
 - დიდი ზომის პორტატული პერსონალური კომპიუტერები (Notebook);
 - მცირე ზომის პორტატული პერსონალური კომპიუტერები (Netbook);
 - პლანშეტური პერსონალური კომპიუტერები (Tablet PC).
3. ჯიბის (PDA).

განსაზღვრებათა სია

განსაზღვრებათა სია სპეციალურად განლაგებული ტერმინებისა და მათი აღწერისაგან (განსაზღვრებისგან) შედგება. ვებგვერდებზე განსაზღვრებების სიის შესაქმნელად <dl>, <dt> და <dd> ტეგები გამოიყენება.

განსაზღვრებათა სია <dl> (ინგლ. *Definition List*) ტეგით იწყება. იგი <dt> (ტერმინი) და <dd> (ტერმინის აღწერა) ტეგებს მოიცავს.

<dt> ტეგი (ინგლ. *Definition Term* - განმსაზღვრელი სიტყვა, ტერმინი) ბლოკური არაწყვილი ტეგია, რომელიც განსაზღვრებათა სიაში ტერმინის ტექსტს მონიშნავს.

<dd> ტეგი (ინგლ. *Definition Description* - განმსაზღვრელი ტერმინის აღწერა) ბლოკური არაწყვილი ტეგია, რომელიც განსაზღვრებათა სიაში განსაზღვრების ტექსტს მონიშნავს.

მაგალითი 2.17:

```
<body bgcolor=" lemonchiffon ">
```

```
<dl>
```

```
<dt>ინტერნეტი
```

```
<dd>- (Internet) არის "მსოფლიო ქსელი", ერთმანეთზე მიერთებული კომპიუტერების საჯაროდ ხელმისაწვდომი ქსელი, სადაც მომხმარებელს, თუ აქვს უფლება, ინფორმაციის მიღება ნებისმიერი კომპიუტერიდან შეუძლია. </dl>
```

```
<dl>
```

```
<dt>ლოკალური ქსელი
```

```
<dd>- (ინგლ. local area network შემოკლებით LAN) მცირე ტერიტორიაზე (ჩვეულებრივ 1-2 კმ-ის რადიუსში) გავრცობილ კომპიუტერულ ქსელს წარმოადგენს. </dl>
```

```
<dl>
```

```
<dt>გლობალური ქსელი
```

```
<dd>- (ინგლ. Wide Area Network, WAN) - კომპიუტერული ქსელი, რომელიც დიდ ტერიტორიასა და დიდი რაოდენობის კომპიუტერებს მოიცავს. </dl>
```

```
<dl>
```

```
<dt>პროვაიდერი
```

```
<dd>- (ინგლ. provider, to provide - მომარაგება, უზრუნველყოფა) ორგანიზაცია ან ფირმა, რომელიც საინფორმაციო სისტემებთან (ინტერნეტთან) მოსარგებლის დაშვებას უზრუნველყოფს.
```

```
</dl>
```

```
<dl>
```

```
<dt>ჰიპერტექსტი (hypertext) ანუ ჰიპერტექსტური სისტემა
```

```
<dd>- სხვადახსვა ტიპის დოკუმენტების სისტემა,
```

რომელშიც ყოველი დოკუმენტი ამავე სისტემიდან რომელიმე სხვა დოკუმენტთანაა დაკავშირებული. </dl>

</body>

მიღებული შედეგი:

ინტერნეტი

- (Internet) არის "მსოფლიო ქსელი", ერთმანეთზე მიერთებული კომპიუტერების საჯაროდ ხელმისაწვდომი ქსელი, სადაც მომხმარებელს, თუ აქვს უფლება, ინფორმაციის მიღება ნებისმიერი კომპიუტერიდან შეუძლია.

ლოკალური ქსელი

- (ინგლ. local area network შემოკლებით LAN) მცირე ტერიტორიაზე (ჩვეულებრივ 1-2 კმ-ის რადიუსში) გავრცობილ კომპიუტერულ ქსელს წარმოადგენს.

გლობალური ქსელი

- (ინგლ. Wide Area Network, WAN) - კომპიუტერული ქსელი, რომელიც დიდ ტერიტორიასა და დიდი რაოდენობის კომპიუტერებს მოიცავს.

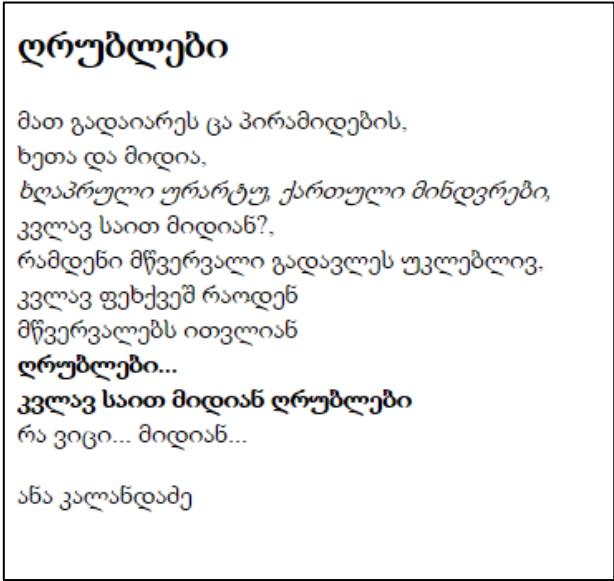
პროვაიდერი

- (ინგლ. provider, to provide - მომარაგება, უზრუნველყოფა) ორგანიზაცია ან ფირმა, რომელიც საინფორმაციო სისტემებთან (ინტერნეტთან) მოსარგებლის დაშვებას უზრუნველყოფს.

ჰიპერტექსტი (Hypertext) ანუ ჰიპერტექსტური სისტემა

- სხვადასხვა ტიპის დოკუმენტების სისტემა, რომელშიც ყოველი დოკუმენტი ამავე სისტემიდან რომელიმე სხვა დოკუმენტთანაა დაკავშირებული.

დავალება 1. შექმენით html ფაილი. ბრაუზერის ფანჯრის სათაურის ზოლში უნდა გამოჩნდეს დოკუმენტის სახელწოდება „ღრუბლები“. ვებგვერდი ფონი და ტექსტის ფერი შეარჩიეთ თქვენი სურვილის მიხედვით. მოცემული დავალების შესრულების შედეგად ბრაუზერის ფანჯარაში მიღებული შედეგი ნაჩვენებია ქვემოთ:



გამოიყენეთ <h2>, <p> , ,
 ტეგები. კოდში კომენტარის სახით დაამატეთ ტექსტი „ანა კალანდაძის პოეზია“.

დავალება 2. შექმენით html ფაილი, რომლისთვისაც ბრაუზერის ფანჯარაში მიღებულ შედეგს ექნება შემდეგი სახე (ვებგვერდის ფონი და ტექსტის ფერი შეარჩიეთ თქვენი სურვილის მიხედვით):

დაპროგრამების ენები

- C++
- C#
- Java
- Python
- PHP

დავალეზა 3. შექმენით ჩადგმული დანომრილი და მარკირებული სიები ქვემოთ მოცემული ნიმუშის მიხედვით:

ქართველი მწერლები

- ილია ჭავჭავაძე
 - მგზავრის წერილები
 - აჩრდილი
 - განდევილი
- აკაკი წერეთელი
 1. პოეზია
 - აღმართ-აღმართ
 - გზაფხული
 2. მოთხრობები
 - ჩემი თავგადასავალი
 - ზაში აჩუკი
- ვაჟა ფშაველა
 - არწივი
 - კიდევაც ვნახავ გზაფხულს
- გალაკტიონ ტაბიძე
 1. გახედე:კახეთი
 2. დროშები ჩქარა

თავი 3

გრაფიკული გამოსახულება HTML დოკუმენტში

- გრაფიკული გამოსახულების ჩასმა **HTML** დოკუმენტში;
- გამოსახულების სწორება ტექსტის მიმართ;
- გამოსახულების მასშტაბირება;
- ტეგი <picture>;
- გამოსახულების რუკა (გამოსახულების დამლა
არეზად).

გრაფიკული გამოსახულების ჩასმა Html დოკუმენტში

ვებგვერდის გრაფიკული ელემენტის განთავსებისათვის (ინგლ. *image* - გამოსახულება) ტეგი გამოიყენება. მისი ატრიბუტებია SRC, ALT, HEIGHT, WIDTH, USEMAP, ISMAP, ALIGN, BORDER, HSPACE, VSPACE. მაგალითად,

```
.
```

- *SRC* ატრიბუტი (ინგლ. *source* - წყარო) დოკუმენტში გამოყენებული გამოსახულების ადგილმდებარეობას განსაზღვრავს. URL აუცილებელი პარამეტრია, რომელიც ბრაუზერს უჩვენებს, სად მდებარეობს სურათი. სურათი გრაფიკულ ფორმატში უნდა იყოს შენახული, მაგალითად, GIF ან JPG ფორმატში. თუ გრაფიკული ფაილი იმავე საქაღალდეში მდებარეობს, სადაც HTML დოკუმენტია, მაშინ მხოლოდ ფაილის სახელის მითითებაა საკმარისი. თუ ფაილი იმავე სერვერზე მდებარეობს, სადაც HTML დოკუმენტი, მაშინ საქაღალდის სახელიც უნდა მივუთითოთ, წინააღმდეგ შემთხვევაში საჭიროა სრული მისამართის მითითება. მაგალითად, .

- *ALT* ატრიბუტი აუცილებელი არ არის და თუ ის მითითებულია, მაშინ, სანამ მიმდინარეობს გამოსახულების ჩატვირთვა, ბრაუზერს ალტერნატიული ტექსტი გამოაქვს;

- *HEIGHT* ატრიბუტი აუცილებელი არ არის და იგი სურათის სიმაღლეს განსაზღვრავს პიქსელებში. თუ მოცემული პარამეტრი არ არის მითითებული, მაშინ სურათი თავისი ორიგინალური სიმაღლით გამოდის, თუმცა ზოგიერთი

ბრაუზერი მოცემულ ატრიბუტს მხარს არ უჭერს. ზოგჯერ სხვადასხვა კომპიუტერზე მონიტორის გარჩევადობის მაჩვენებელი სხვადასხვაა, რის გამოც ამ ატრიბუტის მითითების დროს ყურადღების გამოჩენაა საჭირო;

- *WIDTH* ატრიბუტი აუცილებელი არ არის და იგი სურათის სიგანეს განსაზღვრავს პიქსელებში;

- *ALIGN* ატრიბუტი აუცილებელი არ არის და იგი ეკრანზე სურათის ზუსტი პოზიციონირებისთვის გამოიყენება. მისი მნიშვნელობებია:

- top - ობიექტის ზედა კიდის სწორება სტრიქონის ზედა კიდის მიმართ;
- middle - ობიექტის ცენტრის სწორება სტრიქონის საბაზო ხაზის მიმართ;
- bottom - ობიექტის ქვედა კიდის სწორება სტრიქონის საბაზო ხაზის მიმართ;
- left - ობიექტის სწორება მარცხენა კიდის მიმართ, ამასთან ობიექტის ტექსტით გარს შემოვლანა შესაძლებელი;
- right - ობიექტის სწორება მარჯვენა კიდის მიმართ, ამასთან ობიექტის ტექსტით გარს შემოვლანა შესაძლებელი.

თუ მოცემული პარამეტრი არ არის მითითებული, მაშინ ბრაუზერების უმეტესობა გამოსახულებას მარცხენა მხარეს, ხოლო ტექსტს მარჯვენა მხარეს განათავსებს.

- *BORDER* ატრიბუტი გამოსახულების ჩარჩოს სისქეს განსაზღვრავს;

- *VSPACE* ატრიბუტი გამოსახულების ზემოთ და ქვემოთ ცარიელი არის ზომას მიუთითებს პიქსელებში, რათა ტექსტი გამოსახულებაზე არ გადავიდეს;

- *HSPACE* ატრიბუტი გამოსახულების მარცხნივ და

მარჯვნივ ცარიელი არის ზომას მიუთითებს პიქსელებში, რათა ტექსტი გამოსახულებაზე არ გადავიდეს.

მაგალითი 3.1:

```
<body topmargin=40 leftmargin=50>  
  
</body>
```

ავტომატურად სურათი ვებგვერდის ზედა მარცხენა კუთხეში განთავსდა. მოცემულ მაგალითში სურათის ზომა იყო 485 პიქსელი სიგანეში და 500 პიქსელი - სიმაღლეში.



ქვემოთ მოყვანილია მაგალითი, სადაც სურათის განთავსების ადგილი (ცენტრი) და სურათის ზომებია (სიგანე 281, სიმაღლე 300 პიქსელი) მითითებული, ასევე გაკეთებულია წარწერა სურათის ქვეშ.

მაგალითი 3.2:

```
<body topmargin=40 leftmargin=50>  
<center>  
  
<br><font size=6>  
დაცვარული გვირილა  
</center>  
</font>  
</body>
```



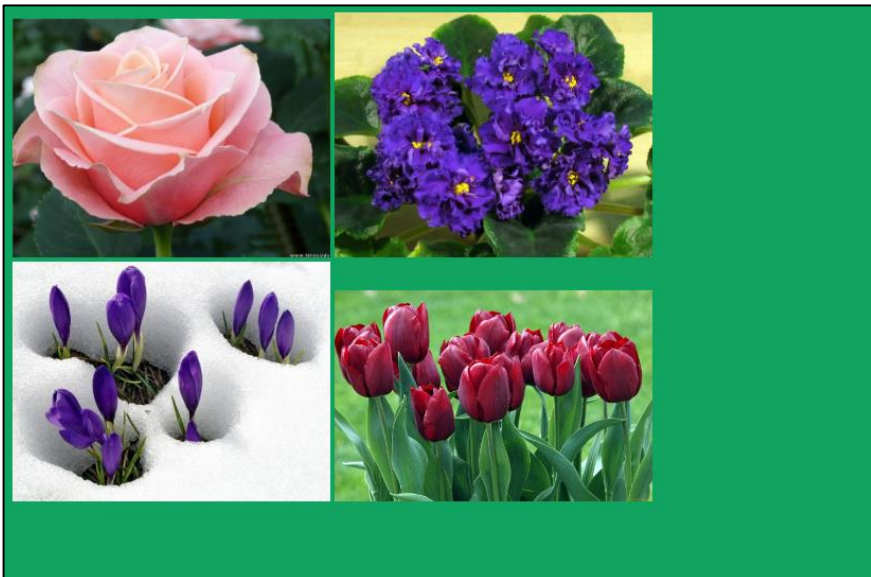
დაცვარული გვირილა

3.3 მაგალითიდან ჩანს, რომ ვებგვერდზე ოთხი სურათია განთავსებული, თითოეული 225 პიქსელი სიმაღლისა და 300 პიქსელი სიგანის.

მაგალითი 3.3:

```
<body bgcolor=#11a360>
```

```
  
 <br>  
  
  
</body>
```



ქვემოთ მოცემულ მაგალითში პირველი სურათი გასწორებულია ვებგვერდის მარჯვენა კიდის მიმართ, ხოლო მეორე - მარცხენა კიდის მიმართ.

მაგალითი 3.4:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>გრაფიკა ტექსტით</title>
```

</head>

<body>

<h1>გვირილა</h1>

 გვირილა ყვავილთა სამეფოს კონკიაა - გარეგნულად მოკრძალებული, მაგრამ ძალიან სასარგებლო, რაც მას დიდ უპირატესობას ანიჭებს ფლორის სხვა უფრო მნიშვნელოვან და ლამაზ წარმომადგენლებთან შედარებით. იშვიათია ადამიანი, ვისაც ეს საოცარი მცენარე არ გამოუყენებია. მისი ნაყენი უებარი სავლებია ყელის, კბილის, ღრძილების ტკივილის დროს. პატარებს მისი ნახარში მუცლის ტკივილს უამებს. ეს მცენარე შედის მრავალი იმ მედიკამენტის შედგენილობაში, რომლებსაც სტომატიტის, გინგივიტის, კოლიკის საწინააღმდეგოდ იყენებენ. ლათინური სახელი მატრიხ გვირილას იმის გამო ეწოდა, რომ უამრავ ქალურ პრობლემას შველოდა. ავიცენა გვირილას დაუძღვრების დროს უცვლელ საშუალებად მიიჩნევდა. გვირილას თანაბარი სიხშირით იყენებენ როგორც ბავშვებისთვის, ისე მოზრდილებისთვის.

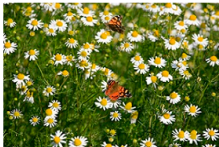
ეს ძალიან მსუბუქი მცენარეული საშუალებაა, რომლის წინაშე უკან იხევს მრავალი სნეულება, მათ შორის - სასუნთქ ორგანოთა ქრონიკული დაავადებები, დიათეზი, დერმატიტი, ეგზემა. გვირილა ერთ-ერთია იმ მცირერიცხოვან სამკურნალო მცენარეთაგან, რომელთა მიღებაც რეკომენდებულია კუჭის მაღალი მჟავიანობის დროს. გვირილის ნაყენი ხსნის სპაზმს, კუჭის ლორწოვანი გარსის შეშუპებას, აძლიერებს ნაღვლის გამომუშავებას, ხელს უწყობს კუჭისა და თორმეტგოჯა

ნაწლავის წყლულის სწრაფ შეხორცებას, ამშვიდებს ცენტრალურ ნერვულ სისტემას. ხალხური მედიცინა გვირილას იყენებს ნაყენის, ნახარშის, ჩაის, აბაზანის, აპლიკაციის, საფენისა და საველების სახით. თანამედროვე ექიმებიც ფართოდ იყენებენ გვირილის სამკურნალო თვისებებს, რეკლამას უწევენ მას როგორც სწრაფმოქმედ საშუალებას სურდოს დროს. კარაბადინებში იოლად იპოვით გვირილისგან დამზადებული საშუალებების რეცეპტებს.

```
<br>
</font>
</body>
</html>
```

გვირილა

გვირილა ყვავილთა სამეფოს კონკია - გარეგნულად მოკრძალებული, მაგრამ ძალიან სასარგებლო, რაც მას დიდ უპირატესობას ანიჭებს ფლორის სხვა უფრო მნიშვნელოვან და ლამაზ წარმომადგენლებთან შედარებით. იშვიათია ადამიანი, ვისაც ეს საოცარი მცენარე არ გამოუყვებია. მისი წაყენი უნებარი სველებია ყელის, კბილის, ღრძილების ტკივილის დროს. პატარებს მისი წახარში მუცლის ტკივილს უშავებს. ეს მცენარე შედის მრავალი იმ მედიკამენტის შედგენილობაში, რომლებსაც სტომატიტის, გინგივიტის, კოლიკის საწინააღმდეგოდ იყენებენ. ლათინური სახელი მატრის გვირილას იმის გამო ეწოდა, რომ უამრავ ქალურ პრობლემას შევლოდა. ავიცნა გვირილას დაუმღებების დროს უცვლელ საშუალებად მიიჩნევა. გვირილას თანაბარი სიხშირით იყენებენ როგორც ბავშვებისთვის, ისე მოზრდილებისთვის.



ეს ძალიან მსუბუქი მცენარეული საშუალებაა, რომლის წინაშე უკან იხევს მრავალი სწველება. მათ შორის - სასუნთქ ორგანოთა ქრონიკული დაავადებები, დიათეზი, დერმატიტი, ეგზემა. გვირილა ერთ-ერთია იმ მცირერიცხოვან სამკურნალო მცენარეთაგან, რომელთა მიღებაც რეკომენდებულია კუჭის მალალი მკვიანობის დროს. გვირილის წაყენი ხსნის საზუმს, კუჭის ლორწოვანი გარსის შემუშვებს, აძლიერებს ნაღლის გამომუშავებას, ხელს უწყობს კუჭისა და თორმეტგოჯა ნაწლავის წყლულის სწრაფ შეხორცებას, ამშვიდებს ცენტრალურ ნერვულ სისტემას. ხალხური მედიცინა გვირილას იყენებს წაყენის, წახარშის, ჩაის, აბაზანის, აპლიკაციის, საფენისა და საველების სახით. თანამედროვე ექიმებიც ფართოდ იყენებენ გვირილის სამკურნალო თვისებებს, რეკლამას უწევენ მას როგორც სწრაფმოქმედ საშუალებას სურდოს დროს. კარაბადინებში იოლად იპოვით გვირილისგან დამზადებული საშუალებების რეცეპტებს.

გამოსახულების მასშტაბირება

ქვემოთ მოყვანილია ვებგვერდის მაგალითი, რომელიც შეიცავს ერთი და იგივე სხვადასხვა ზომის სურათს. ფაილის ზომებია 266 პიქსელი სიგრძეში და 400 პიქსელი სიგანეში. პირველი გამოსახულების ზომა ორჯერ არის გაზრდილი, მეორეს

ტეგი <picture>

ტეგი <picture> საშუალებას გვაძლევს სხვადასხვა მოწყობილობის ან სხვადასხვა ზომის ეკრანზე გამოტანილ იქნეს სხვადასხვა სურათი.

ტეგი <picture> შემოტანილი იქნა HTML5-ში გამოსახულების რესურსების მითითების დროს უფრო მეტი მოქნილობისათვის. <picture> კონტეინერულ ტეგებს შორის რამდენიმე <source> ტეგია განთავსებული, რომელთაგან თითოეული სხვადასხვა გამოსახულებაზე მიუთითებს. ამგვარად, ბრაუზერს შეუძლია აარჩიოს გამოსახულება, რომელიც ყველაზე მეტად შეესაბამება არსებულ მოწყობილობას.

თითოეულ <source> ტეგს აქვს ატრიბუტები, რომლებიც იმ გამოსახულებას აღწერს, რომელიც ყველაზე მეტად შეესაბამება მოწყობილობას.

მაგალითი 3.6. სხვადასხვა ზომის ეკრანისათვის მოცემულია სხვადასხვა გამოსახულება:



```
<picture>
```

```
<source media="(min-width: 650px)" srcset="img_food.jpg">
```

```
<source media="(min-width: 465px)" srcset="img_car.jpg">

</picture>
```

შენიშვნა: <picture> კონტეინერულ ტეგებს შორის ბოლო ტეგი ყოველთვის უნდა იყოს . ეს ტეგი იმ ბრაუზერების მიერ იქნება გამოყენებული, რომელიც <picture> ტეგს მხარს არ უჭერს ან ზემოთ მითითებული არც ერთი <source> ტეგი არ შეესაბამება ამ ბრაუზერს.

<picture> ტეგის გამოყენებას ორი ძირითადი მიზანი აქვს:

1. გამტარუნარიანობა. თუ გაქვთ მცირე ზომის ეკრანი ან მოწყობილობა, მაშინ აუცილებელი არ არის დიდი ზომის გამოსახულების ფაილის ჩამოტვირთვა. ბრაუზერი გამოიყენებს პირველ <source> ტეგს, რომელსაც ატრიბუტების შესაბამისი მნიშვნელობები ექნება და დანარჩენი ტეგების იგნორირებას მოახდენს;

2. ფორმატის მხარდაჭერა. ზოგიერთი ბრაუზერი ან მოწყობილობა შეიძლება გამოსახულების რომელიმე ფორმატს მხარს არ უჭერდეს. <picture> ტეგის გამოყენების შემთხვევაში შეგიძლიათ დაამატოთ ყველა ფორმატის გამოსახულება და ბრაუზერი თვითონ ამოირჩევს პირველ, მისთვის მისაღებ ფორმატს, ხოლო ყველა მომდევნო ფორმატის იგნორირება მოხდება.

მაგალითი 3.7:

```
<picture>
  <source e srcset="img_girl.jpg">
  
</picture>
```

ბრაუზერი გამოიყენებს პირველ, მისთვის მისაღებ გამოსახულების ფორმატს.

გამოსახულების რუკა (გამოსახულების დაშლა არეებად)

გრაფიკული გამოსახულების რუკა (image maps) საშუალებას იძლევა გამოსახულების ცალკეული ნაწილები სხვა დოკუმენტებს ბმულით დაუკავშიროთ (იხ. თავი ბმულები). გამოსახულების ცალკეულ ნაწილებზე ე.წ. აქტიურ არეებზე (hot spots) მაუსით დაწკაპუნების შემთხვევაში მომხმარებელს ამა თუ იმ მოქმედების შესრულება შეუძლია, მაგალითად, სხვა დოკუმენტებზე გადასვლა.

თუ მომხმარებელი გამოსახულებაზე, რომელიც `` ტეგში `ismap` ატრიბუტის საშუალებით განსაზღვრულია როგორც გამოსახულების რუკა, მაუსის მაჩვენებლით პოზიციონირებს და მასზე მარცხენა კლავიშით დააწკაპუნებს, ვებსერვერს გადაეცემა დაწკაპუნების წერტილის კოორდინატები. ამ კოორდინატების მიღების შემდეგ სერვერი რუკაზე ამ კოორდინატების შესაბამის აქტიურ არეებს ეძებს. თუ ასეთი არე არსებობს, მაშინ მოცემული URL მისამართი გააქტიურდება და მომხმარებლის ბრაუზერი ამ მისამართით ახალ გვერდზე გადავა.

ფუნქციონირების აღწერილი ვარიანტი იმ შემთხვევისათვისაა განკუთვნილი, როდესაც გამოსახულების რუკა სერვერზეა განთავსებული, მაგრამ იგი ასევე შეიძლება კლიენტის სერვერზეც იყოს განთავსებული. ამ შემთხვევაში ბრაუზერი საჭირო გვერდის URL მისამართს დამოუკიდებლად არკვევს და სერვერს გადასცემს არა კოორდინატებს, არამედ მის მისამართს.

ჩვენთვის მეორე შემთხვევა უფრო საინტერესოა, ვინაიდან გამოსახულების აქტიური არეების დასადგენად შესასრულებელი პროგრამის (სცენარის) გაშვება სერვერზე არ გვიწევს.

HTML დოკუმენტში გამოსახულების ბმულთან დაკავშირებულ სხვადასხვა არედ დასაშლელად (საიდანაც სხვა დოკუმენტზე ან ვებგვერდზე გადასვლა იქნება შესაძლებელი) `<map>` და `<area>` ტეგები გამოიყენება. გამოსახულება, რომელზეც ვაპირებთ მუშაობას, `` ტეგის საშუალებით უნდა იყოს რეალიზებული, სადაც `usemap` და `src` პარამეტრები იქნება გამოყენებული. მაგალითად,

``

ტეგი `<map>`. `<map>` ტეგი გრაფიკული გამოსახულების რუკის კონფიგურაციის აღწერისათვის გამოიყენება და მას ერთადერთი `name` პარამეტრი აქვს. აქტიურ არეებზე მაუსის მაჩვენებლის მიყვანის შემდეგ ამ არის იდენტიფიცირება მცურავი მოკარნახით შეიძლება. რუკის აღწერისათვის `<map>` და `</map>` კონტეინერული ტეგი გამოიყენება. ამ კონტეინერულ ტეგებს შორის გამოსახულების აქტიური არეები უნდა იყოს აღწერილი, რისთვისაც სპეციალური ტეგი `<area>` გამოიყენება. `<map>` ტეგი `` ან `<input>` ტეგთან `usemap` ატრიბუტით შეიძლება იყოს დაკავშირებული. `<map>` ტეგის `name` პარამეტრის მნიშვნელობა `usemap` პარამეტრის მნიშვნელობას უნდა ემთხვეოდეს.

ტეგი `<area>`. `<area>` ტეგი კლიენტის მხარეს არსებული გამოსახულების რუკის მხოლოდ ერთ-ერთი აქტიური არის შესახებ გვაძლევს ინფორმაციას. მას არ სჭირდება დახურვის ტეგი. `<map>` და `</map>` ტეგებს შორის თითოეული არის აღწერისთვის ცალ-ცალკე `<area>` ტეგი უნდა იქნეს

გამოყენებული. იმ შემთხვევაში, თუ რომელიმე წერტილი ერთდროულად რამდენიმე აქტიურ არეს ეკუთვნის, ის ბმული შესრულდება, რომელიც ამ არეების აღწერის სიაში პირველია. მისი სინტაქსი შემდეგია:

```
<map name="რუკის_სახელი"><area ატრიბუტები></map>
```

<area> ტეგს შემდეგი ატრიბუტები აქვს: shape, coords, href, nohref და title. განვიხილოთ თითოეული მათგანი:

- ატრიბუტი share. გამოსახულებაზე მოსანიშნი არე შეიძლება სხვადასხვა ფორმის იყოს. სწორედ, ამიტომ share ატრიბუტი აქტიური არის ფორმას განსაზღვრავს. მისი შესაძლო მნიშვნელობებია: rect, circle, poly, default. ეს მნიშვნელობები შესაბამისად მართკუთხა, მრგვალ- და მრავალკუთხა ფორმის არეებს გვაძლევს, ხოლო default-ი არის ყველა წერტილს განსაზღვრავს. რუკის არის განმსაზღვრელი ატრიბუტის მითითება აუცილებელია.

შენიშვნა. სერვერზე განთავსებული რუკისაგან განსხვავებით, სადაც default-ი გამოსახულების ყველა იმ წერტილს განსაზღვრავს, რომელიც რომელიმე აქტიურ არეს არ ეკუთვნის, კლიენტის სერვერზე განთავსებული რუკის შემთხვევაში default-ი გამოსახულების ყველა წერტილს განსაზღვრავს. ამიტომ, default-ი აქტიური არეების ნუსხის ბოლოს უნდა იყოს გამოყენებული. მაგალითად, თუ ამ ნუსხიდან პირველ share-ს მნიშვნელობა default-ი ექნება მინიჭებული, მაშინ ის ყველა აქტიური არის ბმულზე გავრცელდება.

- ატრიბუტი coords. coords ატრიბუტის საშუალებით ცალკეული აქტიური არის კოორდინატებს ვუთითებთ, ე. ი. მისი მნიშვნელობა სხვადასხვა წერტილის კოორდინატთა

ნუსხაა, რომლებიც ერთმანეთისგან მძიმით გამოიყოფა და ნატურალური რიცხვებით გამოისახება. კოორდინატის საწყისად გამოსახულების ზედა მარცხენა კუთხე ითვლება, რომელსაც 0,0 მნიშვნელობა შეესაბამება. პირველი რიცხვი ჰორიზონტალურ, ხოლო მეორე ვერტიკალურ კოორდინატს განსაზღვრავს.

მართკუთხა ფორმის არისათვის ანუ shape=rect შემთხვევაში, პირველად იწერება ზედა მარცხენა წერტილის, ხოლო შემდეგ ქვედა მარჯვენა წერტილის კოორდინატი. მაგალითად,

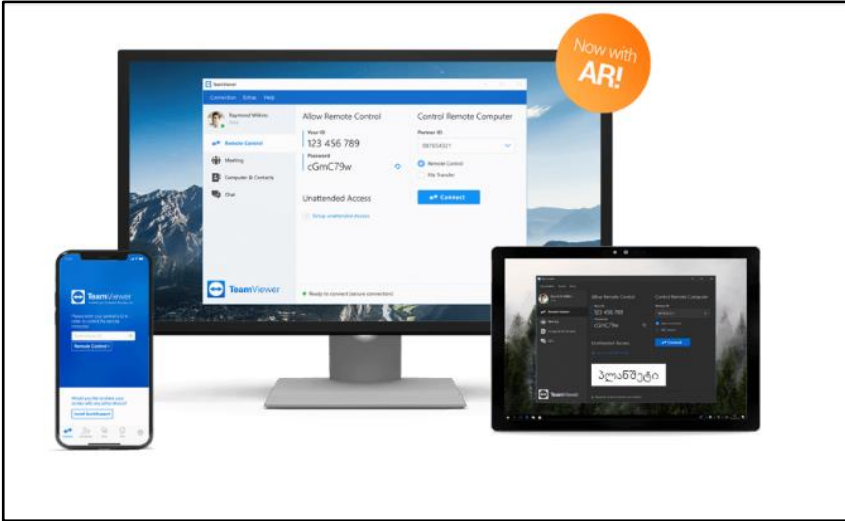
```
<area shape= rect coords="33,60,190,250">
```

განვიხილოთ კონკრეტული HTML დოკუმენტი, სადაც მართკუთხა არეა გამოყენებული.

მაგალითი 3.8:

```
<body>

<map name="tablet">
<area shape=rect coords="550,280,820,450" href=ttt.html
title="პლანშეტი" >
</map>
</body>
```



წრიული ფორმის არისათვის ანუ shape=circle შემთხვევაში, პირველად იწერება ცენტრის კოორდინატი და შემდეგ კი წრის რადიუსი პიქსელებში (ანუ მნიშვნელობა გამოისახება სამი რიცხვით). მაგალითად,

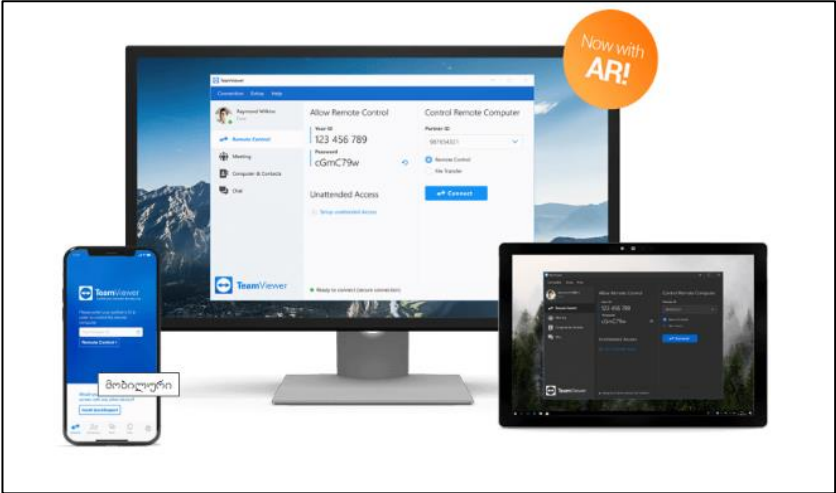
```
<area shape=circle coords="250, 150, 70">
```

განვიხილოთ კონკრეტული HTML დოკუმენტი, სადაც წრიული ტიპის არეა გამოყენებული.

მაგალითი 3.9:

```
<body>

<map name="mobile">
<area shape=circle coords="150,350,70" href=ttt.html
title="მობილური" >
</map>
</body>
```

მრავალკუთხა ფორმის არისათვის ანუ `shape=poly` შემთხვევაში, პირველად იწერება ზედა წერტილის კოორდინატი, შემდეგ თანამიმდევრობით სხვა წერტილის კოორდინატები, ხოლო ბოლო კოორდინატი არ არის აუცილებელი ემთხვეოდეს პირველს. ინტერნეტბრაუზერი ავტომატურად აერთებს ბოლო წერტილს პირველთან. მაგალითად,

```
<area shape=POLY
coords="534,62,699,62,698,236,626,261,534,235">
```

როდესაც `shape=default`, მაშინ იგი არ საჭიროებს კოორდინატების მითითებას.

- ატრიბუტები `href` და `nohref`. `href` ატრიბუტის საშუალებით გამოსახულების რუკის აქტიური არე ამ url მისამართის საშუალებით მითითებულ ბმულს უკავშირდება. თუ `<area>` ტეგში ეს ატრიბუტი არ არის მითითებული ან მითითებულია

nohref ატრიბუტი, მაშინ ეს ნიშნავს, რომ მოცემული აქტიური არე გარკვეულ ბმულს არ უკავშირდება.

- ატრიბუტი alt . ბრაუზერისათვის, რომლებიც <area> ტეგს მხარს არ უჭერს, alt ატრიბუტი გამოსახულებაზე არსებული ყოველი აქტიური არისთვის ალტერნატიულ ტექსტს მიუთითებს. სინამდვილეში, ეს ტექსტი, თავად დოკუმენტის შემქმნელისთვის, კომენტარის როლს ასრულებს.

- ატრიბუტი title. title ატრიბუტით მოკარნახე ტექსტის გამოტანაა შესაძლებელი, რომელიც მოცემულ არეზე მაუსის მაჩვენებლის მიტანის დროს ამოცურდება. მისი სინტაქსია:

```
<area title = "კარნახის ტექსტი">
```

მაგალითი 3.10:

```
<body>

<map name="technic">
<area shape=rect coords="550,280,820,450" href=Index1.html
title="პლანშეტი">
<area shape=circle coords="150,350,70" href=Index2.html
title="მობილური">
</map>
</body>
```

დავალება. შექმენით ვებგვერდი, სადაც:

1. პირველ ხაზზე, ცენტრში, ყველაზე დიდი ზომის სათაური - **საქართველოს ქალაქები**;
2. მეორე ხაზზე მომდევნო დონის სათაური ცენტრში - **თბილისი**;
3. შემდეგ განათავსეთ თბილისის სურათი და ისტორია. სურათი ტექსტის მარცხნივ;

თბილისი საქართველოს დედაქალაქია; მდებარეობს აღმოსავლეთ საქართველოში, თბილისის ქვაბულში, მდინარე მტკვრის ორივე სანაპიროზე, ზღვის დონიდან 380—600 მ სიმაღლეზე, ჩრდილოეთით ესაზღვრება საგურამოს ქედის სამხრეთი მთისწინეთი, აღმოსავლეთით — ივრის ზეგნის ჩრდილო-დასავლეთი მონაკვეთი, დასავლეთით და სამხრეთით კი — თრიალეთის ქედის განშტოებები. ქალაქს 720 კმ² ფართობი უჭირავს და 1.090 მილიონზე მეტი მცხოვრებია. თბილისი კავკასიის რეგიონის მნიშვნელოვანი ინდუსტრიული, სოციალური და კულტურული ცენტრია და ბოლო დროს ერთ-ერთი უმნიშვნელოვანესი სატრანსპორტო კვანძია გლობალური ენერგომატარებლებისა და სავაჭრო პროექტებისთვის.

4. შემდეგ, როგორც განათავსეთ თბილისი, ისეთივე შრიფტით **თელავი**;
5. შემდეგ კი თელავის სურათი და ისტორია. სურათი ტექსტის მარჯვნივ.

თელავი - ქალაქი აღმოსავლეთ საქართველოში, გომბორის ქედის ჩრდილო-აღმოსავლეთ კალთის ძირას და ალაზნის ვაკეზე. კახეთის მხარის და თელავის მუნიციპალიტეტის

ადმინისტრაციული ცენტრი. თბილის-თელავის სარკინიგზო ხაზის ბოლო სადგური. ზღვის დონიდან 550-800 მ, თბილისიდან 158 კმ. ქალაქად გამოცხადდა 1801 წელს. მოსახლეობა 21,8 ათასი (2002 წ.).

ახლანდელ თელავის ტერიტორიაზე პირველი დასახლება გვიანდელი ბრინჯაოს ხანაში გაჩნდა, ელინისტურ ხანაში განვითარდა, ხოლო გვიანდელ ანტიკურ ხანაში მის ბაზაზე ქალაქის ტიპის დასახლება წარმოიქმნა.

თავი 4

ცხრილები

- ცხრილების შექმნის პრინციპი;
- მარტივი ცხრილების შექმნა;
- რთული ცხრილების შექმნა.

ცხრილების შექმნის პრინციპები

ცხრილების შექმნის პრინციპი შემდეგში მდგომარეობს: ვებგვერდზე იქმნება ცხრილი, უჯრების უხილავი საზღვრებით, უჯრებში ჩაწერილი მონაცემები ვებგვერდზე ამ უჯრების შესაბამისად განლაგდება. თითოეული უჯრისთვის შეგიძლიათ დაფორმატების საკუთარი პარამეტრები მიუთითოთ. პრაქტიკაში ხშირად ტექსტის რამდენიმე სვეტად დაყოფა გვჭირდება. ცხრილები სწორედ ამის საშუალებას იძლევა.

ცხრილი იწყება `<table>` ტეგით და მთავრდება `</table>`-ით. ნაგულისხმევი წესის თანახმად, ცხრილს ჩარჩო არ აქვს და უჯრების საზღვრებიც უხილავია.

`<table>` ტეგის ატრიბუტებია: `align`, `border`, `background`, `bgcolor`, `bordercolor`, `cellpadding`, `cellspacing`, `hspace`, `vspace`, `colspec`, `width`.

- ატრიბუტი `align` - ცხრილის მდებარეობას განსაზღვრავს და შემდეგი მნიშვნელობების მიღება შეუძლია: `left` (მარცხნივ), `right` (მარჯვნივ) ან `center` (ცენტრში);

- ატრიბუტი `border` - ცხრილის ჩარჩოს ტიპს განსაზღვრავს. თუ მოცემული ატრიბუტი გამოყენებულია, მაშინ ჩარჩო ყველა უჯრას და მთლიანად ცხრილს გაუკეთდება. `border` ატრიბუტს შეუძლია მიიღოს რიცხვითი მნიშვნელობა, რომელიც ჩარჩოს სისქეს განსაზღვრავს პიქსელებში;

- ატრიბუტი `background` - გამოიყენება ცხრილის ფონად რაიმე სურათის ჩასასმელად;

- ატრიბუტი `bgcolor` - ცხრილის ფონის ფერს განსაზღვრავს;

- ატრიბუტი `bordercolor` - ჩარჩოს ფერს მიუთითებს და მხოლოდ `border` ატრიბუტთან ერთად გამოიყენება;

- ატრიბუტი `cellpadding` - უჯრის ჩარჩოსა და უჯრაში მოთავსებულ ელემენტს (ტექსტი, სურათი) შორის მანძილს განსაზღვრავს პიქსელებში;
- ატრიბუტი `cellspacing` - უჯრებს შორის მანძილს განსაზღვრავს პიქსელებში;
- ატრიბუტი `hspace` - ცხრილის მარჯვნივ და მარცხნივ ცარიელ ადგილს განსაზღვრავს პიქსელებში;
- ატრიბუტი `vspace` - ცხრილის ზემოთ და ქვემოთ ცარიელ ადგილს განსაზღვრავს პიქსელებში;
- ატრიბუტი `colspec` - ფიქსირებული სიგანის სვეტებს განსაზღვრავს. სიგანის მნიშვნელობა როგორც სიმბოლოების რაოდენობით, ისე პროცენტებით შეიძლება იყოს მოცემული;
- ატრიბუტი `width` - განსაზღვრავს ცხრილის სიგანეს. სიგანის მნიშვნელობა შეიძლება როგორც პიქსელებით, ისე პროცენტებით იყოს მოცემული.

ცხრილის სტრიქონებისა და სვეტების შექმნა

ცხრილის სტრიქონებისა და სვეტების შესაქმნელად `<tr>` და `<col>` ტეგები გამოიყენება.

`<tr>` ტეგი (ინგლ. *Table Row* - ცხრილის სტრიქონი) ცხრილის სტრიქონს ქმნის. სტრიქონს შეიძლება `align` (ჰორიზონტალურად სწორება) და `valign` (ვერტიკალურად სწორება) ატრიბუტები ჰქონდეს.

`<col>` (ინგლ. *Column* - სვეტი) ტეგი ცხრილის სვეტის შესაქმნელად გამოიყენება და შეიძლება `width`, `bgcolor`, `align`, `valign` და `id` ატრიბუტები ჰქონდეს.

- ატრიბუტი `width` - სვეტში უჯრის სიგანეს განსაზღვრავს პიქსელებში;

- ატრიბუტი bgcolor - სვეტში უჯრის ფონს განსაზღვრავს;
- ატრიბუტი align - უჯრის შიგთავსის ჰორიზონტალურად სწორების რეჟიმს განსაზღვრავს. მან შეიძლება left (მარცხნივ), center (ცენტრში) და right (მარჯვნივ) მნიშვნელობები მიიღოს;
- ატრიბუტი valign - უჯრის შიგთავსის ვერტიკალურად სწორების რეჟიმს განსაზღვრავს. მან შეიძლება top (ზემოთ), middle (შუაში) და bottom (ქვემოთ) მნიშვნელობები მიიღოს.

ცხრილის თითოეული სტრიქონი იწყება ტეგით <tr> და მთავრდება </tr>-ით.

<tr> ტეგს შეიძლება რამდენიმე ატრიბუტი ჰქონდეს:

- Align = left ცხრილის სტრიქონისთვის ასწორებს ტექსტს მარცხენა მხრიდან; Align="center" (ტექსტს ათავსებს ცენტრში), Align="right" (ტექსტს ასწორებს მარჯვენა მხრიდან).
- Valign - მიუთითებს ტექსტის ვერტიკალურ მდებარეობაზე. valign="top" (ტექსტს უახლოებს უჯრის ზედა ნაწილს); valign="middle" (ტექსტი უჯრის შუაში, ცენტრშია); valign="bottom" (ტექსტი უჯრის ქვედა კიდემია);
- bgcolor - მოცემული სტრიქონისთვის ცხრილის ფერს განსაზღვრავს.

ცხრილის უჯრის შესაქმნელად <td> </td> (ინგლ. *Table Data* - ცხრილური მონაცემი) ტეგი გამოიყენება. <td>-ს ატრიბუტებია:

- colspan-ის მნიშვნელობა განსაზღვრავს ცხრილის რამდენი უჯრა უნდა გაერთიანდეს ჰორიზონტალურად მონიშნულ უჯრასთან ერთად;

- rowspan, რომელიც მიუთითებს ცხრილის რამდენი უჯრა უნდა გაერთიანდეს ვერტიკალურად მონიშნულ უჯრასთან ერთად;
- align უჯრაში ტექსტის განლაგებას ცვლის. დასაშვები მნიშვნელობებია: align = "left"; align="right"; align="center";
- valign ტექსტის ვერტიკალურ მდებარეობას განსაზღვრავს. valign="top" (ტექსტი მიახლოებულია უჯრის ზედა კიდეებთან); valign="middle" (ტექსტი მოთავსებულია ცენტრში); valign="bottom" (ტექსტი უჯრის ქვედა კიდეშია);
- width განსაზღვრავს უჯრის სიგანეს პიქსელებში;
- height განსაზღვრავს უჯრის სიმაღლეს პიქსელებში;
- bgcolor მითითებული უჯრისთვის ცხრილის ფერს მითითებს;
- ატრიბუტი rules - უჯრებს შორის ხაზების გამოტანის წესს განსაზღვრავს;
 - მისი დასაშვები მნიშვნელობებია:
 - none - ხაზი არ არის (მნიშვნელობა ნაგულისხმევი წესის თანახმად მიენიჭება);
 - rows - ხაზები მხოლოდ სტრიქონებს შორის;
 - cols - ხაზები მხოლოდ სვეტებს შორის;
 - all - ხაზები სტრიქონებსა და სვეტებს შორის.

მარტივი და რთული ცხრილების შექმნა

ცხრილის სათაურის შესაქმნელად <th>, </th> (ინგლ. *Table Head* - ცხრილის სათაური) ტეგი გამოიყენება. ცხრილის წარწერის შესაქმნელად <caption> (ინგლ. *caption* - წარწერა) წყვილი ტეგი გამოიყენება. <caption> ტეგი <table> და </table>

ტეგებს შორის და სტრიქონისა და უჯრის აღწერის გარეთ უნდა იყოს განთავსებული. align ატრიბუტი განსაზღვრავს, სად უნდა იყოს წარწერა.

ქვემოთ მოცემულია სამუჯრიანი ერთსტრიქონიანი ცხრილი.

მაგალითი 4.1:

```
<!DOCTYPE html>
<html>
<head>
<title>Table</title></head>
<body>
<table border="1">
<tr>
<td>პირველი უჯრედი</td>
<td>მეორე უჯრედი</td>
<td>მესამე უჯრედი</td>
</tr>
</table>
</body>
</html>
```

პირველი უჯრედი	მეორე უჯრედი	მესამე უჯრედი
----------------	--------------	---------------

4.2 მაგალითში ნაჩვენებია, როგორ შეიძლება შეიქმნას სამი სტრიქონისა და ოთხი სვეტისგან შემდგარი ცხრილი.

მაგალითი 4.2:

```
<!DOCTYPE html>
<html>
<head><title>Table</title></head>
<body>
<table border="1">
<tr>
<td>January</td>
<td>February</td>
<td>March</td>
<td>April</td>
</tr>
<tr>
<td>May</td>
<td>June</td>
<td>July</td>
<td>August</td>
</tr>
<tr>
<td>September</td>
<td>October</td>
<td>November</td>
<td>December</td>
</tr>
</table>
</body>
</html>
```

September	October	November	December
January	February	March	April
May	June	July	August

ქვემოთ მოცემულ პროგრამაში rowspan და colspan ატრიბუტების გამოყენებით ცხრილის უჯრების გაერთიანება განხილულია.

მაგალითი 4.3:

```

<body>
<table border=10 width=80% align=center>
<caption>ცხრილი ვერტიკალურად და ჰორიზონტალურად
გაერთიანებული უჯრედებით <br><br><br>
</caption>
<tr><td width=30% bgcolor=cornsilk>უჯრედი A11 </td>
<td width=200 bgcolor=chartreuse>უჯრედი A12 </td>
<td rowspan=2 bgcolor=aqua>უჯრედი A13+A23 </td></tr>
<tr><td align=center bgcolor=azure> უჯრედი A21 </td>
<td align=right bgcolor=lawnegreen> უჯრედი A22 </td></tr>
<tr><td height=30 valign=top bgcolor=lightgrey> უჯრედი A31
</td>
<td colspan=2 valign=bottom bgcolor=mediumspringgreen>
უჯრედი A32+A33 </td></tr>
</table>
</body>

```

ცხრილი ვერტიკალურად და ჰორიზონტალურად გაერთიანებული უჯრედებით

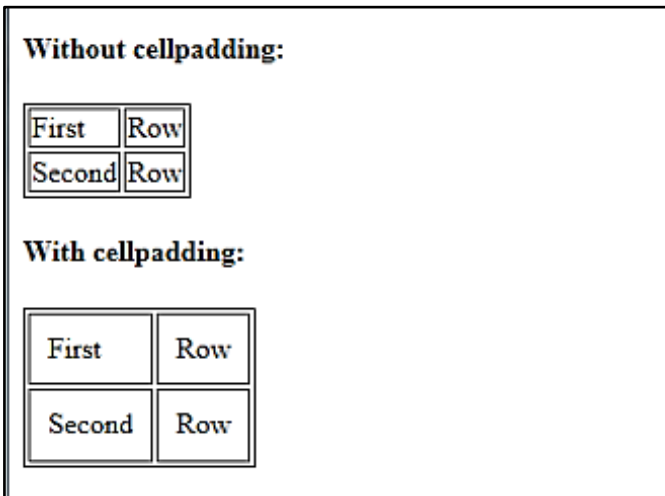
უჯრედი A11	უჯრედი A12	უჯრედი A13+A23
უჯრედი A21	უჯრედი A22	
უჯრედი A31	უჯრედი A32+A33	

ქვემოთ მოცემულ მაგალითში განხილულია cellpadding ატრიბუტის გამოყენება.

მაგალითი 4.4:

```
<!DOCTYPE html>
<html>
<body>
<h4>Without cellpadding:</h4>
<table border="1">
<tr>
<td>First</td>
<td>Row</td>
</tr>
<tr>
<td>Second</td>
<td>Row</td>
</tr>
</table>
<h4>With cellpadding:</h4>
```

```
<table border="1" cellpadding="10">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
</body>
</html>
```



მაგალითი 4.5. ნაჩვენებია cellpadding ატრიბუტის გამოყენება:

```
<!DOCTYPE html>
<html>
<body>
<h4>Without cellspacing:</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>With cellspacing="0":</h4>
<table border="1" cellspacing="0">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>With cellspacing="10":</h4>
<table border="1" cellspacing="10">
```

```
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
</body>
</html>
```

Without cellspacing:

First	Row
Second	Row

With cellspacing="0":

First	Row
Second	Row

With cellspacing="10":

First	Row
Second	Row

ცხრილები ტექსტის სვეტებად განთავსების საშუალებას იძლევა. ასეთი მაგალითი ქვემოთაა მოცემული.

მაგალითი 4.6:

```
<!DOCTYPE html>
<html>
<head>
<title>ტექსტი სვეტებად</title>
</head>
<body bgcolor=#dcdcdc >
<center>
<table width=70% border=0 cellpadding="5">
<tr> <td align=center colspan=2>
<h2>მონაზონ ქალს ოფელიას</h2>
<h3>გალაქტიონ ტაბიძე</h3></td></tr>
<tr>
<td width=50% align=top>
გმართებს ტირილი და ცრემლთა ფრქვევა.<br>
სტიროდე, ადამ, სტიროდე, ევა!<br>
დასწვით, დასდაგეთ ოცნების ფერფლი,<br>
ბოროტი გველის მსხვერპლი ხარ, მსხვერპლი!</td>
<td valign=top>
სევდად იცვალა თრთოლა ციების,<br>
ღმერთი თარეშობს შურისძიების.<br>
სტიროდე, ადამ, სტირიდე, ევა!<br>
გმართებს ტირილი და ცრემლთა ფრქვევა.</td></tr>
</table>
</center>
</body>
</html>
```

მონაზონ ქალს ოფელიას

გალაქტონ ტაბიძე

გმართებს ტირილი და ცრემლთა ფრქვევა.
სტიროდე, ადამ, სტიროდე, ევა!
დასწვით, დასდაგეთ ოცნების ფერფლი,
ბოროტი გველის მსხვერპლი ხარ, მსხვერპლი!

სევდად იცვალა თრთოლა ციების,
ღმერთი თარეშობს შურისძიების.
სტიროდე, ადამ, სტირიდე, ევა!
გმართებს ტირილი და ცრემლთა ფრქვევა.

4.7 მაგალითში განხილულია რთული ცხრილის შექმნა.

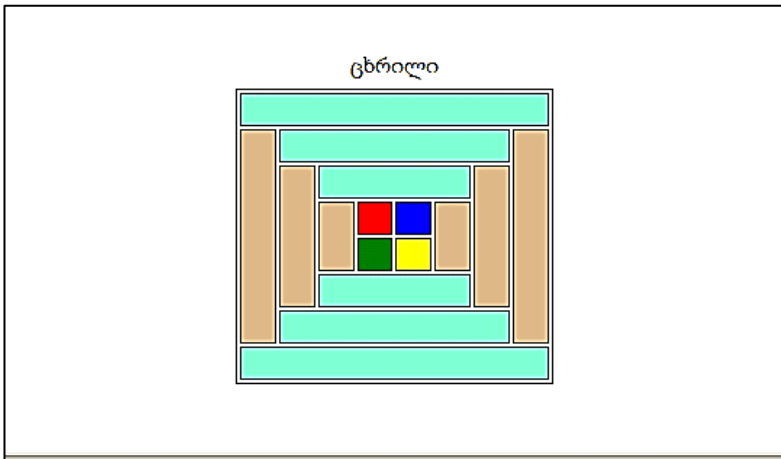
მაგალითი 4.7:

```
<!DOCTYPE html>
<html>
<head>
<title> რთული ცხრილი </title>
</head>
<body><table border=1 bordercolor=black width=30%
align=center>
<caption>ცხრილი</caption> <br><br><br>
<tr><td colspan=8 bgcolor=aquamarine> &nbsp;</td></tr>
<tr><td rowspan=6 bgcolor=burlywood> &nbsp;</td>
<td colspan=6 bgcolor=aquamarine>&nbsp;</td>
<td rowspan=6 bgcolor=burlywood> &nbsp;</td></tr>
<tr><td rowspan=4 bgcolor=burlywood> &nbsp;</td>
<td colspan=4 bgcolor=aquamarine>&nbsp;</td>
<td rowspan=4 bgcolor=burlywood> &nbsp;</td></tr>
<tr><td rowspan=2 bgcolor=burlywood> &nbsp;</td>
<td colspan=1 bgcolor=red>&nbsp;</td>
<td colspan=1 bgcolor=blue>&nbsp;</td>
<td rowspan=2 bgcolor=burlywood> &nbsp;</td></tr>
```

```

<tr><td bgcolor=green>&nbsp;</td>
<td bgcolor=yellow>&nbsp;</td></tr>
<tr><td colspan=4 bgcolor=aquamarine>&nbsp;</td>
<tr><td colspan=6 bgcolor=aquamarine>&nbsp;</td>
<tr><td colspan=8 bgcolor=aquamarine>&nbsp;</td>
</table>
</body>
</html>

```



დავალბა 1. შექმენით თქვენთვის სასურველი სახის ვებგვერდი, სადაც ნებისმიერ ადგილას განათავსებთ შემდეგ კალენდარს:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

დავალბა 2. შექმენით ცხრილი, რომელშიც განათავსებული ტექსტი ქვემოთაა მოცემული:



კეთილი იყოს თქვენი მობრძანება ჩემს ვებგვერდზე

გილოცავთ

გილოცავთ

გილოცავთ

გილოცავთ

გილოცავთ საქართველო დამოუკიდებლობის დღეს!

26 მაისს საქართველო დამოუკიდებლობის დღეს აღნიშნავს. ეს ქვეყნის ერთ-ერთი უმნიშვნელოვანესი დღესასწაულია, რომელიც 1918 წლიდან იღებს სათავეს.

1917 წლის რუსეთში მომხდარ ბურჟუაზიულ რევოლუციას მონარქიის დაშობა და იმპერიის დაშლა მოჰყვა. საქართველოში ამავე პერიოდში შეიქმნა ეროვნული საბჭო, რომელიც, ფაქტობრივად, ქვეყანას მართავდა. ეროვნული საბჭოს დადგენილებით, 1918 წლის 26 მაისს საქართველო დამოუკიდებელ რესპუბლიკად გამოცხადდა. რესპუბლიკის თავმჯდომარედ აირჩიეს ნიკეორ დანიანი.

1918 წლის 26 მაისს მიღებულ იქნა "საქართველოს სახელმწიფოებრივი დამოუკიდებლობის აქტი".

საქართველოს დემოკრატიულმა რესპუბლიკამ, რომლის დამოუკიდებლობა ევროპის წამყვანმა ქვეყნებმა ცნეს, არსებობა არასრული სამი წლის თავზე, საბჭოთა რუსეთის მიერ განხორციელებული ოკუპაციის შემდეგ, შეწყვიტა.

გასული საუკუნის 90-იან წლებში საქართველომ შეძლო დაშლის პირას მისული საბჭოთა კავშირიდან თავის დაღწევა. 70-წლიანი ტყვეობის შემდეგ, 1991 წლის 9 აპრილს, საქართველოს ხელისუფლებაში მოსულმა ეროვნულმა მალეზმა, 31 მარტის საყოველთაო რეფერენდუმის საფუძველზე, საქართველოს სახელმწიფოებრივი დამოუკიდებლობა აღადგინეს.



26 მაისს საქართველო დამოუკიდებლობის დღეს აღნიშნავს. ეს ქვეყნის ერთ-ერთი უმნიშვნელოვანესი დღესასწაულია, რომელიც 1918 წლიდან იღებს სათავეს.

1917 წელს რუსეთში მომხდარ ბურჟუაზიულ რევოლუციას მონარქიის დამხობა და იმპერიის დაშლა მოჰყვა. საქართველოში ამავე პერიოდში შეიქმნა ეროვნული საბჭო, რომელიც ფაქტობრივად ქვეყანას მართავდა. ეროვნული საბჭოს დადგენილებით, 1918 წლის 26 მაისს საქართველო დამოუკიდებელ რესპუბლიკად გამოცხადდა. რესპუბლიკის თავმჯდომარედ აირჩიეს ნოე ჟორდანიას.

1918 წლის 26 მაისს მიღებულ იქნა “საქართველოს სახელმწიფოებრივი დამოუკიდებლობის აქტი”.

საქართველოს დემოკრატიულმა რესპუბლიკამ, რომლის დამოუკიდებლობა ევროპის წამყვანმა ქვეყნებმა აღიარეს, არსებობა არასრული სამი წლის თავზე, საბჭოთა რუსეთის მიერ განხორციელებული ოკუპაციის შემდეგ შეწყვიტა.

გასული საუკუნის 90-იან წლებში საქართველომ შეძლო დაშლის პირას მისული საბჭოთა კავშირიდან თავის დაღწევა. 70-წლიანი ტყვეობის შემდეგ, 1991 წლის 9 აპრილს, საქართველოს ხელისუფლებაში მოსულმა ეროვნულმა ძალებმა, 31 მარტის საყოველთაო რეფერენდუმის საფუძველზე, საქართველოს სახელმწიფოებრივი დამოუკიდებლობა აღადგინეს.

თავი 5

ბმულები

- ბმულების დოკუმენტში ჩასმა;
- შიგა და გარე ბმულები;
- დოკუმენტიდან დოკუმენტზე გადასასვლელი ბმულის შექმნა;
- ტექსტიდან ტექსტზე გადასასვლელი ბმულის შექმნა;
- ბმულის შექმნა არსებულ Web-გვერდზე;
- ბმულის შექმნა სურათიდან;
- ტეგი <iframe>.

ბმულები

ბმულები ერთ-ერთი მნიშვნელოვანი კომპონენტია, იგი ვებგვერდებს მომხმარებლისთვის უფრო საინტერესოს ხდის. ბმულების ანუ, როგორც ზოგჯერ მას უწოდებენ, ჰიპერბმულების დახმარებით ვებგვერდის სტრუქტურირება და მისი სხვა გვერდებთან ან იმავე დოკუმენტის სხვა ნაწილებთან დაკავშირება ხდება, რაც ინფორმაციის სწრაფ და მოსახერხებელ მიღებას უზრუნველყოფს. ინტერნეტს დიდი პოპულარობა სწორედ ბმულებმა მოუტანა, რადგან მისი დახმარებით მომხმარებელს ძალზე მარტივად შეუძლია ერთი გვერდიდან მეორეზე გადასვლა.

ბმულების დოკუმენტში ჩასმა

HTML ენაში ბმულების დოკუმენტში ჩასასმელად `<a>` `` ტეგი გამოიყენება. `<a>` ტეგს შემდეგი ატრიბუტები აქვს: href, name, accesskey, target.

ამ ატრიბუტის გამოყენების დროს ნებისმიერი ტექსტი, რომელიც `<a>` და `` ტეგებს შორის არის მოთავსებული, ვებბრაუზერის მიერ სპეციალურად გამოიყოფა (ჩვეულებრივ სტანდარტულად ხაზგასმული გამოდის და ლურჯი ფერითაა გამოყოფილი). როდესაც მაუსის მაჩვენებელს რომელიმე ბმულს მივუახლოვებთ, იგი შეიცვლება და ხელის ფორმას მიიღებს.

- ატრიბუტი href აუცილებლად უნდა იყოს მითითებული. მის მნიშვნელობას იმ დოკუმენტის URL მისამართი წარმოადგენს, რომელზეც ბმული მიუთითებს.

- ატრიბუტი name ატრიბუტ href-ის ნაცვლად დოკუმენტში შიგა ბმულების დროს მიეთითება.

- ატრიბუტი `accesskey` საშუალებას გვამძლევს ე. წ. "ცხელი კლავიში" მივუთითოთ, რომელზეც ხელის დაჭერით ბმულზე გადასვლა მოხდება. მისი სინტაქსია:

```
<a href = " http://www.gtu.ge/" accesskey="კლავიშის სახელი"> ბმული</a>
```

ბმული, რომელიც ელექტრონული ფოსტით შეტყობინების მითითებულ მისამართზე გაგზავნის საშუალებას იძლევა:

```
<a href="mailto:name@gmail.com"> ბმული</a>
```

ბმული, რომელიც ელექტრონული ფოსტით ავტომატურად შეყვანილი შეტყობინების თემისა და ტექსტის გაგზავნის საშუალებას იძლევა:

```
<a href="mailto:name@gmail.com, subject=თემა&body = ტექსტი"> ბმული</a>
```

- ატრიბუტი `target` განსაზღვრავს ჩასატვირთავ URL მისამართის ბრაუზერში ჩატვირთვის ტიპს. მისი მნიშვნელობებია: `_blank`, `_new`, `_parent`, `_self`, `_top`.

`blank`, `new` მნიშვნელობების შემთხვევაში URL მისამართი ჩაიტვირთება ბრაუზერის ახალ ფანჯარაში.

`parent`, `self`, `top` მნიშვნელობის შემთხვევაში URL მისამართი ბრაუზერის იმავე ფანჯარაში ჩაიტვირთება.

შიგა ბმულები

შიგა ბმულები მიმდინარე ვებსაიტზე განთავსებულ სხვა ვებგვერდებზე მიუთითებს, ამიტომ შიგა ბმულების შექმნის დროს `<a>` ტეგის `href` ატრიბუტში ძებნის ფარდობითი მისამართის მითითებაა საკმარისი.

ფარდობით მისამართში ათვლის წერტილი ის საქაღალდეა, რომელშიც ეს ფაილია მოთავსებული, ამიტომ თუ

ფაილები ერთ საქალაქდემია, საკმარისია მხოლოდ ფაილის სახელის მითითება.

გარე ბმულები

გარე ბმულები იმ ვებგვერდებზე მიუთითებს, რომლებიც მიმდინარე ვებსაიტზე არ არის განთავსებული, ამიტომ გარე ბმულების შექმნისას <a> ტეგის href ატრიბუტში URL მისამართის სრულად მითითება აუცილებელია.

დოკუმენტიდან დოკუმენტზე გადასვლა

მაგალითი 5.1:

```
<!DOCTYPE html>
<html>
<head>
<title> Link 2
</title>
</head>
<body>
<h1 align=center>
<a href="index.html">საქართველოს ქალაქები</a>
</h1>
<br>თბილისი
<br>ქუთაისი
<br>ბათუმი
<br>სოხუმი
<br>მცხეთა
<br>თელავი
```

</body>

</html>

საქართველოს ქალაქები

თბილისი
ქუთაისი
ბათუმი
სოხუმი
მცხეთა
თელავი

დოკუმენტის შიგნით გადასვლა

როგორც უკვე აღვნიშნეთ, ბმულები შეიძლება როგორც სხვა დოკუმენტებზე, ასევე ერთი და იგივე დოკუმენტის ცალკეულ ფრაგმენტებზე გაკეთდეს. ეს დოკუმენტის შიგნით ერთი ფრაგმენტიდან მეორეზე სწრაფად გადასვლის საშუალებას იძლევა. ასეთი გადასვლის ორგანიზებისათვის სპეციალური ღუზები გამოიყენება, რომლებიც ჩვეულებრივ ფრაგმენტის პირველ სტრიქონში ან სათაურში განთავსდება.

ღუზის ჩასმისთვის ისევ <a> ტეგი გამოიყენება, მაგრამ href ატრიბუტის ნაცვლად მასში name ატრიბუტი უნდა მიეთითოს. ამ ატრიბუტის მნიშვნელობა ღუზის სახელია. ის მხოლოდ ლათინური ასოებისა და ციფრებისაგან უნდა შედგებოდეს და ჰარს არ უნდა შეიცავდეს.

დაყენებულ ღუზაზე გადასასვლელად URL მისამართის

ბოლოს ღუზის სახელის მითითება აუცილებელია, რომლის წინ # სიმბოლო იქნება ჩაწერილი.

ბმული შეიძლება იყოს როგორც ტექსტი, ისე გამოსახულება. ამასთან, HTML კოდი და გამოსახულება ერთ საქაღალდეში უნდა იყოს განთავსებული.

მაგალითი 5.2:

```
<!DOCTYPE html>
<html>
<head>
<title> Link 1 </title>
</head>
<body>
<h1 align=center> საქართველოს ქალაქები</h1>
<br><a href="#თბილისი">თბილისი</a>
<br><a href="#ქუთაისი">ქუთაისი</a>
<br><a href="#ბათუმი">ბათუმი</a>
<br><a href="#სოხუმი">სოხუმი</a>
<br><a href="#მცხეთა">მცხეთა</a>
<br><a href="#თელავი">თელავი</a>
<br><br>
<a name = "#თბილისი"> თბილისი </a>
<p align="justify">

```

თბილისი საქართველოს დედაქალაქია; მდებარეობს აღმოსავლეთ საქართველოში, თბილისის ქვაბულში, მდინარე მტკვრის ორივე სანაპიროზე, ზღვის დონიდან 380-600 მ

სიმაღლეზე, ჩრდილოეთით ესაზღვრება საგურამოს ქედის სამხრეთი მთისწინეთი, აღმოსავლეთით - ივრის ზეგნის ჩრდილო-დასავლეთი მონაკვეთი, დასავლეთით და სამხრეთით კი - თრიალეთის ქედის განშტოებები. ქალაქს 720 კმ² ფართობი უჭირავს და 1.090 მილიონზე მეტი მცხოვრებია.

თბილისი კავკასიის რეგიონის მნიშვნელოვანი ინდუსტრიული, სოციალური და კულტურული ცენტრია და ბოლო დროს ერთ-ერთი უმნიშვნელოვანესი სატრანსპორტო კვანძია გლობალური ენერგომატარებლებისა და სავაჭრო პროექტებისთვის.

 ქუთაისი

<p align="justify">

ქუთაისი - ქალაქი საქართველოში, იმერეთის მხარის ადმინისტრაციული ცენტრი. მდებარეობს მდინარე რიონზე. ოფიციალურად ქალაქად გამოცხადდა 1811 წელს, თუმცა უფრო ადრინდელ პერიოდში (730 წ., 980-1072 წწ.) იგი მოიხსენიება საქართველოს მნიშვნელოვან ქალაქთა რიცხვში. მოსახლეობა 192.500 (2010 წ.). ქუთაისი მნიშვნელობით საქართველოს მეორე სამრეწველო და კულტურული ქალაქია. აქ მდებარეობდა მძიმე, მსუბუქი და კვების მრეწველობის მრავალი საწარმო, კულტურის ობიექტები, თეატრები, გალერეები. ქუთაისი განათლების ერთ-ერთი ცენტრია საქართველოში, ქალაქში მდებარეობს 35 ბაგა-ბაღი, 46 საჯარო და 20 კერძო სკოლა, აკაკი წერეთლის სახელობის სახელმწიფო უნივერსიტეტი, საქართველოს სუბტროპიკული მეურნეობის სახელმწიფო

უნივერსიტეტი, ქუთაისის სამართლისა და ეკონომიკის უნივერსიტეტი, ქუთაისის სამედიცინო ინსტიტუტი «ქუთაისი», თბილისის ვანო სარაჯიშვილის სახელობის სახელმწიფო კონსერვატორიის ქუთაისის ფილიალი, თბილისის ეკონომიკურ ურთიერთობათა სახელმწიფო უნივერსიტეტი - ქუთაისის ჰუმანიტარულ-სოციალურ მეცნიერებათა ფაკულტეტი, ქ. ქუთაისის იოანე პეტრიწის სახელობის უნივერსიტეტი, ქუთაისის პედაგოგიური ინსტიტუტი „ლამპარი“, ქ. ქუთაისის ხელოვნებისა და პედაგოგიკის ინსტიტუტი „კოლხა“, ქ. ქუთაისის საერო ინსტიტუტი «გელათი», ქ. ქუთაისის საინჟინრო ინსტიტუტი, ქუთაისის კვებისა და ფარმაცევტული მრეწველობის ინსტიტუტი, ქალაქ ქუთაისის დამოუკიდებელი უნივერსიტეტი, ბორის ონიანის სახელობის ჰუმანიტარული და სამართლის საერო ინსტიტუტი, საქართველოს ტურიზმის მეცნიერებათა აკადემიის ქუთაისის სახელობის მრავალპროფილიანი ინსტიტუტი, ქ. ქუთაისის ჰუმანიტარული და სამართლის ინსტიტუტი, ქ. ქუთაისის ნ. მუსხელიშვილის სახელობის სახელმწიფო ტექნიკური უნივერსიტეტი და რამდენიმე პროფსასწავლებელი.

</p>

 ბათუმი

<p align="justify">

ბათუმი - ქალაქი საქართველოში, აჭარის ავტონომიური რესპუბლიკის ადმინისტრაციული ცენტრი, ქალაქი 1878 წლიდან. მოსახლეობა 121,8 ათასი (2002 წ.).

ბათუმი მსხვილი საერთაშორისო ნავსადგურია შავი ზღვის სამხრეთ-აღმოსავლეთ სანაპიროზე, ბაქო-ბათუმის რკინიგზისა და ნავთობსადენის ბოლო პუნქტი და საავტომობილო გზების კვანძი. ქალაქი გაშენებულია ღრმა, კარგად დაცული ბუნებრივი ნავსაყუდელის ბათუმის ყურის ნაპირას; მდებარეობს ზღვის დონიდან 5 მ, თბილისიდან 350 კმ-ში (რკინიგზით).

ქალაქი გადაჭიმულია ჩრდილო-აღმოსავლეთიდან სამხრეთ-დასავლეთისაკენ 7 კმ-ზე. სიგანე ზღვის ნაპირიდან გორაკ-ბორცვებამდე 1-2,5 კმ. 1937 წ. აქედან განაშენიანებულია 1300 ჰა-ზე მეტი. ბათუმის ძირითადი ნაწილი სამხრეთიდან ეკვრის ყურეს და გაშენებულია კახაბრის ვაკის ჩრდილოეთ მონაკვეთზე, დანარჩენი ნაწილი ყურის აღმოსავლეთით და ჩრდილო-აღმოსავლეთით - მდინარეების ბარცხანისა და ყოროლისწყლის გასწვრივ.

</p>

 სოხუმი

<p align="justify">

სოხუმი (ადრინდელი ქართული სახელწოდება ცხუმი) - ქალაქი საქართველოში, აფხაზეთის ავტონომიური რესპუბლიკის დედაქალაქი და სოხუმის მუნიციპალიტეტის ადმინისტრაციული ცენტრი, შავიზღვისპირა კურორტი.

სოხუმი გაშენებულია უმთავრესად მდინარეების - გუმისთისა და კელასურის ქვემო დინებათა შორის მოქცეულ დაბლობ-ვაკეზე, მისი სამხრეთ-აღმოსავლეთი ნაწილი მდინარეებს - ბესლეთსა და კელასურს შორის მდებარე ვიწრო სანაპირო დაბლობ-ვაკეზე. ქალაქის დასახლება ვრცელდება გორაკ-ბორცვებზე, რომლებიც ზღვისპირა ვაკეს ჩრდილო-აღმოსავლეთით ესაზღვრება. სოხუმის ფართობია 23,3 კმ².

</p>

 მცხეთა

<p align="justify">

მცხეთა ქალაქი (1956 წლიდან) საქართველოში, მცხეთა-მთიანეთის მხარის და მცხეთის რაიონის (1930 წლიდან) ადმინისტრაციული ცენტრი. მცხეთა გამოცხადებულია ქალაქ-მუზეუმად და იუნესკოს მსოფლიო კულტურული მემკვიდრეობის ნუსხაშია შეტანილი. მოსახლეობა 7,7 ათასი (2002 წ.).

მდებარეობს მდინარეების - მტკვრისა და არაგვის შესაყართან, მტკვრის ორივე და არაგვის მარჯვენა ნაპირზე. რკინიგზის სადგური თბილის-სამტრედიის ხაზზე. ზღვის დონიდან - 480 მ, თბილისიდან - 21 კმ (რკინიგზით).

</p>

 თელავი

<p align="justify">


```

```

თელავი, ქალაქი აღმოსავლეთ საქართველოში, გომბორის ქედის ჩრდილო-აღმოსავლეთ კალთის ძირას და ალაზნის ვაკეზე. კახეთის მხარის და თელავის მუნიციპალიტეტის ადმინისტრაციული ცენტრი. თბილის-თელავის სარკინიგზო ხაზის ბოლო სადგური. ზღვის დონიდან 550-800 მ, თბილისიდან 158 კმ. ქალაქად გამოცხადდა 1801 წელს. მოსახლეობა 21,8 ათასი (2002 წ.).

ახლანდელი თელავის ტერიტორიაზე პირველი დასახლება გვიანდელი ბრინჯაოს ხანაში გაჩნდა, ელინისტურ ხანაში განვითარდა, ხოლო გვიანდელ ანტიკურ ხანაში მის ბაზაზე წარმოიქმნა ქალაქის ტიპის დასახლება.</p>

```
</body>
```

```
</html>
```

საქართველოს ქალაქები

[თბილისი](#)
[ქუთაისი](#)
[ზაზუმი](#)
[სოხუმი](#)
[მცხეთა](#)
[თელავი](#)

თბილისი



თბილისი საქართველოს დედაქალაქია; მდებარეობს აღმოსავლეთ საქართველოში, თბილისის ქვაბულში, მდინარე მტკვრის ორივე სანაპიროზე, ზღვის დონიდან 380-600 მ სიმაღლეზე, ჩრდილოეთით ესაზღვრება საგურამოს ქედის სამხრეთი მთისწინეთი, აღმოსავლეთით - ივრის ზეგნის ჩრდილო-დასავლეთი მონაკვეთი, დასავლეთით და სამხრეთით კი - თრიალეთის ქედის განშტოებები. ქალაქს 720 კმ² ფართობი უჭირავს და 1.090 მილიონზე მეტი მცხოვრებია. თბილისი კავკასიის რეგიონის მნიშვნელოვანი ინდუსტრიული, სოციალური და კულტურული

ცენტრია და ბოლო დროს ერთ-ერთი უმნიშვნელოვანესი სატრანსპორტო კვანძია გლობალური ენერგომატარებლებისა და სავაჭრო პროექტებისთვის.

ქუთაისი



ქუთაისი - ქალაქი საქართველოში, იმერეთის მხარის ადმინისტრაციული ცენტრი. მდებარეობს მდინარე რიონზე. ოფიციალურად ქალაქად გამოცხადდა 1811 წელს, თუმცა უფრო ადრინდელ პერიოდში (730 წ., 980-1072 წწ.) იგი მოიხსენიება საქართველოს მნიშვნელოვან ქალაქთა რიცხვში. მოსახლეობა 192.500 (2010 წ.). ქუთაისი მნიშვნელობით საქართველოს მეორე სამრეწველო და კულტურული ქალაქია. აქ მდებარეობდა მიმიე, მსუბუქი და კვების მრეწველობის მრავალი საწარმო, კულტურის ობიექტები, თეატრები,

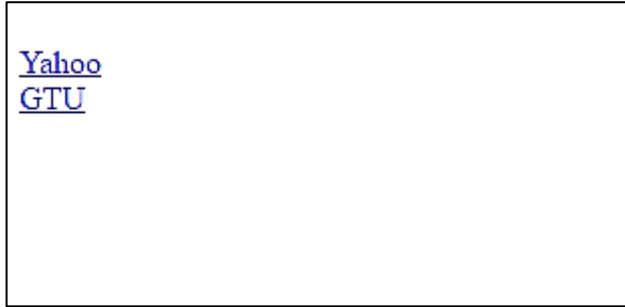
გალერეები. ქუთაისი განათლების ერთ-ერთი ცენტრია საქართველოში, ქალაქში მდებარეობს 35 ზაგა-ბაღი, 46 საჯარო და 20 კერძო სკოლა, აკაკი წერეთლის სახელობის სახელმწიფო უნივერსიტეტი, საქართველოს

დოკუმენტიდან არსებულ Web-გვერდზე გადასვლა

ქვემოთ განხილულ მაგალითში ნაჩვენებია, როგორ შეიძლება ჩვენ მიერ შექმნილი HTML დოკუმენტიდან უკვე არსებულ Web-გვერდზე გადასვლა.

მაგალითი 5.3:

```
<body>  
<br><a href = "http://www.yahoo.com">Yahoo</a>  
<br><a href = "http://www.gtu.ge"> GTU </a>  
</body>
```



სხვა დოკუმენტზე ან სურათზე გადასვლა შეიძლება სურათიდანაც. პრინციპი ისეთივეა, როგორც ტექსტის შემთხვევაში.

მაგალითი 5.4:

```
<body>
<br>
თბილისი
<p align="justify">
<a href="tbilisi.jpg">
 </a>
თბილისი საქართველოს დედაქალაქია; მდებარეობს
აღმოსავლეთ საქართველოში, თბილისის ქვაბულში, მდინარე
მტკვრის ორივე სანაპიროზე, ზღვის დონიდან 380-600 მ
სიმაღლეზე, ჩრდილოეთით ესაზღვრება საგურამოს ქედის
სამხრეთი მთისწინეთი, აღმოსავლეთით - ივრის ზეგნის
ჩრდილო-დასავლეთი მონაკვეთი, დასავლეთით და
სამხრეთით კი - თრიალეთის ქედის განშტოებები. ქალაქს 720
კმ² ფართობი უჭირავს და 1.090 მილიონზე მეტი
მცხოვრებია.<br>
```

თბილისი კავკასიის რეგიონის მნიშვნელოვანი ინდუსტრიული, სოციალური და კულტურული ცენტრია და ბოლო დროს ერთ-ერთი უმნიშვნელოვანესი სატრანსპორტო კვანძი გახდა გლობალური ენერგომატარებლებისა და სავაჭრო პროექტებისთვის.

სოხუმი

<p align="justify">

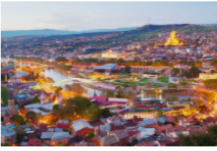
სოხუმი, (ადრინდელი ქართული სახელწოდება ცხუმი) - ქალაქი საქართველოში, აფხაზეთის ავტონომიური რესპუბლიკის დედაქალაქი და სოხუმის მუნიციპალიტეტის ადმინისტრაციული ცენტრი, შავიზღვისპირა კურორტი.

სოხუმი გაშენებულია უმთავრესად მდინარეების - გუმისთისა და კელასურის ქვემო დინებათა შორის მოქცეულ დაბლობ-ვაკეზე, მისი სამხრეთ-აღმოსავლეთი ნაწილი მდინარეებს - ბესლეთსა და კელასურს შორის მდებარე ვიწრო სანაპირო დაბლობ-ვაკეზე. ქალაქის დასახლება ვრცელდება გორაკ-ბორცვებზე, რომლებიც ზღვისპირა ვაკეს ჩრდილო-აღმოსავლეთით ესაზღვრება. სოხუმის ფართობია 23,3 კმ².

</p>

</body>

თბილისი



თბილისი საქართველოს დედაქალაქია; მდებარეობს აღმოსავლეთ საქართველოში, თბილისის ქვაბულში, მდინარე მტკვრის ორივე სანაპიროზე, ზღვის დონიდან 380-600 მ სიმაღლეზე, ჩრდილოეთით ესაზღვრება საგურამოს ქედის სამხრეთი მთისწინეთი, აღმოსავლეთით - ივრის ზეგნის ჩრდილო-დასავლეთი მონაკვეთი, დასავლეთით და სამხრეთით კი - თრიალეთის ქედის განშტოებები. ქალაქს 720 კმ² ფართობი უჭირავს და 1.090 მილიონზე მეტი მცხოვრებია.

თბილისი კავკასიის რეგიონის მნიშვნელოვანი ინდუსტრიული, სოციალური და კულტურული ცენტრია და ბოლო დროს ერთ-ერთი უმნიშვნელოვანესი სატრანსპორტო კვანძი გახდა გლობალური ენერჯომატარებლებისა და სავაჭრო პროექტებისთვის.

სოხუმი



სოხუმი, (ადრინდელი ქართული სახელწოდება ცხუმი) - ქალაქი საქართველოში, აფხაზეთის ავტონომიური რესპუბლიკის დედაქალაქი და სოხუმის მუნიციპალიტეტის ადმინისტრაციული ცენტრი, შავიზღვისპირა კურორტი. სოხუმი გამეწეულია უმთავრესად მდინარეების - გუმისთისა და კელასურის ქემო დინებათა შორის მოქცეულ დაბლობ-ვაკეზე, მისი სამხრეთ-აღმოსავლეთი ნაწილი მდინარეებს - ბესლეთსა და კელასურს შორის მდებარე ვიწრო სანაპირო დაბლობ-ვაკეზე. ქალაქის დასახლება

ვრცელდება გორაკ-ბორცვებზე, რომლებიც ზღვისპირა ვაკეს ჩრდილო-აღმოსავლეთით ესაზღვრება. სოხუმის ფართობია 23,3 კმ².

ტეგი <iframe>

<iframe> ტეგი ვებგვერდზე ვებგვერდის გამოსატანად გამოიყენება. ფრეიმებს, რომლებიც დოკუმენტში <iframe> ტეგის გამოყენებით არის ჩასმული მცურავი ფრეიმი ეწოდება.

<iframe> ტეგს შემდეგი ატრიბუტები აქვს: src, width, height, style (ეს ატრიბუტი გამოიყენება CSS-ში).

• ატრიბუტი SRC - მოცემულ ფრეიმში ჩასატვირთავ დოკუმენტის მისამართს აწვდის. მაგალითად:

```
<iframe src="URL"></iframe>
```

• ატრიბუტი width - ფრეიმის სიგანეს განსაზღვრავს პიქსელებში;

• ატრიბუტი height - ფრეიმის სიმაღლეს განსაზღვრავს პიქსელებში. მაგალითად,

```
<iframe src="demo_iframe.htm" height="200" width="300">
</iframe>
```

ფრეიმის სიგანისა და სიმაღლის განსაზღვრისათვის შეიძლება გამოვიყენოთ CSS-ი, მაგალითად,

```
<iframe src="demo_iframe.htm" style="height:200px;
width:300px;" > </iframe>
```

ჩუმათობის პრინციპით ფრეიმებს აქვს ჩარჩო. ჩარჩოს მოსახსნელად უნდა დავუმატოთ style ატრიბუტი და გამოვიყენოთ CSS-ის border თვისება. მაგალითად,

```
<iframe src="demo_iframe.htm" style="border:none;">
</iframe>
```

CSS-ის დახმარებით აგრეთვე შეიძლება შეიცვალოს ფრეიმის ჩარჩოს ზომა, სტილი და ფერი. მაგალითად,

```
<iframe src="demo_iframe.htm" style="border: 2px solid
red;"></iframe>
```

დავალება. შექმენით ვებგვერდი „კომპიუტერის მოწყობილობის მოკლე დახასიათება“, სადაც ძირითად გვერდზე განათავსეთ კომპიუტერის სურათი, რომლის საშუალებითაც ტექსტიან ვებგვერდზე მოხდება მიმართვა, ხოლო ამ გვერდის შიგნით ბმულებად კომპიუტერის ძირითადი მოწყობილობების დასახელება იქნება გამოყენებული. ძირითად გვერდზე კომპიუტერის სურათის ქვეშ გამოიტანეთ ჰიპერმიმართვა Microsoft-ის ძირითად გვერდზე.

თავი 6

ფორმები

- ფორმების შექმნის პრინციპები;
- მართვის ელემენტების ფორმაზე განთავსება;
- მარტივი ფორმების შექმნა;
- რთული ფორმების შექმნა;
- მულტიმედია;
- აუდიო- და ვიდეორგოლის ჩასმა.

ფორმების შექმნის პრინციპები

ფორმები მსოფლიო აბლაბუდაში ინტერაქტიული ურთიერთქმედებისთვის ერთ-ერთ ყველაზე პოპულარული საშუალებაა. ფორმების საშუალებით მომხმარებლისგან ინფორმაციის მიღებაა შესაძლებელი. HTML-ის დახმარებით შეგიძლიათ მარტივი ფორმები შექმნათ, რომლებიც „დიახ“ და „არა“ ტიპის პასუხებს ითვალისწინებს. ასევე, შეგიძლიათ შექმნათ რთული ფორმები შეკვეთებისთვის ან იმისთვის, რომ მკითხველისგან რაიმე კომენტარი ან კეთილი სურვილები მოისმინოთ.

ფორმები რამდენიმე ველისგან შედგება, სადაც მომხმარებელს შეუძლია გარკვეული ინფორმაცია შეიტანოს ან რომელიმე პარამეტრი აირჩიოს. მას შემდეგ, რაც მომხმარებელი გააგზავნის ინფორმაციას, ის სერვერზე მოთავსებული პროგრამის (სკრიპტის) მიერ დამუშავდება. სკრიპტი მოკლე პროგრამაა, რომელიც თითოეული ფორმის დასამუშავებლად სპეციალურადაა შექმნილი.

HTML-ის დახმარებით მხოლოდ ფორმის ვიზუალურ მხარეს განვსაზღვრავთ. იმისათვის, რომ ფორმის ღილაკებმა რეალურად იმუშაოს, სხვა ტექნოლოგია, რაიმე პროგრამა ან სკრიპტია საჭირო, რომელიც ჩვენს ფორმას უნდა მივაბათ.

ფორმები ინფორმაციის შეტანისთვისაა განკუთვნილი. ყველა ფორმა <form> (ინგლ. form - ფორმა) ტეგით იწყება და </form> ტეგით მთავრდება. ფორმას შეიძლება შემდეგი ატრიბუტები ჰქონდეს: name, action, method, target.

- ატრიბუტი name ფორმის სახელს განსაზღვრავს. თუ დოკუმენტში რამდენიმე ფორმა არ არის გამოყენებული, მაშინ მისი მითითება აუცილებელი არ არის;

- ატრიბუტი action უზრუნველყოფს ფორმის მონაცემების

შესაბამისი ფაილისთვის დასამუშავებლად გადაგზავნას.

- ატრიბუტი `method` ფორმის პარამეტრების გადაცემის მეთოდს განსაზღვრავს. იგი `get` ან `post` მნიშვნელობებს ღებულობს.

- ატრიბუტი `target` განსაზღვრავს ფანჯარას, რომელშიც გაგზავნილი ფორმის დამუშავების შედეგები დაბრუნდება. მასში ფანჯრის სახელი ცხადად უნდა ჩაიწეროს ან ერთ-ერთი შემდეგი პარამეტრი უნდა მიეთითოს:

- `blank` - დოკუმენტი ახალ ფანჯარაში გაიხსნება;
- `self` - დოკუმენტი იგივე ფანჯარაში გაიხსნება, რომელშიც მიმდინარე დოკუმენტია გახსნილი;
- `parent` - დოკუმენტი იმ ფრეიმში გაიხსნება, რომელიც მიმდინარე დოკუმენტის შემცველი ფრეიმის მიმართ მშობლიური ფრეიმია (თუ მშობლიური ფრეიმი არ არსებობს, მაშინ დოკუმენტი მიმდინარე ფრეიმში გაიხსნება);
- `top` - დოკუმენტი მიმდინარე ფანჯარაში გაიხსნება და მთელ ეკრანს დაიკავებს.

კოდის მუშაობის შედეგი ბრაუზერში არ გამოჩნდება, რადგანაც HTML-ში მხოლოდ ფორმის კარკასი იქმნება. იმისათვის, რომ ფორმა გამოჩნდეს, საჭიროა ფორმის ელემენტების შექმნა.

მაგალითი 6.1:

```
<!DOCTYPE html>
<html>
<head>
<title>forms1</title>
```

```

</head>
<body>
<form action ="formdata.php" method="POST" name ="form1">
<p>
<label>Input your name (max. 10 symbols): <input type="text"
name="name" size="30" maxlength="10">
</label>
</p>
<p> <label>Input your password (max. 10 symbols): <input
type="password" name="password" size="30" maxlength="10">
</label>
</p>
<p> <label> Male: <input type="radio" name="mf" value="Male">
</label>
<label> Female: <input type="radio" name="mf" value="Female"
checked> </label> </p>
<p> What kind of films do you like?</p>
<p> <label> <input type="checkbox" name="melodrama">
Melodramatic </label>
<label> <input type="checkbox" name="science fiction" checked>
Science fiction </label>
<label> <input type="checkbox"
name="documentary">Documentary </label>
<label> <input type="checkbox" name="adventure"> Adventure
</label> </p>
<p> Who is your favorite actor? </p>
<select name: "actor" size ="4">
<option value ="Ninidze" selected> Merab Ninidze

```

```
<option value ="Kipshidze"> Zurab Kipshidze
<option value ="Medea"> Medea Djaparidze
<option value ="Archvadze"> Tengiz Archvadze
<option value ="Lia"> Lia Eliava
<option value ="Koberidze"> Otar Koberidze
</select>
<p>Your comments:</p>
<textarea rows="7" cols="50" name="comments"> </textarea>
<br>
<br>
<input type="submit" value="OK">
<input type="reset" value="Cancel">
</form>
</body>
</html>
```

ბრაუზერის ფანჯარაში დავინახავთ:

Input your name (max. 10 symbols):

Input your password (max. 10 symbols):

Male: Female:

What kind of films do you like?

Melodramatic Science fiction Documentary Adventure

Who is your favorite actor?

Your comments:

მართვის ელემენტების ფორმაზე განთავსება

მონაცემთა შესატანი ველის და მართვის სხვა ელემენტების შესაქმნელად `<input>` (ინგლ. input - შეტანა) ტეგი გამოიყენება. ამ ტეგს აქვს ძირითადი ატრიბუტები: name, value, size, disabled, checked, maxlength, src, type.

- ატრიბუტი name - მონაცემთა შესატანი ველის სახელს განსაზღვრავს. მოცემული სახელი განიხილება, როგორც ველის უნიკალური იდენტიფიკატორი, რომლის მიხედვითაც

შემდგომში შეიძლება მომხმარებლის მიერ ამ ველში შეტანილი მონაცემები მივიღოთ;

- ატრიბუტი checked - ალმებით მოსანიშნად და გადამრთველებისათვის გამოიყენება. იგი უჩვენებს, რომ ვებგვერდის გახსნისას ამ ატრიბუტის მქონე ველი მოინიშნება;

- ატრიბუტი src - გამოსახულების URL-მისამართს მიუთითებს (image ატრიბუტთან ერთად გამოიყენება);

- ატრიბუტი type - მართვის ელემენტის ტიპს განსაზღვრავს. თუ იგი არ იქნება მითითებული, მაშინ გამოდის ერთსტრიქონიანი მონაცემთა შესატანი ველი. ყველა დანარჩენი ტიპი აუცილებლად უნდა მიეთითოს; type ატრიბუტმა შეიძლება შემდეგი მნიშვნელობები მიიღოს:

- Checkbox, რომელიც მონაცემთა შესატან ველს ალმით მოსანიშნ ველად გარდაქმნის;
- Hidden, რომელიც მართვის ელემენტს მალავს - მას ბრაუზერი არ ასახავს და მომხმარებელს ნაგულისხმევი წესით მინიჭებული მნიშვნელობის შეცვლის უფლებას არ აძლევს. მართვის ასეთი ელემენტი პროგრამისათვის უცვლელი ინფორმაციის გადასაცემად გამოიყენება, მაგალითად, მომხმარებლის იდენტიფიკატორი ან პაროლი;
- image საშუალებას იძლევა მართვის ელემენტის სახელს გამოსახულება დაუკავშირდეს;
- password მონაცემთა შესატან ველს გარეგნულად არ ცვლის, ბრაუზერი მომხმარებლის მიერ შეტანილ მნიშვნელობებს ეკრანზე არ ასახავს;
- radio მონაცემთა შესატან ველს ისეთ გადამრთველად

გადააქცევს, რომელიც საშუალებას იძლევა name ატრიბუტის ერთი და იგივე მნიშვნელობის და value ატრიბუტით მოცემულ გადამრთველის სხვადასხვა მნიშვნელობას შორის მხოლოდ ერთი მნიშვნელობა აირჩეს;

- text ერთსტრიქონიან მონაცემთა შესატან ველს აღწერს. მაქსიმალური სიგრძის მნიშვნელობისა და მონაცემთა შესატანი ველის განსაზღვრისათვის maxlength და size ატრიბუტები გამოიყენება;
- reset მონაცემთა შესატან ველს ღილაკად გარდაქმნის, რომელზეც მაუსის დაწკაპუნებით ფორმის მართვის ყველა ელემენტი მათთვის ნაგულისხმევი წესით მინიჭებულ მნიშვნელობას მიიღებს;
- submit მონაცემთა შესატან ველს ღილაკად გარდაქმნის, რომელზეც მაუსის დაწკაპუნებით ხდება ინფორმაციის გაგზავნა სერვერზე.

ფორმაში მრავალსტრიქონიანი ტექსტური ველების შესაქმნელად, რომელიც მომხმარებელს დიდი მოცულობის ინფორმაციის შეტანის საშუალებას აძლევს, <textarea> ტეგი გამოიყენება. ამ ტეგს name, rows და cols ატრიბუტები აქვს.

- ატრიბუტი name - მონაცემთა შეტანის ველის სახელს განსაზღვრავს;
- ატრიბუტი rows - მონაცემთა შეტანის ველის სიმაღლეს განსაზღვრავს სიმბოლოებში;
- ატრიბუტი cols - მონაცემთა შეტანის ველის სიგანეს განსაზღვრავს სიმბოლოებში;

იმისათვის, რომ მონაცემთა შეტანის ველში ფორმის გახსნისას სისტემაში ნაგულისხმევი წესით რაიმე ტექსტი გამოვიდეს, საჭიროა იგი <textarea> და </textarea> ტეგებს შორის

ჩავსვათ.

- ატრიბუტი value - საშუალებას იძლევა მართვის ელემენტს მიენიჭოს საწყისი მნიშვნელობა ნაგულისხმევი წესით ან მნიშვნელობა, რომელიც name ატრიბუტს შესაბამისი გადამრთველის დაყენების საშუალებით მიენიჭება (გადამრთველის ტიპისათვის მოცემული ატრიბუტი აუცილებელია).

მაგალითი 6.2:

```
<!DOCTYPE html>
<html>
<body>
<form action="">
First name:<br>
<input type="text" name="firstname" value="John">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
</body>
</html>
```

მიღებული შედეგი:

First name:
<input type="text" value="John"/>
Last name:
<input type="text"/>

readonly ატრიბუტი მიუთითებს, რომ input ველის მხოლოდ წაკითხვაა შესაძლებელი.

მაგალითი 6.3:

```
<!DOCTYPE html>
<html>
<body>
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" readonly>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
</body>
</html>
```

ბრაუზერის ფანჯარაში მივიღებთ ზუსტად იმავე შედეგს, რაც წინა მაგალითის შემთხვევაში გვქონდა, იმ განსხვავებით, რომ input ველში მოთავსებული ტექსტის შეცვლას მომხმარებელი ვერ შეძლებს.

- disabled ატრიბუტის მითითება input ველისათვის ნიშნავს იმას, რომ ეს ველი გათიშულია, ის უმოქმედოა. ფორმის მონაცემების გაგზავნისას ველში არსებული მონაცემი არ იგზავნება.

მაგალითი 6.4:

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

```

<body>
<form action="">
First name:<br>
<input type="text" name="firstname" value ="John" disabled>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
</body>
</html>

```

ბრაუზერის ფანჯარაში მიღებული შედეგი ვიზუალურად ისეთივეა, როგორც წინა ორ შემთხვევაში.

- ატრიბუტი `size` - მონაცემთა შესატანი ველის ზომას განსაზღვრავს. ეს ატრიბუტი განსაზღვრავს ველის სიგრძეს და არა მასში შესატანი სიმბოლოების რაოდენობას. თუ მონაცემთა შესატან ველში შეტანილი სიმბოლოების რაოდენობა ველის სიგრძეს აღემატება, მაშინ იქ გადაფურცვლის ზოლი გაჩნდება.

მაგალითი 6.5:

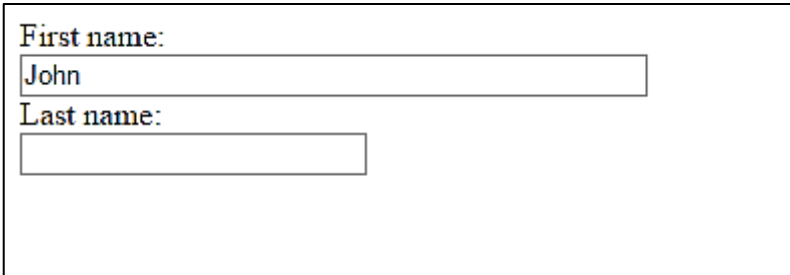
```

<!DOCTYPE html>
<html>
<body>
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" size="40">
<br>
Last name:<br>
<input type="text" name="lastname">

```

```
</form>
</body>
</html>
```

მიღებული შედეგი:



First name:
John
Last name:

- ატრიბუტი `maxlength` - მომხმარებლის მიერ მონაცემთა შესატან ველში შესატანი სიმბოლოების დასაშვები რაოდენობა. ამ რაოდენობაზე მეტი სიმბოლოს შეტანის მცდელობისას ბრაუზერი ხმოვან სიგნალს გამოსცემს და ზედმეტი სიმბოლოს შეტანის საშუალებას არ მოგვცემს.

მაგალითი 6.6:

```
<!DOCTYPE html>
<html>
<body>
<form action="">
First name:<br>
<input type="text" name="firstname" maxlength="10">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

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

ქვემოთ ნაჩვენებია ფორმის ელემენტების გარეთ ჩარჩოს შექმნის მაგალითი.

მაგალითი 6.7:

```
<!DOCTYPE html>
<html>
<head>
<title>forms4</title>
</head>
<body>
<form action="">
<fieldset>
<legend>Personal information:</legend>
Name: <input type="text" size="30"><br><br>
E-mail: <input type="text" size="30"><br><br>
Date of birth: <input type="text" size="10">
</fieldset>
</form>
</body>
</html>
```

Personal information: _____

Name:

E-mail:

Date of birth:

ქვემოთ განხილულ მაგალითში ფორმაში სიების შესაქმნელად <select> წყვილი ტეგი გამოიყენება, რომელიც საშუალებას იძლევა მრავალი შესაძლო მნიშვნელობიდან ერთი ან რამდენიმე მნიშვნელობა ამოირჩეს. ნაგულისხმევი წესის თანახმად, სიის ველში მისი პირველი ელემენტი აისახება. სიის ელემენტი <option> ტეგის საშუალებით იქმნება.

მაგალითი 6.8:

```

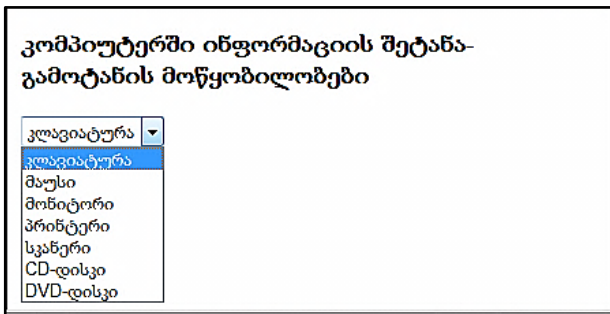
<!DOCTYPE html>
<html>
<head>
<title>ფორმები</title>
</head>
<body>
<h3 align=left>კომპიუტერში ინფორმაციის შეტანა-
გამოტანის მოწყობილობები</h3>
<form>
<select name=group>
<option> კლავიატურა
<option> მაუსი
<option> მონიტორი

```

```

<option> პრინტერი
<option> სკანერი
<option> CD-დისკი
<option> DVD-დისკი
</select>
</form>
</body>
</html>

```



<optgroup> ტეგის საშუალებით სიებში მონაცემთა რაიმე ნიშნით დაჯგუფება შესაძლებელი.

მაგალითი 6.9:

```

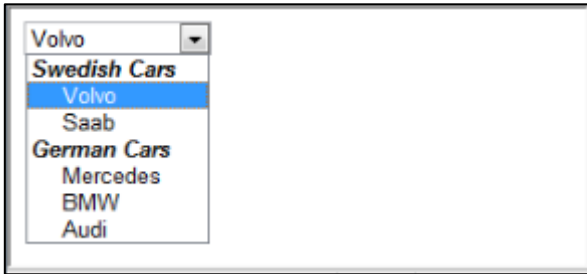
<!DOCTYPE html>
<html>
<body>
<select>
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">

```

```

<option value="mercedes">Mercedes</option>
<option value="BMW">BMW</option>
<option value="audi">Audi</option>
</optgroup>
</select>
</body>
</html>

```



Input ტეგის ახალი ატრიბუტები

HTML 5 ვერსიაში დაემატა input-ის ახალი ატრიბუტები:

- autofocus ატრიბუტი შესაბამის input ველში ავტომატურად მოათავსებს კურსორს. ქვემოთ განხილულ მაგალითში კურსორი ავტომატურად მოათავსდება First name ველში.

მაგალითი 6.10:

```

<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  First name:<input type="text" name="fname" autofocus><br>
  Last name: <input type="text" name="lname"><br>

```



```

    <input type="submit">
  </form>
  <p><strong>Note:</strong> The autofocus attribute of the input
tag is not supported in Internet Explorer 9 and earlier versions.</p>
</body>
</html>

```

• form ატრიბუტი განსაზღვრავს, რომელ ფორმას შეიძლება მიეკუთვნოს <input> ელემენტი. თუ ის მიეკუთვნება რამდენიმე ფორმას, მაშინ ფორმის id მნიშვნელობების ჩამოწერა უნდა მოხდეს და ისინი ერთმანეთისგან ჰარით გამოიყოს. ქვემოთ განხილულ მაგალითში <input> ელემენტი ფორმის ტეგებს გარეთაა მოთავსებული, მაგრამ ის მაინც მიეკუთვნება ამ ფორმას.

მაგალითი 6.11:

```

<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php" id="form1">
First name: <input type="text" name="fname"><br>
<input type="submit" value="Submit">
</form>
<p>The "Last name" field below is outside the form element, but
still part of the form.</p>
Last name: <input type="text" name="lname" form="form1">
</body>
</html>

```

- `formtarget` ატრიბუტი განსაზღვრავს სად უნდა გამოჩნდეს პასუხი ფორმის მონაცემების გაგზავნის შემდეგ. `formtarget` ატრიბუტი გადაფარავს `<form>` ელემენტის `target` ატრიბუტს. `formtarget` შეიძლება გამოყენებული იყოს `type="submit"` და `type="image"` ელემენტებთან.

მაგალითი 6.12:

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit as normal">
  <input type="submit" formtarget="_blank" value="Submit to a
new window/tab">
</form>
<p><strong>Note:</strong> The formtarget attribute of the
input tag is not supported in Internet Explorer 9 and earlier
versions.</p>
</body>
</html>
```

- `height` და `width` ატრიბუტები განსაზღვრავს `<input type="image">` ელემენტის სიმაღლეს და სიგანეს. ქვემოთ მოცემულ მაგალითში გამოსახულება ასრულებს ფორმის მონაცემების გაგზავნის დილაკის როლს. სწორედ მისთვის არის განსაზღვრული სიგანე და სიმაღლე.

მაგალითი 6.13:

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="image" src="img_submit.gif" alt="Submit"
  width="48" height="48">
</form>
<p><b>Note:</b> The input type="image" sends the X and Y
coordinates of the click that activated the image button.</p>
</body>
</html>
```

- list ატრიბუტი მიუთითებს <datalist> ელემენტზე, რომელიც შეიცავს <input> ელემენტისთვის წინასწარ განსაზღვრულ მონაცემებს.

ქვემოთ მოცემულ მაგალითში განხილულია <input> ელემენტი <datalist>-ში წინასწარ განსაზღვრული მონაცემებით.

მაგალითი 6.14:

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php" method="get">
<input list="browsers" name="browser">
<datalist id="browsers">
  <option value="Internet Explorer">
```

```

<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>
<input type="submit">
</form>
<p><b>Note:</b> The datalist tag is not supported in Internet
Explorer 9 and earlier versions, or in Safari.</p>
</body>
</html>

```

- min და max ატრიბუტები <input> ელემენტისთვის მინიმალურ და მაქსიმალურ მნიშვნელობებს განსაზღვრავს. min და max ატრიბუტების გამოყენება შესაძლებელია input-ის შემდეგ მნიშვნელობებთან: number, range, date, datetime-local, month, time და week.

მაგალითი 6.15:

```

<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
  Quantity (between 1 and 5):

```

```

    <input type="number" name="quantity" min="1"
max="5"><br>
    <input type="submit">
</form>
<p><strong>Note:</strong> The max and min attributes of the
input tag is not supported in Internet Explorer 9 and earlier versions,
or in Firefox.</p>
<p><strong>Note:</strong> The max and min attributes will
not work for dates and time in Internet Explorer 10, since IE 10 does
not support these input types.</p>
</body>
</html>

```

• multiple ატრიბუტი მიუთითებს იმაზე, რომ მომხმარებელს შეუძლია შეიტანოს ერთზე მეტი მონაცემი <input> ელემენტში. multiple ატრიბუტის გამოყენება შესაძლებელია input-ის შემდეგ მნიშვნელობებთან: email და file.

ქვემოთ განხილულ მაგალითში, ფაილის ატვირთვის ველიდან, შესაძლებელია რამდენიმე ფაილის ატვირთვა.

მაგალითი 6.16:

```

<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  Select images: <input type="file" name="img" multiple>
  <input type="submit">
</form>

```

```
<p>Try selecting more than one file when browsing for files.</p>
<p><strong>Note:</strong> The multiple attribute of the input tag is not supported in Internet Explorer 9 and earlier versions.</p>
</body>
</html>
```

• pattern (regexp) ატრიბუტი განსაზღვრავს რეგულარულ გამოსახულებას, რომელსაც შედარდება <input> ელემენტის მნიშვნელობა.

ქვემოთ განხილულ მაგალითში input ელემენტი უნდა შეიცავდეს ლათინური ანბანის მხოლოდ სამ ასოს (არ უნდა შეიცავდეს რიცხვებს და არც სპეციალურ სიმბოლოებს).

მაგალითი 6.17:


```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  Country code: <input type="text" name="country_code"
pattern="[A-Za-z]{3}" title="Three letter country code">
  <input type="submit">
</form>
<p><strong>Note:</strong> The pattern attribute of the input tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>
</body>
</html>
```

- placeholder ატრიბუტი იძლევა მინიშნებას input ველში შესატანი მონაცემისთვის.

მაგალითი 6.18:

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  <input type="text" name="fname" placeholder="First
name"><br>
  <input type="text" name="lname" placeholder="Last
name"><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

მიღებული შედეგი:



The image shows a rendered HTML form. It consists of a rectangular container with a thin black border. Inside the container, there are two text input fields stacked vertically. The top field has the placeholder text "First name" and a small cursor icon on the left. The bottom field has the placeholder text "Last name" and a small cursor icon on the left. Below these two fields is a rectangular button with the text "Submit" centered on it.

- required ატრიბუტი განსაზღვრავს შეტანის ველის შევსების აუცილებლობას, ფორმის მონაცემების გაგზავნამდე. ველის შეუვსებლად დატოვების შემთხვევაში, გამოვა შეტყობინება შეცდომის შესახებ.

required ატრიბუტის გამოყენება შესაძლებელია input-ის შემდეგ მნიშვნელობებთან: text, search, url, tel, email, password, date pickers, number, checkbox, radio და file.

მაგალითი 6.19:

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  Username: <input type="text" name="username" required>
  <input type="submit">
</form>
<p><strong>Note:</strong> The required attribute of the input
tag is not supported in Internet Explorer 9 and earlier versions, or in
Safari.</p>
</body>
</html>
```

ლილაკის შექმნა

მიუხედავად იმისა, რომ ფორმებში ლილაკების შესაქმნელად <input> ტეგი გამოიყენება, HTML 5-ში ამ მიზნით კიდევ <button> ტეგის გამოყენებაა შესაძლებელი. <button> აქვს შემდეგი ატრიბუტები: type, name, value, autofocus, disabled, form, formaction, formenctype, formmethod და formnovalidate.

- ატრიბუტი type ლილაკის ტიპს განსაზღვრავს. მას შემდეგი მნიშვნელობების მიღება შეუძლია: button - ჩვეულებრივი, submit - ფორმის გაგზავნისა და reset - გაგზავნის შეწყვეტის ლილაკები;

- ატრიბუტი name ღილაკის სახელს განსაზღვრავს;
- ატრიბუტი value ღილაკის წარწერას განსაზღვრავს;
- ატრიბუტი autofocus მიუთითებს, რომ ღილაკი გვერდის ჩატვირთვისთანავე გახდეს აქტიური;
- ატრიბუტი disabled მიუთითებს, რომ ღილაკი უნდა იყოს უმოქმედო (გამორთული);
- form ატრიბუტის მნიშვნელობა იმ ფორმის სახელია, რომელსაც აღნიშნული ღილაკი ეკუთვნის;
- formaction ატრიბუტის მნიშვნელობა ის URL მისამართია, რომელზეც ფორმის მონაცემები უნდა გადაიგზავნოს. გამოიყენება მაშინ, როდესაც type=submit;
- ატრიბუტი formenctype მიუთითებს, თუ როგორ უნდა მოხდეს ფორმის მონაცემების კოდირება მისი გადაგზავნის წინ (გამოიყენება მხოლოდ type=submit შემთხვევაში). მისი შესაძლო მნიშვნელობებია: application, x-www-form-urlencoded, multipart/form-data text და plain;
- ატრიბუტი formmethod ფორმის მონაცემების გაგზავნის მეთოდს მიუთითებს. გამოიყენება მხოლოდ, როცა type=submit. მისი შესაძლო მნიშვნელობებია: get და post;
- ატრიბუტი formnovalidate მიუთითებს იმაზე, რომ ფორმის მონაცემები გაგზავნის წინ არ უნდა შემოწმდეს. გამოიყენება, მხოლოდ, როცა type=submit.

ქვემოთ გადამრთველების, ალმით მოსანიშნი ველის და ღილაკების შექმნის კოდი და მიღებული შედეგია მოცემული:

მაგალითი 6.20:

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

```
<head><title>input1.html</title></head>
<body>
<form name = "anketa" method = "post" action =
"http://mysite.ge/script">
<b> შენი სახელი </b> <br>
<input type = text name = "შენი სახელი" size = 37 > <br> <br>
<b> თქვენ სქესი: </b>
მამრობითი <input type = radio name = „სქესი“ value =
“Male”>
მდედრობითი <input type = radio name = „სქესი“ value =
“Female”>
<br> <br>
<b> რა სახის ფილმები მოგწონთ?
</b> <br>
<input type = checkbox name = "მხატვრული" value = "yes">
მხატვრული <br>
<input type = checkbox name = "თრილერი" value =
"yes">თრილერი <br>
<input type = checkbox name = "სათავგადასავლო" value =
"yes"> სათავგადასავლო <br>
<input type = checkbox name = "მელოდრამა" value = "yes">
მელოდრამა <br>
<input type = checkbox name = "დოკუმენტური" value =
"yes"> დოკუმენტური <br> <br>
<b> თქვენი E-mail: </b> <br>
<input type = text name = "email" size = 37”><br><br>
<input type = Submit value = "გაგზავნა"><input type = reset
value = "გაუქმება">
```

```
</form>
```

```
</body>
```

```
</html>
```

შენი სახელი

თქვენ ხართ: მამრობითი მდედრობითი

რა სახის ფილმები მოგწონთ?

- მხატვრული
- თრილერი
- სათავგადასავლო
- მელოდრამა
- დოკუმენტური

თქვენი E-mail:

როგორც ხედავთ, ყველაფერში „დამნაშავე“ თქვენთვის უკვე ნაცნობი ტეგი `<input>`-ია, მხოლოდ ამჯერად მისი ატრიბუტი ღებულობს `checkbox` (`type=checkbox`) მნიშვნელობას. მიაქციეთ ყურადღება, რომ თითოეულ „ალამს“ თავისი უნიკალური სახელი აქვს, სამაგიეროდ მნიშვნელობა (`value`) ყველასათვის ერთნაირია.

თუ გსურთ, რომელიმე ალამი ან გადამრთველი ავტომატურად იყოს მონიშნული, ამ „ალმისთვის“ ან გადამრთველისთვის კოდში შეიტანეთ ატრიბუტი `checked`:

```
<input type=checkbox name=“adventure” value=“yes” checked>  
adventure <br>
```

შედეგად ფორმაში, რომელიც საიტის დამთვალიერებელს შესავსებად წარედგინება, პუნქტი adventure მონიშნული იქნება. რა თქმა უნდა, მომხმარებელს შეუძლია მოაცილოს ეს აღნიშვნა და შემდეგ შეარჩიოს ის, რომელიც სურს.

მულტიმედია

მულტიმედია პირველ რიგში აუდიო- და ვიდეომასალაა. მულტიმედია ვებდოზინის დანართში ვებგვერდებზე განთავსებული აუდიო- და ვიდეორგოლებია.

უკანასკნელ პერიოდამდე აუდიო- და ვიდეორგოლების ვებგვერდებზე განთავსება საკმაოდ დიდი HTML კოდის ჩაწერით ან დამატებითი პროგრამით იყო შესაძლებელი. ამჟამად, HTML-5 გამოსვლის შემდეგ ეს პროცესი გამარტივდა და მხოლოდ ჩვენთვის ნაცნობი ტეგი გამოიყენება.

ფაილების ფორმატი და კოდირების ფორმატები

მულტიმედიური ფაილების მრავალი ფორმატი არსებობს. როგორც ინტერნეტ-გრაფიკის შემთხვევაში, აქაც ბრაუზერი ყველა ფორმატს არ უჭერს მხარს. მულტიმედიური ფაილების კოდირება თითქმის ყოველთვის სხვადასხვა ფორმატით ხდება. პრაქტიკულად, მულტიმედიური ფაილების კოდირების ყველა ფორმატი მის შეკუმშვას უჭერს მხარს. ამის გამო, მათი ზომები შედარებით მცირდება, რაც მათი ქსელით გადაცემის სიჩქარეზე დადებითად მოქმედებს. ქვემოთ ჩამოთვლილია და მოკლედ აღწერილი ვებდოზინში გამოყენებული და ბრაუზერების მიერ მხარდაჭერილი მულტიმედიური ფაილების ფორმატები.

- *WAV* - მულტიმედიურ ფორმატებს შორის ყველაზე ძველი. გასული საუკუნის 90-იანი წლების დასაწყისში Microsoft-ის მიერ აუდიომონაცემების შესანახად იყო დამუშავებული და ამჟამადც გამოიყენება. ასეთი ფორმატის ფაილებს wav გაფართოება აქვს;
- *OGG* - შედარებით ახალი ფორმატი. იგი Xiph.org არაკომერციული ორგანიზაციის მიერ აუდიო- და ვიდეომონაცემების შესანახად დაახლოებით თორმეტი წლის წინ დამუშავდა. ამ ტიპის ფაილებს ogg (უნივერსალური ფორმატი), oga (აუდიოფაილების) და ogv (ვიდეოფაილების) გაფართოება აქვს. ბოლო ორი გაფართოება ძალზე იშვიათად გვხვდება;
- *MP4* - შედარებით ახალი ფორმატია. ეს ფორმატი Motion Picture Expert Group (მოდრავი გამოსახულების საკითხების ექსპერტთა ჯგუფი) ორგანიზაციის მიერ 1998 წელს აუდიო- და ვიდეომონაცემთა შენახვის მიზნით იყო დამუშავებული. ამ ფორმატის ფაილებს mp4 გაფართოება აქვს;
- *QuickTime* - ფორმატი ძალიან ძველია და Apple ფირმის მიერ 1989 წელს აუდიო- და ვიდეომონაცემების შესანახად იყო დამუშავებული. ასეთი ფორმატის ფაილებს mov გაფართოება აქვს;
- *PCM* (Pulse-Coded Modulation - იმპულსურ-კოდური მოდულაცია) - კოდირების ყველაზე მარტივი და ყველაზე ძველი ფორმატი. ის შემჭიდროებასაც კი არ უწევს მხარდაჭერას. აუდიომონაცემების კოდირებისათვის გამოიყენება;

- *Vorbis* - კოდირების თანამედროვე ფორმატი. დამუშავებულია 2002 წელს Xiph.org ორგანიზაციის მიერ. გამოიყენება აუდიომონაცემების კოდირებისათვის;
- *AAC* (Advanced Audio Coding - აუდიოს განვითარებული კოდირება) - კოდირების არც ისე ახალი ფორმატი. 1997 წელს დაამუშავა Motion Picture Expert Group ორგანიზაციამ. გამოიყენება აუდიომონაცემების კოდირებისათვის.
- *Theora* - კოდირების თითქმის ყველაზე „ახალგაზრდა“ ფორმატი. ისიც რამდენიმე წლის წინ Xiph.org ორგანიზაციამ დაამუშავა. გამოიყენება ვიდეომონაცემების კოდირებისათვის;
- *H.264* - ძალიან „ახალგაზრდა“ ფორმატია. 2003 წელს Motion Picture Expert Group და Video Coding Experts Group (ვიდეოს კოდირების ექსპერტთა ჯგუფი) ორგანიზაციამ წარმოადგინა ვიდეო მონაცემების კოდირებისათვის.

თითქმის ყველა ეს ფორმატი ღიაა, გამონაკლისი მხოლოდ QuickTime ფორმატის ფაილებია, რომლებიც Apple ფირმას მიეკუთვნება და H.264 კოდირების ფორმატი, რომელიც ასზე მეტი პატენტითაა დაცული.

პირველ ცხრილში მოყვანილია ბრაუზერებისა და კოდირების ფორმატების ურთიერთმხარდაჭერა.

როგორც ცხრილიდან ჩანს, სხვადასხვა ბრაუზერი სხვადასხვა ფორმატს უჭერს მხარს, ამან კი ვებგვერდების დამპროექტებლებს შეიძლება გარკვეული პრობლემები შეუქმნას.

აუდიო	ვიდეო	Firefox	Opera	Safari	Chrom e
აუდიო ინფორმაციის შემცველი ფაილები					
WAV-PCM		*	*		*
OGG-Vorbis		*	*		*
MOV-AAC				*	
აუდიო- და ვიდეოინფორმაციის შემცველი ფაილები					
OGG-Vorbis	OGG- Theora	*	*		*
MOV-AAC	MOV-H.264			*	
MP4-AAC	MP4-H.264			*	

MIME ტიპები

ქსელის საშუალებით სულ სხვადასხვა მონაცემი გადაიცემა: ვებგვერდები, გრაფიკული გამოსახულება, აუდიო- და ვიდეოფაილები, არქივი, შესრულებადი ფაილები და სხვა. ეს მონაცემები სხვადასხვა პროგრამისათვისაა გათვალისწინებული. ამასთან, მიმღები პროგრამა შეიძლება სხვადასხვანაირად მოიქცეს. ასე, მაგალითად, ვებგვერდის ან გრაფიკული გამოსახულების მიღების შემთხვევაში მიმღები პროგრამა მას ეკრანზე გამოიტანს, ხოლო თუ ეს ფაილი არქივი ან შესრულებადი ფაილია, მაშინ მას გახსნის ან დისკზე შეინახავს.

ქსელის საშუალებით გადაცემულ ყველა მონაცემს განსაკუთრებული აღნიშვნა სჭირდება, რომელიც ცალსახად მის ბუნებაზე მიუთითებს. ამ აღნიშვნას MIME (Multipurpose Internet Mail Extensions, ინტერნეტის ფოსტის მრავალმიზნობრივი

გაფართოება) ტიპი ეწოდება. პროგრამების მონაცემებს ამ ტიპს მისი გამგზავნი ანიჭებს, მაგალითად, ვებსერვერი. ხოლო მიმღები პროგრამა MIME ტიპის მიხედვით განსაზღვრავს, მოცემული ბრაუზერი მხარს უჭერს თუ არა მიღებულ მონაცემებს და თუ უჭერს, რა უყოს მას.

ვებგვერდის MIME ტიპია text/html, GIF ფორმატის გრაფიკული გამოსახულების - image/gif., შესრულებადი ფაილის - application/x-msdownload, ხოლო ZIP არქივის - application/x-zip-compressed. თავისი MIME ტიპები აქვს მულტიმედიურ ფაილებსაც.

მე-2 ცხრილში მოყვანილია მულტიმედიური ფაილების ფორმატების MIME ტიპები, რომლებსაც მხარს უჭერს თანამედროვე ბრაუზერები.

ცხრილი 2

ფაილის ფორმატი	MIME ტიპი
WAV	audio/wave audio/wav audio/x-wav audio/x-pn-wav
OGG	audio/ogg (აუდიოფაილებისათვის) video/ogg (ვიდეოფაილებისათვის) application/ogg
MP4	video/mp4
QuickTime	video/quicktime

როგორც ცხრილიდან ჩანს ერთი ფორმატის ფაილებს შეიძლება რამდენიმე MIME ტიპი ჰქონდეს. ჩვეულებრივ, ამ ტიპებიდან, ძირითადად, როგორც ყველაზე საპატიო, პირველი ტიპი გამოიყენება.

აუდიორგოლის ჩასმა

ვებგვერდზე აუდიორგოლის ჩასასმელად HTML-5 ენა წყვილ `<audio>` ტეგს იყენებს. ფაილის ინტერნეტ-მისამართს, რომელშიც მოცემული აუდიორგოლი ინახება, SRC ატრიბუტის საშუალებით მიეთითება:

```
<audio src="sound.wav"></audio>
```

ბრაუზერმა, `<audio>` ტეგს როგორც კი შეხვდება, შეიძლება აუდიო-ფაილი მაშინვე ჩატვირთოს და გაუშვას ან მხოლოდ ჩატვირთოს და არაფერი არ გააკეთოს. მან ასევე შეიძლება გამოიტანოს მართვის ელემენტები, რომლის საშუალებითაც მომხმარებელს შეუძლია აუდიოფაილი გაუშვას, შეაჩეროს, წინ და უკან გადაახვიოს, არეგულიროს ხმა. ყოველივე ეს `<audio>` ტეგის ატრიბუტების საშუალებით რეგულირდება.

`<audio>` ტეგი ვებგვერდზე ბლოკურ ელემენტს ქმნის. ასე, რომ აუდიორგოლის აბზაცში ჩასმას ვერ შევძლებთ. სამაგიეროდ, ის რომ ცალკე აბზაცად იყოს გამოყოფილი, ამისათვის არავითარი დამატებითი მოქმედება არ არის ჩასატარებელი.

ბრაუზერი აუდიორგოლის ჩუმათობის პრინციპით გაშვებას ვერ უზრუნველყოფს, ამისათვის `<audio>` ტეგში განსაკუთრებული ატრიბუტი - `AUTOPLAY` უნდა მივუთითოთ. მას არ გააჩნია მნიშვნელობა - მთავარია იგი ეწეროს ტეგში და აუდიორგოლი ჩატვირთვისთანავე შესრულებაზე გაეშვება:

```
<p>ეხლა თქვენ გაიგონებთ ხმას!</p>
```

```
<audio src="sound.ogg" AUTOPLAY></audio>
```

თუ `<audio>` ტეგში მნიშვნელობის გარეშე `CONTROLS` ატრიბუტს მივუთითებთ, მაშინ ბრაუზერი აუდიორგოლის გაშვების მართვის ელემენტებს გამოიტანს. მასში შედის

გაშვებისა და გაჩერების ღილაკები, დროის სკალა და ხმის რეგულატორი:

```
<p>დააჭირეთ გამშვებ ღილაკს, რათა გაიგოთ ხმა.</p>
<audio src="sound.ogg" controls></audio>
```

თუ <audio> ტეგში მნიშვნელობის გარეშე autobuffer ატრიბუტს მივუთითებთ, მაშინ ბრაუზერი ვებგვერდის ჩატვირთვის შემდეგ მაშინვე აუდიორგოლსაც ჩატვირთავს, რათა შეეყოვნება არ მოხდეს. ამასთან, უნდა გვახსოვდეს, რომ ამ ატრიბუტის გამოყენება მხოლოდ იქ შეიძლება, სადაც ატრიბუტი autoplay არ არის გამოყენებული.

ქვემოთ მოყვანილია აუდიორგოლის გამოტანის მაგალითი:

პროგრამა 1.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>AUDIO ტეგი</title>
</head>
<body>
<h1>AUDIO ტეგი</h1>
<p>ტეგი AUDIO ვებგვერდზე აუდიორგოლის ჩასასმელად გამოიყენება.</p>
<h2>მაგალითი:</h2>
<pre>&lt;AUDIO src=&quot;Kalimba.wav&quot;
CONTROLS&gt;&lt;/AUDIO&gt; </pre>
<h2>შედეგი:</h2>
```

```
<audio src="Kalimba.wav" controls></audio>
</body>
</html>
```

AUDIO ტეგი

ტეგი AUDIO ვებგვერდზე აუდიორგოლის ჩასასმელად გამოიყენება.

მაგალითი:

```
<AUDIO src="Kalimba.wav" CONTROLS></AUDIO>
```

შედეგი:



შექმნილი ვებგვერდი და აუდიორგოლი ერთი და იგივე საქალაქოში უნდა შევინახოთ.

ვიდეორგოლის ჩასმა

ვებგვერდზე ვიდეორგოლის ჩასასმელად წყვილი <video> ტეგი გამოიყენება. ვიდეოფაილის ინტერნეტ-მისამართი ამ ტეგის src ატრიბუტით მიეთითება:

```
<video src="film.ogg"></video>
```

ამ ტეგის შეხვედრის შემთხვევაში, ბრაუზერი ვებგვერდის ამ ადგილზე ვიდეორგოლის ჩვენების პანელს გამოიტანს. <video> ტეგში მითითებული ატრიბუტების

მიხედვით მას შეუძლია მაშინვე ჩატვირთოს ვიდეორგოლი და გაუშვას, მხოლოდ ჩატვირთოს ან საერთოდ არაფერი არ გააკეთოს. ასევე, მას შეუძლია ვებგვერდზე ვიდეორგოლის გაშვების მართვის ელემენტები გამოიტანოს.

როგორც <audio> ტეგი, ასევე <video> ტეგი ვებგვერდის ბლოკურ ელემენტს ქმნის და autoplay, controls და autobuffer ატრიბუტი გააჩნია. მაგალითად,

```
<video src="film.ogg" autoplay controls></video>
```

თუ ვიდეორგოლის ჩვენება ჯერ არ არის გაშვებული, მაშინ ჩვენების პანელში მისი პირველი კადრი ან საერთოდ არაფერი არ იქნება გამოტანილი (კონკრეტული მისი ქმედება ბრაუზერზეა დამოკიდებული). მაგრამ ჩვენ შეგვიძლია მივუთითოთ ის გრაფიკული გამოსახულება, რომელიც პირველად, ფარდის სახით იქნება გამოტანილი. ამისათვის, <video> ტეგის poster ატრიბუტი გამოიყენება. მისი მნიშვნელობა საჭირო გრაფიკული ფაილის ინტერნეტ-მისამართზე მიუთითებს. მაგალითად,

```
<video src="film.ogg" controls poster="filmposter.jpg"> </video>
```

შენიშვნა. თუ საჭირო ფორმატის ვიდეორგოლს ვერ მოძებნით, მაშინ შეიძლება თქვენ შექმნათ, სხვა ფორმატში შენახული ვიდეორგოლის კოდირების შეცვლის საშუალებით. ამისათვის SUPER © უტილიტა გამოიყენება, რომელიც შემდეგ ინტერნეტ მისამართზე შეიძლება იპოვოთ:

<http://www.erightsoft.com/SUPER.html>.

ქვემოთ ვებგვერდზე ვიდეორგოლის ჩასმის მაგალითია მოყვანილი:

პროგრამა 2.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-
8">
<title>VIDEO ტეგი</title>
</head>
<body>
<h1>VIDEO ტეგი</h1>
<p>VIDEO ტეგი ვებგვერდზე ვიდეორგოლის ჩასასმელად
გამოიყენება.</P>
<h6>მაგალითი:</h6>
<pre>&lt;VIDEO SRC=&quot; Wildlife.wmv&quot;
CONTROLS&gt;&lt;/VIDEO&gt; </pre>
<h6>შედეგი:</h6>
<video src="wildlife.wmv" controls></video>
</body>
</html>
```

დავალბა. შექმენით დეტექტივის მოყვარულთა კლუბის სარეგისტრაციო გვერდი შემდეგი ნიმუშის მიხედვით:

დეტექტივების მოყვარულთა კლუბის სარეგისტრაციო გვერდი

ანკეტის შევსების შემდეგ შეგემოვებათ ისარგებლოთ ელექტრონული ბიბლიოთეკით

	შეიყვანეთ რეგისტრაციის სახელი
	შეიყვანეთ პაროლი
	დადასტურეთ პაროლი

თქვენი ასაკი:

- 20-მდე
- 20-30
- 30-50
- 50-ის ზემოთ

რომელ ენაზე კითხულობთ:

- ქართული
- ინგლისური
- გერმანული
- ფრანგული
- რუსული

წიგნის რომელი ფორმატია თქვენთვის მისაღები:

PDF ▼

თქვენი საყვარელი ავტორი:

OKგაუქმება

თავი 7

CSS3

- CSS კოდის HTML დოკუმენტთან დაკავშირების მეთოდები;
- HTML ელემენტისთვის სტილის დანიშვნა id და class იდენტიფიკატორების საშუალებით;
- სტილების გარე ცხრილის გამოყენება;
- HTML ფერები.

CSS კოდის HTML დოკუმენტთან დაკავშირება

CSS ტექნოლოგია HTML-ის ნებისმიერ ტეგთან მუშაობის საშუალებას იძლევა. HTML-ის გამოყენებით ვებგვერდის სტრუქტურა უკვე ავაგეთ, CSS კი მოცემული სტრუქტურის დახვეწასა და მისი დიზაინის სრულყოფაში დაგვეხმარება.

CSS კოდის HTML დოკუმენტში ჩართვა სამი ხერხითაა შესაძლებელი.

style ატრიბუტი

HTML დოკუმენტის მიმართ CSS-ის გამოყენება HTML ობიექტის style ატრიბუტის საშუალებით.

CSS სტილების ჩაწერის სინტაქსი ასეთია:

HTML ობიექტის თვისება: მნიშვნელობა;

გამოვიყენოთ ატრიბუტი style დოკუმენტის ფონის შესაქმნელად. მას ასეთი სახე ექნება:

```
<body style="background-color:#ffff05;">
```

სადაც background-color CSS-ის საშუალებით ჩაწერილ HTML ობიექტის თვისებაა, რომლის მნიშვნელობაა - #ffff05.

style ატრიბუტის საშუალებით ფონის შესაქმნელად CSS-ის კოდი ჩავსვათ უკვე არსებულ დოკუმენტში.

მაგალითი 7.1:

```
<!DOCTYPE html>  
<html>  
<head>  
<title> my first programm</title>
```


</head>

<body style="background-color:#ffff05;" topmargin=20
bottommargin=20 leftmargin=20 rightmargin=20>

<h2> ვარდი </h2>

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10-12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენტფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2-15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ვარდისფერი, წითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუჭაა, რომელიც ხორცოვან ცრუნაყოფშია.

</body>

</html>

ვარდი

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10-12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენტფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთბუზა (2-15 სმ დიამეტრის). არომატული ან უსუნო; ფერი - ვარდისფერი, წითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასმნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუჭაა, რომელიც ხორცოვან ცრუნაყოფმა.

CSS-ის საშუალებით აბზაცში ტექსტის დაფორმატება შეიძლება შემდეგნაირად განხორციელდეს:

```
<p style="color:red;font-weight:bold;">
```

სადაც color და font-weight CSS-ის საშუალებით ჩაწერილ HTML ობიექტის თვისებებია, რომელთა მნიშვნელობები შესაბამისად red და bold-ია. აღნიშნული ბრძანება აბზაცში ტექსტისთვის წითელი ფერის დაყენებას და მის გამუქებას უზრუნველყოფს.

მაგალითი 7.2:

```
<!DOCTYPE html>
<html>
<title> my first programm</title>
</head>
<body style="background-color:#ffff05;" topmargin=20
bottommargin=20 leftmargin=20 rightmargin=20>
<h2> ვარდი </h2>
<p align=center style="color:red; font-weight:bold;">
```

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10-12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენჭფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2-15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ვარდისფერი, წითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუქაა, რომელიც ხორცოვან ცრუნაყოფშია.

</body>

</html>

ვარდი

ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10-12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენჭფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2-15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ვარდისფერი, წითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუქაა, რომელიც ხორცოვან ცრუნაყოფშია.

<style> ტეგი

განვიხილოთ CSS კოდის HTML დოკუმენტში ჩართვის მეორე მეთოდი. თქვენ უკვე გამოიყენეთ CSS კოდის ჩასმა style ატრიბუტის მეშვეობით. ახლა HTML დოკუმენტში CSS კოდი <style> ტეგის მეშვეობით ჩავსვათ.

HTML დოკუმენტში CSS კოდის ჩასმისას <style> ტეგი-კონტეინერი <head> განყოფილებაში თავსდება. <style> ტეგში მოთავსებული სელექტორების სინტაქსი შემდეგნაირია:

```
სელექტორი
{
    თვისება: მნიშვნელობა;
}
```

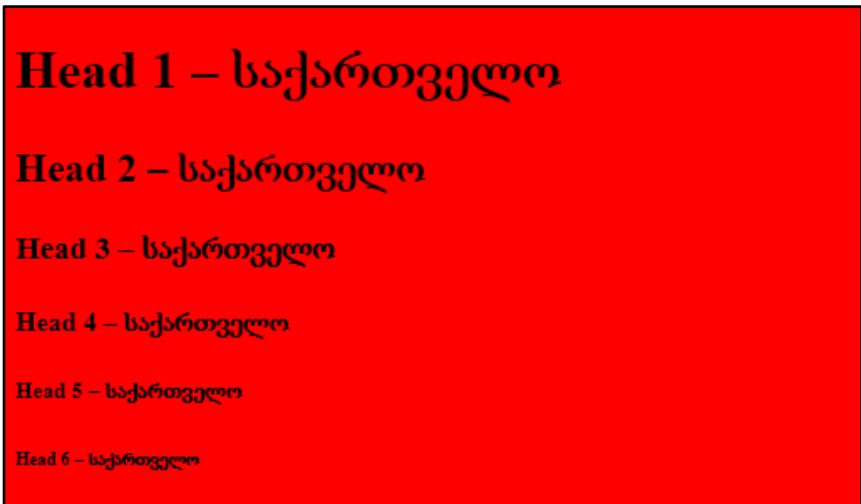
სელექტორი მიუთითებს, თუ რომელი HTML ტეგის მიმართ გამოიყენება ამ სელექტორის ტანში გაწერილი თვისებები. ფიგურულ ფრჩხილებში იწერება სელექტორის ტანი, რომელიც კონკრეტული სელექტორის (HTML ტეგის) CSS სტილებს განსაზღვრავს.

მაგალითი 7.3:

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

```
<body>
<h1> Head 1 – საქართველო </h1>
<h2> Head 2 – საქართველო </h2>
<h3> Head 3 – საქართველო </h3>
<h4> Head 4 – საქართველო </h4>
<h5> Head 5 – საქართველო </h5>
<h6> Head 6 – საქართველო </h6>
</body>
</html>
```

ჩვენს შემთხვევაში, სელექტორი არის body და აქვს თვისება background-color, რომელიც გვერდის ფონის ფერის შერჩევას უზრუნველყოფს. მოცემულ შემთხვევაში მისი მნიშვნელობაა red.

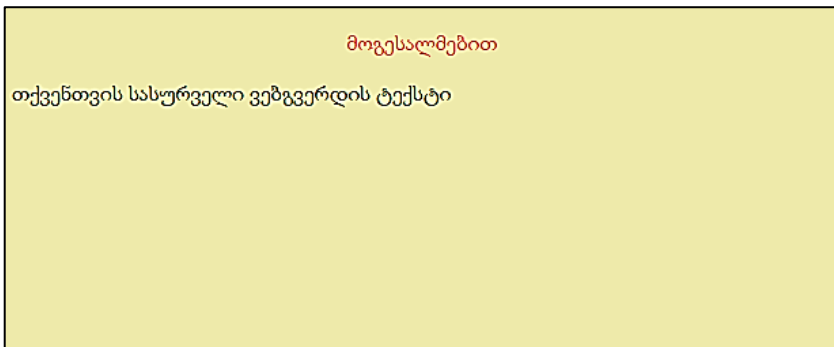


HTML ელემენტისთვის სტილის დანიშვნა ასევე id და class იდენტიფიკატორების საშუალებითაც შეიძლება მოხდეს.

განვიხილოთ ID იდენტიფიკატორის მაგალითი.

მაგალითი 7.4:

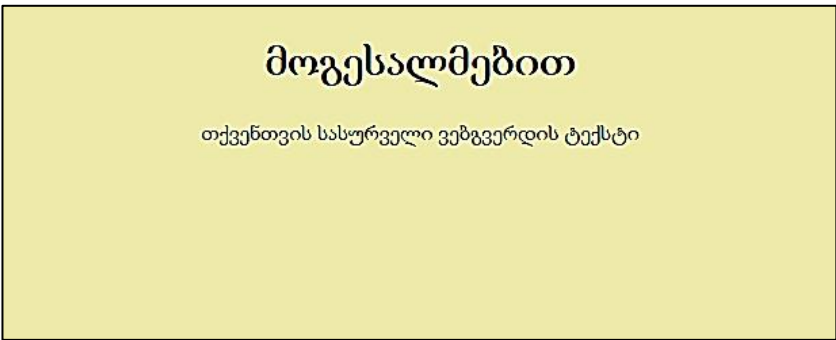
```
<!DOCTYPE html>
<html>
<head>
<title> formatting </title>
<style>
#style1
{text-align:center;
color:red;}
</style>
</head>
<body bgcolor=#eeeeaa>
<p id="style1">მოგესალმებით</p>
<p>თქვენთვის სასურველი ვებგვერდის ტექსტი</p>
</body>
</html>
```



განვიხილოთ class იდენტიფიკატორის მაგალითი.

მაგალითი 7.5:

```
<!DOCTYPE html>
<html>
<head>
<title> formatting </title>
<style>
.center
{text-align:center;}
</style>
</head>
<body bgcolor=#eeeeaa>
<h1 class="center">მოგესალმებით</h1>
<p class="center">თქვენთვის სასურველი ვებგვერდის
ტექსტი</p>
</body>
</html>
```



ამ მაგალითში, ყველა HTML ელემენტი, რომლისთვისაც class="center" ცენტრში მოთავსდება, რისი განსაზღვრაც <style>... </style> ტეგებში მოთავსებული ბრძანებით ხდება:

```
.center {text-align:center;}
```

ცენტრში მოთავსდება სათაური და აბზაცი.

ასევე შესაძლებელია, რომ კლასის მოქმედება მხოლოდ განსაზღვრულ HTML ელემენტზე გავრცელდეს, თუ center კლასი მხოლოდ h1 ელემენტისთვის იქნება განსაზღვრული,

h1.center {text-align:center;}

ამ შემთხვევაში მხოლოდ h1 ელემენტისთვის იმოქმედებს, მიუხედავად იმისა, რომ class მითითებულია სათაურშიც და აბზაცშიც.

მაგალითი 7.6:

```
<!DOCTYPE html>
<html>
<head>
<title> formatting </title>
<style>
h1.center
{text-align:center;}
</style>
</head>
<body bgcolor=#eeeeaa>
<h1 class="center">მოგესალმებით</h1>
<p class="center">თქვენთვის სასურველი ვებგვერდის
ტექსტი</p>
</body>
</html>
```


მოგესალმებით

თქვენთვის სასურველი ვებგვერდის ტექსტი

სტილების გარე ცხრილის გამოყენება

ეს მეთოდი ე. წ. სტილების გარე ცხრილზე მიმართავს გულისხმობს. სტილების გარე ცხრილი .css გაფართოების მქონე ტექსტური ფაილია.

დავუშვათ, გვაქვს სტილების ცხრილი, რომელსაც style.css დავარქვით და ის style საქალაქდემია მოთავსებული. ამოცანა იმაში მდგომარეობს, რომ HTML დოკუმენტიდან (default.html) სტილების ცხრილზე (style.css) მიმართვა შეიქმნას. ეს შეიძლება HTML კოდის ერთი სტრიქონის საშუალებით გაკეთდეს:

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

ეს სტრიქონი HTML დოკუმენტში <head> და </head> ტეგებს შორის უნდა განთავსდეს. იგი ბრაუზერს მიუთითებს, რომ მან HTML ფაილის ეკრანზე ასახვის წესი CSS ფაილიდან უნდა აიღოს. ყველაზე მნიშვნელოვანი ის არის, რომ სტილების ერთ ცხრილს შეიძლება რამდენიმე HTML დოკუმენტმა მიმართოს. სტილების ცხრილში შეტანილი ცვლილება ყველა იმ HTML დოკუმენტში აისახება, რომლებიც სტილების ამ ცხრილს მიმართავს.

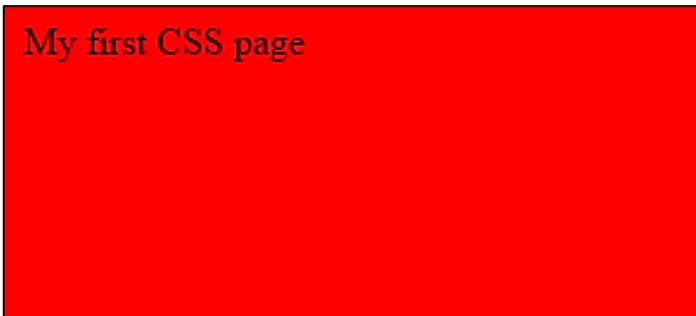
მაგალითი 7.7:

```
<!DOCTYPE html>
<html>
<head>
<title>
სტილების გარე ცხრილთან დაკავშირება
</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
My first CSS page
</body>
</html>
```

Style.css-ში ვწერთ კოდს:

```
body {
background-color: #ff0000;
}
```

მიღებული შედეგი მოცემულია ქვემოთ:



იმ შემთხვევაში, როდესაც HTML დოკუმენტიდან სტილების გარე ცხრილზე ხდება მიმართვა და იმავე დოკუმენტშივე გვაქვს <style>...</style> ტეგების გამოყენებით CSS კოდი ჩასმული, მაშინ ყველაზე მაღალი პრიორიტეტი <style> ატრიბუტით განსაზღვრულ სტილებს აქვს, რაც ნიშნავს, რომ ის <head>... </head> ტეგის შიგნით „გადაფარავს“, გარე სტილების ფაილში განსაზღვრულ ან ბრაუზერის მიერ ნაგულისხმევი წესით განსაზღვრულ სტილებს.

თუ სტილების გარე ცხრილთან მიმართვა <head>... </head> ტეგის შიგნით მოთავსებული სტილების ცხრილის შემდეგაა განთავსებული, მაშინ გარე სტილების ცხრილი შიგა ცხრილების სტილს „გადაფარავს“.

მაგალითი 7.8:

```
<!DOCTYPE html>
<html>
<head>
<title>practcss1_7.html</title>
<link rel="stylesheet" type="text/css" href="style.css"/>
<style>
h3 { text-align:right;
font-size:20pt;
}
</style>
</head>
<body>
<h3>
My first CSS page
```

```
</h3>
</body>
</html>
```

style.css:

```
h3 {
color:red;
text-align:left;
font-size:8pt;
}
```

მიღებული შედეგი:



My first CSS page

სტილების გარე ცხრილში h3 სელექტორს აქვს სამი თვისება:

```
h3 {
color:red;
text-align:left;
font-size:8pt;
}
```

იგივე h3 სელექტორისთვის სტილების შიგა ცხრილში შემდეგი თვისებები გვაქვს:

```
h3 {  
  text-align:right;  
  font-size:20pt;  
}
```

ასეთ შემთხვევაში მოხდება ფერის თვისების მემკვიდრეობით გადაცემა სტილების გარე ცხრილიდან შიგა ცხრილისთვის, ტექსტის დაშორება კიდიდან და შრიფტის ზომა სტილების შიგა ცხრილიდან აიღება. ამის გამო, h3 სელექტორის თვისებებს ექნება სახე:

```
color:red;  
text-align:right;  
font-size:20pt;
```

HTML ფერები

ფერების მითითება HTML-ში წინასწარ განსაზღვრული ფერების სახელებით ან RGB, HEX, HSL, RGBA, HSLA მნიშვნელობებით ხორციელდება.

ფერთა სახელები

როგორც ზემოთ აღვნიშნეთ, HTML-ში ფერი შეიძლება მითითებული იყოს ფერის სახელით:



Red (წითელი)

Orange (ნარინჯისფერი)

DodgerBlue (ღია ლურჯი)

MediumSeaGreen (მწვანე ზღვისფერი)

Gray (ნაცრისფერი)

SlateBlue (მუქი ლურჯი)

Violet (იისფერი)

LightGray (ღია ნაცრისფერი)

HTML-ი მხარს უჭერს ფერთა 140 სტანდარტულ სახელს. ფერები შეიძლება მითითებული იყოს როგორც ფონად, ისე ტექსტისა და ჩარჩოს ფერად. მაგალითად:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<h1 style="color:Tomato;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

ფერთა მნიშვნელობები

HTML-ში ფერები შეიძლება მითითებული იყოს RGB, HEX, HSL, RGBA и HSLA მნიშვნელობების გამოყენებით:

ქვემოთ მოყვანილ მაგალითში მნიშვნელობებით მოცემულია ფერი "Tomato":

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

ქვემოთ მოყვანილ მაგალითში მნიშვნელობებით მოცემულია ფერი "Tomato" 50% გამჭვირვალობით:

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

ფერის RGB მნიშვნელობა

HTML-ში ფერი შეიძლება მითითებული იყოს, როგორც ფერის RGB მნიშვნელობა, შემდეგი ფორმატით:

rgb(წითელი, მწვანე, ლურჯი)

ფერის თითოეული პარამეტრი (წითელი, მწვანე და ლურჯი) განსაზღვრავს 0-დან 255-მდე ფერის ინტენსივობას.

მაგალითად, rgb (255, 0, 0) განსაზღვრავს წითელ ფერს, რადგანაც წითელი ფერისათვის ყველაზე დიდი მნიშვნელობაა მითითებული (255), ხოლო დანარჩენები 0-ის ტოლია.

შავი ფერის მისათითებლად ყველა ფერის პარამეტრი უნდა იყოს 0-ის ტოლი, მაგალითად, rgb (0, 0, 0).

თეთრი ფერის მისათითებლად ყველა პარამეტრი უნდა იყოს 255, მაგალითად, rgb (255, 255, 255).

ქვემოთ მოვიყვანოთ მაგალითები ფერის RGB პარამეტრის სხვადასხვა მნიშვნელობისათვის:



rgb(255, 99, 71)

აქ მითითებულია წითელი - 255, მწვანე - 99 და ლურჯი - 71.



`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(60, 179, 113)`

`rgb(238, 130, 238)`

`rgb(255, 165, 0)`

`rgb(106, 90, 205)`

ნაცრისფრის სხვადასხვა ტონალობის მიღება ხშირად სამივე პარამეტრის ერთნაირი მნიშვნელობებით განისაზღვრება, მაგალითად:

`rgb(0, 0, 0)`

`rgb(60, 60, 60)`

rgb(120, 120, 120)

rgb(180, 180, 180)

rgb(240, 240, 240)

rgb(255, 255, 255)

ფერის HEX მნიშვნელობა

HTML-ში ფერი შეიძლება თექვსმეტობითი კოდით იყოს მითითებული:

#rrggbb

სადაც rr (წითელი), gg (მწვანე) ი ხხ (ლურჯი) თექვსმეტობით მნიშვნელობაა 00-დან ff-მდე (0-255 მნიშვნელობის ანალოგიურია).

მაგალითად, # ff0000 მნიშვნელობა შეესაბამება წითელ ფერს, ვინაიდან წითლისათვის მითითებულია ყველაზე დიდი მნიშვნელობა (ff), ხოლო სხვებისთვის - ყველაზე დაბალი (00). მაგალითად:



#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

შესაბამისად, ნაცრისფრის სხვადასხვა ტონალობისათვის გვექნება:

#000000

#3c3c3c

#787878

#b4b4b4

#f0f0f0

#ffffff

ფერის HSL მნიშვნელობა

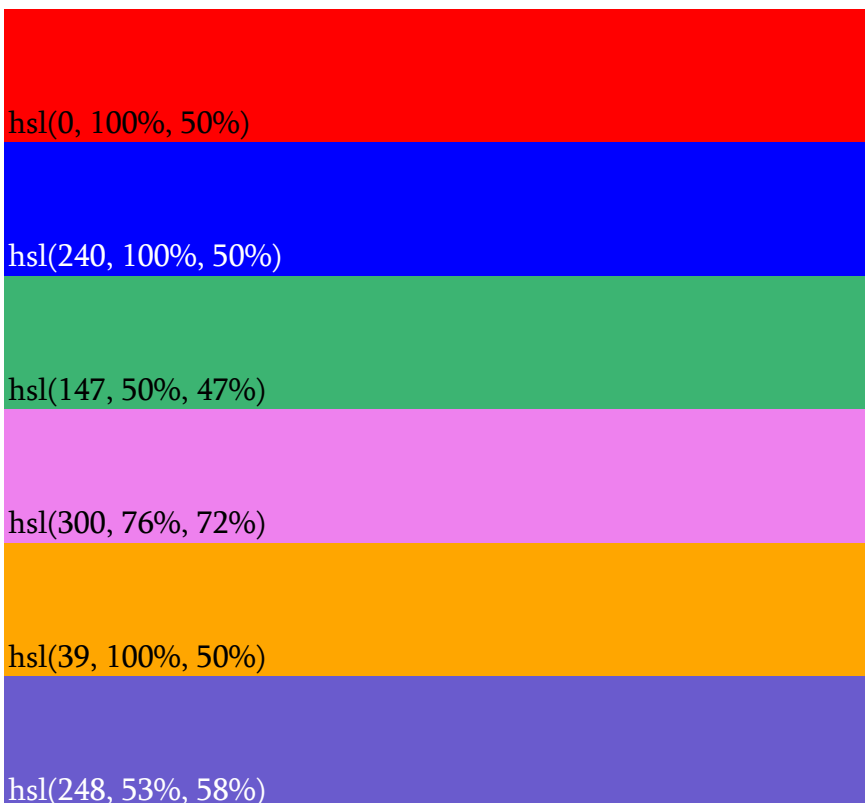
HTML-ში ფერი შეიძლება მითითებული იყოს ტონალობის, ინტენსივობის და სიკაშკაშის (HSL) ფორმით:

hsl(hue, saturation, lightness)

ტონალობა - ფერთა წრის გრადუსული ზომა. 0 - წითელი, 120 - მწვანე, 240 - ლურჯი.

ინტენსივობა - პროცენტული მაჩვენებელი, 0% უჩვენებს ნაცრისფრის ინტენსივობას ამ ფერში, ხოლო 100% არის მთლიანად მოცემული ფერი.

სიკაშკაშე - პროცენტული მაჩვენებელი, 0% - ჩამუქებული, 50% - არც კაშკაშა და არც ჩაბნელებული, 100% - თეთრი.



hsl(0, 100%, 50%)

hsl(240, 100%, 50%)

hsl(147, 50%, 47%)

hsl(300, 76%, 72%)

hsl(39, 100%, 50%)

hsl(248, 53%, 58%)

Saturation გაჯერებულობა

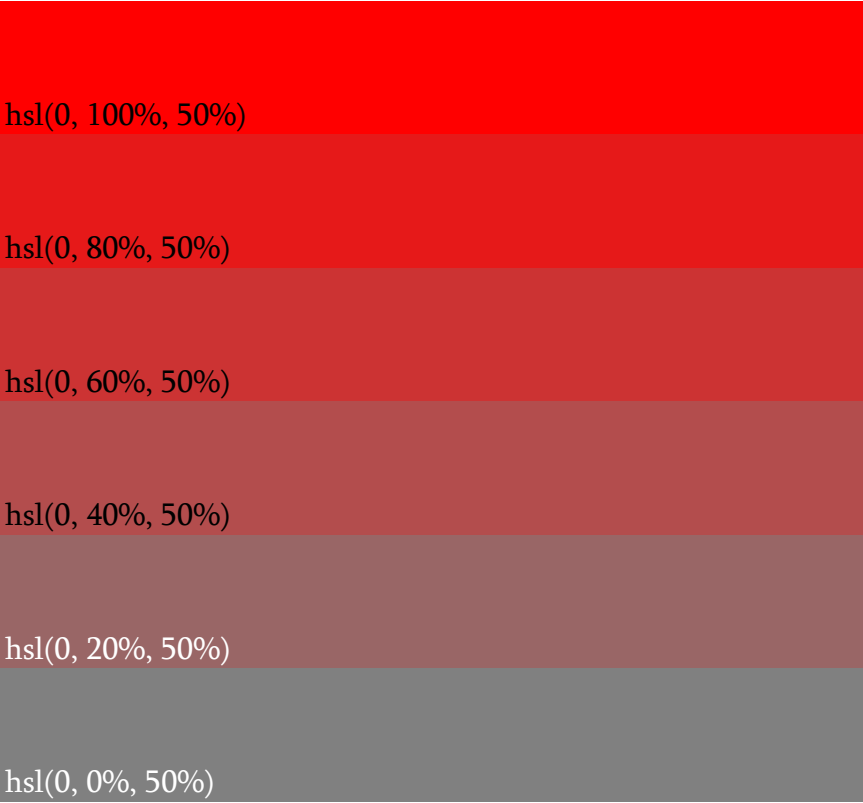
ფერის გაჯერებულობა შეიძლება აღიწეროს, როგორც ფერის ინტენსივობა.

100% სრულად მოცემული ფერია ნაცრისფრის ტონალობის გარეშე.

50% - 50% ნაცრისფერი, მაგრამ მასში კიდევ ჩანს მოცემული ფერი.

0% - სრულად ნაცრისფერი, მასში არ ჩანს საწყისი ფერი.

მაგალითი:



hsl(0, 100%, 50%)

hsl(0, 80%, 50%)

hsl(0, 60%, 50%)

hsl(0, 40%, 50%)

hsl(0, 20%, 50%)

hsl(0, 0%, 50%)

Lightness სიკაშკაშე

ფერის სიკაშკაშე შეიძლება აღწერილი იყოს, როგორც სინათლის ის რაოდენობა, რომელიც ამ ფერს გინდათ მისცეთ. 0% აღნიშნავს, რომ სინათლე საერთოდ არა აქვს (შავია), 50% აღნიშნავს რომ სინათლე 50% აქვს (არც ჩაბნელებულია და არც ძალიან განათებული), ხოლო 100% სინათლე აღნიშნავს სრულ სიკაშკაშეს (თეთრი). მაგალითი:

hsl(0, 100%, 0%)

hsl(0, 100%, 25%)

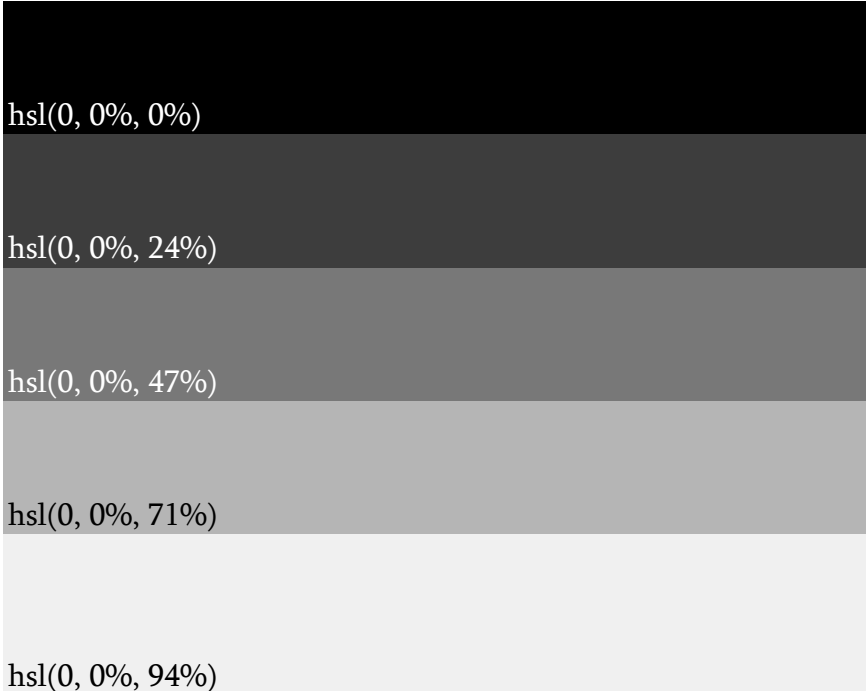
hsl(0, 100%, 50%)

hsl(0, 100%, 75%)

hsl(0, 100%, 90%)

hsl(0, 100%, 100%)

ნაცრისფერი ტონალობა განისაზღვრება ფერის ნულოვანი ტონალობით და სიკაშკაშის 0%-დან 100%-მდე რეგულირებით, რათა მიღებულ იქნეს მუქი და ნათელი სიკაშკაშის მქონე ნაცრისფერი:



`hsl(0, 0%, 0%)`

`hsl(0, 0%, 24%)`

`hsl(0, 0%, 47%)`

`hsl(0, 0%, 71%)`

`hsl(0, 0%, 94%)`

`hsl(0, 0%, 100%)`

ფერის RGBA მნიშვნელობა

ფერის RGBA მნიშვნელობა ფერთა RGB მნიშვნელობის ალფა-არხით გაფართოებაა, რომელიც ფერის გაუმჭვირობას განსაზღვრავს.

ფერის RGBA მნიშვნელობის მითითება შემდეგნაირად ხდება:

`rgba(red, green, blue, alpha)`

ალფა პარამეტრი რიცხვითი მნიშვნელობაა 0,0-დან (სრულად გამჭვირვალე) 1,0-მდე სრულად გაუმჭვირი):

`rgba(255, 99, 71, 0)`

`rgba(255, 99, 71, 0.2)`

`rgba(255, 99, 71, 0.4)`

`rgba(255, 99, 71, 1)`

ფერის HSLA მნიშვნელობა

ფერის HSLA მნიშვნელობა ფერთა HSL მნიშვნელობის ალფა-არხით გაფართოებაა, რომელიც ფერის გაუმჭვირობას განსაზღვრავს.

ფერის HSLA მნიშვნელობის მითითება შემდეგნაირად ხდება:

hsla(hue, saturation, lightness, alpha)

აღფა პარამეტრი რიცხვით მნიშვნელობაა 0,0-დან (სრულად გამჭვირვალე) 1,0-მდე სრულად გაუმჭვირი):

hsla(9, 100%, 64%, 0)

hsla(9, 100%, 64%, 0.2)

hsla(9, 100%, 64%, 0.4)

hsla(9, 100%, 64%, 0.8)

hsla(9, 100%, 64%, 1)

თავი 8

ფონური გამოსახულება, ტექსტის დაფორმატება და შრიფტი CSS3-ში

- ფონური გამოსახულება CSS-ში;
- ტექსტის დაფორმატება CSS-ში;
- შრიფტები.

ფონური გამოსახულება CSS-ში

CSS-ში ფონური გამოსახულების სხვადასხვა ეფექტის მისაღებად background-ის სხვადასხვა თვისება გამოიყენება:

background-color თვისება ელემენტის ფონის ფერს აღწერს.

background-image თვისება ფონური გამოსახულების ჩასასმელად გამოიყენება.

background-repeat თვისება ფონური გამოსახულების გამეორებადობას მართავს და მას 4 მნიშვნელობის მიღება შეუძლია:

ა) background-repeat:repeat-x (სურათი ჰორიზონტალზე მეორდება);

ბ)background-repeat:repeat-y (სურათი ვერტიკალზე მეორდება);

გ)background-repeat:repeat (სურათი მეორდება ჰორიზონტალზეც და ვერტიკალზეც);

დ) background-repeat:no-repeat (სურათი არ მეორდება).

background-attachment თვისება განსაზღვრავს, რომ ფონური სურათი დაფიქსირებულია თუ გვერდის შემცველობასთან ერთად მოძრაობს.

background-attachment-ს ორი მნიშვნელობა აქვს:

```
background-attachment: fixed;
```

```
background-attachment: scroll;
```

მნიშვნელობა scroll - გამოსახულება მოძრაობს გვერდთან ერთად;

მნიშვნელობა fixed - გამოსახულება ბლოკირებულია.

background-position თვისება საშუალებას გვაძლევს ფონური სურათი ეკრანის ნებისმიერ ადგილას განვათავსოთ. ფონური გამოსახულების მდებარეობის მისათითებლად შესაძლებელია კოორდინატების გამოყენება. მაგალითად, "100px 200px" მნიშვნელობა ფონურ გამოსახულებას ბრაუზერის ფანჯრის მარცხენა კიდიდან 100 და ზემოდან 200 პიქსელის დაშორებით განათავსებს. მდებარეობის მითითება შეიძლება პროცენტებშიც ან გამოიყენება სიტყვები: top, bottom, center, left და right

მაგალითი 8.1: background-color თვისების გამოყენება დოკუმენტის ფონის ფერისა და სათაურის ფონის ფერის მისათითებლად.

```
<!DOCTYPE html>
<html>
<head>
<title>practcss 8_1.html</title>
<style>
Body {
background-color:#b0c4de;
}
h1 {
color:red; background-color:yellow;
}
</style>
</head>
<body>
<h1 align="center">What is CSS!</h1>
```

<p>Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.</p>

<p>

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[2] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers.

</p>

<p>

CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold," leaving pure semantic HTML markup that

asserts "this text is a level 1 heading" without formatting code such as a `<bold>` tag indicating how such text should be displayed.

```
</p>  
</body>  
</html>
```

მიღებული შედეგი მოცემულია ქვემოთ:

What is CSS!

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[2] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers.

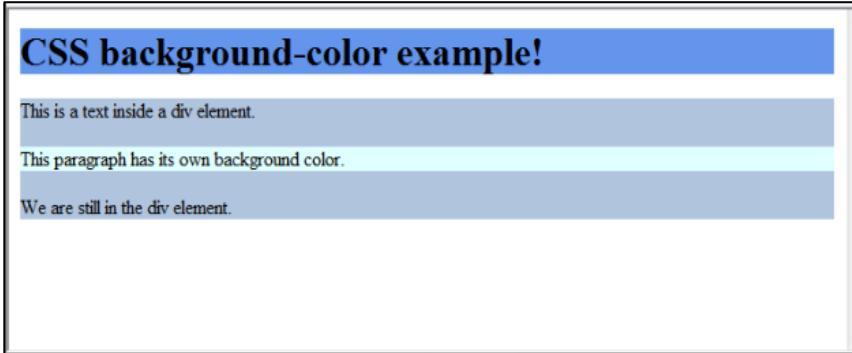
CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold," leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a tag indicating how such text should be displayed.

`<body>`-ის შიგნით თავსდება HTML დოკუმენტის მთელი შემცველობა. ამიტომ, მთელი გვერდის ფონის შესაცვლელად ელემენტ `<body>`-ის მიმართ უნდა გამოვიყენოთ `background-color` თვისება. ამ თვისების გამოყენება სხვა ელემენტების, მათ შორის სათაურისა და ტექსტის მიმართაცაა შესაძლებელი. ყურადღება მიაქციეთ, რომ `<h1>` ტეგისთვის ორი თვისების - სათაურისა და სათაურის ფონის ფერის მითითება მოხდა.

მაგალითი 8.2: `h1`, `p` და `div` ელემენტების მიმართ `background-color` თვისების გამოყენება.

```
<!DOCTYPE html>
<html>
<head>
<title>practcss 8_2.html</title>
<style>
h1{
background-color:#6495ed;
}
P {
background-color:#e0ffff;
}
div{
background-color:#b0c4de;
}
</style>
</head>
<body>
<h1>CSS background-color example!</h1>
<div>This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element. </div>
</body>
</html>
```

მივიღებთ შედეგს:



ამ მაგალითში h1, p და div ელემენტებისთვის ფონის ფერია მითითებული. div ბლოკური ელემენტის შიგნით მოთავსებული p ელემენტისთვის ტექსტის ფერი განისაზღვრა მისთვის დანიშნული სტილის შესაბამისად, ხოლო იმ ტექსტებისთვის, რომლებიც div ელემენტს მიეკუთვნება, განისაზღვრა იმ სტილის შესაბამისად, რომელიც div-ს დაენიშნა.

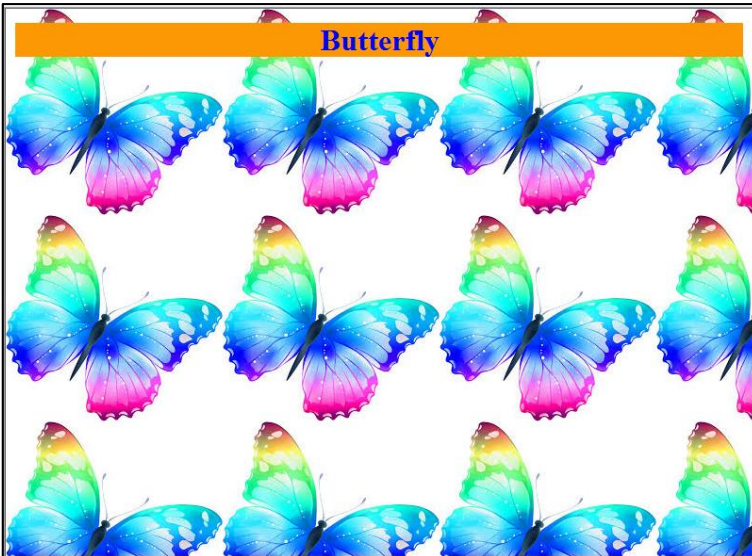
background-image თვისება ფონური გამოსახულების ჩასასმელად გამოიყენება.

მაგალითი 8.3: ფონური გამოსახულების ჩასმა.

```
<!DOCTYPE html>
<html>
<head>
<title>practcss 8_3.html</title>
<style>
body {
background-image: url("butterfly.jpg");
}
h1 {
color: blue;
```

```
background-color: #fc9804;
}
</style>
</head>
<body>
<h1 align="center" >
Butterfly
</h1>
</body>
</html>
```

მიღებული შედეგი:



სურათის ფონად ჩასასმელად <body> ტეგში გამოიყენება background-image თვისება და სურათის ადგილმდებარეობა მიეთითება:

```
url("butterfly.jpg").
```

ეს ნიშნავს, რომ ფაილი იმავე საქალაქდებშია, რომელშიც არის ჩვენი HTML დოკუმენტი. რა თქმა უნდა, შესაძლებელია სხვა საქალაქდებში ან ინტერნეტში არსებული სურათებისადმი მიმართვაც, ასეთ შემთხვევაში მისი სრული მისამართი უნდა მიეთითოს:

```
url("../images/butterfly.jpg")
```

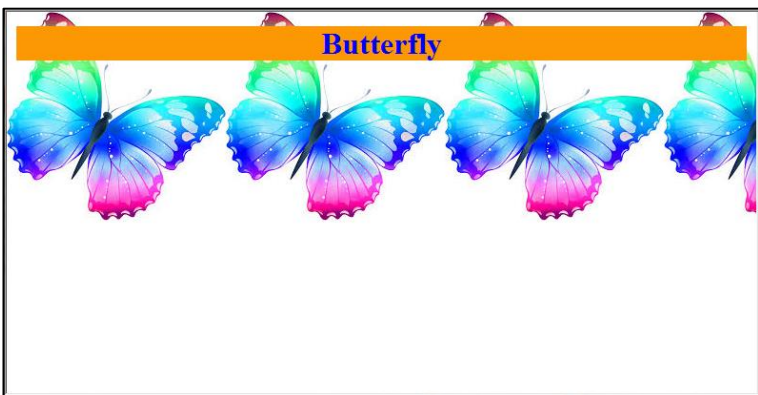
ან

```
url("http://www.teabesotemo.ge/butterfly.jpg")
```

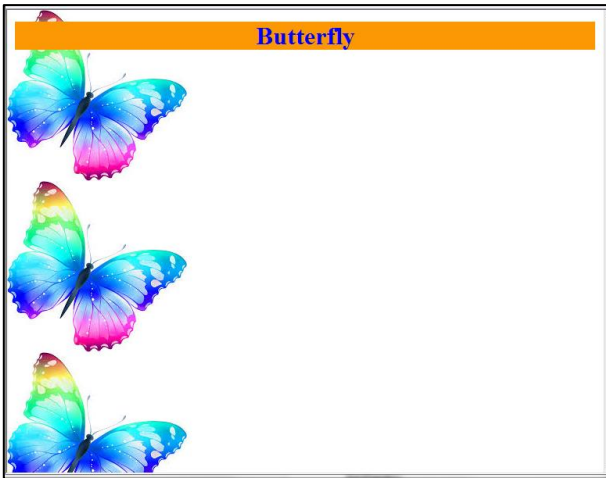
url("images/butterfly.jpg") მისამართის მითითება ნიშნავს, რომ images საქალაქდე იმავე საქალაქდებშია მოთავსებული, რომელშიც HTML დოკუმენტი.

background-repeat თვისება ფონური გამოსახულების გამეორებას მართავს. ამ თვისებას ოთხი მნიშვნელობის მიღება შეუძლია:

ა) background-repeat:repeat-x (ფონური გამოსახულება მეორდება ჰორიზონტალზე);



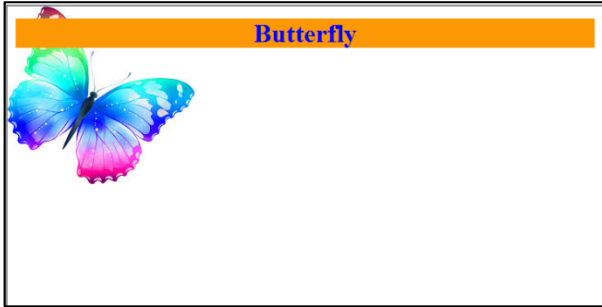
ბ) background-repeat:repeat-y (ფონური გამოსახულება მეორდება ვერტიკალზე);



გ) background-repeat:repeat (ფონური გამოსახულება მეორდება ჰორიზონტალზეც და ვერტიკალზეც).

ეს შემთხვევა უკვე განვიხილეთ ზემოთ განხილულ მაგალითში, როდესაც ფონური გამოსახულება ვერტიკალზეც და ჰორიზონტალზეც repeat თვისების მითითების გარეშე გამეორდა, რადგანაც სისტემაში ჩუმათობის პრინციპის თანახმად მიღებულია, რომ გამოსახულება გამეორდეს და მთელი ეკრანი დაფაროს ჰორიზონტალურადაც და ვერტიკალურადაც.

დ) background-repeat:no-repeat (ფონური გამოსახულება არ მეორდება).



background-attachment თვისება განსაზღვრავს, ფონური სურათი დაფიქსირებულია (ბლოკირებულია) თუ გვერდის შემცვლობასთან ერთად მოძრაობს.

ცხრილში background-attachment-ის ორი მნიშვნელობაა ნაჩვენები.

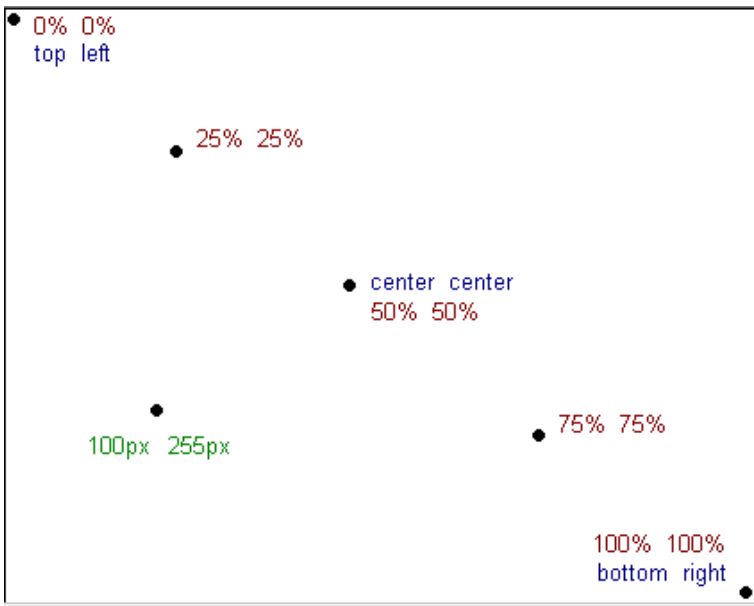
მნიშვნელობა	აღწერა
Background-attachment: scroll	გამოსახულება მოძრაობს გვერდთან ერთად
Background-attachment: fixed	გამოსახულება ბლოკირებულია

თუ თქვენ style.css ფაილს ქვემოთ მოცემული სახე ექნება, მაშინ გამოსახულება დაფიქსირებული იქნება:

```
body {
background-color: #ffcc66;
background-image: url("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
}
h1
```

```
{  
  color: #990000; background-color: #FC9804;  
}
```

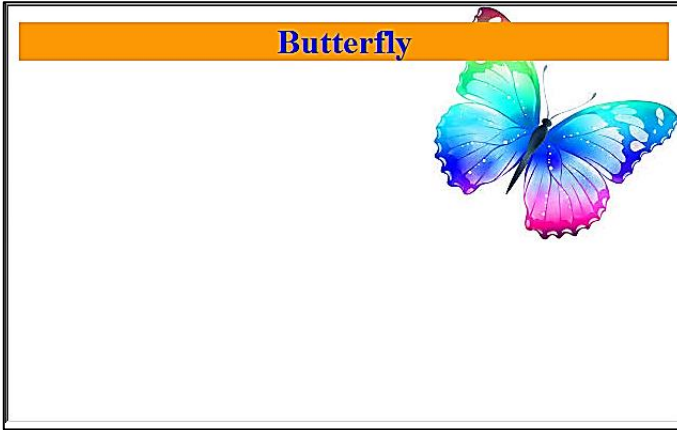
background-position თვისება საშუალებას გვაძლევს ფონური სურათი ეკრანის ნებისმიერ ადგილას განვათავსოთ. ჩუმათობის პრინციპით, ფონური გამოსახულება ეკრანის ზედა მარცხენა კუთხეში განთავსდება. ფონური გამოსახულების სასურველი ადგილმდებარეობის მისათითებლად რამდენიმე ხერხი არსებობს. ყველა ხერხი კოორდინატების მითითებას ეფუძნება. მაგალითად, "100px 200px" ფონურ გამოსახულებას ბრაუზერის ფანჯრის მარცხენა კიდიდან 100 და ზემოდან 200 პიქსელის დაშორებით განათავსებს. კოორდინატების მითითება ასევე პროცენტებშია შესაძლებელი ან გამოიყენება სიტყვები: top, bottom, center, left და right.



მაგალითი 8.4: ეკრანზე გამოსახულების მდებარეობის მითითება background-position თვისების საშუალებით.

```
<!DOCTYPE html>
<html>
<head>
<title>practcss 8_4.html</title>
<style>
body { background-image: url("butterfly.jpg");
background-repeat: no-repeat;
background-attachment:fixed;
background-position: right top;
}
h1 {
color: blue;
background-color: #fc9804;
}
</style>
</head>
<body>
<h1 align="center" >Butterfly</h1>
</body>
</html>
```

მიღებული შედეგი :



როგორც სურათიდან ჩანს, გამოსახულება მარჯვენა ზედა კუთხეშია მოთავსებული, რასაც ბრძანება `background-position: right top` უზრუნველყოფს. გამოსახულება არ მეორდება, რადგან მოცემული სურათისთვის `no-repeat` თვისება გვაქვს მინიჭებული.

მეტი მოხერხებულობისათვის უფრო მოკლე ჩანაწერის გაკეთება შესაძლებელი:

```
body {  
  background: url("butterfly.jpg")no-repeat fixed right top;  
}
```

თვისებების შემოკლებული ჩანაწერის გამოყენებისას იგულისხმება, რომ ისინი ქვემოთ მოცემული თანამიმდევრობითაა ჩაწერილი:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`

- background-position.

Background-სთვის ზემოთ აღწერილ შემოკლებულ ჩანაწერში background-color თვისებაა გამოტოვებული, ამას მნიშვნელობა არა აქვს, მთავარია მათი ჩაწერის დადგენილი თანამიმდევრობა იყოს დაცული.

ტექსტის დაფორმატება CSS-ში

color თვისება ტექსტის ფერის მისათითებლად გამოიყენება. ფერი CSS-ში უმეტესწილად განისაზღვრება:

- HEX (თექვსმეტობითი კოდი) მნიშვნელობით (მაგ., „#ff0000”);
- RGB მნიშვნელობით (მაგ., “rgb (255,0,0)”);
- ფერის დასახელებით (მაგ., “red”).

მაგალითი 8.5:

```
<!DOCTYPE html>
<html>
<head>
<title> head </title>
<style>
body
{
  color:green;
}
h1{
  color:red;
}
.ex
{
```

```
color:blue;
}
</style>
</head>
<body>
<h1> head 1 – საქართველო </h1>
<h2> head 2 – საქართველო </h2>
<h3> head 3 – საქართველო </h3>
<h4> head 4 – საქართველო </h4>
<h5> head 5 – საქართველო </h5>
<h6 class="ex"> head 6 – საქართველო </h6>
</body>
</html>
```

head 1 – საქართველო

head 2 – საქართველო

head 3 – საქართველო

head 4 – საქართველო

head 5 – საქართველო

head 6 – საქართველო

text-align თვისება ტექსტის გვერდის კიდეების მიმართ გასასწორებლად გამოიყენება.

მაგალითი 8.6:

```
<!DOCTYPE html>
<html>
<head>
<title> My students </title>
</head>
<style>
h1 {text-align:center;}
h2{text-align:right;}
</style>
<body>
<h1 align = left>ჩემი სტუდენტები</h1>
<br>
<h2 align = left> გოგონები </h2>
მარი <br>
თათია <br>
სალომე <br>
<hr width = 50% size = 6 align = left color = red>
<h2 align = left> ბიჭები </h2>
შოთიკო <br>
ლაშა <br>
გიორგი<br>
<hr width = 50% size = 6 align = left color =navy>
<body>
</html>
```

ჩემი სტუდენტები

გოგონები

მარი
თათია
სალომე

ბიჭები

შოთიკო
ლაშა
გიორგი

text-decoration თვისება.

text-decoration: none; - ბმულისთვის ხაზგასმის მოცილებას,

text-decoration: overline; - ტექსტის ზემოთ ხაზის გავლებას,

text-decoration: line-through; - ტექსტის გადახაზვას,

text-decoration: underline; - ტექსტისთვის ხაზგასმას უზრუნველყოფს.

მაგალითი 8.7:

```
<!DOCTYPE html>
<html>
<head>
<title>decoration</title>
<style>
h1 {text-decoration:overline;}
h2 {text-decoration: line-through;}
```

```
h3 {text-decoration:underline;}
</style>
</head>
<body>
<h1>სასურველი ტექსტი1</h1>
<h2>სასურველი ტექსტი2</h2>
<h3>სასურველი ტექსტი3</h3>
</body>
</html>
```

სასურველი ტექსტი1

~~სასურველი ტექსტი2~~

სასურველი ტექსტი3

text-transform თვისება.

capitalize საშუალებას იძლევა მთავრული ასოებით დაიწეროს სიტყვის დასაწყისი.

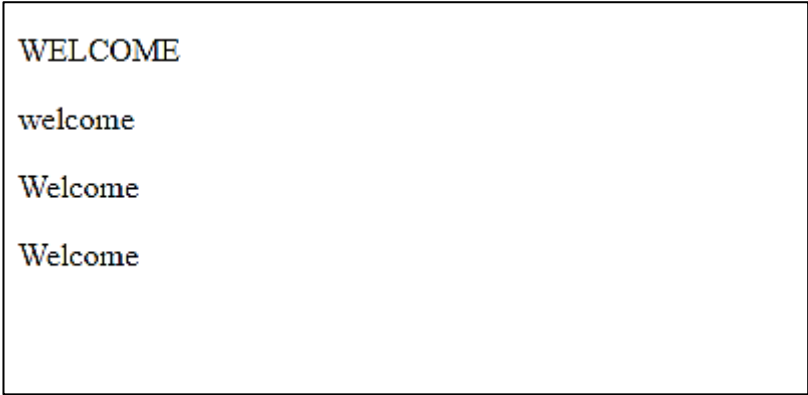
uppercase ყველა სიმბოლოს ზედა რეგისტრში კონვერტირებას უზრუნველყოფს.

lowercase ყველა სიმბოლოს ქვედა რეგისტრში კონვერტირებას უზრუნველყოფს.

none მნიშვნელობა გულისხმობს, რომ ტრანსფორმაცია არ ხდება. ბრაუზერში ტექსტი ისევე აისახება, როგორც HTML კოდში იყო მოცემული.

მაგალითი 8.8:

```
<!DOCTYPE html>
<html>
<head>
<title>transform</title>
<style>
p.uppercase {text-transform:uppercase;}
p.lowercase {text-transform:lowercase;}
p.capitalize {text-transform:capitalize;}
p.none {text-transform:none}
</style>
</head>
<body>
<p class="uppercase">Welcome</p>
<p class="lowercase">WELCOME</p>
<p class="capitalize">welcome</p>
<p class="none"> Welcome</p>
</body>
</html>
```



text-indent თვისება იმისათვის გამოიყენება, რომ პირველი სტრიქონისთვის სააბზაცო შეწევა განსაზღვროს.

text-indent თვისების მნიშვნელობა განისაზღვრება პიქსელებში, პროცენტებში ან სანტიმეტრებში. მაგალითად:

```
text-indent:50px  
text-indent:30%  
text-indent:2cm
```

მაგალითი 8.9:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="content-type" content="text/html;  
charset=utf-8" />  
<title>აბზაცი</title></head>  
<style>  
h1 {text-align:center; color:red;}  
p {text-indent:60px;}  
</style>
```

<body bgcolor=#eaaaac>

<h1> საქართველო </h1>

<p>საქართველო - ევროპისა და აზიის გასაყარზე, კერძოდ, კავკასიაში მდებარეობს. დასავლეთიდან მას ესაზღვრება შავი ზღვა, ჩრდილოეთით - რუსეთის ფედერაცია, სამხრეთ-აღმოსავლეთით - აზერბაიჯანი, სამხრეთით - სომხეთი და თურქეთი. საქართველო იმ უძველეს სატრანსპორტო გზაჯვარედინზე მდებარეობს, რომელიც აკავშირებდა და აკავშირებს ჩრდილოეთისა და სამხრეთის, დასავლეთისა და აღმოსავლეთის ქვეყნებს. სწორედ მასზე გადიოდა ევროპა - აზიის დამაკავშირებელი სატრანსპორტო მაგისტრალი - ძველი აბრემუმის გზა. საქართველო შავი ზღვით უკავშირდება შავიზღვისპირეთის ქვეყნებს, ხოლო ბოსფორისა და დარდანელის სრუტეებით - ხმელთაშუა ზღვის აუზის ქვეყნებს; გიბრალტარის სრუტით შესაძლებელია კავშირი მთელს მსოფლიოსთან, მდინარე დუნაის მეშვეობით კი - აღმოსავლეთ და ცენტრალური ევროპის ქვეყნებთან.

</body>

</html>

საქართველო

საქართველო - ევროპისა და აზიის გასაყარზე, კერძოდ, კავკასიაში მდებარეობს. დასავლეთიდან მას ესაზღვრება შავი ზღვა, ჩრდილოეთით - რუსეთის ფედერაცია, სამხრეთ-აღმოსავლეთით - აზერბაიჯანი, სამხრეთით - სომხეთი და თურქეთი. საქართველო იმ უძველეს სატრანსპორტო გზაჯვარედინზე მდებარეობს, რომელიც აკავშირებდა და აკავშირებს ჩრდილოეთისა და სამხრეთის, დასავლეთისა და აღმოსავლეთის ქვეყნებს. სწორედ მასზე გადიოდა ევროპა - აზიის დამაკავშირებელი სატრანსპორტო მაგისტრალი - ძველი აბრემუმის გზა. საქართველო შავი ზღვით უკავშირდება შავიზღვისპირეთის ქვეყნებს, ხოლო ბოსფორისა და დარდანელის სრუტეებით - ხმელთაშუა ზღვის აუზის ქვეყნებს; გიბრალტარის სრუტით შესაძლებელია კავშირი მთელს მსოფლიოსთან, მდინარე დუნაის მეშვეობით კი - აღმოსავლეთ და ცენტრალური ევროპის ქვეყნებთან.

letter-spacing თვისება სიმბოლოებს შორის ინტერვალის დაყენებას უზრუნველყოფს.

word-spacing თვისება სიტყვებს შორის ინტერვალის მითითების საშუალებას იძლევა.

სიმბოლოებსა და სიტყვებს შორის ინტერვალის მითითება პიქსელებში ხდება.

მაგალითი 8.10:

```
<!DOCTYPE html>
<html>
<head>
<title>მარკირებული სია</title>
<style>
li {letter-spacing:6px;}
h2 {word-spacing:30px};
</style>
</head>
<body bgcolor=snow>
<h2>სამაგიდო კომპიუტერები, თავის მხრივ,
შემდეგ ჯგუფებად იყოფა:</h2>
<br>
<ul>
<li>უნივერსალური ტიპის კომპიუტერები
<li>სპეციალური დანიშნულების კომპიუტერები
<li>სამაგიდო მინიკომპიუტერები
<li>სერვერები
</ul>
</body>
```

</html>

სამაგიდო კომპიუტერები, თავის მხრივ, შემდეგ ჯგუფებად იყოფა:

- უნივერსალური ტიპის კომპიუტერები
- სპეციალური დანიშნულების კომპიუტერები
- სამაგიდო მინიკომპიუტერები
- სერვერები

line-height თვისება აბზაცში სტრიქონებს შორის მანძილის დაყენების საშუალებას იძლევა.

სტრიქონებს შორის დაშორება შეიძლება როგორც პროცენტებში, ისე პიქსელებში მიეთითოს, შესაძლებელია ეს მნიშვნელობა გამოხატული იყოს რიცხვებშიც:

```
line-height:1;
line-height:4;
line-height: 25px;
```

მაგალითი 8.11:

```
<!DOCTYPE html>
<html >
<head>
<title> გრაფიკული ფონი</title>
<style>
h1 {letter-spacing:7px;}
p {line-height:30px;}
</style>
```

</head>

<body background=rose.jpg bgproperties=fixed text=yellow topmargin=20 bottommargin=20 leftmargin=20 rightmargin=20 >

<h1>ვარდი და ია</h1>

<h2 align = "center"> ვარდი </h2>

<p align = "justify"> ვარდი (ლათ. Rosa) - მცენარის გვარი ვარდისებრთა ოჯახისა. ცნობილია დაახლოებით 400 ველური სახეობა (ასკილი), რომლებიც გავრცელებულია ჩრდილოეთ ნახევარსფეროში. საქართველოში 25-მდე სახეობაა. ვარდი მარადმწვანე ან ფოთოლმცვივანი ბუჩქია, რომლის სიმაღლე 15 სმ-დან 2 მ-მდე აღწევს. ზოგიერთ სახეობას გრძელი (10-12 მ-მდე) ხვიარა ტოტები აქვს, ღერო დაფარულია ეკლებით. ფოთოლი კენჭფრთისებრია, ღეროზე მორიგეობით განლაგებული. ყვავილი მარტივია ან ბუთხუზა (2-15 სმ დიამეტრის), არომატული ან უსუნო; ფერი - ვარდისფერი, წითელი, თეთრი, ყვითლი, ნარინჯისფერი ან იასამნისფერი. შეკრებილია ქოლგისებრ-საგველა ყვავილედებად ან ერთეულია. ნაყოფი ერთთესლიანი (ზოგჯერ რამდენიმეთესლიანი) კაკლუქაა, რომელიც ხორცივან ცრუნაყოფშია.<p>

<h2 align = "center">ია </h2>

<p align="justify"> ია (ლათ. Viola) - მცენარეთა გვარი ისებრთა ოჯახისა. მრავალწლოვანი, იშვიათად ერთწლოვანი ბალახები, ზოგჯერ ნახევრად ბუჩქები. აქვს მორიგეობით გაწყობილი ან ფესვთან როზეტად შეკრებილი ფოთლები, იისფერი, ყვითელი და თეთრი მარტოული ყვავილები. გვირგვინის ქვედა ფურცელი წარზიდულია დეზად, რომელშიც სანექტრეა მოქცეული. ზოგიერთს უვითარდება ვავილები.

ნაყოფი სამსაგდულიანი კოლოფია. ზოგ სახეობას ახასიათებს ამფიკარპია. ცნობილია 500-მდე სახეობა. უმრავლესობა გავრცელებულია ჩრდილოეთ ნახევარსფეროს ზომიერ სარტყელში. საქართველოში გავრცელებულია 26 სახეობა, მათგან 3 კავკასიისა და 2 საქართველოს ენდემია.

საქართველოში ია იზრდება ტყეებსა და ბუჩქნარებში, სუბალპურ და ალპურ მდელოებზე, კლდეების ნაპრალებში, ჩამონაზვავებზე, ზოგიერთი სახეობა (Viola alba) სარეველაც არის, ასევე დეკორატიულიც (იაჟუჟუნა - Viola tricolor და სხვა), ზოგს (სურნელოვანი ია - Viola odorata) პარფიუმერიაში იყენებენ.

</body>
</html>



შრიფტები

font-family თვისება მოცემული ელემენტის შრიფტების პრიორიტეტულ სიაზე მიუთითებს. თუ სიაში მოცემული პირველი შრიფტი იმ კომპიუტერზე, რომლიდანაც მოცემულ საიტზე ხდება წვდომა, არ არის დაყენებული, მაშინ მომდევნო შრიფტზე გადასვლა მოხდება.

შრიფტების კატეგორიისათვის სახელების ორი ტიპი გამოიყენება: font-family და generic family.

Font-family (ხშირად უბრალოდ "შრიფტი" ეწოდება). ეს არის, მაგალითად, "Arial", "Times New Roman" ან "Tahoma".

Generic family შეიძლება აღიწეროს, როგორც ფონტების ჯგუფი, რომელსაც დამახასიათებელი საერთო შტრიხები აქვს. მაგალითად, sans-serif, serif, monospace.

სიის ბოლოს რეკომენდებულია generic family ტიპის მითითება. ასეთ შემთხვევაში, თუ ჩამოთვლილი შრიფტებიდან კომპიუტერში არც ერთი არ არის, ვებგვერდზე იმავე ოჯახის რომელიმე შრიფტი აისახება.

მაგალითი 8.12:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  font-family: "Times New Roman", Times, serif;
}
p.b {
  font-family: Arial, Helvetica, sans-serif;
```

```

}
</style>
</head>
<body>
<h1>CSS შრიფტები</h1>
<p class="a">This is a paragraph, shown in the Times New
Roman font.</p>
<p class="b">This is a paragraph, shown in the Arial font.</p>
</body>
</html>

```

CSS შრიფტები

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

font-style თვისება უმეტესწილად შრიფტის დახრილი (italic) თვისების მისათითებლად გამოიყენება, შრიფტის სტილს შეუძლია ქვემოთ მოცემული სამი მნიშვნელობიდან ერთ-ერთი მიიღოს: normal (ჩვეულებრივი ტექსტი), italic (დახრილი ტექსტი), oblique (ესეც დახრილი, ძალიან ჰგავს italic-ს, მაგრამ გამოიყენება იშვიათად).

მაგალითი 8.13:

```

<!DOCTYPE html>
<html>
<head>

```

```
<style>
p.a {
  font-style: normal;
}
p.b {
  font-style: italic;
}
p.c {
  font-style: oblique;
}
</style>
</head>
<body>
<p class="a">This is a paragraph in normal style.</p>
<p class="b">This is a paragraph in italic style.</p>
<p class="c">This is a paragraph in oblique style.</p>
</body>
</html>
```

This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

font-variant თვისება normal და small-caps ვარიანტებიდან ერთ-ერთის ამოსარჩევად გამოიყენება. small-caps შრიფტი

დაბალი რეგისტრის სიმბოლოების ნაცვლად პატარა მთავრულ ასოებს (upper case) იყენებს.

მაგალითი 8.14:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-variant: normal;
}
p.small {
  font-variant: small-caps;
}
</style>
</head>
<body>
<p class="normal">We are students.</p>
<p class="small">We are students.</p>
</body>
</html>
```

We are students.

WE ARE STUDENTS.

font-weight თვისება შრიფტის სისქეს აღწერს. შრიფტი შეიძლება იყოს normal ან bold. შრიფტის სისქის აღსაწერად შესაძლებელია რიცხვითი მნიშვნელობის მითითებაც.

```
p {font-family: arial, verdana, sans-serif; font-weight: bold;}
```

font-size თვისების საშუალებით შრიფტის ზომა დგება. შრიფტის ზომის აღსაწერად სხვადასხვა ერთეული გამოიყენება. მაგალითად:

```
p {font-size: 30px;} ბაზური ზომები:16px;  
p {font-size: 12pt;} ბაზური ზომები:12pt;  
p {font-size: 120%;} ბაზური ზომები:100%;  
p {font-size: 1em;} ბაზური ზომები:1em;
```

მაგალითი 8.15:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.a {  
    font-style:normal;font-weight:bold;  
}  
p.b {  
    font-style:italic; font-size:120%;  
}  
p.c {  
    font-style:oblique;  
}  
</style>  
</head>
```

```
<body>
<p class="a">This is a paragraph in normal style.</p>
<p class="b">This is a paragraph in italic style.</p>
<p class="c">This is a paragraph in oblique style.</p>
</body>
</html>
```

This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

CSS-ში font-ის შემოკლებული ჩაწერაც არის შესაძლებელი.

```
p
{
font-style: italic;
font-weight: bold;
font-size: 30px;
font-family: arial, sans-serif;
}
```

ზემოთ მოცემული ჩანაწერის ნაცვლად შეიძლება დავწეროთ:

```
p { font: italic bold 30px arial, sans-serif; }.
```

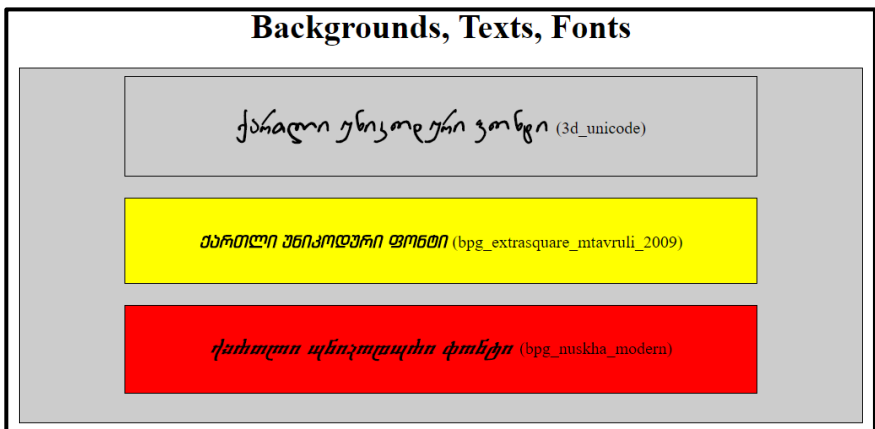
დავალება. გამოიყენეთ ფონის და ფონტის დაფორმატების თვისებები (გამოიყენეთ @font-face) და ააგეთ (სურათი 8.1) მოყვანილი ვებკომპონენტები. ვებგვერდი შემდეგნაირად გააფორმეთ:

✓ კონტეინერის ფონი ვერცხლისფერია, საზღვრები 2px შავი ფერის. განთავსებულია ბრაუზერის ცენტრში, ზომები 800x100, მიუთითეთ ცვალებადი სიმაღლე;

✓ პირველი div ტეგი - ფონი გამჭვირვალე. ფონტი - მუქი დახრილი; ზომა - 26 px, დაყენებულია 3d_unicode.ttf ფონტი (მოიძიეთ ინტერნეტში); ზომა - 600x50.

✓ მეორე div ტეგი - ფონი ყვითელი. ფონტი - მუქი დახრილი; ზომა - 16px, დაყენებულია bpg_extrasquare_mtavruli_2009.ttf ფონტი (მოიძიეთ ინტერნეტში); ზომა - 600x50.

✓ მესამე div ტეგი - ფონი წითელი. ფონტი - მუქი დახრილი ზომა - 16px, დაყენებულია bpg_nuskha_modern.ttf ფონტი (მოიძიეთ ინტერნეტში); ზომა - 600x50.



თავი 9

ჩარჩო CSS3-ში

- ჩარჩოს ტიპები CSS-ში;
- ჩარჩოს სისქე და ფერი;
- ჩარჩოს „მომრგვალებული კუთხეების“ შექმნა;
- საზღვრის ველების განსაზღვრა;
- ჩარჩოსა და ელემენტს შორის დაშორების განსაზღვრა.

ჩარჩოს ტიპები (border-style)

ჩარჩოს ტიპის მისათითებლად border-style თვისება გამოიყენება. მასში შეიძლება შემდეგი მნიშვნელობები იყოს მითითებული:

- dotted - განსაზღვრავს წერტილოვან ჩარჩოს;
- dashed - განსაზღვრავს წყვეტილ ჩარჩოს;
- solid - განსაზღვრავს უწყვეტ ჩარჩოს;
- double - განსაზღვრავს ორმაგ ჩარჩოს;
- groove - განსაზღვრავს 3D ჩაღრმავებული ეფექტის მქონე ჩარჩოს. ეფექტი ჩარჩოს ფერზეა დამოკიდებული;
- ridge - განსაზღვრავს 3D ამობურცული ეფექტის მქონე ჩარჩოს. ეფექტი ჩარჩოს ფერზეა დამოკიდებული;
- inset - განსაზღვრავს 3D ჩარჩოში ჩაღრმავების ეფექტს. ეფექტი ჩარჩოს ფერზეა დამოკიდებული;
- outset - განსაზღვრავს 3D ჩარჩოში ამოზნექილობის ეფექტს. ეფექტი ჩარჩოს ფერზეა დამოკიდებული;
- none - ჩარჩო არ არის;
- hidden - ჩარჩო დამალულია.

border-style თვისებას შეიძლება ოთხი არგუმენტი (ზედა ჩარჩო, მარჯვენა ჩარჩო, ქვედა ჩარჩო და მარცხენა ჩარჩო) ჰქონდეს ანუ შესაძლებელია ჩარჩოს სხვადასხვა ნაწილისთვის სხვადასხვა სტილი მიეთითოს.

მაგალითი 9.1:

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```

p {
border-color:aqua;
border-width:thick;
}
p.dotted
{
border-style: dotted;
}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>

```

```
</head>
```

```
<body>
```

```
<h2>border-style თვისება</h2>
```

<p>მოცემული თვისება განსაზღვრავს, როგორ აისახება ჩარჩოს გვერდები:</p>

```
<p class="dotted"> dotted - წერტილოვანი ჩარჩო.</p>
```

```
<p class="dashed">dashed - წყვეტილი ჩარჩო.</p>
```

```
<p class="solid">solid - უწყვეტი ჩარჩო.</p>
```

```
<p class="double">double - ორმაგი ჩარჩო.</p>
```

```
<p class="groove">groove - ჩაღრმავებული ჩარჩო.</p>
<p class="ridge"> ridge - ამობურცული ჩარჩო.</p>
<p class="inset"> inset - ჩარჩოში ჩაღრმავებული.</p>
<p class="outset"> outset - ჩარჩოში ამობურცული.</p>
<p class="none"> none - ჩარჩო არ არის.</p>
<p class="hidden"> hidden - დამალული ჩარჩო.</p>
<p class="mix"> mix - ჩარჩო შერეული საზღვრებით.</p>
</body>
</html>
```

border-style თვისება

მოცემული თვისება განსაზღვრავს, როგორ აისახება ჩარჩოს გვერდები:

dotted - წერტილოვანი ჩარჩო.

dashed - წყვეტილი ჩარჩო.

solid - უწყვეტი ჩარჩო.

double - ორმაგი ჩარჩო.

groove - ჩაღრმავებული ჩარჩო.

ridge - ამობურცული ჩარჩო.

inset - ჩარჩოში ჩაღრმავებული.

outset - ჩარჩოში ამობურცული.

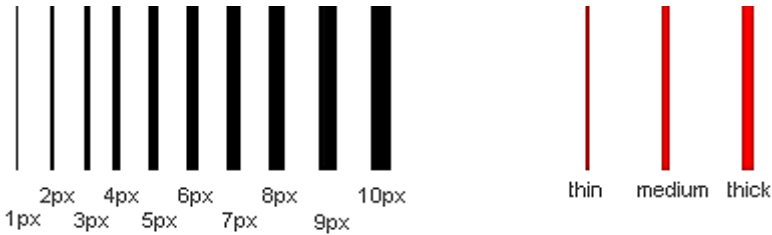
none - ჩარჩო არ არის.

hidden - დამალული ჩარჩო.

mix - ჩარჩო შერეული საზღვრებით.

ჩარჩოს სისქე (border-width)

ჩარჩოს სისქე border-width თვისებით განისაზღვრება, რომელსაც შეიძლება ხაზის სისქის რიცხვითი მნიშვნელობა მივანიჭოთ (px, pt, cm - სმ, mm - მმ და სხვა) ან შეიძლება ჰქონდეს ერთ-ერთი თვისება: thin (წვრილი), medium (საშუალო) და thick (სქელი).



border-width თვისება ოთხივე მხარის ჩარჩოს სისქეს განსაზღვრავს.

border-width თვისებას შეიძლება ჰქონდეს ოთხი არგუმენტი (ზედა, მარჯვენა, ქვედა და მარცხენა ჩარჩო) ანუ შესაძლებელია მიეთითოს სხვადასხვა სისქე ჩარჩოს ზედა, ქვედა, მარჯვენა ან მარცხენა ნაწილისთვის.

ჩარჩოს ფერი (border-color)

ჩარჩოს ფერს border-color თვისება განსაზღვრავს.

CSS3-ის border-radius თვისება

CSS3-ში ნებისმიერი ელემენტისათვის „მომრგვალებული კუთხეების“ შესაქმნელად border-radius თვისება შეგიძლიათ გამოიყენოთ.

მაგალითი 9.2:

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners1
{
  border-radius: 25px;
  background: #800000;
  padding: 10px;
  width: 150px;
  height: 100px;
}
#rcorners2
{
  border-radius: 25px;
  border: 2px solid #800000;
  padding: 10px;
  width: 150px;
  height: 100px;
}
#rcorners3
{
  border-radius: 25px;
  background: url(oceano.jpg);
  background-position: left top;
  background-repeat: repeat;
  padding: 10px;
```

```

width: 150px;
height: 100px;
}
</style>
</head>
<body>
<p>border-radius თვისება საშუალებას გაძლევთ ელემენტს
მომრგვალებული კუთხეები დაამატოთ.</p>
<p> მომრგვალებული კუთხეები ფონის მქონე
ელემენტისთვის: </p>
<p id="rcorners1"> მართკუთხედი მომრგვალებული
კუთხეებით </p>
<p> მომრგვალებული კუთხეები ჩარჩოიანი
ელემენტისათვის: </p>
<p id="rcorners2"> მართკუთხედი მომრგვალებული
კუთხეებით </p>
<p> მომრგვალებული კუთხეები ფონურგამოსახულებიანი
ელემენტისათვის: </p>
<p id="rcorners3"> მართკუთხედი მომრგვალებული
კუთხეებით </p>
</body>
</html>

```

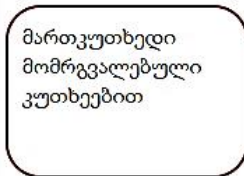
მიღებულ შედეგს შემდეგი სახე ექნება:

border-radius თვისება საშუალებას გაძლევთ ელემენტს მომრგვალებული კუთხეები დაამატოთ.

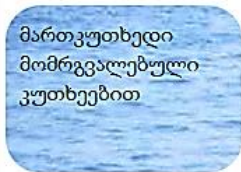
მომრგვალებული კუთხეები ფონის მქონე ელემენტისთვის:



მომრგვალებული კუთხეები ჩარჩოიანი ელემენტისათვის:



მომრგვალებული კუთხეები ფონურგამოსახულებიანი ელემენტისათვის:



უნდა აღინიშნოს, რომ border-radius თვისება შემდეგი ოთხი თვისების - border-top-left-radius, border-top-right-radius, border-bottom-right-radius და border-bottom-left-radius-ის შემოკლებული ჩანაწერია.

მომხმარებელს თუ სურვილი აქვს, შეუძლია თითოეული კუთხისთვის ცალ-ცალკე მიუთითოს რადიუსი შემდეგი წესის გამოყენებით:

1. თუ მითითებულია ოთხივე მნიშვნელობა, პირველი მნიშვნელობა მიეკუთვნება ზედა მარცხენა კუთხეს, მეორე -

ზედა მარჯვენა კუთხეს, მესამე - ქვედა მარჯვენა კუთხეს და მეოთხე - ქვედა მარცხენა კუთხეს.

2. თუ მითითებულია სამი მნიშვნელობა: პირველი მნიშვნელობა მიეკუთვნება ზედა მარცხენა კუთხეს, მეორე - ზედა მარჯვენა კუთხეს და ქვედა მარცხენა კუთხეს, მესამე - ქვედა მარჯვენა კუთხეს.
3. თუ მითითებულია ორი მნიშვნელობა: პირველი მნიშვნელობა მიეკუთვნება ზედა მარცხენა კუთხეს და ქვედა მარჯვენა კუთხეს, ხოლო მეორე - ზედა მარჯვენა კუთხეს და ქვედა მარცხენა კუთხეს.
4. ერთი მნიშვნელობა მიუთითებს, რომ ოთხივე კუთხე თანაბრად იქნება მომრგვალებული.

შემდეგ მაგალითში სამი სხვადასხვა შემთხვევა გვაქვს განხილული:

1. ოთხი მნიშვნელობა - border-radius: 15px 50px 30px 5px;
2. სამი მნიშვნელობა - border-radius: 15px 50px 30px;
3. ორი მნიშვნელობა - border-radius: 15px 50px.

მაგალითი 9.3:

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners4
{
border-radius: 15px 50px 30px 5px;
background: #800000;
padding: 10px;
```

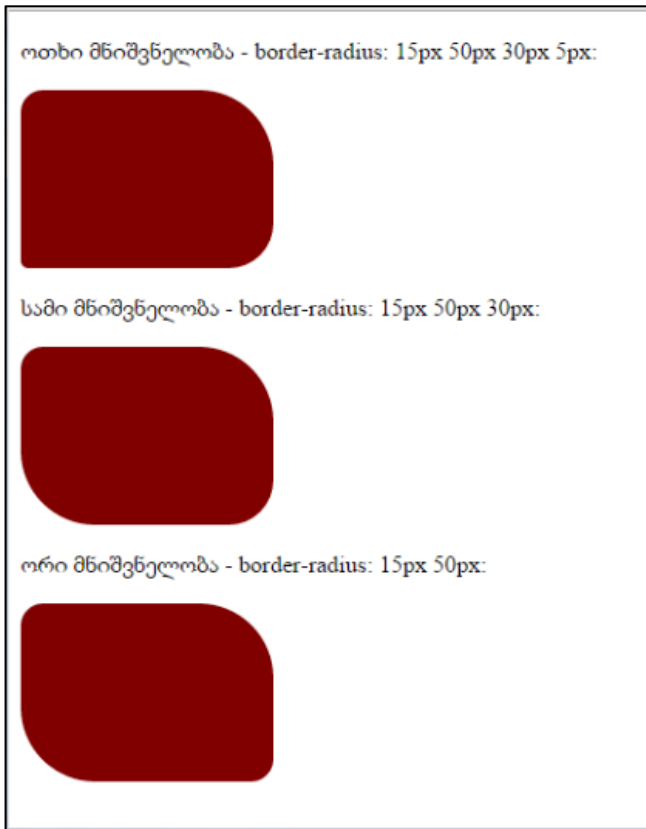
```

width: 150px;
height: 100px;
}
#rcorners5
{
border-radius: 15px 50px 30px;
background: #800000;
padding: 10px;
width: 150px;
height: 100px;
}
#rcorners6
{
border-radius: 15px 50px;
background: #800000;
padding: 10px;
width: 150px;
height: 100px;
}
</style>
</head>
<body>
<p>
ოთხი მნიშვნელობა - border-radius: 15px 50px 30px 5px:
</p>
<p id="rcorners4"></p>
<p>
სამი მნიშვნელობა - border-radius: 15px 50px 30px:

```

```
</p>
<p id="rcorners5"></p>
<p>ორი მნიშვნელობა - border-radius: 15px 50px:</p>
<p id="rcorners6"></p>
</body>
</html>
```

შედეგად მივიღებთ:



ასევე შეგიძლიათ შექმნათ ელიფსური კუთხეები:

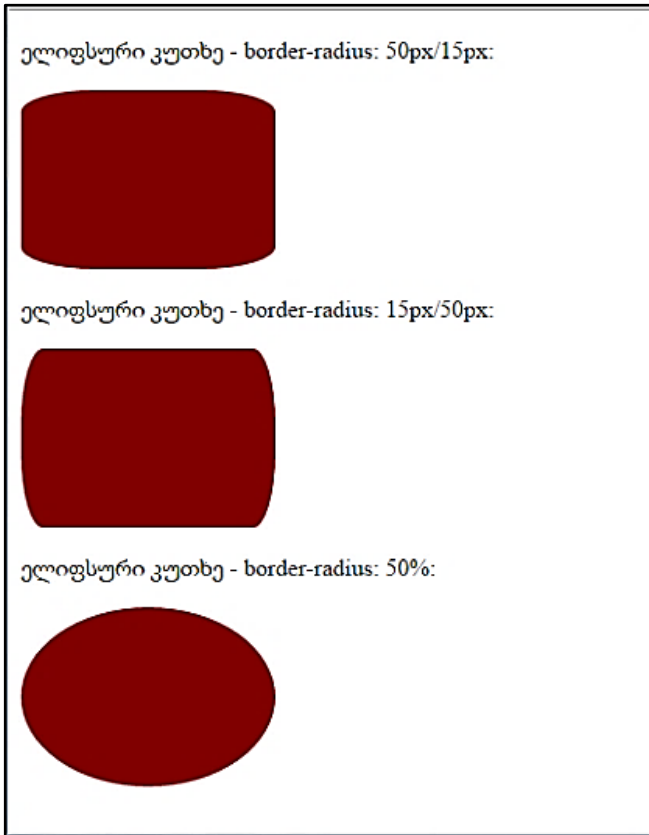
მაგალითი 9.4:

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners7 {
  border-radius: 50px/15px;
  background: #800000;
  padding: 10px;
  width: 150px;
  height: 100px; }
#rcorners8 {
  border-radius: 15px/50px;
  background: #800000;
  padding: 10px;
  width: 150px;
  height: 100px; }
#rcorners9 {
  border-radius: 50%;
  background: #800000;
  padding: 10px;
  width: 150px;
  height: 100px; }
</style>
</head>
<body>
<p>ელიფსური კუთხე - border-radius: 50px/15px:</p>
<p id="rcorners7"></p>
```



```
<p>ელიფსური კუთხე - border-radius: 15px/50px:</p>
<p id="rcorners8"></p>
<p>ელიფსური კუთხე - border-radius: 50%:</p>
<p id="rcorners9"></p>
</body>
</html>
```

მიღებულ შედეგს შემდეგი სახე ექნება:



CSS სხვა თვისებების მსგავსად, შემოკლებული ჩანაწერი შესაძლებელია border თვისებისთვისაც გვქონდეს. ვთქვათ, გვაქვს ჩანაწერი

```
P
{
border-width: 1px;
border-style: solid;
border-color: blue;
}
```

იგი შემოკლებით ასე შეიძლება ჩაიწეროს:

```
P
{
border: 1px solid blue;
}
```

ელემენტის საზღვრის ველების (margin) განსაზღვრა

ელემენტს აქვს ოთხი მხარე: right, left, top და bottom. საზღვრის ველები (margin) განსაზღვრავს თითოეული მხარიდან მეზობელ ელემენტამდე ან დოკუმენტის კიდეებამდე დაშორებას.

Css კოდი ველების (margin) განსაზღვრისთვის ასე გამოიყურება:

```
body {
margin-top: 100px; margin-right: 40px; margin-bottom: 10px;
margin-left: 70px;
```

```
}
```

ან იგივე შემოკლებით შეიძლება ასეც ჩაიწეროს:

```
body {  
  margin: 100px 40px 10px 70px;  
}
```

ამ დროს ჩაწერას ზედა ველიდან ვიწყებთ და საათის ისრის მიმართულებით ვაგრძელებთ.

შეგიძლიათ ელემენტის საზღვრის ველები თითქმის ნებისმიერი ელემენტისთვის დააყენოთ. მაგალითად, შესაძლებელია საზღვრის ველების განსაზღვრა ყველა პარაგრაფისთვის:

```
Body {  
  margin: 100px 40px 10px 70px;  
}  
  
p {  
  margin: 5px 50px 5px 50px;  
}
```

ჩარჩოსა და ელემენტს შორის დაშორების განსაზღვრა

ჩარჩოსა და ელემენტის შემცველობას შორის შიგა დაშორებას padding ელემენტი განსაზღვრავს.

padding-ის გამოყენება მარტივ მაგალითზე შეიძლება ვაჩვენოთ, სადაც ყველა სათაურს ფერადი ფონი აქვს.

```
h1 {  
  background: yellow;
```

```

}
h2 {
  background: orange;
}

```

Headings and padding

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevis in manibus non est et mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigitur quotiens, sit prior, Pacuvius docti.

Hos ediscit et hos arto stipata

Indignor quicquam reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit, vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

Quod si tam Graecis novitas

Quod si tam Graecis novitas invisae fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque viritum. Ut primum positis nugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc arsit equorum, marmoris aut eboris fabros aut aeris amavit, tibicinibus, nunc est gavisus tragoedis; puella.

მართალია, padding-ის საშუალებით სათაურისთვის საიტის კიდებიდან დაცილებას განვსაზღვრავთ, მაგრამ ამავე კოდით საზღვრის ველების სიდიდე სათაურის მთელი ტექსტისთვის ავტომატურად განისაზღვრება:

```

h1 {
  background: yellow; padding: 20px 20px 20px 80px; }
h2 {
  background: orange; padding-left: 120px;
}

```

ამ კოდის ჩაწერის შემდეგ ვებგვერდს შემდეგი სახე ექნება:

Headings and paddings

Ennius et sapiens et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevis in manibus non est et mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigunt quotiens, sit prior, Pacuvius docti.

Hos ediscit et hos arto stipata

Indignor quicquam reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament perisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

Quod si tam Graecis novitas

Quod si tam Graecis novitas invisa fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque virum. Ut primum positus nugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum studiis, nunc arsit equorum, marmoris aut

ელემენტების სიგანე და სიმაღლე

width თვისება ელემენტის სიგანეს განსაზღვრავს, მაგალითად:

```
div.box {  
  width: 250px;  
  border: 4px solid navy;  
  background: yellow;  
  color: red; }
```

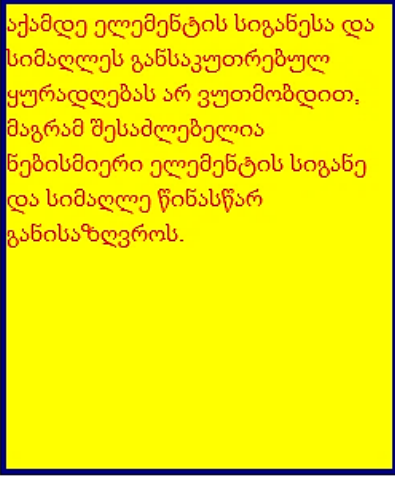
შედეგად ტექსტური ბლოკი 250 პიქსელი სიგანის, 4 პიქსელი მუქი ლურჯი ჩარჩოთი, ყვითელი ფონითა და წითელი შრიფტით გვექნება:

აქამდე ელემენტის სიგანესა და სიმაღლეს განსაკუთრებულ ყურადღებას არ ვუთმობდით, მაგრამ შესაძლებელია ნებისმიერი ელემენტის სიგანე და სიმაღლე წინასწარ განისაზღვროს.

როდესაც ელემენტის (მოცემულ შემთხვევაში ტექსტური ბლოკი) სიმაღლე მითითებული არ არის, მაშინ ის მასში მოცემული ტექსტით განისაზღვრება. ზოგადად, ელემენტის სიმაღლე შეიძლება height თვისებით განვსაზღვროთ.

```
div.box {  
  height: 300px;  
  width: 250px;  
  border: 4px solid navy;  
  background: yellow;  
  color: red; }
```

შედეგად ასეთი სახის ტექსტური ბლოკი მიიღება:



დავალება. გამოიყენეთ საზღვრების დაფორმატების თვისებები და ააგეთ ქვემოთ მოყვანილი ვებკომპონენტები. ვებგვერდი გააფორმეთ შემდეგნაირად:

✓ კონტეინერის ფონი ვერცხლისფერია. საზღვრები - 2px, შავი ფერის. განთავსებულია ბრაუზერის ცენტრში, ზომები - 800x130;

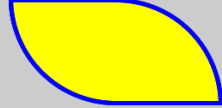
✓ div ტეგების ფონი ყვითელია. ზომებია 200x100-ზე, გარდა პირველი ჰორიზონტალურ ზოლში მოცემული მესამე div ტეგისა, რომლის ზომებია 0x0;

✓ განსხვავებული კომპონენტებისთვის გამოიყენეთ განსხვავებული ფერის და სისქის საზღვრები;

✓ პირველ ჰორიზონტალურ ზოლში მოცემული პირველი div ტეგისათვის გამოყენებულია წითელი წერტილოვანი საზღვარი, მეორესთვის ზედა საზღვარი წერტილოვანი წითელი ფერისაა, ქვედა წყვეტილი - ლურჯი, მარცხენა მწვანე - უწყვეტი, ხოლო მარცხნივ საზღვარი არა აქვს. მესამე div ტეგში საზღვრები სხვადასხვა ფერისაა (მარცხენა - მწვანე, ზედა - წითელი, მარჯვენა - ყვითელი და ქვედა - ლურჯი, სიგანე - 30 px. (მოყვანილია საზღვრების შეერთების დემონსტრირება);

✓ მეორე ზოლში ყველა div ტეგისათვის მომრგვალების რადიუსი 100 px-ია.

Borders



თავი 10

გრადიენტი CSS3-ში

- ხაზოვანი გრადიენტი;
- გრადიენტის მიმართულების კონტროლი კუთხეების გამოყენებით;
- გამჭვირვალობის გამოყენება გრადიენტებში;
- რადიალური გრადიენტი ფერების თანაბარი განაწილებით;
- რადიალური გრადიენტი ფერების არათანაბარი განაწილებით.

ფერთა გადასვლა (გრადიენტი) CSS3-ში

CSS3-ში ფერთა გადასვლა ანუ გრადიენტი ორ ან მეტ ფერს შორის რბილად გადასვლას უზრუნველყოფს.

CSS3-ში ორი ტიპის გრადიენტი განიხილება:

- ხაზოვანი გრადიენტი (მიმართულია ქვემოთ, ზემოთ, მარცხნივ, მარჯვნივ, დიაგონალურად);
- რადიალური გრადიენტი (განისაზღვრება ცენტრის მიმართ).

ხაზოვანი გრადიენტი

ხაზოვანი გრადიენტის შესაქმნელად მინიმუმ ორი ფერი უნდა განისაზღვროს. ეს ორი ფერი ის ფერებია, რომელთა შორისაც გადასვლა უნდა მოხდეს. ასევე გრადიენტის ეფექტთან ერთად უნდა განისაზღვროს გრადიენტის საწყისი წერტილი, მიმართულება (ან კუთხე).

ქვემოთ მოყვანილია ხაზოვანი გრადიენტის მაგალითი, სადაც ფერები ზემოდან ქვემოთ იცვლება.

მაგალითი 10.1:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 { height: 200px;
background: linear-gradient(red, green);
}
</style>
```

```
</head>
<body>
<h3>ხაზოვანი გრადიენტი -ზემოდან ქვემოთ</h3>
<p>მოცემული ხაზოვანი გრადიენტი იწყება ზემოდან. იგი იწყება წითელი ფერით და თანდათან გადადის მწვანეში:</p>
<div id="grad1"></div>
</body>
</html>
```



ანალოგიურად, თუ ზემოთ მოყვანილ პროგრამაში სტილს შემდეგნაირად ჩავწერთ:

```
#grad1 { height: 200px;
background: linear-gradient(to right, red , green);
}
```

მივიღებთ ხაზოვან გრადიენტს მარცხნიდან მარჯვნივ, ხოლო თუ შემდეგ ჩანაწერს გავაკეთებთ:

```
#grad1 { height: 200px;
```

```
background: linear-gradient(to bottom right, red , green);
}
```

მაშინ მივიღებთ დიაგონალურ ხაზოვან გრადიენტს ზედა მარცხენა კუთხიდან ქვედა მარჯვენა კუთხისკენ.

გრადიენტის მიმართულების კონტროლი კუთხეების გამოყენებით

თუ გსურთ გრადიენტის მიმართულების კონტროლი, მაშინ წინასწარ მოცემული მიმართულების ნაცვლად (ქვემოთ, ზემოთ, მარცხნივ, მარჯვნივ და ა. შ.) უნდა განსაზღვროთ კუთხე. მისი სინტაქსი იქნება:

```
background: linear-gradient(კუთხე, ფერი1, ფერი2);
```

ამ შემთხვევაში კუთხე უნდა ჩავწეროთ, როგორც კუთხე ჰორიზონტალურ ხაზსა და გრადიენტს შორის საათის ისრის საპირისპირო მიმართულებით.

მაგალითი 10.2:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 100px;
  background: linear-gradient(0deg, red, green);
}
#grad2 {
  height: 100px;
```

```

    background: linear-gradient(90deg, red, green);
  }
#grad3 {
  height: 100px;
  background: linear-gradient(180deg, red, green);
  }
#grad4 {
  height: 100px;
  background: linear-gradient(-90deg, red, green);
  }

```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h3>ხაზოვანი გრადიენტი - სხვადასხვა კუთხის
```

გამოყენებით

```
</h3>
```

```
<div id="grad1" style="color:white;text-align:center;">
```

0 გრადუსი

```
</div>
```

```
<br>
```

```
<div id="grad2" style="color:white;text-align:center;">
```

90 გრადუსი

```
</div><br>
```

```
<div id="grad3" style="color:white;text-align:center;">
```

180 გრადუსი

```
</div>
```

```
<br>
```

```
<div id="grad4" style="color:white;text-align:center;">
```

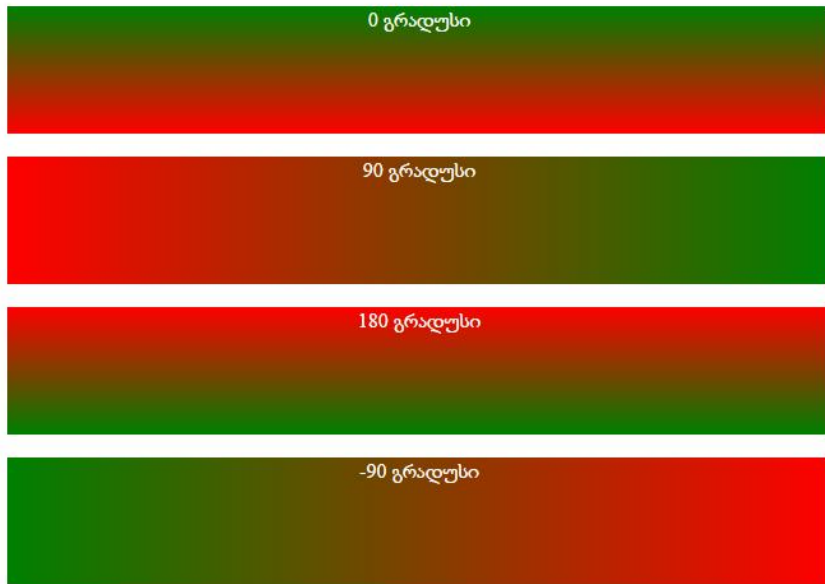
-90 გრადუსი

</div>

</body>

</html>

ხაზოვანი გრადიენტი - სხვადასხვა კუთხის გამოყენებით



ქვემოთ მოყვანილია რამდენიმე ფერის (ცისარტყელას ფერების) გამოყენებით ხაზოვანი გრადიენტის მაგალითი.

მაგალითი 10.3:

<!DOCTYPE html>

<html>

<head>

<style>

```
#grad1 { height: 55px;
background: linear-gradient(to right, red, orange, yellow,
green, blue, indigo, violet);
}
</style>
</head>
<body>
<div id="grad1" style="text-align:center;margin:auto;
color:#888888;font-size:40px;font-weight:bold">
გრადიენტის ფონი
</div>
</body>
</html>
```



გამჭვირვალობის გამოყენება გრადიენტებში

გრადიენტში ასევე შეიძლება გამოყენებულ იქნეს გამჭვირვალობაც. გამჭვირვალობის დასამატებლად RGBA () ფუნქცია გამოიყენება. RGBA () ფუნქციის ბოლო პარამეტრმა შეიძლება მიიღოს მნიშვნელობები 0-დან 1-მდე და ეს ფერის გამჭვირვალობას განსაზღვრავს: 0 - სრულად გამჭვირვალე, ხოლო 1 - სრულად გაუმჭვირ ფერს გულისხმობს.

მაგალითი 10.4:

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

```

<head>
<title>
gradients
</title>
<style>
#grad1 {
    height: 200px;
    background: linear-gradient(to right, rgba(255,0,0,0),
rgba(255,0,0,1));
}
</style>
</head>
<body>
<h3>
ხაზოვანი გრადიენტი გამჭვირვალობის გამოყენებით
</h3>
<p>
გამჭვირვალობის დასამატებლად RGBA () ფუნქცია
გამოიყენება, რათა მოხდეს ფერის განსაზღვრა.
    RGBA () ფუნქციის ბოლო პარამეტრმა შეიძლება მიიღოს
მნიშვნელობები 0-დან 1-მდე და ეს ფერის გამჭვირვალობას
განსაზღვრავს: 0 სრულ გამჭვირვალობას, ხოლო 1 სრულ ფერს
(გაუმჭვირობას) აჩვენებს.
</p>
<div id="grad1"></div>
</body>
</html>

```

მიღებული შედეგი:

ხაზოვანი გრადიენტი გამჭვირვალობის გამოყენებით

გამჭვირვალობის დასამატებლად RGBA () ფუნქცია გამოიყენება, რათა მოხდეს ფერის განსაზღვრა. RGBA () ფუნქციის ბოლო პარამეტრმა შეიძლება მიიღოს მნიშვნელობები 0-დან 1-მდე და ეს ფერის გამჭვირვალობას განსაზღვრავს: 0 სრულ გამჭვირვალობას, ხოლო 1 სრულ ფერს (გაუმჭვირობას) აჩვენებს.



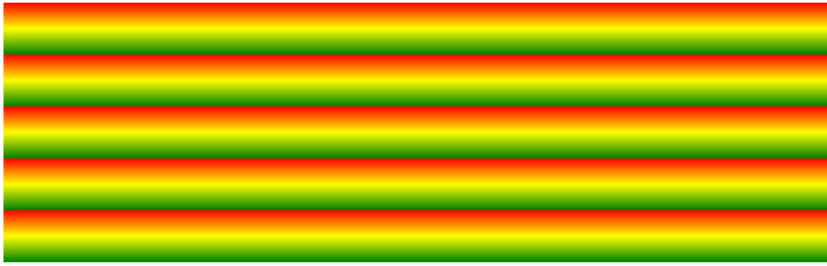
ხაზოვანი გრადიენტების გამეორებისათვის repeating-linear-gradient() ფუნქცია გამოიყენება.

მაგალითი 10.5:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 { height: 200px;
background: repeating-linear-gradient(red, yellow 10%,
green 20%);
}
</style>
</head>
<body>
<h3>გამეორებადი ხაზოვანი გრადიენტი</h3>
<div id="grad1"></div>
```

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

გამეორებადი ხაზოვანი გრადიენტი



რადიალური გრადიენტი ფერების თანაბარი განაწილებით

რადიალური გრადიენტის დროს პირველ რიგში მისი ცენტრი განისაზღვრება. მისი შექმნისთვის მინიმუმ ორი ფერია საჭირო:

```
background: radial-gradient(ფორმა, სასტარტო ფერი, ..., ბოლო  
ფერი);
```

თუ ფორმა არ არის მითითებული, მაშინ გრადიენტს ელიფსის ფორმა ექნება. თუ გვინდა, რომ მას წრის ფორმა ჰქონდეს, მაშინ circle-ს ვუთითებთ.

მაგალითი 10.6:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#grad1 {
```

```
height: 150px;  
width: 200px;  
background: radial-gradient(red, green, blue);  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h3>
```

რადიალური გრადიენტი - ფერების თანაბარი
განაწილებით

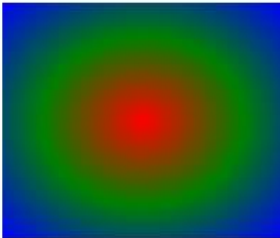
```
</h3>
```

```
<div id="grad1"></div>
```

```
</body>
```

```
</html>
```

რადიალური გრადიენტი - ფერების თანაბარი განაწილებით

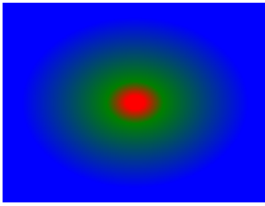


რადიალური გრადიენტი ფერების არათანაბარი განაწილებით

მაგალითი 10.7:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 150px;
  width: 200px;
  background: radial-gradient(red 5%, green 15%, blue 60%);
}
</style>
</head>
<body>
<h3>
რადიალური გრადიენტი - ფერების არათანაბარი
განაწილებით
</h3>
<div id="grad1"></div>
</body>
</html>
```

რადიალური გრადიენტი - ფერების არათანაბარი განაწილებით

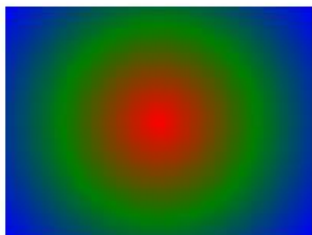


ქვემოთ მოცემულ მაგალითში ნაჩვენებია წრიული ფორმის რადიალური გრადიენტის შექმნა.

მაგალითი 10.8:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 { height: 150px;
width: 200px;
background: radial-gradient(circle, red, green, blue);
}
</style>
</head>
<body>
<h3>წრიული ფორმის რადიალური გრადიენტი - ფერების
თანაბარი განაწილებით</h3>
<div id="grad1"></div>
</body>
</html>
```

წრიული ფორმის რადიალური გრადიენტი - ფერების
თანაბარი განაწილებით



დავალება. გამოიყენეთ ფონში გრადიენტის ჩასმის თვისებები და ააგეთ მოყვანილი ვებკომპონენტები. ვებგვერდი გააფორმეთ შემდეგნაირად:

✓ კონტეინერის ფონი ვერცხილსფერია. საზღვრები 2px - შავი ფერის. განთავსებულია ბრაუზერის ცენტრში, ზომები - 800x130;

✓ გრადიენტები ყველგან სამფეროვანია (წითელი, მწვანე, ლურჯი);

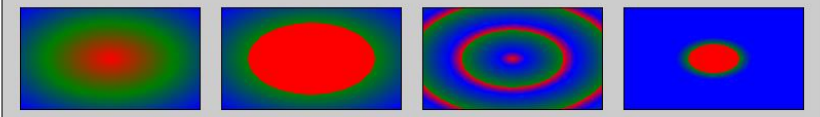
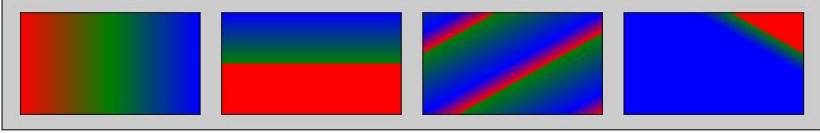
✓ პირველ კონტეინერში მოყვანილია ოთხი განსხვავებული წრფივი გრადიენტი:

- 1) ჰორიზონტალური (მარცხნიდან მარჯვნივკენ მიმართული) - თანაბარი პროცენტული განაწილებით;
- 2) ვერტიკალური (ქვემოდან ზემოთ მიმართული) - 40%, 10%, 50% პროცენტული განაწილებით;
- 3) დიაგონალური (ზედა მარცხენა წვეროსკენ მიმართული) - განმეორებადი, 50%, 10%, 40% პროცენტული განაწილებით;
- 4) დიაგონალური (ქვედა მარცხენა წვეროსკენ მიმართული) - 20%, 20%, 30% პროცენტული განაწილებით.

✓ მეორე კონტეინერში მოყვანილია ოთხი განსხვავებული წრიული გრადიენტი განსხვავებული პროცენტული შენარევით:

- 1) თანაბარი პროცენტული განაწილებით;
- 2) 40%, 10%, 50% პროცენტული განაწილებით;
- 3) განმეორებადი 50%, 10%, 40% პროცენტული განაწილებით;
- 4) 20%, 20%, 30% პროცენტული განაწილებით.

Gradients



თავი 11

ჩრდილის ეფექტები და გადასვლები CSS3-ში

- ჩრდილის ეფექტები ტექსტისათვის;
- ჩრდილის რამდენიმე ეფექტის ერთდროული გამოყენება;
- ჩრდილის ეფექტი სხვადასხვა ელემენტისთვის;
- გადასვლები CSS3-ში;
- ობიექტის რამდენიმე თვისების ერთდროული ცვლილება;
- გადასვლის სიჩქარე;
- გადასვლის ეფექტი დაყოვნებით;
- გადასვლის ეფექტი და გარდაქმნა.

ჩრდილის ეფექტები CSS3-ში

CSS3-ში შესაძლებელია ჩრდილის ეფექტების დამატება ტექსტისა და ზოგიერთი ელემენტისთვის:

- text-shadow - ჩრდილის ეფექტი ტექსტისთვის;
- box-shadow - ჩრდილის ეფექტი სხვადასხვა ელემენტისთვის.

ჩრდილის ეფექტები ტექსტისთვის

CSS3-ში ტექსტისთვის ჩრდილის ეფექტები text-shadow თვისებით ხორციელდება. თუ ამ თვისების არგუმენტებად მხოლოდ ჰორიზონტალური და ვერტიკალური ჩრდილის ზომის პარამეტრებს მივუთითებთ, მაშინ ჩრდილი შავი ფერის იქნება. ჩრდილის ფერის შესაცვლელად იგივე თვისებაში მესამე პარამეტრად ფერი უნდა დავამატოთ.

მაგალითი 11.1:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px red;
}
</style>
</head>
<body>
<h1>ჩრდილის ეფექტები ტექსტისათვის!</h1>
</body>
```

```
</html>
```

ჩრდილის ეფექტები ტექსტისათვის!

ჩრდილის ეფექტებში შეიძლება დამატებული იყოს ბუნდოვანების ეფექტი. შესაბამისი პარამეტრი ფერის წინ დამატება და პიქსელებში გამოისახება.

მაგალითი 11.2:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px 5px red;
}
</style>
</head>
<body>
<h1>ჩრდილის ეფექტები ტექსტისათვის!</h1>
</body>
</html>
```

ჩრდილის ეფექტები ტექსტისათვის!

მაგალითი 11.3: თეთრი ტექსტი შავი ჩრდილით

```

<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: white;
  text-shadow: 2px 2px 4px #000000;
}
</style>
</head>
<body>
<h1>ჩრდილის ეფექტები ტექსტისათვის!</h1>
</body>
</html>

```

ჩრდილის ეფექტები ტექსტისათვის!

მაგალითი 11.4: ტექსტის წითელი ჩრდილი ნეონის ნათების ეფექტით.

```

<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 0 0 3px #FF0000;
}
</style>
</head>

```

```
<body>
<h1>ჩრდილის ეფექტები ტექსტისათვის!</h1>
</body>
</html>
```

ჩრდილის ეფექტები ტექსტისათვის!

ჩრდილის რამდენიმე ეფექტის ერთდროული გამოყენება

ტექსტს შეიძლება ერთდროულად ჩრდილის რამდენიმე ეფექტი დავუმატოთ, ამისათვის საჭიროა text-shadow თვისებას არგუმენტებად ერთმანეთისაგან მძიმით გამოყოფილი ჩრდილების სია დავურთოთ. ქვემოთ მოცემულ მაგალითში განხილულია ტექსტის ორი - ლურჯი და წითელი ჩრდილების შექმნა ნეონის ეფექტით.

მაგალითი 11.5:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 { text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;}
</style>
</head>
<body>
<h1>ჩრდილის ეფექტები ტექსტისათვის!</h1>
</body>
</html>
```

ჩრდილის ეფექტი ტექსტისათვის!

მაგალითი 11.6: თეთრი ტექსტი შავი, ლურჯი და მუქი ლურჯი ჩრდილებით.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 { color: white;
text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px
darkblue;}
</style>
</head>
<body>
<h1>ჩრდილის ეფექტი ტექსტისათვის!</h1>
</body>
</html>
```

ჩრდილის ეფექტი ტექსტისათვის!

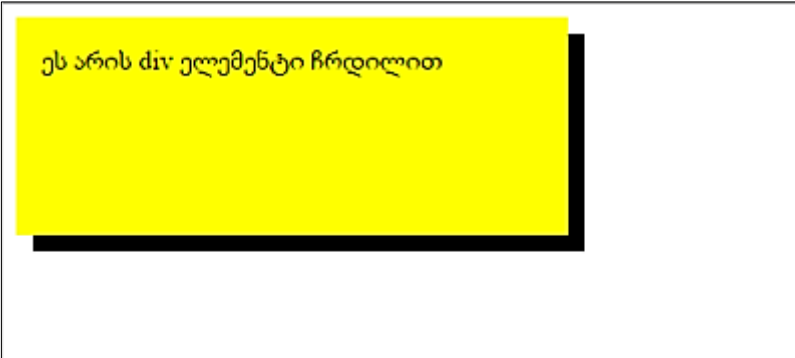
ჩრდილის ეფექტი სხვადასხვა ელემენტისთვის

ელემენტისთვის ჩრდილის შესაქმნელად CSS3-ში box-shadow თვისება გამოიყენება. მისი გამოყენების დროს, როგორც ტექსტის შემთხვევაში, ჰორიზონტალური და ვერტიკალური ჩრდილების ზომები, აგრეთვე ბუნდოვანების ეფექტი და ჩრდილის ფერი უნდა მივუთითოთ. თუ ჩრდილის ფერი

მითითებული არ არის, მაშინ ნაგულისხმევი წესის თანახმად, ჩრდილი შავი ფერის იქნება.

მაგალითი 11.7:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: yellow;
  box-shadow: 10px 10px;
}
</style>
</head>
<body>
<div>ეს არის div ელემენტი ჩრდილით</div>
</body>
</html>
```



თუ გვსურს ყვითელი <div> ელემენტის ჩრდილის ფერი იყოს ნაცრისფერი, მაშინ ჩანაწერი მიიღებს სახეს:

```
div {  
  box-shadow: 10px 10px grey;  
}
```

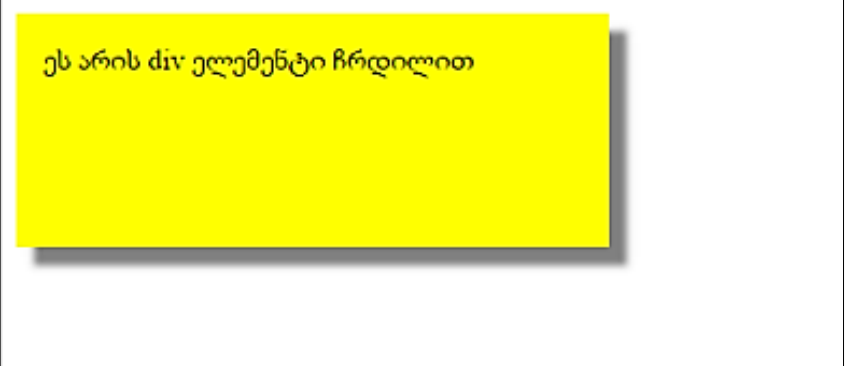
ქვემოთ მოცემულ მაგალითში ფერს ბუნდოვანების ეფექტის აქვს დამატებული.

მაგალითი 11.8:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div {  
  width: 300px;  
  height: 100px;  
  padding: 15px;  
  background-color: yellow;  
  box-shadow: 10px 10px 5px grey;
```



```
}  
</style>  
</head>  
<body>  
<div>ეს არის div ელემენტი ჩრდილით</div>  
</body>  
</html>
```



ეს არის div ელემენტი ჩრდილით

გადასვლები CSS3-ში

თვისებების მნიშვნელობების ცვლილება (ერთი მნიშვნელობიდან მეორეზე გადასვლა) დროის გარკვეულ მონაკვეთში ნელა უნდა განხორციელდეს.

იმისათვის, რომ გადასვლის ეფექტი შევქმნათ, საჭიროა CSS-ის თვისებას დაუმატოთ ეფექტი და მისი მოქმედების დრო.

თუ ეფექტის მოქმედების დრო მითითებული არ არის, მაშინ არავითარი გადასვლა არ მოხდება, ვინაიდან ნაგულისხმევი წესის თანახმად, ეს დრო 0-ის ტოლია.

შემდეგ მაგალითში მოცემულია <div> ელემენტი ზომებით 100px * 100px. <div> ელემენტს ასევე მითითებული აქვს გადასვლის ეფექტი, რომლის მიხედვითაც 2 წამის განმავლობაში მისი სიგანე უნდა გაიზარდოს.

გადასვლის ეფექტის განხორციელება მაშინ დაიწყება, როდესაც CSS თვისების (width) ცვლილება იქნება მითითებული. პროგრამაში width თვისების ახალი მნიშვნელობა მივუთითოთ და შემდეგ მაუსის მაჩვენებელი <div> ელემენტთან მივიყვანოთ.

მაგალითი 11.9:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s; /*სტანდარტული სინტაქსი */
}
div:hover {
  width: 300px; }
</style>
</head>
<body>
<div></div>
<p>გადასვლის ეფექტის სანახავად მაუსის მაჩვენებელი
მიიტანეთ ელემენტთან.</p>
```

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

მიაქციეთ ყურადღება, რომ როდესაც მაუსის მაჩვენებელს ელემენტს მოვაცილებთ, იგი თანდათან საწყის მდგომარეობას დაუბრუნდება.

ობიექტის რამდენიმე თვისების ერთდროული ცვლილება

შემდეგ მაგალითში გადასვლის ეფექტი დავამატოთ სიგანისა და სიმაღლის ერთდროულად შეცვლის მიზნით. დავუშვათ, გვინდა სიგანის ცვლილება 2 წამში, ხოლო სიმაღლის - 4 წამში. <div> ელემენტში განსახორციელებელ ცვლილებას შემდეგი სახე ექნება:

```
div {
  transition: width 2s, height 4s; /*სტანდარტული სინტაქსი*/
}
```

<div> ელემენტის ახალი მნიშვნელობა იქნება:

```
div:hover {
  width: 300px;
  height: 300px; }
```

თუ წინა მაგალითში ამ ცვლილებებს შევიტანთ, მაშინ ელემენტთან მაუსის მიტანის შემთხვევაში მისი სიგანისა და სიმაღლის ცვლილება ერთდროულად მოხდება.

გადასვლის სიჩქარე

გადასვლის ეფექტის სიჩქარე შემდეგი თვისებით რეგულირდება: transition-timing-function.

transition-timing-function თვისებას შემდეგი მნიშვნელობების მიღება შეუძლია:

- ease - გადასვლის ეს ეფექტი განსაზღვრავს გადასვლას ნელი სტარტიდან უფრო სწრაფ და ბოლოს ისევ ნელ მოქმედებაზე;
- linear - გადასვლის ეს ეფექტი განსაზღვრავს ერთი და იმავე სიჩქარით გადასვლას თავიდან ბოლომდე;
- ease-in - გადასვლის ეს ეფექტი განსაზღვრავს გადასვლას ნელი სტარტით;
- ease-out - გადასვლის ეს ეფექტი განსაზღვრავს გადასვლას ნელი დასასრულით;
- ease-in-out - გადასვლის ეს ეფექტი განსაზღვრავს გადასვლას ნელი სტარტით და დასასრულით.

მაგალითად, გადასვლის შესაბამისი ეფექტები შეიძლება ასე ჩაიწეროს:

```
#div1 {transition-timing-function: linear;}  
#div2 {transition-timing-function: ease;}  
#div3 {transition-timing-function: ease-in;}  
#div4 {transition-timing-function: ease-out;}  
#div5 {transition-timing-function: ease-in-out;}
```

გადასვლის ეფექტი დაყოვნებით

გადასვლის ეფექტი დაყოვნებით (წამებში) transition-delay თვისებით ხორციელდება.

შემდეგ მაგალითში გადასვლის ეფექტის დაწყების წინ 1-წამიანი დაყოვნება ხდება

```
div {
```

```
transition-delay: 1s; }
```

გადასვლის ეფექტი და გარდაქმნა

ქვემოთ მოცემულ მაგალითში გადასვლის ეფექტს დამატებული აქვს გარდაქმნაც, რის შედეგადაც კვადრატის ზომების გადიდება მის ბრუნვასთან ერთად ხორციელდება.

მაგალითი 11.10:

```
<!DOCTYPE html>
<html>
<head>
<style>
div { width: 100px;
      height: 100px;
      background: red;
      transition: width 2s, height 2s, transform 2s; }
div:hover { width: 300px;
            height: 300px;
            transform: rotate(180deg); }
</style>
</head>
<body>
<div></div>
<p>გადასვლის ეფექტის სანახავად მაუსის მაჩვენებელი
მიიტანეთ ელემენტთან. </p>
</body>
</html>
```

დავალბა 1. გამოიყენეთ text-shadow, box-shadow თვისებები და ააგეთ მოყვანილი ვებკომპონენტები. ვებგვერდი გააფორმეთ შემდეგნაირად:

✓ კონტეინერის ფონი ვერცხლისფერია. საზღვრები 2px - შავი ფერის. განთავსებულია ბრაუზერის ცენტრში, ზომები 750x100;

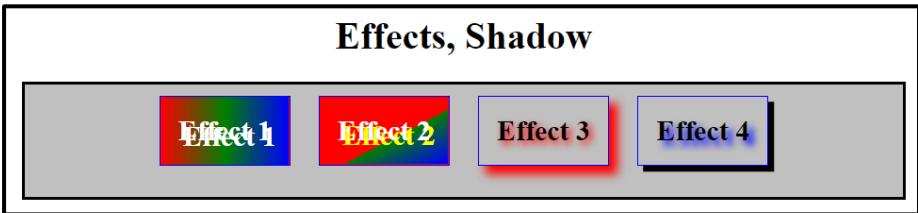
✓ ტექსტი “Effect” ჩაწერილი h2 ტეგებში;

✓ “Effect 1” - ფონზე სამფეროვანი (წითელი, მწვანე, ლურჯი) ჰორიზონტალური წრფივი გრადიენტი. ტექსტი - თეთრი, ჩრდილოვანი ეფექტი 4px, 5px;

✓ “Effect 2” - ფონზე სამფეროვანი (წითელი, მწვანე, ლურჯი) დიაგონალური წრფივი გრადიენტი. ტექსტი თეთრი - ყვითელი ფერის ჩრდილოვანი ეფექტი 4px, 4px;

✓ “Effect 3” - ფონი გამჭვირვალე. საზღვარი - ლურჯი. ტექსტი შავი, წითელი ფერის ჩრდილოვანი ეფექტი 3px, 3px - ბუნდოვანების მაჩვენებელი 8px, ობიექტის ჩრდილოვანი ეფექტი წითელი 7px, 7px ბუნდოვანების მაჩვენებელი 12px;

✓ “Effect 4” - ფონი გამჭვირვალე. საზღვარი - ლურჯი. ტექსტი შავი, ლურჯი ფერის ჩრდილოვანი ეფექტი 5px, 5px - ბუნდოვანების მაჩვენებელი 8px. ობიექტის ჩრდილოვანი ეფექტი 5px, 5px.

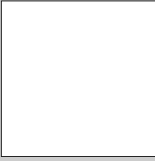


დავალეზა 2. გამოიყენეთ transition თვისება და ააგეთ div ტეგების ანიმაციები შემდეგი პირობების გათვალისწინებით:.

- ✓ კონტეინერების ფონი ვერცხლისფერია. განთავსებულია ბრაუზერის ცენტრში, ზომები 800x200;
- ✓ პირველი div ტეგის ზომებია 150x150-ზე, თეთრი ფონით, შავი საზღვრით;
- ✓ მეორე div ტეგის ზომებია 50x50-ზე, თეთრი ფონით, შავი საზღვრით;
- ✓ პირველი div ტეგის ანიმაცია: ანიმაცია ხორციელდება კვადრატზე მაუსის მიტანისას. კვადრატი გადაადგილდება 500px-ით მარცხნიდან მარჯვნივ, ზომებში მცირდება 50x50-მდე, იცვლის ფონის ფერს (ხდება ყვითელი), ანიმაცია გრძელდება 8 წამის განმავლობაში.
- ✓ მეორე div ტეგის ანიმაცია: ანიმაცია ხორციელდება კვადრატზე მაუსის მიტანით. კვადრატი იცვლის ზომებს: სიმაღლე - 150px, სიგანე - 700px, იცვლის ფონის ფერს (ხდება წითელი), ანიმაცია გრძელდება 6 წამის განმავლობაში.

ქვემოთ სურათზე მოცემულია ანიმაციის საწყისი წერტილი:

Transitions



ეს არის პირველი კვადრატის ანიმაციის მიმდინარეობა:

Transitions



ეს კი მეორე კვადრატის ანიმაციის მიმდინარეობა:

Transitions



თავი 12

გამჭვირვალობა, ელემენტების პოზიციონირება და ფენები CSS3

- CSS-ში სურათის გამჭვირვალობა;
- ელემენტების სასურველ ადგილას განთავსება;
- ელემენტების პოზიციონირება;
- ფარდობითი პოზიციონირება;
- ფენების შექმნა Z-ინდექსის საშუალებით.

CSS-ში სურათის გამჭვირვალობა

CSS-ში გამჭვირვალე სურათების შექმნა მარტივია.

შევქმნათ გამჭვირვალე გამოსახულება. ამისათვის CSS3-ის opacity თვისება გამოვიყენოთ.

საწყისი გამოსახულება:



გარდაქმნის შემდეგ ის ასეთი გახდება:



მაგალითი 12.1:

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
img {
  opacity: 0.4;
}
</style>
</head>
<body>
<h3>სურათის გამჭვირვალობა</h3>

</body>
</html>

```

opacity თვისებას შეუძლია მიიღოს მნიშვნელობები 0.0-დან 1.0-მდე. რაც მცირეა მისი მნიშვნელობა, მით უფრო გამჭვირვალეა გამოსახულება.

ეხლა განვიხილოთ მაგალითი, როდესაც გამჭვირვალობის ეფექტს გამოსახულებაზე მაუსის მაჩვენებლის მიახლოების დროს გამოვიყენებთ:

მაგალითი 12.2:

```

<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.4;
}

```

```

img:hover {
  opacity: 1.0;}
</style>
</head>
<body>
<h1>სურათის გამჭვირვალობა</h1>

</body>
</html>

```

ამ კოდის პირველი ბლოკი წინა მაგალითის მსგავსია, მაგრამ მეორე ბლოკიც დავამატეთ, რომლის მიხედვითაც შემდეგი მოქმედება ხორციელდება: თუ მომხმარებელი მაუსს გამოსახულებასთან მიიყვანს, გამოსახულება გაუმჭვირი გახდება ანუ `opacity:1;` გამოსახულებიდან მაუსის მაჩვენებლის გადაადგილების შემთხვევაში გამოსახულება ისევ გამჭვირვალე ხდება.

CSS3-ის იმავე თვისების გამოყენებით შესაძლებელია გამოსახულების გამჭვირვალე ნაწილზე ტექსტი განვათავსოთ.

მაგალითი 12.3:

```

<!DOCTYPE html>
<html>
<head>
<style>
div.background {
  background: url(roze.jpg) repeat;
  border: 2px solid black;
}

```

```
div.transbox {
  margin: 75px;
  background-color: #ffffff;
  border: 1px solid black;
  opacity: 0.6;
}
```

```
div.transbox p {
  margin: 5%;
  font-weight: bold;
  color: #000000;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="background">
```

```
<div class="transbox">
```

<p>ეს ტექსტი გამოსახულების გამჭვირვალე ნაწილშია
განთავსებული

```
</p>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```



ელემენტების სასურველ ადგილას განთავსება

ელემენტების სასურველ ადგილას განსათავსებლად float თვისება გამოიყენება. დავუშვათ, გვინდა ვებგვერდზე სურათი მარცხნივ და ტექსტი მარჯვნივ განვათავსოთ. ასეთ შემთხვევაში HTML კოდში გვექნება:

```
<div id="picture">  
    
</div>  
<p> text ...</p>
```

CSS კოდს კი შემდეგი სახე ექნება:

```
#picture {  
  float:left;  
  width: 150px;  
}
```


A floating image



Bill Gates sold a whopping \$1.5 billion of stock in Microsoft this year making him the top corporate insider who sold the most stocks in his company, according to a new research.

The Microsoft founder was closely followed by other tech executives such as Google co-founders Sergey Brin and Lawrence Page who each sold more than \$800 million of company's stocks in 2015, according to Scoop.com.

Brin took up two positions in the top ten by selling shares not only in Google but also in Alphabet which the company acquired in October this year.

New to the list from the first half of the year, WhatsApp CEO and co-founder Jan Koum sold \$561 million worth of stock to land in fourth place.

While former Twitter CEO Evan William sold \$359 million of shares in the company he helped to build.

float თვისება შეიძლება გამოვიყენოთ ასევე დოკუმენტის სვეტების სახით გამოსატანად.

HTML კოდი:

```
<div id="column1">  
<p>ტექსტი...</p>  
</div>  
<div id="column2">  
<p>ტექსტი...</p>  
</div>  
<div id="column3">  
<p>ტექსტი...</p>  
</div>
```

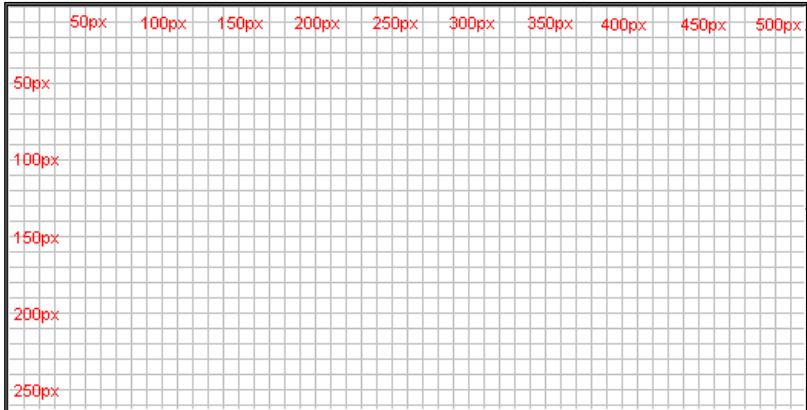
CSS კოდი:

```
#column1 {  
    float:left;  
    width: 33%;  
}  
#column2 {  
    float:left;  
    width: 33%;  
}  
#column3 {  
    float:left;  
    width: 33%;  
}
```

ელემენტების პოზიციონირება

CSS პოზიციონირების საშუალებით შესაძლებელია ელემენტის ზუსტად საჭირო ადგილას განთავსება.

ბრაუზერის ფანჯარა წარმოვადგინოთ, როგორც კოორდინატთა სისტემა:



CSS პოზიციონირების პრინციპი იმაში მდგომარეობს, რომ კოორდინატთა სისტემაში ელემენტი შეგიძლიათ მოათავსოთ იქ, სადაც გნებავთ.

მაგალითისათვის დოკუმენტის ოთხივე კუთხეში აბსოლუტური პოზიციონირების გზით ოთხი ბლოკი მოვათავსოთ. ამისათვის შემდეგი სახის CSS კოდი უნდა დავწეროთ:

მაგალითი 12.4 (CSS კოდი)

```
#box1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
#box2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}
```

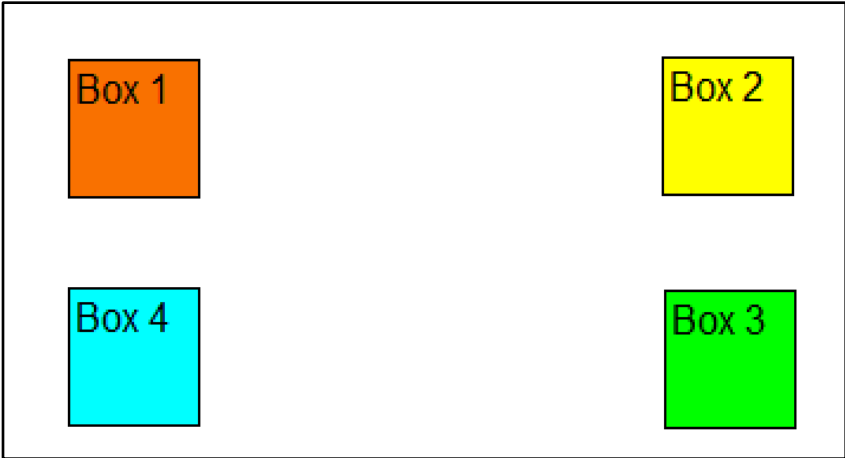
```
#box3 {
    position: absolute;
    bottom: 50px;
    right: 50px;
}
#box4 {
    position: absolute;
    bottom: 50px;
    left: 50px;
}
```

მაგალითი 12.5 (HTML კოდი):

```
<!DOCTYPE HTML>
<html>
<head>
<title>ელემენტების პოზიციონირება</title>
</head>
<body>
<link rel="stylesheet" type="text/css" href="style.css">
<div id="box1">
    
</div>
<div id="box2">
    
</div>
<div id="box3">
    
</div>
<div id="box4">
```

```

</div>
</body>
</html>
```



ფარდობითი პოზიციონირება

ელემენტის ფარდობითი პოზიციონირებისთვის position თვისებას relative მნიშვნელობა მივანიჭოთ. ფარდობითი პოზიციონირების დროს ელემენტის მდებარეობის შეცვლა მისი არსებული პოზიციის მიმართ (მის პოზიციასთან ფარდობით) ხდება.

მაგალითი 12.6:

```
<!DOCTYPE html>
<html>
<head>
<title>Position</title>
<style>
.home {
```

```

        border: solid 1px black;
        width: 350px;
        height: 200px;
        margin: auto;
        background-color: #f6f6f6;
    }
    .position1 {
        width: 50px;
        height: 50px;
        background-color: #ff0000;
    }
    .position2 {
        width: 50px;
        height: 50px;
        background-color: #008000;
    }
</style>
</head>
<body>
    <div class='home'>
        <div class='position1'></div>
        <div class='position2'></div>
    </div>
</body>
</html>

```

მიღებული შედეგი იქნება:



მოცემულ კოდში მოვახდინოთ ჩასწორება.

მაგალითი 12.7:

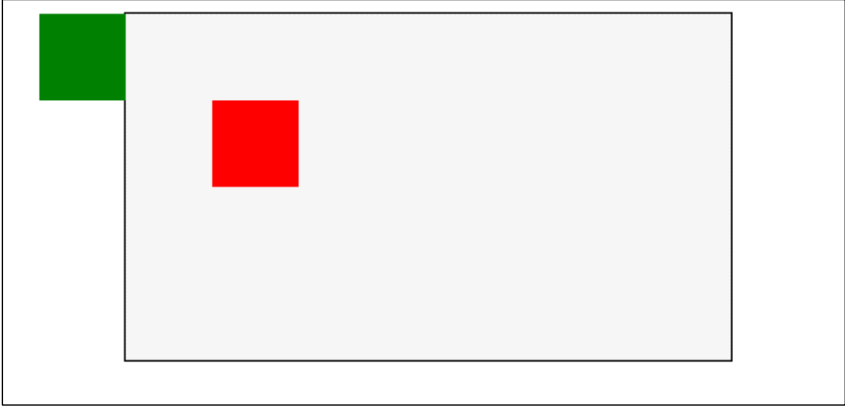
```
<!DOCTYPE html>
<html>
<head>
<title>Position</title>
<style>
  .home {
    border: solid 1px black;
    width: 350px;
    height: 200px;
    margin: auto;
    background-color: #f6f6f6;  }
  .position1 {
    width: 50px;
    height: 50px;
    background-color: #ff0000;
    position: relative;
```

```

        top:50px;
        left:50px;    }
.position2 {
    width: 50px;
    height: 50px;
    background-color: #008000;
    position: relative;
    right:50px;
    bottom:50px; }
</style>
</head>
<body>
    <div class='home'>
        <div class='position1'></div>
        <div class='position2'></div>
    </div>
</body>
</html>

```

ჩასწორების შედეგად მივიღებთ:



კოდში შეტანილი ჩასწორებები შემდეგ ცვლილებებს გამოიწვევს:

- `top:50px;` გადაადგილებს ობიექტს საწყისი პოზიციიდან ზემოდან ქვემოთ 50 პიქსელით.
- `bottom:50px;` გადაადგილებს ობიექტს საწყისი პოზიციიდან ქვემოთ ზემოთ 50 პიქსელით.
- `left:50px;` გადაადგილებს ობიექტს საწყისი პოზიციიდან მარცხნიდან მარჯვნივ 50 პიქსელით.
- `right:50px;` გადაადგილებს ობიექტს საწყისი პოზიციიდან მარჯვნიდან მარცხნივ 50 პიქსელით.

მაგალითი 12.8:

```
<!DOCTYPE HTML>
<html>
<head>
<title>ფარდობითი პოზიციონირება</title>
<style>
  .div1{
```

```

        width:500px;
        height:350px;
        background-color:#CCC;
        margin:auto;}
    .div2{
        width:250px;
        height:150px;
        background-color:#F00;
        top:20px;;
        right:-30px;
        position:relative;}
    .div3{
        width:300px;
        height:250px;
        background-color:#00C;
        top:5px;
        right:30px;
        position:absolute;}
    .img1{
        border:2px solid #F00;
        position:relative;
        left:200px;}
</style>
</head>
<body>
<div class="div3">

```

Conventionally, a computer consists of at least one processing element, typically a central processing unit (CPU), and some form of

memory. The processing element carries out arithmetic and logic operations, and a sequencing and control unit can change the order of operations in response to stored information. Peripheral devices allow information to be retrieved from an external source, and the result of operations saved and retrieved.

</div>

<div class="div1">

A personal computer is a general-purpose computer whose size, capabilities and original sale price make it useful for individuals, and is intended to be operated directly by an end-user with no intervening computer operator. This contrasts with the batch processing or time-sharing models that allowed larger, more expensive minicomputer and mainframe systems to be used by many people, usually at the same time. A related term is "PC" that was initially an acronym for "personal computer", but later became used primarily to refer to the ubiquitous Wintel platform.

<div class="div2">

A computer is a general-purpose device that can be programmed to carry out a set of arithmetic or logical operations automatically. Since a sequence of operations can be readily changed, the computer can solve more than one kind of problem.

</div>

<p>

Software applications for most personal computers include, but are not limited to, word processing, spreadsheets, databases, web browsers and e-mail clients, digital media playback, games and myriad personal productivity and special-purpose software applications. Modern personal computers often have connections to the Internet,

allowing access to the World Wide Web and a wide range of other resources. Personal computers may be connected to a local area network (LAN), either by a cable or a wireless connection. A personal computer may be a desktop computer or a laptop, netbook, tablet or a handheld PC.

```

```

```
<p>
```

Early computer owners usually had to write their own programs to do anything useful with the machines, which even did not include an operating system. The very earliest microcomputers, equipped with a front panel, required hand-loading of a bootstrap program to load programs from external storage (paper tape, cassettes, or eventually diskettes). Before very long, automatic booting from permanent read-only memory became universal. Today's users have access to a wide range of commercial software, freeware and free and open-source software, which are provided in ready-to-run or ready-to-compile form. Software for personal computers, such as applications and video games, are typically developed and distributed independently from the hardware or OS manufacturers, whereas software for many mobile phones and other portable systems is approved and distributed through a centralized online store.

```
</p>
```

```
<p>
```

Since the early 1990s, Microsoft operating systems and Intel hardware have dominated much of the personal computer market, first with MS-DOS and then with Windows. Popular alternatives to Microsoft's Windows operating systems include Apple's OS X and free


open-source Unix-like operating systems such as Linux and BSD. AMD provides the major alternative to Intel's processors.

</p>
</body>
</html>

A personal computer is a general-purpose computer whose size, capabilities and original sale price make it useful for individuals, and is intended to be operated directly by an end-user with no intervening computer operator. This contrasts with the batch processing or time-sharing models that allowed larger, more expensive main-computer and mainframe systems to be used by many people, usually at the same time. A related term is "PC" that was initially an acronym for "personal computer", but later became used primarily to refer to the ubiquitous Wintel platform.

A personal computer is a general-purpose device that can be programmed to carry out a set of arithmetic or logical operations automatically. Since a sequence of operations can be readily changed, the computer can solve many different kinds of problems without any special hardware.

Software applications for most personal computers include, but are not limited to, word processing, spreadsheets, databases, web browsers and e-mail clients, digital media playback, games and myriad personal productivity and special-purpose software applications. Modern personal computers often have connections to the Internet, allowing access to the World Wide Web and a wide range of other resources. Personal computers may be connected to a local area network (LAN), either by a cable or a wireless connection. A personal computer may be a desktop computer or a laptop, netbook, tablet or a handheld PC.



Early computer owners usually had to write their own programs to do anything useful with the machines, which even did not include an operating system. The very earliest microcomputers, equipped with a front panel, required hand-loading of a bootstrap program to load programs from external storage (paper tape, cassettes, or eventually diskettes). Before very long, automatic booting from permanent read-only memory became universal. Today's users have access to a wide range of commercial software, firmware and free and open-source software, which are provided in ready-to-run or ready-to-compile form. Software for personal computers, such as applications and video games, are typically developed and distributed independently from the hardware or OS manufacturers, whereas software for many mobile phones and other portable systems is approved and distributed through a centralized online store.

Since the early 1990s, Microsoft operating systems and Intel hardware have dominated much of the personal computer market, first with MS-DOS and then with Windows. Popular alternatives to Microsoft's Windows operating systems include Apple's OS X and free open-source Unix-like operating systems such as Linux and BSD. AMD provides the major alternative to Intel's processors.

ფარდობითი პოზიციონირებისას ახალი პოზიცია ორიგინალი გამოსახულების ზედა მარცხენა კუთხიდან აითვლება.

ფენების შექმნა Z-ინდექსის საშუალებით

განვიხილოთ შრეების (layers) შექმნის საკითხი. ეს ნიშნავს, რომ ელემენტები ისე უნდა დავალაგოთ, რომ ერთმანეთი გადაფაროს. ამისათვის, თითოეულ ელემენტს უნდა მივანიჭოთ ნომერი (z-index). ელემენტი უფრო მეტი ნომრით ფარავს უფრო ნაკლები ნომრის მქონე ელემენტს.

მაგალითი 12.9:

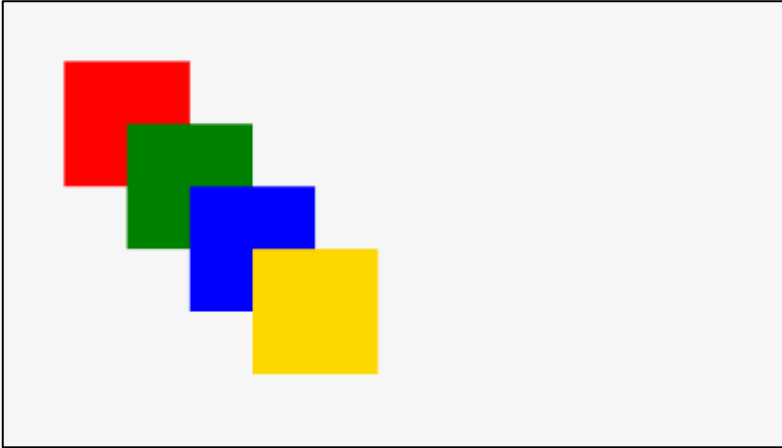
```
<!DOCTYPE html>
<html>
<head>
  <title>Z-Index</title>
<style>
  .home {
    border: solid 1px black;
    width: 450px;
    height: 250px;
    margin: auto;
    padding: 40px;
    background-color: #f6f6f6;  }
  .zindex1 {
    width: 80px;
    height: 80px;
    background-color: #ff0000;
    position: relative;  }
  .zindex2 {
    width: 80px;
    height: 80px;
    background-color: #008000;
    position: relative;
    top: -40px;
    left: 40px;  }
  .zindex3 {
    width: 80px;
    height: 80px;
    background-color: #0000ff;
```

```

        position: relative;
        top:-80px;
        left:80px;    }
.zindex4 {
    width: 80px;
    height: 80px;
    background-color: #ffd700;
    position: relative;
    top:-120px;
    left:120px;    }
</style>
</head>
<body>
    <div class='home'>
        <div class='zindex1'></div>
        <div class='zindex2'></div>
        <div class='zindex3'></div>
        <div class='zindex4'></div>
    </div>
</body>
</html>

```

შედეგად მიიღება:



შემოვიტანოთ z-index თვისებები და ჩავასწოროთ კოდი ისე, როგორც ქვემოთ მოცემულ მაგალითშია განხილული.

მაგალითი 12.10:

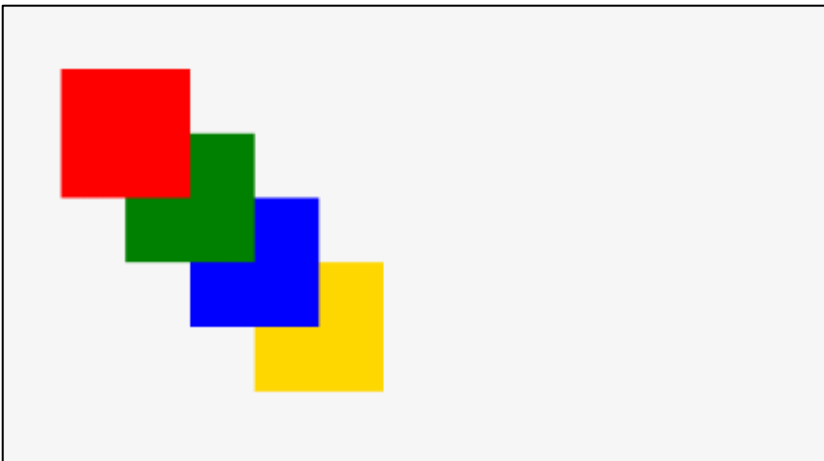
```
<!DOCTYPE html>
<html>
<head>
  <title>Z-Index</title>
<style>
  .home{
    border: solid 1px black;
    width: 450px;
    height: 250px;
    margin: auto;
    padding: 40px;
    background-color: #f6f6f6;
  }
  .zindex1{
```



```
        width: 80px;
        height: 80px;
        background-color: #ff0000;
        position: relative;
        z-index: 4;
    }
    .zindex2{
        width: 80px;
        height: 80px;
        background-color: #008000;
        position: relative;
        top:-40px;
        left: 40px;
        z-index: 3;
    }
    .zindex3{
        width: 80px;
        height: 80px;
        background-color: #0000ff;
        position: relative;
        top:-80px;
        left:80px;
        z-index: 2;
    }
    .zindex4{
        width: 80px;
        height: 80px;
        background-color: #ffd700;
```

```
        position: relative;
        top:-120px;
        left:120px;
        z-index: 1;
    }
</style>
</head>
<body>
    <div class='home'>
        <div class='zindex1'></div>
        <div class='zindex2'></div>
        <div class='zindex3'></div>
        <div class='zindex4'></div>
    </div>
</body>
</html>
```

ჩასწორების შემდეგ ფენები შემდეგნაირად დალაგდება:



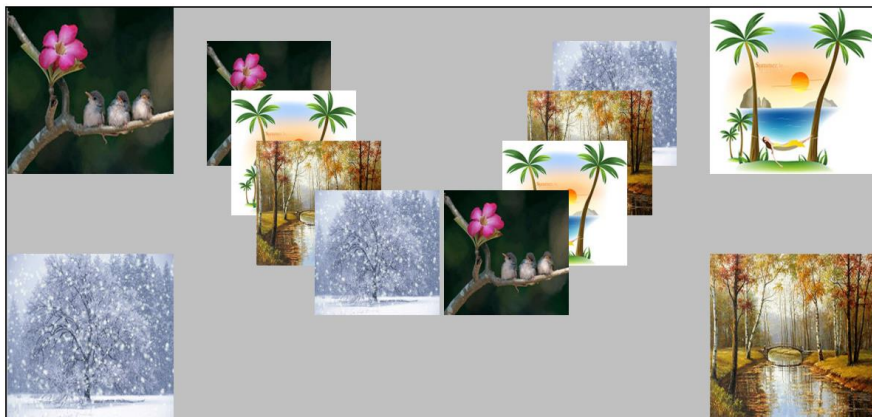
დავალება. გამოიყენეთ position და z-index თვისებები და ააგეთ ქვემოთ მოცემული ვებგვერდი შესაბამისი წყობით. ვებგვერდი გააფორმეთ შემდეგნაირად:

✓ კონტეინერის ფონი ვერცხლისფერია, საზღვრები 2px - შავი ფერის, განთავსებულია ბრაუზერის ცენტრში, ზომები 1050x500;

✓ კონტეინერის კუთხეებში მდებარე ფოტოების ზომებია 200x200;

✓ შიგა ფოტოების ზომებია 150x150.

position, z-index



თავი 13

ანიმაცია CSS3-ში

- CSS-ში ანიმაცია;
- დაყოვნება ანიმაციაში;
- ანიმაციის შესრულების მიმართულება;
- ანიმაციის შესრულების სიჩქარე;
- ანიმაციის შესრულების შემოკლებული ჩანაწერი.

ანიმაცია CSS3-ში

CSS3-ში ანიმაციის შესაქმნელად @keyframes თვისება გამოიყენება. ეს თვისება კონტეინერს წარმოადგენს, რომელშიც დოკუმენტის გაფორმებისათვის საჭირო სხვადასხვა თვისება უნდა მოთავსდეს.

მას შემდეგ, რაც ანიმაცია შეიქმნება, აუცილებელია შეიქმნას ელემენტი, რომლის ანიმაციაც გვინდა. ამ ელემენტისთვის უნდა მივუთითოთ ანიმაციის სახელი (animation-name) და დრო (animation-duration), რომლის განმავლობაშიც ეს ანიმაცია შესრულდება.

ქვემოთ მოცემული კოდი საშუალებას იძლევა ბლოკის ფონის ფერი მასში მითითებული მნიშვნელობების მიხედვით შეიცვალოს.

მაგალითი 13.1:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example
{
```

```

0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<p>
<b>შენიშვნა:
</b>
ანიმაციის დამთავრების შემდეგ იგი კვლავ საწყის
მდგომარეობას უბრუნდება.
</p>
<div></div>
</body>
</html>

```

შენიშვნა: ანიმაციის დამთავრების შემდეგ იგი კვლავ საწყის მდგომარეობას უბრუნდება.



ასევე, შესაძლებელია ელემენტისთვის შეიცვალოს როგორც ფონის ფერი, ისე მისი მდებარეობა.

მაგალითი 13.2:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```


დაყოვნება ანიმაციაში

animation-delay თვისება ანიმაციის მითითებული დროით დაყოვნებას იწვევს.

animation-iteration-count თვისება განსაზღვრავს, თუ რამდენჯერ უნდა შესრულდეს ანიმაცია. თუ ამ თვისებაში რიცხვის მაგივრად მითითებული იქნება „infinite“, მაშინ ანიმაცია უსასრულოდ გამეორდება.

მაგალითი 13.3:

```
<!DOCTYPE html>
<html>
<head>
<title>ანიმაცია</title>
<style>
@keyframes anim
{
from {margin-left: 3px;}
to {margin-left: 500px;}
}
#wrap1{
border:2px #000 solid;
background-color:#7F0055;
height:100px;
width:100px;
animation-name: anim;
animation-duration: 4s;
animation-iteration-count: 3;
}
```

```
</style>
</head>
<body>
<div id="wrap1"></div>
<p><b>ეს ანიმაცია მეორდება სამჯერ</b></p>
</body>
</html>
```



ეს ანიმაცია მეორდება სამჯერ

ანიმაციის შესრულების მიმდინარეობა არა მხოლოდ from და to სიტყვების მეშვეობით, არამედ %-ითაც შეიძლება განისაზღვროს.

%-ის მეშვეობით შეგიძლიათ ანიმაციის მიმდინარეობა უფრო ზუსტად გააკონტროლოთ.

მაგალითი 13.4:

```
<!DOCTYPE html>
<html>
<head>
<title>ანიმაცია - 2</title>
<style>
@keyframes anim {
```

```

    0% {margin-left:3px;margin-top:3px;background-color:#
f09900;}
    30% {margin-left:3px;margin-top:250px;background-
color:#ffc608;}
    60% {margin-left:500px;margin-top:250px;background-
color:black;}
    100% {margin-left:3px;margin-top:3px;background-color:#
f09900;}
}
#wrap1 {
border:2px #000 solid;
background-color:# f09900;
height:100px;
width:100px;
animation:anim 6s 3;
-webkit-animation:anim 6s 3;
}
</style>
</head>
<body>
<div id="wrap1"></div>
</body>
</html>

```

ანიმაციის შესრულების მიმართულება

ანიმაციის შესრულების მიმართულებას animation-direction თვისება განსაზღვრავს.

მაგალითი 13.5:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
  animation-direction: reverse;
}
@keyframes example {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

შემდეგ მაგალითში გამოყენებულია თვისება alternate, რომლის მიხედვითაც ანიმაცია ჯერ ერთი მიმართულებით მოძრაობს, შემდეგ - უკუმიმართულებით, შემდეგ ისევ საწყისი მიმართულებით და ა. შ.

მაგალითი 13.6:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 4;
  animation-direction: alternate; }
@keyframes example {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;} }
</style>
</head>
<body>
```

```
<div></div>
</body>
</html>
```

ანიმაციის შესრულების სიჩქარე

animation-timing-function თვისება ანიმაციის შესრულების სიჩქარეს განსაზღვრავს.

animation-timing-function თვისებას შემდეგი მნიშვნელობების მიღება შეუძლია:

- ease - ანიმაციის ეს ეფექტი განსაზღვრავს მის შესრულებას ნელი სტარტიდან უფრო სწრაფ, საბოლოოდ ისევ ნელ მოქმედებაზე (ეს ეფექტი არსებობს ნაგულისხმევი წესის თანახმად);
- linear - ანიმაციის ეს ეფექტი განსაზღვრავს მის შესრულებას ერთი და იგივე სიჩქარით თავიდან ბოლომდე;
- ease-in - ანიმაციის ეს ეფექტი განსაზღვრავს მის შესრულებას ნელი სტარტით;
- ease-out - ანიმაციის ეს ეფექტი განსაზღვრავს მის შესრულებას ნელი დასასრულით;
- ease-in-out - ანიმაციის ეს ეფექტი განსაზღვრავს მის შესრულებას ნელი სტარტით და დასასრულით.

მაგალითი 13.7:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
```

```
width: 100px;
height: 50px;
background-color: red;
font-weight: bold;
position: relative;
  animation: mymove 5s infinite;      }
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
@keyframes mymove {
  from {left: 0px;}
  to {left: 300px;}
}
</style>
</head>
<body>
<div id="div1">linear</div>
<div id="div2">ease</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">ease-in-out</div>
</body>
</html>
```

ანიმაციის შესრულების შემოკლებული ჩანაწერი

დავუშვათ, ანიმაციის შესრულების დროს შემდეგი 6 თვისების გათვალისწინება გვსურს:

```
div {  
  animation-name: example;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

ანიმაციის იგივე ეფექტის შესრულება შემდეგი შემოკლებული ჩანაწერთაც მიიღწევა:

```
div {  
  animation: example 5s linear 2s infinite alternate;  
}
```


დავალება. გამოიყენეთ animation თვისება და ააგეთ div ტეგების ანიმაციები. ვებგვერდის გასაფორმებლად გამოიყენეთ შემდეგი პირობები:

✓ პირველი პატარა div ტეგის ფონი მწვანეა, საზღვრები - წითელი, ზომები 50x50;

✓ მეორე პატარა div ტეგის ფონი ყვითელია, საზღვრები - ლურჯი, ზომები 50x50;

✓ მესამე პატარა div ტეგის ფონი ლურჯია, საზღვრები - წითელი, ზომები 50x50;

✓ კონტეინერის ფონი ვერცხლისფერია, განთავსებულია ბრაუზერის ცენტრში, ზომები 700x300;

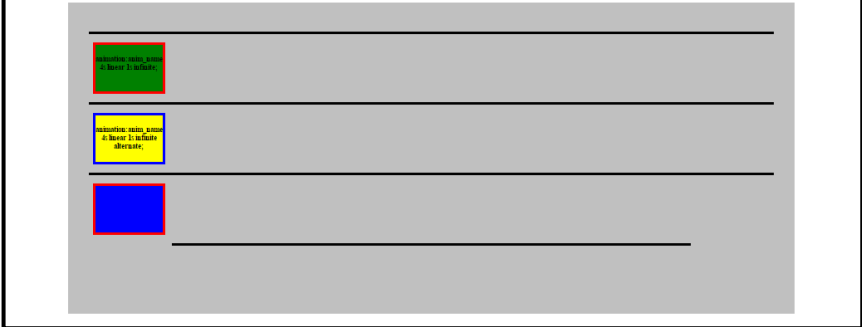
✓ პირველი კვადრატის ანიმაცია: div ტეგის სიგანე იზრდება ხაზის ბოლომდე, ანიმაცია მიმდინარეობს ერთი მიმართულებით, უწყვეტად;

✓ მეორე კვადრატის ანიმაცია: div ტეგის სიგანე იზრდება ხაზის ბოლომდე და ისევ მცირდება, ანიმაცია მიმდინარეობს ორმხრივი მიმართულებით, უწყვეტად;

✓ მესამე კვადრატის ანიმაცია: კვადრატი მოძრაობს მეოთხე ხაზის გარშემო, ანიმაცია მიმდინარეობს ორმხრივი მიმართულებით, უწყვეტად.

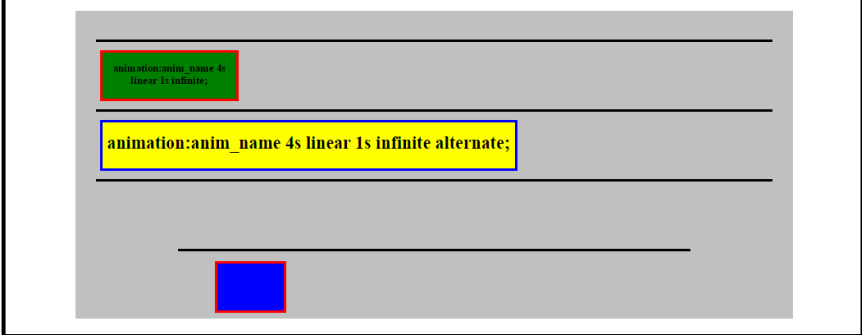
ქვემოთ სურათზე მოცემულია ანიმაციის საწყისი წერტილი:

animation: anim_name durement timing start direction delay



აქ კი ანიმაციის მიმდინარეობა:

animation: anim_name durement timing start direction delay



თავი 14

CSS ტრანსფორმაციები

- 2D ტრანსფორმაციები;
- 3D ტრანსფორმაციები.

CSS ტრანსფორმაციები

CSS-ში 2D და 3D ტრანსფორმაციების განხორციელება შესაძლებელია.

2D ტრანსფორმაციები

`translate(x,y)` ფუნქციის მეშვეობით შეგიძლიათ ელემენტი პიქსელების მითითებული რაოდენობით ჰორიზონტალისა და ვერტიკალის გასწვრივ გადაანაცვლოთ.

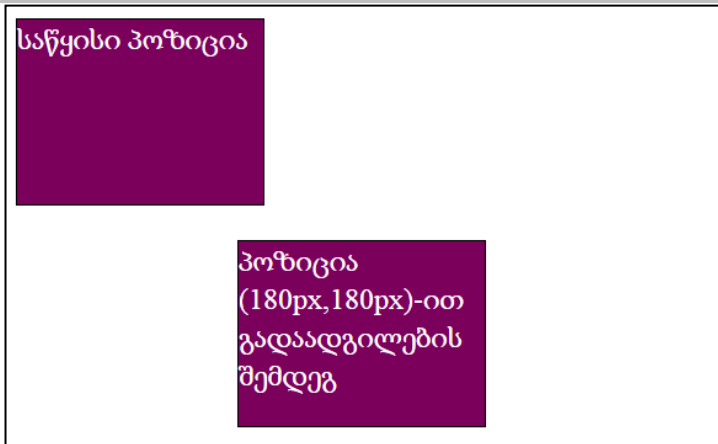
მაგალითი 14.1:

```
<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>
<style>
#e1,#e2 {
position:absolute;
top:10px;
left:10px;
background-color:#7A005C;
color:white;
width:200px;
height:150px;
font-size:1.5em;
border:1px #000 solid;
}
#e2 {
transform: translate(180px,180px);
}
```

```

</style>
</head>
<body>
<div id='el1'>
საწყისი პოზიცია
</div>
<div id='el2'>პოზიცია (180px,180px)-ით გადაადგილების
შემდეგ</div>
</body>
</html>

```



rotate ფუნქციის დახმარებით შეგიძლიათ მოაბრუნოთ ელემენტი გრადუსების მითითებული რაოდენობით საათის ისრის მოძრაობის მიმართულებით.

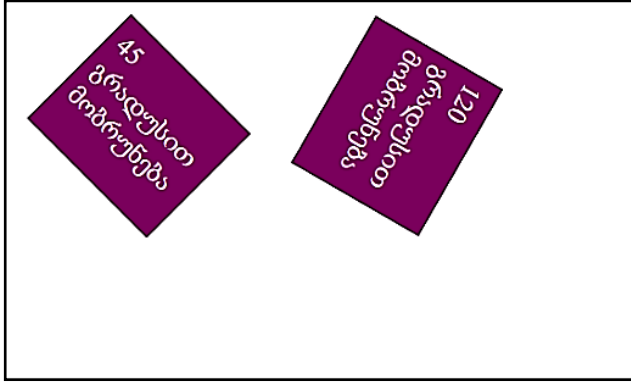
მაგალითი 14.2:

```

<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>

```

```
<style>
#e13,#e14 {
margin:40px;
background-color:#7A005C;
width:130px;
height:110px;
font-size:1.5em;
border:1px #000 solid;
color:white;
padding:10px;
float:left;
}
#e13 {
transform: rotate(45deg);
}
#e14 {
transform: rotate(120deg);
}
</style>
</head>
<body>
<div id='e13'>45 გრადუსით მობრუნება</div>
<div id='e14'>120 გრადუსით მობრუნება </div>
</body>
</html>
```



scale (x, y) მეთოდის საშუალებით ელემენტი შეიძლება გაიწელოს სიგანესა და სიმაღლეში.

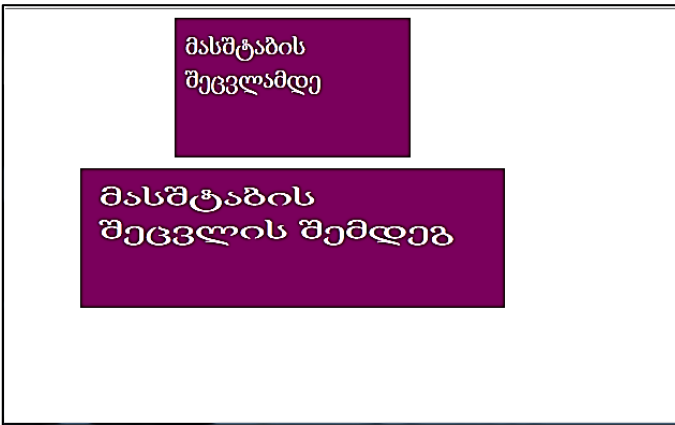
მაგალითი 14.3:

```
<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>
<style>
#e15,#e16 {
margin-left:150px;
margin-bottom:10px;
background-color:#7A005C;
width:200px;
height:110px;
font-size:1.5em;
color:white;
padding:10px;
}
```

```

#e16 {
transform:scale(1.8,1);
</style>
</head>
<body>
<div id='e15'>მასშტაბის შეცვლამდე </div>
<div id='e16'>მასშტაბის შეცვლის შემდეგ </div>
</body>
</html>

```



skewX(x) მეთოდის მეშვეობით შეგიძლიათ დახაროთ ელემენტი გრადუსების მითითებული რაოდენობით X ღერძის მიმართ.

მაგალითი 14.4:

```

<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>
<style>

```

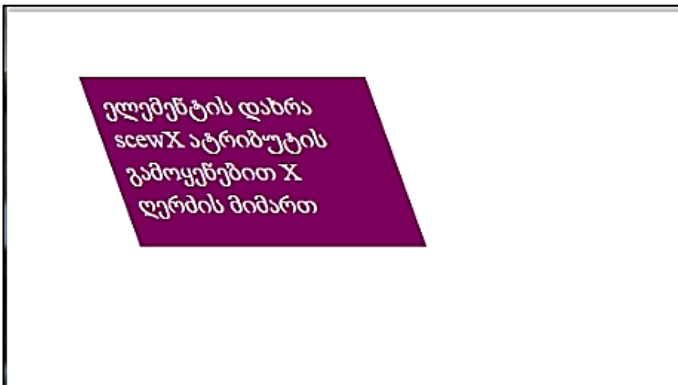


```

#el8 {
margin-left:70px;
margin-top:50px;
background-color:#7A005C;
width:200px;
height:110px;
font-size:1.2em;
color:white;
padding:10px;
transform:skewX(20deg);
</style>
</head>
<body>
<div id='el8'>ელემენტის დახრა scwX ატრიბუტის
გამოყენებით X ღერძის მიმართ </div>
</body>
</html>

```

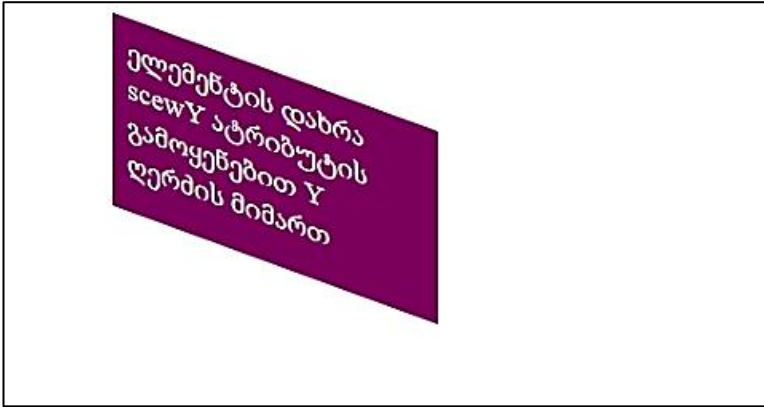
ელემენტის დახრა scwX ატრიბუტის გამოყენებით X ღერძის მიმართ



skewY(x) მეთოდის მეშვეობით შეგიძლიათ დახაროთ ელემენტი გრადუსების მითითებული რაოდენობით Y ღერძის მიმართ.

მაგალითი 14.5:

```
<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>
<style>
#el9 {
margin-left:70px;
margin-top:50px;
background-color:#7A005C;
width:200px;
height:110px;
font-size:1.2em;
color:white;
padding:10px;
transform:skewY(20deg);
</style>
</head>
<body>
<div id='el9'>ელემენტის დახრა scewY ატრიბუტის
გამოყენებით Y ღერძის მიმართ </div>
</body>
</html>
```

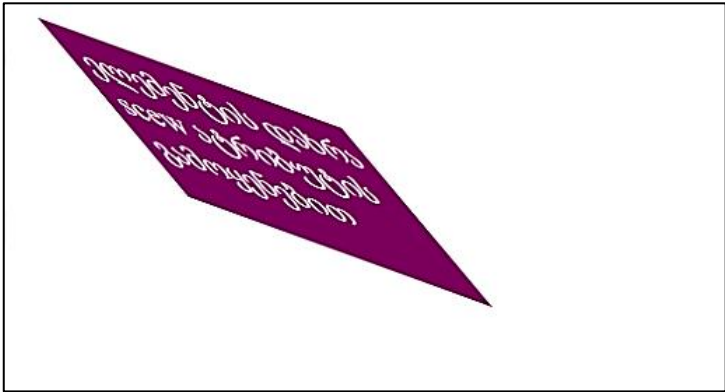


`skew(x,y)` მეთოდის მეშვეობით შეგიძლიათ დახაროთ ელემენტი გრადუსების მითითებული რაოდენობით X და Y ღერძის მიმართ ერთდროულად.

მაგალითი 14.6:

```
<!DOCTYPE html>
<head>
<title>CSS გარდაქმნები</title>
<style>
#e110 {
margin-left:70px;
margin-top:50px;
background-color:#7A005C;
width:200px;
height:110px;
font-size:1.5em;
color:white;
padding:10px;
transform:skew(40deg,20deg);
```

```
</style>
</head>
<body>
  <div id='el10'>ელემენტის დახრა scew ატრიბუტის
გამოყენებით </div>
</body>
</html>
```



matrix() მეთოდი 2D ტრანსფორმაციების ყველა მეთოდის კომბინაციაა. მას ექვსი პარამეტრი აქვს. პირველი პარამეტრი არის ჰორიზონტალურად გაზრდის კოეფიციენტი, მეორე - X ღერძის მიმართ დახრის კოეფიციენტი, მესამე - Y ღერძის მიმართ დახრის კოეფიციენტი, მეოთხე - ვერტიკალურად გაზრდის კოეფიციენტი, მეხუთე - პიქსელების მითითებული რაოდენობით გადაადგილება X ღერძის მიმართ, ხოლო მეექვსე - გადაადგილება Y ღერძის მიმართ.

მაგალითი 14.7:

```
<!DOCTYPE html>
<head>
```

```

<title>CSS გარდაქმნები</title>
<style>
div
{
margin-left:150px;
margin-bottom:10px;
background-color:#7A005C;
border: 5px solid black;
width:200px;
height:110px;
font-size:1.3em;
color:white;
padding:10px;
}
div#myDiv1
{
    -webkit-transform: matrix(1, -0.3, 0, 1, 0, 30); /* Chrome-
ობათვის */
    -ms-transform: matrix(1, -0.3, 0, 1, 0, 30); /* IE-ობათვის */
}
div#myDiv2 {
    -webkit-transform: matrix(1, 0, 0.5, 1, 220, 0); /* Chrome-
ობათვის */
    -ms-transform: matrix(1, 0, 0.5, 1, 220, 0); /* IE-ობათვის */
}
</style>
</head>
<body>

```

```
<h3>
```

matrix() მეთოდი 2D ტრანსფორმაციების ყველა მეთოდის ერთ მეთოდში გაერთიანებაა.

```
</h3>
```

```
<div>
```

ეს არის საწყისი ელემენტი.

```
</div>
```

```
<div id="myDiv1">
```

matrix() მეთოდის გამოყენების შემდეგ.

```
</div>
```

```
<div id="myDiv2">
```

matrix() მეთოდის სხვა პარამეტრების გამოყენების შემდეგ.

```
</div>
```

```
</body>
```

```
</html>
```

matrix() მეთოდი 2D ტრანსფორმაციების ყველა მეთოდის ერთ მეთოდში გაერთიანება.

ეს არის საწყისი ელემენტი.

matrix() მეთოდის გამოყენების შემდეგ.

matrix() მეთოდის სხვა პარამეტრების გამოყენების შემდეგ.

3D ტრანსფორმაციები

rotateX () მეთოდის საშუალებით ხდება ელემენტის მობრუნება მისი X ღერძის გარშემო მოცემული კუთხით.

მაგალითი 14.8:

```
<!DOCTYPE html>
<head>
<title>CSS transform</title>
<style>
div{
width:100px;
height:175px;
```

```
background-color:lightgreen;
border:1px solid black;}
div#div2{
transform: rotateX(120deg);
```

```
</style>
```

```
</head>
```

```
<body>
```

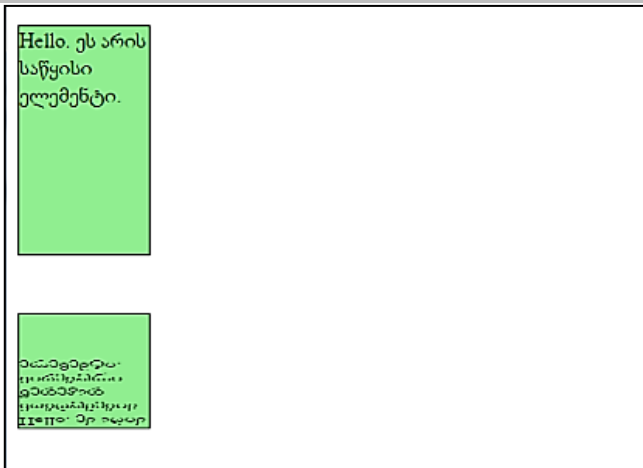
```
<div>Hello. ეს არის საწყისი ელემენტი.</div>
```

<div id="div2">Hello. ეს არის მობრუნების შედეგად მიღებული ელემენტი.

```
</div>
```

```
</body>
```

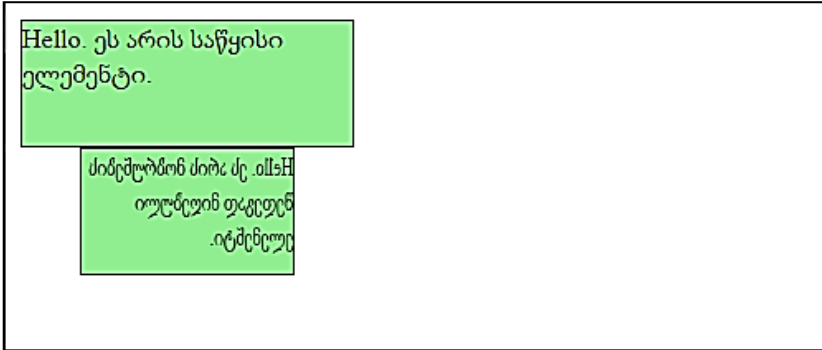
```
</html>
```



rotateY () მეთოდის საშუალებით ხდება ელემენტის მობრუნება მისი Y ღერძის გარშემო მოცემული კუთხით.

მაგალითი 14.9:

```
<!DOCTYPE html>
<head>
<title>CSS transform</title>
<style>
div {
width:200px;
height:75px;
background-color:lightgreen;
border:1px solid black; }
div#div2
{
transform:rotateY(130deg);
}
</style>
</head>
<body>
<div>Hello. ეს არის საწყისი ელემენტი.</div>
<div id="div2">Hello. ეს არის მოზრუნების შედეგად
მიღებული ელემენტი.
</div>
</body>
</html>
```



rotateZ () მეთოდის საშუალებით ხდება ელემენტის მობრუნება Z ღერძის გარშემო მოცემული კუთხით.

მაგალითი 14.10:

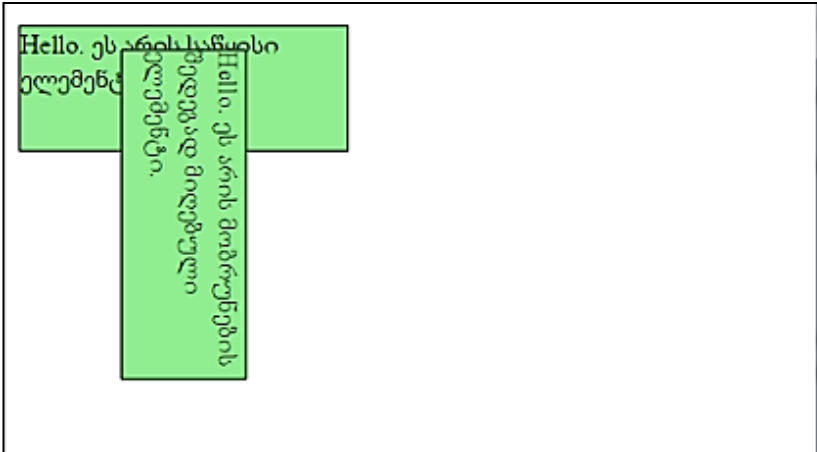
```

<!DOCTYPE html>
<head>
<title>CSS transform</title>
<style>
div {width:200px;
      height:75px;
      background-color:lightgreen;
      border:1px solid black; }
div#div2 {
transform:rotateZ(90deg);
</style>
</head>
<body>
<div>Hello. ეს არის საწყისი ელემენტი.</div>
<div id="div2">Hello. ეს არის მობრუნების შედეგად
მიღებული ელემენტი.

```

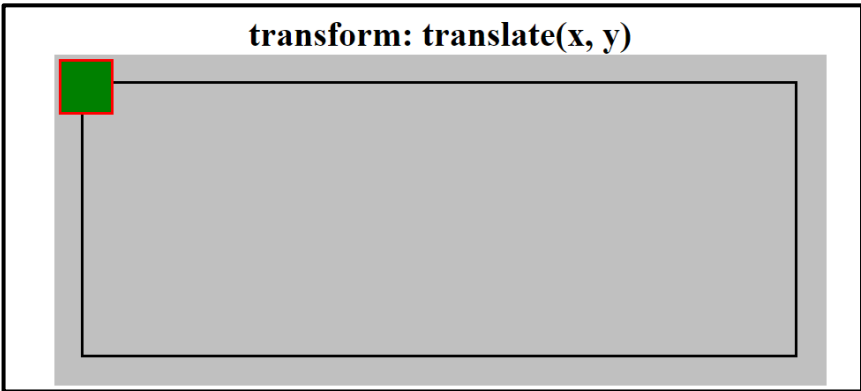
```
</div>  
</body>  
</html>
```

Z ღერძის გარშემო 90° მობრუნებით მივიღებთ:

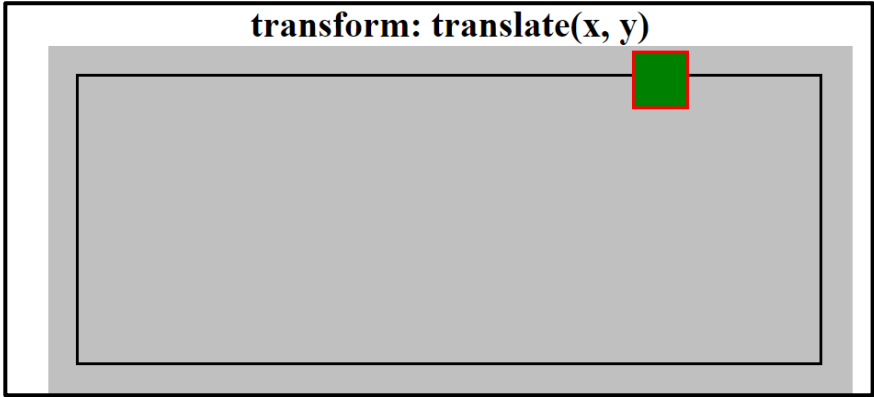


დავალება 1. გამოიყენეთ transform თვისების translate(x, y) მნიშვნელობა და ააგეთ div ტეგის გადაადგილების ანიმაცია მართკუთხედზე. ვებგვერდის გაფორმებისათვის შემდეგი პარამეტრები გამოიყენეთ:

- ✓ პატარა div ტეგების ფონი მწვანეა, საზღვრები - წითელი, ზომები 50x50;
- ✓ კონტეინერის ფონი ვერცხლისფერია, განთავსებულია ბრაუზერის ცენტრში, ზომები 600x250;
- ✓ მართკუთხედი აგებულია div ტეგის საშუალებით, საზღვრებია შავი, საზღვრის სიგანე 2px;
- ✓ პატარა კვადრატი მიჰყვება მართკუთხედის საზღვრებს, ანიმაცია უწყვეტ პროცესად ერთი მიმართულებით მიმდინარეობს;
- ✓ ანიმაციის საწყისი წერტილი მოცემულია სურათზე,



ხოლო ანიმაციის მიმდინარეობა მოცემულია ქვემოთ.

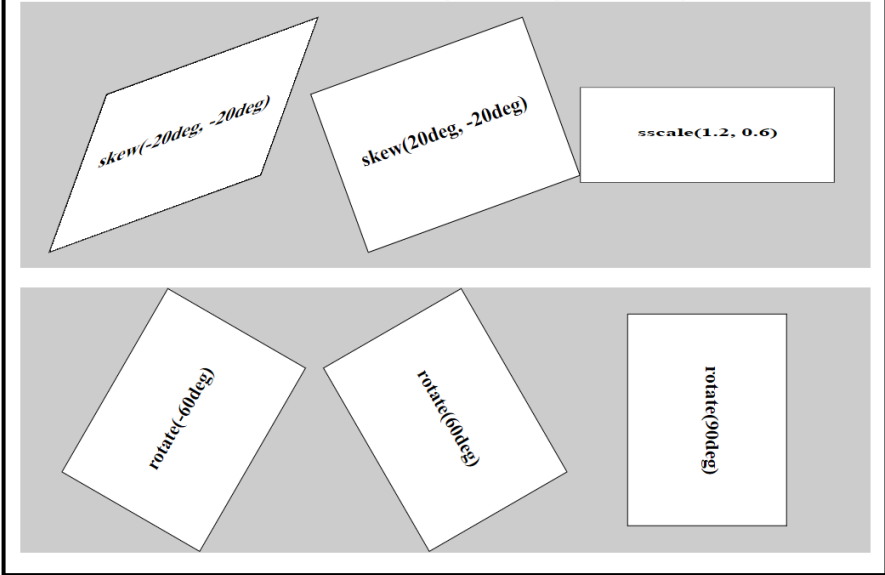


დავალება 2. გამოიყენეთ transform თვისების `stew()`, `scale()`, `rotate()` მნიშვნელობები და ააგეთ შესაბამისი სტრუქტურის `div` ტეგები.

ვებგვერდი შემდეგნაირად გააფორმეთ:

- ✓ `div` ტეგების ფონი თეთრია, საზღვრები - შავი, ზომები 200x150;
- ✓ კონტეინერების ფონი ვერცხლისფერია, განთავსებულია ბრაუზერის ცენტრში, ზომები 800x250;
- ✓ `transform` თვისებების მნიშვნელობები ჩაწერილია `div` ტეგებში.

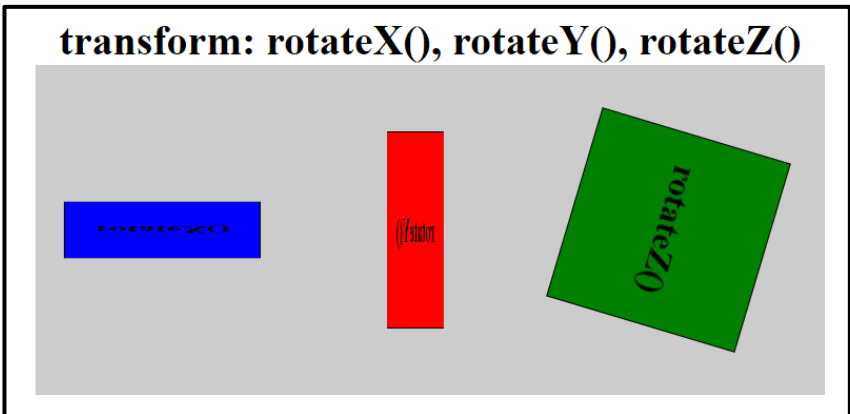
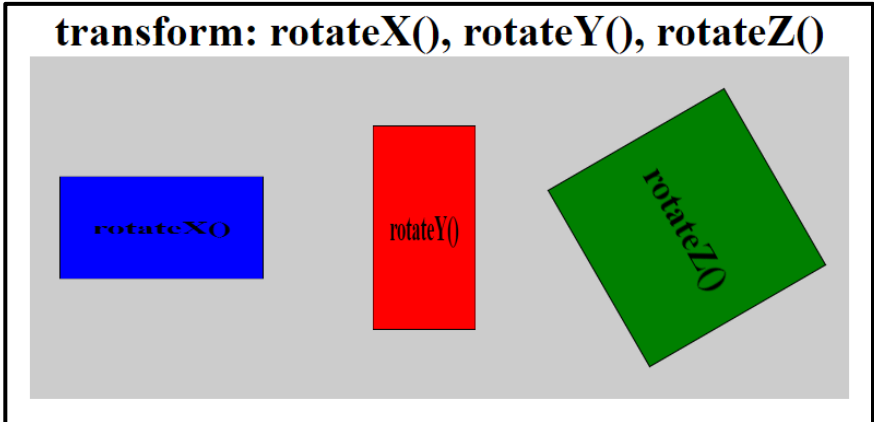
transform: skew(), scale(), rotate()



დავალეზა 3. გამოიყენეთ 3D transform თვისების rotateX(), rotateY(), rotateZ() მნიშვნელობები და ააგეთ შესაბამისი ანიმაციები div ტეგებისთვის. ვებგვერდი შემდეგნაირად გააფორმეთ:

- ✓ პირველი div ტეგის ფონი ლურჯია, ზომები - 150x150;
- ✓ მეორე div ტეგის ფონი წითელია, ზომები - 150x150;
- ✓ მესამე div ტეგის ფონი მწვანეა, ზომები - 150x150;
- ✓ კონტეინერის ფონი ვერცხლისფერია, განთავსებულია ბრაუზერის ცენტრში, ზომები 650x250;
- ✓ პირველი კვადრეტი ბრუნავს X ღერძის გარშემო, ანიმაცია უწყვეტ პროცესად ორმხრივი მიმართულებით მიმდინარეობს;

- ✓ მეორე კვადრატი ბრუნავს Y ღერძის გარშემო, ანიმაცია უწყვეტ პროცესად ორმხრივი მიმართულებით მიმდინარეობს;
- ✓ მესამე კვადრატი ბრუნავს Z ღერძის გარშემო, ანიმაცია უწყვეტ პროცესად ორმხრივი მიმართულებით მიმდინარეობს.
- ✓ ქვემოთ სურათებზე მოცემულია ანიმაციის საწყისი წერტილი და ანიმაციის მიმდინარეობა.



თავი 15

განლაგებები Flexbox

- Flexbox კონტეინერი;
- Flexbox კონტეინერში მოთავსებული ელემენტები.

Flexbox კონტეინერი

HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox ერთ-ერთი თანამედროვე მიდგომაა. იმისათვის, რომ გამოვიყენოთ Flexbox მოდელი, პირველ რიგში საჭიროა განისაზღვროს ე.წ. კონტეინერი, რისთვისაც display თვისებას მივანიჭოთ flex მნიშვნელობა.

მაგალითი 15.1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Flexbox</title>
  <style>
    .flex-container{
      display: flex;
      background-color: aqua;
      width: 400px;
      margin: auto;
    }

    .flex-container > div{
      margin: 10px;
      padding: 10px;
      background-color: azure;
      font-size: 35px;
    }
  </style>
```

```
</head>
<body>
  <div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
  </div>
</body>
</html>
```

ზემოთ მოყვანილი მაგალითიდან ჩანს, რომ გვაქვს ერთი კონტეინერი (ცისფერი არე) ხუთი ელემენტით. მივიღებთ შესაბამის შედეგს:



როგორც შედეგიდან ჩანს, div ტეგები - ელემენტები განთავსებულია კონტეინერში მარჯვნიდან მარცხნივ.

display თვისების flex მნიშვნელობის მქონე კონტეინერის შიგნით არსებული ელემენტებისათვის შეიძლება გამოყენებულ იქნეს რამდენიმე მნიშვნელოვანი თვისება.

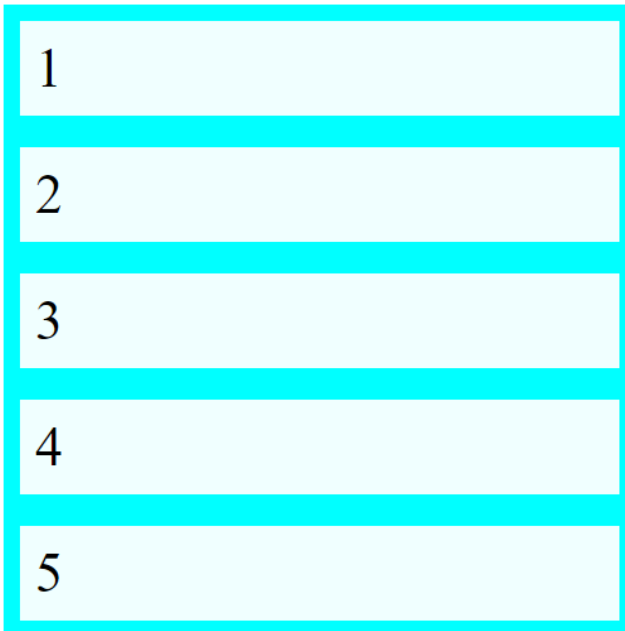
flex-direction თვისება განსაზღვრავს კონტეინერში არსებული ელემენტების მიმართულებას. მას აქვს ოთხი მნიშვნელობა: row - ელემენტები განლაგდება სტრიქონში (ეს მნიშვნელობა ჩუმათობის პრინციპითა იგულისხმება), row-reverse - სტრიქონში განლაგდება უკუმიმართულებით,

column - განლაგდება სვეტში, column-reverse - სვეტში განლაგდება უკუმიმართულებით.

მაგალითი 15.2. თუ 15.1 მაგალითში .flex-container სელექტორს შემდეგი სახით შევცვლით:

```
.flex-container{  
    display: flex;  
    flex-direction: column;  
    background-color: aqua;  
    width: 400px;  
    margin: auto;  
}
```

შესაბამისად, ელემენტები დალაგდება სვეტში და შემდეგ შედეგს მივიღებთ:

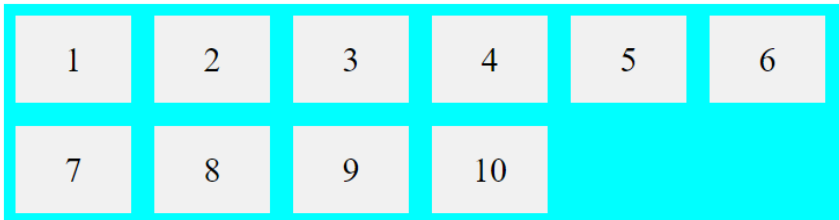


flex-wrap თვისება განსაზღვრავს მოხდება თუ არა კონტეინერში მყოფი ელემენტების წყვეტა. თუ მას wrap მნიშვნელობას მივანიჭებთ, მაშინ კონტეინერში მოთავსებული ელემენტები რამდენიმე ხაზზე გადანაწილდება.

მაგალითი 15.3.

```
<!DOCTYPE html>
<html>
<head>
<title>The flex-wrap Property</title>
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: aqua;
}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
```

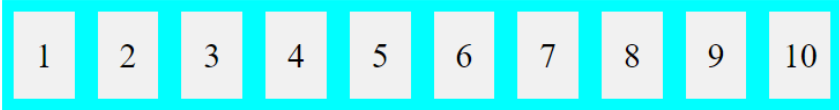
```
<div>1</div>
<div>2</div>
<div>3</div>
<div>4</div>
<div>5</div>
<div>6</div>
<div>7</div>
<div>8</div>
<div>9</div>
<div>10</div>
</div>
</body>
</html>
```



მაგალითი 15.4. თუ იგივე მაგალითში `.flex-container` სელექტორს შემდეგი სახით შევცვლით:

```
.flex-container {
  display: flex;
  flex-wrap: nowrap;
  background-color: aqua;
}
```

მივიღებთ:



თუ flex-wrap თვისებას wrap-reverse მნიშვნელობას მივანიჭებთ, მაშინ კონტეინერში მოთავსებული ელემენტები უკუმიმართულებით რამდენიმე ხაზზე გადანაწილდება.

flex-flow - თვისება აერთიანებს flex-direction და flex-wrap თვისებებს.

justify-content - განსაზღვრავს კონტეინერში მყოფი ელემენტების ჰორიზონტალურ სწორებას. მას აქვს შემდეგი მნიშვნელობები: center, flex-start, flex-end, space-between, space-around, space-evenly.

მაგალითი 15.5. თუ 15.1 მაგალითში .flex-container სელექტორს შემდეგი სახით შევცვლით:

```
.flex-container{
  display: flex;
  background-color: aqua;
  width: 400px;
  margin: auto;
  justify-content: space-evenly;
}
```

მაშინ კონტეინერში მყოფი ელემენტები ჰორიზონტალურად თანაბრად იქნება ერთმანეთისგან დაშორებული.

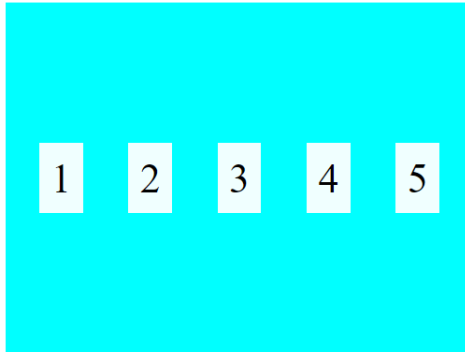


`align-items` - განსაზღვრავს კონტეინერში მყოფი ელემენტების ვერტიკალურ სწორებას. მას აქვს შემდეგი მნიშვნელობები: `center`, `flex-start`, `flex-end`, `baseline`, `stretch`.

მაგალითი 15.6. 15.5 მაგალითში `.flex-container` სელექტორში შემდეგი ცვლილება შევიტანოთ:

```
.flex-container{  
  display: flex;  
  background-color: aqua;  
  width: 400px;  
  height: 300px;  
  margin: auto;  
  justify-content: space-evenly;  
  align-items: center;  
}
```

ამ შემთხვევაში კონტეინერში მყოფი ელემენტები განთავსდება კონტეინერის ცენტრში.



Flexbox კონტეინერში მოთავსებული ელემენტები

display თვისების flex მნიშვნელობის მქონე კონტეინერის შიგნით არსებულ ელემენტებს განსხვავებული თვისებები აქვს მინიჭებული.

განვიხილოთ order თვისება, რომელიც უზრუნველყოფს ელემენტების გადალაგებას.

მაგალითი 15.7:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Flexbox</title>
  <style>
    .flex-container{
      display: flex;
      background-color: aqua;
      width: 400px;
```

```

margin: auto;
justify-content: space-evenly;
align-items: center;
}

.flex-container > div{
margin: 10px;
padding: 10px;
background-color: azure;
font-size: 35px;
}
</style>
</head>
<body>
<div class="flex-container">
<div style="order:2;">1</div>
<div style="order:3;">2</div>
<div style="order:1">3</div>
</div>
</body>
</html>

```

შედეგად ვნახავთ რომ .flex-container-ში მოთავსებული ელემენტები order თვისების მნიშვნელობების შესაბამისად გადალაგდება.



საინტერესო თვისებაა flex-grow, რომელიც უზრუნველყოფს კონტეინერში არსებული ელემენტების მასშტაბს.

მაგალითი 15.8. ზემოთ მოყვანილი კოდის html ნაწილი შემდეგი სახით შევცვალოთ:

```
<div class="flex-container">  
  <div style="flex-grow:1;">1</div>  
  <div style="flex-grow:3;">2</div>  
  <div style="flex-grow:2;">3</div>  
</div>
```

მივიღებთ:



შედეგიდან ჩანს, რომ კონტეინერის სიგანე flex-grow თვისების მნიშვნელობების შესაბამისად დაყოფილია 6 ტოლ ნაწილად. აქედან პირველ ობიექტს უკავია 1, მეორეს - 3, ხოლო მესამეს - 2 ნაწილი.

flex-shrink თვისება გვიჩვენებს, ამ თვისების მქონე ელემენტი რამდენად უფრო შემჭიდროვდება სხვა ელემენტებთან შედარებით. მისი მნიშვნელობა არის რიცხვი, რომელიც ჩუმათობის პრინციპით 1-ის ტოლია.

მაგალითი 15.9. 15.7 მაგალითში კოდის ტანის ნაწილი შევცვალოთ შემდეგი კოდით:

```
<div class="flex-container">
<div>1</div>
<div>2</div>
<div style="flex-shrink: 0">3</div>
<div>4</div>
<div>5</div>
<div>6</div>
</div>
```

მივიღებთ

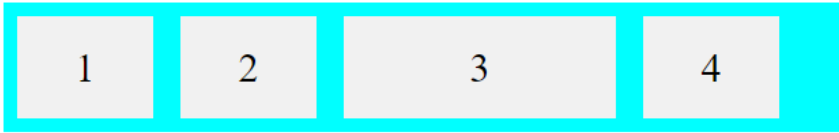


flex-base თვისება ელემენტის საწყის სიგრძეს გვიჩვენებს.

მაგალითი 15.10. მესამე ელემენტის საწყის სიგრძედ დავაყენოთ 200 პიქსელი:

```
<div class="flex-container">
<div>1</div>
<div>2</div>
<div style="flex-basis: 200px">3</div>
<div>4</div>
</div>
```

მივიღებთ შემდეგ შედეგს:



flex თვისება flex-grow, flex-shrink და flex-basic თვისებების შემოკლებული ჩანაწერია.

მაგალითად, მესამე ელემენტი არ იზრდება (0), არაა შემჭიდროებული (0) და საწყისი სიგრძეა 200 პიქსელი:

```
<div style="flex: 0 0 200px">3</div>
```

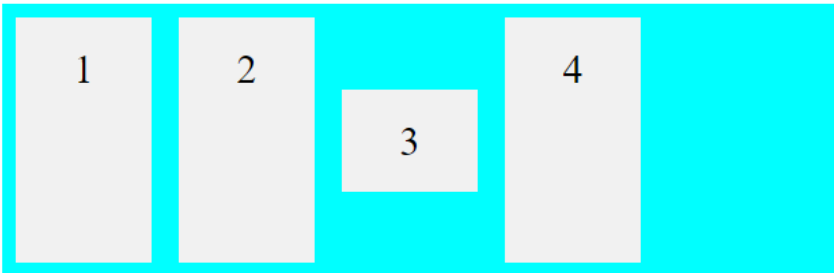
align-self თვისება განსაზღვრავს ცალკეული ელემენტების სწორებას კონტეინერის შიგნით. ეს თვისება ცვლის კონტეინერის align-items თვისების მიერ მინიჭებულ მნიშვნელობას.

მაგალითი 15.11. align-self თვისების უკეთ დემონსტრირების მიზნით კონტეინერის სიმაღლე 200 პიქსელი ავიღოთ, ხოლო მესამე ელემენტი კონტეინერის ცენტრში განვათავსოთ:

```
<!DOCTYPE html>
<html>
<head>
<title>The flex-wrap Property</title>
<style>
.flex-container {
  display: flex;
  height: 200px;
  background-color: aqua;
}

.flex-container > div {
  background-color: #f1f1f1;
```

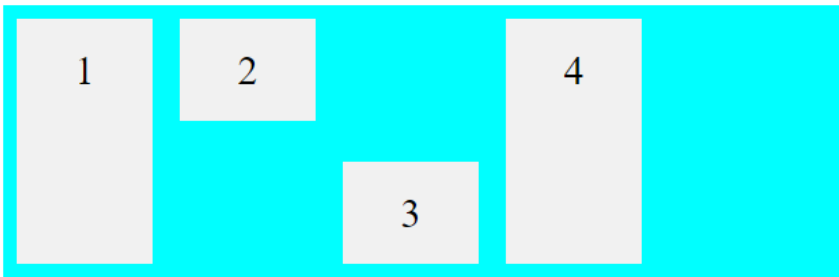
```
width: 100px;
margin: 10px;
text-align: center;
line-height: 75px;
font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
<div>1</div>
<div>2</div>
<div style="align-self: center">3</div>
<div>4</div>
</div>
</body>
</html>
```



მაგალითი 15.12. მეორე ელემენტი კონტეინერის ზემო ნაწილში, ხოლო მესამე ელემენტი ქვემო ნაწილში განთავსდეს, ამისათვის კოდის ტანის ნაწილი შევცვალოთ შემდეგი კოდით:

```
<div class="flex-container">
  <div>1</div>
  <div style="align-self: flex-start">2</div>
  <div style="align-self: flex-end">3</div>
  <div>4</div>
</div>
```

მივიღებთ:



თანამედროვე ვებგვერდის დიზაინში Flexbox ფაქტობრივად ყველგან გვხვდება, ის გამოიყენება ასევე ე.წ. ყველა მოწყობილობაზე ადაპტირებულ ვებგვერდებთან.

თავი 16

მედიამოთხოვნები

- CSS3 Media Queries (მედიამოთხოვნები);
- მედიამოთხოვნების ზოგიერთი მაგალითი;
- ადაპტირებული Flexbox;
- სემანტიკური ტეგები HTML-ში.

CSS3 Media Queries (მედიამოთხოვნები)

CSS2-ში შემოტანილ იქნა მედიამოთხოვნები.

@media წესებმა, რომელიც CSS2-ში იყო შემოღებული, სხვადასხვა მედიატიპის სტილისთვის სხვადასხვა წესის განსაზღვრის საშუალება მოგვცა.

მაგალითად, შეიძლება სტილების ერთი ნაკრები გვექნოდნა კომპიუტერის ეკრანისთვის, პლანშეტისთვის, მობილურისთვის და ა.შ.

მედიამოთხოვნები CSS3-ში CSS2-ის მედიატიპების იდეას ეფუძნება: ნაცვლად იმისა, რომ ეძებოს მოწყობილობის ტიპი, ისინი მოწყობილობის შესაძლებლობებს ითვალისწინებს.

მედიამოთხოვნები სხვადასხვა მახასიათებლის შესამოწმებლად გამოიყენება, როგორცაა:

- დათვალეების ფანჯრის სიგანე და სიმაღლე;
- მოწყობილობის სიგანე და სიმაღლე;
- ორიენტაცია (გამოიყენება ძირითადად პლანშეტებისა და მობილურებისათვის);
- გარჩევადობა.

მედიამოთხოვნების გამოყენება ძალზე პოპულარული მეთოდია სტილების ცხრილების მისაწოდებლად პლანშეტებში, iPhone-სა და Androids-ში.

მედიამოთხოვნები შედგება მედიატიპისაგან და შეიძლება ერთ ან რამდენიმე გამოსახულებას შეიცავდეს, რომელიც true ან false მნიშვნელობას ღებულობს.

```
@media not|only მედიატიპი and (expressions) {  
    CSS-კოდი;  
}
```

მოთხოვნების შედეგი ჭეშმარიტია, თუ მედიატიპი დოკუმენტის მოწყობილობაზე ასახვის ტიპს შეესაბამება და მედია-მოთხოვნების ყველა გამოსახულების მნიშვნელობაა true. როდესაც მედიამოთხოვნების მნიშვნელობა არის true, მისი შესაბამისი სტილები ან სტილების წესები ჩვეულებრივად გამოიყენება.

თუ მხოლოდ კონკრეტულ მოწყობილობას იყენებთ, მაშინ მედიატიპის მითითება აუცილებელია და თუ არ იყენებთ, მაშინ all მნიშვნელობა იგულისხმება.

ასევე შეიძლება მედიასაშუალებებისთვის სხვადასხვა სტილის ცხრილი გქონდეთ:

```
<link rel="stylesheet" media="მედიატიპი and|not|only (expressions)" href="print.css">
```

CSS3 მედიატიპები

მნიშვნელობა	აღწერა
all	გამოიყენება მედიატიპის ნებისმიერი მოწყობილობისათვის
print	გამოიყენება პრინტერისთვის
screen	გამოიყენება კომპიუტერების, პლანშეტების, სმარტფონებისა და სხვათა შემთხვევაში
speech	გამოიყენება ელექტრონული წიგნების ხმამაღლა წამკითხველის დროს

მედიამოთხოვნების ზოგიერთი მაგალითი

მედიამოთხოვნების გამოყენების ერთ-ერთი საშუალებაა ის, რომ სტილების ცხრილში CSS-ის ალტერნატიული სტილების განყოფილება გქონდეთ.

შემდეგ მაგალითში ფონის ფერი ღია მწვანით იცვლება, მაშინ როდესაც დათვალიერების არე 480 პიქსელს აღემატება (თუ დათვალიერების არე 480 პიქსელზე ნაკლებია, მაშინ ფონი ვარდისფერია).

მაგალითი 16.1:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: pink;
}
@media screen and (min-width: 480px) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
<body>
<h1>ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის
ზომა!</h1>
```

```
<p>მედიამოთხოვნა მხოლოდ იმ შემთხვევაში  
ამოქმედდება, როდესაც მედიატიპის დათვალიერების არე  
480px-ს აღემატება. </p>
```

```
</body>
```

```
</html>
```

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

მედიამოთხოვნა მხოლოდ იმ შემთხვევაში ამოქმედდება, როდესაც მედიატიპის დათვალიერების არე 480px-ს აღემატება.

შემდეგ მაგალითში არსებული მენიუ, დათვალიერების არის სიგანის 420 პიქსელზე მეტად გაზრდის შემთხვევაში, ტექსტის მარცხნივ გადაინაცვლებს, ხოლო შემცირების შემთხვევაში ტექსტის ზემოთ გადაადგილდება.

მაგალითი 16.2:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
<style>  
.wrapper {overflow:auto;}
```

```

#main {margin-left: 4px;}
#leftsidebar {float: none;width: auto;}
#menulist {margin:0;padding:0;}
.menuitem { background:#CDF0F6;
            border:1px solid #d4d4d4;
            border-radius:4px;
            list-style-type:none;
            margin:4px;
            padding:2px; }
@media screen and (min-width: 420px) {
  #leftsidebar {width:200px;float:left;}
  #main {margin-left:216px;}
}
</style>
</head>
<body>
<div class="wrapper">
  <div id="leftsidebar">
    <ul id="menulist">
      <li class="menuitem">Menu-item 1</li>
      <li class="menuitem">Menu-item 2</li>
      <li class="menuitem">Menu-item 3</li>
      <li class="menuitem">Menu-item 4</li>
      <li class="menuitem">Menu-item 5</li>
    </ul>
  </div>
  <div id="main">

```

```
<h1>ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!</h1>
```

```
<p>თუ დათვალიერების არის სიგანე 420 პიქსელზე მეტია, მაშინ მენიუ ტექსტის მარცხნივ განთავსდება. თუ დათვალიერების არის სიგანე 420 პიქსელზე ნაკლებია, მაშინ მენიუ ტექსტის ზემოთ განთავსდება.</p>
```

```
</div>  
</div>  
</body>  
</html>
```

როდესაც დათვალიერების არის სიგანე 420 პიქსელზე მეტია:

Menu-item 1
Menu-item 2
Menu-item 3
Menu-item 4
Menu-item 5

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

თუ დათვალიერების არის სიგანე 420 პიქსელზე მეტია, მაშინ მენიუ ტექსტის მარცხნივ განთავსდება. თუ დათვალიერების არის სიგანე 420 პიქსელზე ნაკლებია, მაშინ მენიუ ტექსტის ზემოთ განთავსდება.

ხოლო როდესაც დათვალიერების არის სიგანე 420 პიქსელზე ნაკლებია, მაშინ შედეგი ასეთი იქნება:

Menu-item 1

Menu-item 2

Menu-item 3

Menu-item 4

Menu-item 5

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

თუ დათვალიერების არის სიგანე 420 პიქსელზე მეტია, მაშინ მენიუ ტექსტის მარცხნივ განთავსდება. თუ დათვალიერების არის სიგანე 420 პიქსელზე ნაკლებია, მაშინ მენიუ ტექსტის ზემოთ განთავსდება.

განვიხილოთ მედიამოთხოვნის რამდენიმე მაგალითი:

პირველ რიგში ისეთი სია განვიხილოთ, სადაც ელემენტებად ელექტრონული ფოსტის მისამართები იქნება გამოყენებული.

ყურადღება მიაქციეთ data-email ატრიბუტს. HTML 5-ში ატრიბუტი პრეფიქსით data შეგვიძლია ინფორმაციის შესანახად გამოვიყენოთ.

დავწეროთ კოდი, რომელშიც გათვალისწინებული იქნება ფანჯრის ზომები: იმ შემთხვევაში, როცა ფანჯრის სიგანე 420 პიქსელსა და 599 პიქსელს შორისაა, მაშინ ელექტრონული ფოსტის ნიშანი ყველა ბმულის წინ გამოიყენება. თუ დათვალიერების არის სიგანე 600 პიქსელსა და 900 პიქსელს შორისაა, მაშინ ბმულების წინ გამოჩნდება ტექსტი "Email:". თუ დათვალიერების არის სიგანე 900 პიქსელზე მეტია, მაშინ ბმულების შემდეგ გამოჩნდება მათი ელექტრონული ფოსტის ზუსტი

მისამართიც, ხოლო თუ დათვალიერების არის სიგანე 1051 პიქსელზე მეტია, მაშინ ყველა ბმულის წინ ელექტრონული ფოსტის ნიშანიც დაემატება.

საბოლოოდ, მიღებულ კოდს ქვემოთ მოცემული სახე ექნება.

მაგალითი 16.3:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
}
ul li a {
  color: green;
  text-decoration: none;
  padding: 3px;
  display: block;
}
@media screen and (max-width: 599px) and (min-width: 420px),
(min-width: 1051px) {
  ul li a {
    padding-left: 30px;
    background: url(email-icon.jpg) left center no-repeat;
  }
}
```

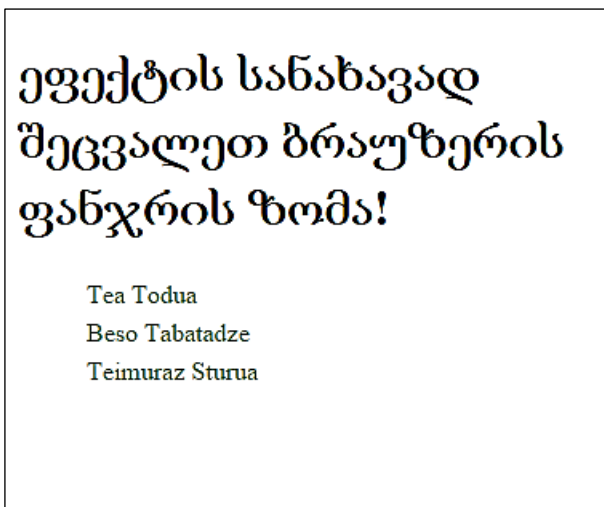
```

    @media screen and (max-width: 900px) and (min-width: 600px)
    {
        ul li a:before {
            content: "Email: ";
            font-style: italic;
            color: #607080;
        }
    }
    @media screen and (min-width: 901px) {
        ul li a:after {
            content: " (" attr(data-email) ")";
            font-size: 12px;
            font-style: italic;
            color: #607080;
        }
    }
</style>
</head>
<body>
    <h1>ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის
ზომა!</h1>
    <ul>
        <li><a data-
email="tea_todua@yahoo.com" href="mailto:tea_todua@yahoo.com">
Tea Todua</a></li>
        <li><a data-
email="besotabatadze84@gmail.com" href="mailto:besotabatadze84@
gmail.com">Beso Tabatadze</a></li>

```




```
<li><a data-  
email="temuri.sturua@gmail.com"href="mailto:temuri.sturua@gmail.  
com">Teimuraz Sturua</a></li>  
</ul>  
</body>  
</html>
```

როდესაც დათვალიერების არის სიგანე 420 პიქსელზე ნაკლებია, მაშინ შედეგი ასეთი იქნება:



ამ კოდის შესრულების შედეგად, როდესაც დათვალიერების არის სიგანე 420 პიქსელსა და 599 პიქსელს შორისაა მოთავსებული, მაშინ შედეგი ასეთი იქნება:

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

-  Tea Todua
-  Beso Tabatadze
-  Teimuraz Sturua

როდესაც დათვალიერების არის სიგანე 600 პიქსელსა და 900 პიქსელს შორისაა მოთავსებული, მაშინ შედეგი ასეთი იქნება:

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

Email: Tea Todua
Email: Beso Tabatadze
Email: Teimuraz Sturua

როდესაც დათვალიერების არის სიგანე 901 პიქსელსა და 1051 პიქსელს შორისაა მოთავსებული, მაშინ შედეგი ასეთი იქნება:

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

Tea Todua (tea_todua@yahoo.com)
Beso Tabatadze (besotabatadze54@gmail.com)
Teimuraz Sturua (teimurazsturua@gmail.com)

როდესაც დათვალიერების არის სიგანე 1051 პიქსელზე მეტია, მაშინ შედეგი ასეთი იქნება:

ეფექტის სანახავად შეცვალეთ ბრაუზერის ფანჯრის ზომა!

 Tea Todua (tea_todua@yahoo.com)
 Beso Tabatadze (besotabatadze54@gmail.com)
 Teimuraz Sturua (teimurazsturua@gmail.com)

ადაპტირებული Flexbox

CSS Media Queries შესწავლისას ვნახეთ, რომ სხვადასხვა ზომის ეკრანისა და სხვადასხვა მოწყობილობისათვის სხვადასხვა მაკეტის შესაქმნელად მედიამოთხოვნების გამოყენება შეგვიძლია.

მაგალითად, თუ გინდათ დიდი ზომის მქონე ეკრანისათვის შექმნათ მაკეტი ორი სვეტისგან და შედარებით მცირე ზომის ეკრანისათვის, როგორცაა პლანშეტი და ტელეფონი, ერთი სვეტით, შეგვიძლიათ შეცვალოთ მოქნილობის მიმართულება სტრიქონიდან სვეტზე გარკვეული ზომის

მიღწევის შემდეგ. ქვემოთ მაგალითში ეს წერტილი არის 600 პიქსელი.

მაგალითი 16.4:

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}
.flex-container {
  display: flex;
  flex-direction: row;
  font-size: 30px;
  text-align: center;
}
.flex-item-left {
  background-color: #f1f1f1;
  padding: 10px;
  flex: 50%;
}
.flex-item-right {
  background-color: aqua;
  padding: 10px;
  flex: 50%;
}

@media (max-width: 600px) {
```

```

.flex-container {
  flex-direction: column;
}

```

</style>

</head>

<body>

<h1> Flexbox კონტეინერი</h1>

<p> HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox ერთ-ერთი თანამედროვე მიდგომაა. </p>

<p>იმისათვის, რომ გამოვიყენოთ Flexbox მოდელი პირველ რიგში საჭიროა განისაზღვროს ე.წ. კონტეინერი, რისთვისაც display თვისებას მივანიჭოთ flex მნიშვნელობა. </p>

<p> შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია, მიმართულება იცვლება.</p>

```

<div class="flex-container">
  <div class="flex-item-left">1</div>
  <div class="flex-item-right">2</div>
</div>
</body>
</html>

```

თუ ეკრანის ზომა აღემატება 600 პიქსელს, მაშინ მივიღებთ:

Flexbox კონტეინერი

HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox – ერთ-ერთი თანამედროვე მიდგომაა.

იმისათვის, რომ გამოვიყენოთ Flexbox მოდელი პირველ რიგში საჭიროა განისაზღვროს ე.წ. კონტეინერი, რისთვისაც display თვისებას მივანიჭოთ flex მნიშვნელობა.

შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია, მიმართულება იცვლება.



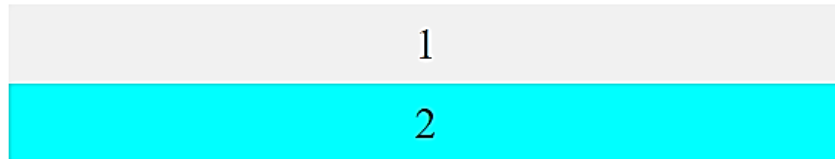
ხოლო ეკრანის ზომა როგორც კი გახდება 600 პიქსელზე ნაკლები, მაშინ ის მიიღებს შემდეგ სახეს:

Flexbox კონტეინერი

HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox – ერთ-ერთი თანამედროვე მიდგომაა.

იმისათვის, რომ გამოვიყენოთ Flexbox მოდელი პირველ რიგში საჭიროა განისაზღვროს ე.წ. კონტეინერი, რისთვისაც display თვისებას მივანიჭოთ flex მნიშვნელობა.

შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია, მიმართულება იცვლება.



სხვა მეთოდით იგივე შედეგის მისაღებად flex თვისების პროცენტული მნიშვნელობის ცვლილება უნდა მოვახდინოთ. ამასთან ყურადღება მიაქციეთ, ამ მაგალითმა რომ იმუშაოს, კონტეინერში უნდა ჩავრთოთ flex-wrap: wrap;

მაგალითი 16.5:

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}
.flex-container {
  display: flex;
  flex-wrap: wrap;
  font-size: 30px;
  text-align: center;
}
.flex-item-left {
  background-color: #f1f1f1;
  padding: 10px;
  flex: 50%;
}
.flex-item-right {
  background-color: aqua;
  padding: 10px;
  flex: 50%;
```

```
}
```

```
@media (max-width: 600px) {  
  .flex-item-right, .flex-item-left {  
    flex: 100%;  
  }  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> Flexbox კონტეინერი</h1>
```

```
<p> HTML ვებგვერდზე შესაბამისი ელემენტების  
განლაგების რამდენიმე საშუალება არსებობს, მათ შორის  
flexbox ერთ-ერთი თანამედროვე მიდგომაა. </p>
```

```
<br>
```

```
<p><b> შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა  
დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია,  
მიმართულება იცვლება.</b></p>
```

```
<div class="flex-container">
```

```
  <div class="flex-item-left">1</div>
```

```
  <div class="flex-item-right">2</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

მივიღებთ ანალოგიურ შედეგებს:

Flexbox კონტეინერი

HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox – ერთ-ერთი თანამედროვე მიდგომაა.

შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია, მიმართულება იცვლება.

1

2

Flexbox კონტეინერი

HTML ვებგვერდზე შესაბამისი ელემენტების განლაგების რამდენიმე საშუალება არსებობს, მათ შორის flexbox – ერთ-ერთი თანამედროვე მიდგომაა.

შეცვალეთ ბრაუზერის ფანჯრის ზომა, რათა დაინახოთ, რომ როცა ეკრანის ზომა 600 პიქსელზე ნაკლებია, მიმართულება იცვლება.

1

2

HTML-ის სემანტიკური ტეგები

რა არის სემანტიკური ტეგი?

სემანტიკური ტეგი ზუსტად აღწერს მის მნიშვნელობას როგორც ბრაუზერისათვის, ისე დამმუშავებლისათვის.

არასემანტიკური ტეგის მაგალითია <div> და ტეგი, რომლებიც არაფერს ამბობს მის შინაარსზე.

სემანტიკური ტეგის მაგალითია <form>, <table> და <article> ტეგები, რომლებიც ზუსტად განსაზღვრავს მის შინაარსს.

სემანტიკური ტეგები HTML-ში

მრავალი ვებსაიტი შეიცავს HTML-კოდს, მაგალითად:

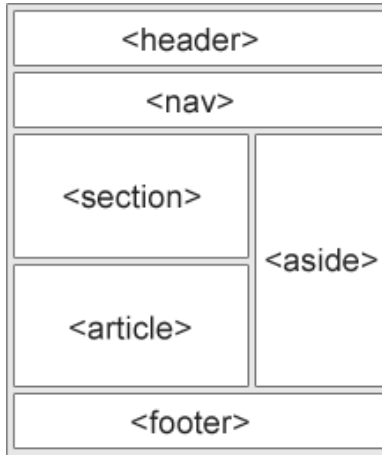
```
<div id = "nav">  
<div class = "header">  
<div id = "footer">
```

ნავიგაციის, ზედა და ქვედა კოლონტიტულების აღსანიშნავად.

HTML-ში არის ზოგიერთი სემანტიკური ტეგი, რომლებიც შეიძლება გამოყენებული იყოს ვებსაიტის სხვადასხვა ნაწილის განსაზღვრისათვის. ეს ტეგებია:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>

- `<time>`



ტეგი `<section>`

ტეგი `<section>` დოკუმენტში განსაზღვრავს სექციას. სექცია არის შინაარსის თემატური დაჯგუფება სათაურით.

ჩვეულებრივ ვებგვერდი შეიძლება იყოს დაყოფილი სექციებად - შესავალი, შინაარსი და საკონტაქტო ინფორმაცია.

მაგალითი 16.6. ორი სექცია ერთ დოკუმენტში:

```

<!DOCTYPE html>
<html>
<body>

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an
international organization working on issues regarding the

```

conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>

</section>

<section>

<h1>WWF's Panda symbol</h1>

<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p>

</section>

</body>

</html>

WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

WWF's Panda symbol

The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.

ტეგი <article>

ტეგი <article> დამოუკიდებელ, ავტონომიურ შინაარსს განსაზღვრავს.

სტატიას დამოუკიდებლად უნდა ჰქონდეს აზრი და ვებსაიტის სხვა ნაწილებისაგან დამოუკიდებლად მისი გავრცელების საშუალება უნდა იყოს.

ტეგი <article> შეიძლება გამოყენებული იყოს, როგორც შეტყობინება ფორუმზე, ბლოგის შეტყობინება ან საგაზეთო სტატია.

ქვემოთ მოცემულია სამი სტატიის მაგალითი დამოუკიდებელი, ავტონომიური შინაარსით.

მაგალითი 16.7:

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

<h1>The article element</h1>

<article>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by Google,
released in 2008. Chrome is the world's most popular web browser
today!</p>
</article>

<article>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser developed
by Mozilla. Firefox has been the second most popular web browser
since January, 2018.</p>
</article>

<article>
```

```
<h2>Microsoft Edge</h2>
```

```
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>
```

```
</article>
```

```
</body>
```

```
</html>
```

The article element

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

Mozilla Firefox

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.

Microsoft Edge

Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.

მაგალითი 16.8. `<article>` ტეგის სტილისათვის გამოყენებულია CSS:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.all-browsers {
```

```
margin: 0;
```



```

padding: 5px;
background-color: lightgray;
}
.all-browsers > h1, .browser {
margin: 10px;
padding: 5px;
}
.browser {
background: white;
}
.browser > h2, p {
margin: 4px;
font-size: 90%;
}
</style>
</head>
<body>
<article class="all-browsers">
<h1>Most Popular Browsers</h1>
<article class="browser">
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google,
released in 2008. Chrome is the world's most popular web browser
today!</p>
</article>
<article class="browser">
<h2>Mozilla Firefox</h2>

```

```
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>
```

```
</article>
```

```
<article class="browser">
```

```
<h2>Microsoft Edge</h2>
```

```
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.</p>
```

```
</article>
```

```
</article>
```

```
</body>
```

```
</html>
```

Most Popular Browsers

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

Mozilla Firefox

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.

Microsoft Edge

Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.

ტეგი <header>

ტეგი <header> საწყისი შინაარსის ან სანავიგაციო ბმულების ნაკრების კონტეინერული ტეგია. ტეგი <header> ჩვეულებრივ შეიცავს:

- ერთ ან რამდენიმე სათაურის ტეგს (<h1> - <h6>);
- ლოგოს ან ნიშნაკს;
- საავტორო ინფორმაციას.

შენიშვნა. ერთ HTML დოკუმენტში შეიძლება რამდენიმე <header> ტეგი იყოს. მაგრამ <header> ტეგი არ შეიძლება მოთავსებული იყოს <footer>, <address> ტეგში ან სხვა <header> ტეგში.

მაგალითი 16.9. სათაური <article> ტეგისთვის:

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

<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's
natural environment, and build a future in which humans live in
harmony with nature.</p>
</article>

</body>
</html>
```

What Does WWF Do?

WWF's mission:

WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.

ტეგი <footer>

ტეგი <footer> განსაზღვრავს დოკუმენტის ან სექციის ქვედა კოლონტიტულს. ტეგი <footer> შეიძლება შეიცავდეს:

ინფორმაციას ავტორის შესახებ;

ინფორმაციას საავტორო უფლების შესახებ;

საკონტაქტო ინფორმაციას;

საიტის რუკას;

საიტის დასაწყისში გადასასვლელ ბმულს;

საიტთან დაკავშირებულ დოკუმენტებს.

ერთ დოკუმენტში შეგიძლიათ რამდენიმე <footer> ტეგი გქონდეთ.

მაგალითი 16.10:

```
<!DOCTYPE html>
<html>
<body>
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com
</a></p>
</footer>
```

```
</body>
```

```
</html>
```

Author: Hege Refsnes

hege@example.com

ტეგი <nav>

ტეგი <nav> სანავიგაციო ბმულების ნაკრებს განსაზღვრავს. ყურადღება მიაქციეთ, რომ არ არის სავალდებულო დოკუმენტში გამოყენებული ყველა ბმული <nav> ტეგში იყოს განთავსებული. <nav> ტეგი მხოლოდ ძირითადი სანავიგაციო ბმულებისთვისაა განკუთვნილი.

მაგალითი 16.11:

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

<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

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

[HTML](#) | [CSS](#) | [JavaScript](#) | [jQuery](#)

ტეგი <aside>

ტეგი <aside> განსაზღვრავს რაიმე შინაარსს, იმ შინაარსის გარდა, რომელშიც ისაა განთავსებული (მაგალითად, გვერდით პანელს).

<aside> ტეგით განსაზღვრული შინაარსი ირიბად მაინც უნდა იყოს დაკავშირებული იმ შინაარსთან, რომელშიც ის არის განთავსებული.

მაგალითი 16.12. განსაზღვრეთ რაიმე შინაარსი, იმ შინაარსის გარდა, რომელშიც ისაა განთავსებული:

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

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
```

```
<aside>
  <h4>Epcot Center</h4>
  <p>Epcot is a theme park at Walt Disney World Resort
featuring exciting attractions, international pavilions, award-winning
fireworks and seasonal special events.</p>
</aside>

</body>
</html>
```

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

Epcot Center

Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.

მაგალითი 16.13. <aside> ტეგის სტილისათვის გამოყენებულია CSS:

```
<!DOCTYPE html>
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
```

```
background-color: lightgray;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>My family and I visited The Epcot center this summer. The  
weather was nice, and Epcot was amazing! I had a great summer  
together with my family!</p>
```

```
<aside>
```

```
<p>The Epcot center is a theme park at Walt Disney World  
Resort featuring exciting attractions, international pavilions, award-  
winning fireworks and seasonal special events.</p>
```

```
</aside>
```

```
<p>My family and I visited The Epcot center this summer. The  
weather was nice, and Epcot was amazing! I had a great summer  
together with my family!</p>
```

```
<p>My family and I visited The Epcot center this summer. The  
weather was nice, and Epcot was amazing! I had a great summer  
together with my family!</p>
```

```
</body>
```

```
</html>
```


My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.

ტეგები <figure> და <figcaption>

ტეგი <figure> ავტონომიურ შინაარსზე მიუთითებს, მაგალითად, ილუსტრაციები, დიაგრამები, ფოტოები, კოდების სია და ა. შ.

ტეგი <figcaption> სათაურს განსაზღვრავს <figure> ტეგისათვის. ტეგი <figcaption> შეიძლება იყოს როგორც პირველი, ისე ბოლო <figure> ტეგის შვილობილი ტეგი.

მაგალითი 16.14.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>მოინახულეთ საქართველო</h2>
```

```
<p> კაცხის მაცხოვრის სახელობის ეკლესია მდებარეობს  
ჭიათურის რაიონის სოფელ კაცხში, მდ. კაცხურას ხეობაში.
```

```
ტაძარი აგებულია XI საუკუნის დასაწყისში. მისი ძირითადი  
ნაწილი 1010-1014 წლებშია აშენებული, ბაღვაშთა ფეოდალური  
საგვარეულოს მიერ (ნაგებობის მთავარ შესასვლელში გვხვდება
```

ბაღვაშთა გერბის შემადგენელი ფიგურები: ლომი და „მოხუცი ანგელოზი“). </p>

<figure>

<figcaption>სურ.1 - კაცხი, ჭიათურა,

საქართველო.</figcaption>

</figure>

</body>

</html>

მოინახულეთ საქართველო

კაცხის მაცხოვრის სახელობის ეკლესია მდებარეობს ჭიათურის რაიონის სოფელ კაცხში, მდ. კაცხურას ხეობაში. ტაძარი აგებულია XI საუკუნის დასაწყისში. მისი ძირითადი ნაწილი 1010-1014 წლებშია აშენებული, ბაღვაშთა ფეოდალური საგვარეულოს მიერ (ნაგებობის მთავარ შესასვლელში გვხვდება ბაღვაშთა გერბის შემადგენელი ფიგურები: ლომი და „მოხუცი ანგელოზი“).



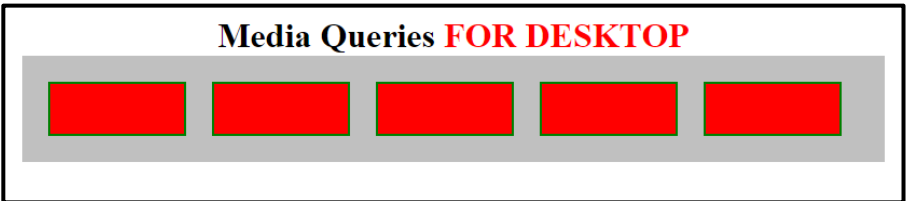
სურ.1 - კაცხი, ჭიათურა, საქართველო.

 ტეგი განსაზღვრავს გამოსახულებას, ხოლო ტეგი <figcaption> - მის სათაურს.

დავალება. შესრულების პროცესში გამოიყენეთ @media მოთხოვნები. ააგეთ ქვემოთ მოცემული div ტეგების სტრუქტურა Desktop-ისთვის, როდესაც ბრაუზერის სიგანე მეტია 850 px-ზე:

✓ ტექსტი “Media Queries” შავია, ტექტი “FOR DESKTOP” წითელი;

✓ პატარა div ტეგების ფონი წითელია, საზღვრები - მწვანე;
✓ კონტეინერის ფონი ვერცხლისფერია, განთავსებულია ბრაუზერის ცენტრში, სიგანე - 800 px. კონტეინერში შინაარსის განაწილებისთვის გამოიყენეთ flexBox თვისებები.

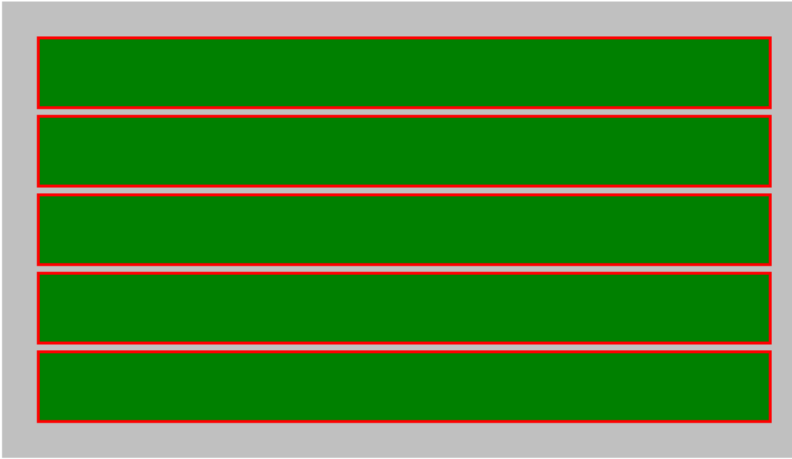


ზემოთ მოყვანილი div ტეგების სტრუქტურა ბრაუზერის სიგანის შემცირებისას, როდესაც სიგანე ნაკლებია 850 px-ზე, მიიღებს სახეს:

✓ “FOR DESKTOP” იცვლება და ხდება მწვანე ფერის “FOR MOBILE”, ტექსტი “Media Queries” არ იცვლება;

✓ პატარა div ტეგების ფონი მწვანეა, საზღვრები - წითელი;
✓ კონტეინერის თვისებები არ იცვლება.

Media Queries **FOR MOBILE**



**დანართი 1. ძირითადი მათემატიკური სიმბოლოების
დასახელება, მათი ათობითი და თექვსმეტობითი კოდები**

Char	Dec	Hex	Entity	Name
∀	8704	2200	∀	ნებისმიერი
∁	8705	2201		დამატება
∂	8706	2202	∂	კერძო დიფერენციალი
∃	8707	2203	∃	არსებობს
∄	8708	2204		არ არსებობს
∅	8709	2205	∅	ცარიელი სიმრავლე
∈	8712	2208	∈	ეკუთვნის ელემენტი
∉	8713	2209	∉	არ ეკუთვნის ელემენტი
∋	8715	220B	∋	შეიცავს წევრს
∌	8716	220C		არ შეიცავს წევრს
∏	8719	220F	∏	N რიცხვის ნამრავლი
∑	8721	2211	∑	N რიცხვის ჯამი
-	8722	2212	−	მინუსი
+	8723	2213		მინუს პლუსი
/	8725	2215		გაყოფა
*	8727	2217	∗	გამრავლება
√	8730	221A	√	კვადრატული ფესვი
³ √	8731	221B		კუბური ფესვი
⁴ √	8732	221C		მეოთხე რიგის ფესვი
∝	8733	221D	∝	პროპორციულია
∞	8734	221E	∞	უსასრულობა
⊥	8735	221F		მართი კუთხე
∠	8736	2220	∠	კუთხე
⊚	8737	2221		ჩვეულებრივი კუთხე

⚡	8738	2222		სფერული კუთხე
	8739	2223		იყოფა
†	8740	2224		არ იყოფა
	8741	2225		პარალელური
‡	8742	2226		არ არის პარალელური
∧	8743	2227	∧	ლოგიკური „და“
∨	8744	2228	∨	ლოგიკური „ან“
∩	8745	2229	∩	თანაკვეთა
∪	8746	222A	∪	გაერთიანება
∫	8747	222B	∫	ინტეგრალი
∬	8748	222C		ორმაგი ინტეგრალი
∭	8749	222D		სამმაგი ინტეგრალი
—:	8761	2239		მეტობა
≐:	8762	223A		გეომეტრიული პროპორციულობა
≈	8776	2248	≈	მიახლოებითი ტოლობა
≇	8777	2249		მიახლოებით არ უდრის
≍	8781	224D		ეკვივალენტურია
≠	8800	2260	≠	არ უდრის
≡	8801	2261	≡	იდენტურია
≢	8802	2262		არ არის იდენტური
≤	8804	2264	≤	ნაკლებია ან ტოლი
≥	8805	2265	≥	მეტია ან ტოლი
≪	8810	226A		ბევრად ნაკლები
≫	8811	226B		ბევრად მეტი
≠	8813	226D		არ არის ეკვივალენტური
≠	8814	226E		არ არის ნაკლები

➤	8815	226F		არ არის მეტი
⚔	8816	2270		არ არის ნაკლები ან ტოლი
⚕	8817	2271		არ არის მეტი ან ტოლი
⊂	8834	2282	⊂	ქვესიმრავლე
⊃	8835	2283	⊃	ზესიმრავლე
⊄	8836	2284	&nsb;	არ არის ქვესიმრავლე
⊇	8837	2285		არ არის ზესიმრავლე
⊆	8838	2286	⊆	ტოლია ან ქვესიმრავლე
⊇	8839	2287	⊇	ტოლია ან ზესიმრავლე
⊄	8895	22BF		მართკუთხა სამკუთხედი

დანართი 2. HTML-ში სემანტიკური ტეგების სია

ტეგი	მნიშვნელობა
<article>	განსაზღვრავს დამოუკიდებელ, ავტონომიურ შინაარსს
<aside>	განსაზღვრავს რაიმე შინაარსს, იმ შინაარსის გარდა, რომელშიც ისაა განთავსებული
<details>	განსაზღვრავს დამატებით დეტალებს, რომელთა ნახვა ან დამალვა შეუძლია მომხმარებელს
<figcaption>	განსაზღვრავს <figure> ტეგით განსაზღვრული ობიექტის სათაურს
<figure>	განსაზღვრავს ავტონომიურ შინაარსს, მაგალითად, ილუსტრაციას, დიაგრამას, ფოტოს, კოდების სიას და ა. შ.
<footer>	განსაზღვრავს დოკუმენტისათვის ან სექტორისათვის ქვედა კოლონტიტულს
<header>	განსაზღვრავს დოკუმენტისათვის ან სექტორისათვის სათაურს
<main>	განსაზღვრავს დოკუმენტის ძირითად შინაარსს

<mark>	განსაზღვრავს მონიშნულ / გამოყოფილ ტექსტს
<nav>	განსაზღვრავს სანავიგაციო ზმულებს
<section>	განსაზღვრავს სექტორს დოკუმენტში
<summary>	განსაზღვრავს <details> ტეგისთვის სათაურს
<time>	განსაზღვრავს თარიღს / დროს

ლიტერატურა

1. თ. სტურუა, თ. თოდუა, ბ. ტაბატაძე. HTML5&CSS3. თბილისი, „ტექნიკური უნივერსიტეტი“, 2016 წ.,
https://gtu.ge/book/ims/HTML5_CSS3.pdf
2. თ. სტურუა, ქ. ნანობაშვილი, მ. დოლიძე. Web-გვერდების დაპროექტება HTML ენის გამოყენებით, თბილისი, „ტექნიკური უნივერსიტეტი“, 2013 წ.,
<http://dSPACE.nplg.gov.ge/bitstream/1234/22443/1/WebGverdebisDaproeqtebaHTMLEnisGamoyenebit.pdf>
3. თ. სტურუა, ქ. ნანობაშვილი, მ. დოლიძე. Web-გვერდების დაპროექტება (HTML) მეთოდური მითითებები ლაბორატორიული სამუშაოს შესასრულებლად, თბილისი, „ტექნიკური უნივერსიტეტი“, 2012 წ.,
<http://dSPACE.nplg.gov.ge/bitstream/1234/3939/1/WebgverdebiDaproeqteba.pdf>
4. WebDevelopmentTutorial. <https://aditi-softskill.org/web-extras/library/WebDevelopment/WebDevelopmentTutorial.docx>
5. <https://ka.wikipedia.org>
6. https://www.w3schools.com/html/html5_semantic_elements.asp
7. http://w3schools-fa.ir/css/css_colors.html
8. <https://blog.eduonix.com/web-programming-tutorials/design-simple-website-using-flexbox/>
9. <http://ww2.cs.fsu.edu/~faizian/cgs3066/resources/Lecture13-HTML%20Forms%20and%20Javascript%20Validation.pdf>
10. <https://startuptalky.com/web-browsers-market-share/>
11. <https://stackoverflow.com/questions/53189666/how-to-horizontally-stretch-elements-when-using-flex-box>

რედაქტორი ლ. მამალაძე

გადაეცა წარმოებას 05.10.2021. ხელმოწერილია დასაბეჭდად
10.11.2021. ქაღალდის ზომა 60X84 1/16 პირობითი ნაბეჭდი თაბახი 25.
№3434.

საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“,
თბილისი, კოსტავას 77



Verba volant,
scripta manent