

ქ. ნანობაშვილი, თ. სტურუა, მ. დოლიძე

# Web-გვერდების დაპროექტება

## HTML ენის გამოყენებით



საქართველოს ტექნიკური უნივერსიტეტი

ქ. ნანობაშვილი, თ. სტურუა, მ. დოლიძე

# Web-გვერდების დაპროექტება HTML ენის გამოყენებით



დამტკიცებულია სახელმძღვანელოდ  
სტუ-ს სარედაქციო-საგამომცემლო  
საბჭოს მიერ. 06.06.2013, ოქმი №3

თბილისი

2013

მოცემულია **HTML** ენის ელემენტარული საფუძვლებიდან დაწყებული და რთული პრაქტიკული საკითხებით დამთავრებული, აგრეთვე, ვებ-გვერდების ინტერნეტში განთავსებისა და ინფორმაციის ძებნის საკითხები. ასევე, მოყვანილია **Web**-გვერდების შექმნის მრავალი მაგალითი და შესასრულებლად გამზადებულ პროგრამათა ტექსტები.

განკუთვნილია ინფორმატიკისა და მართვის სისტემების ფაკულტეტის პროფესიული სწავლებისა და ბაკალავრიატის სტუდენტებისათვის, აგრეთვე, **HTML** ენის გამოყენებით **Web**-გვერდების შექმნით დაინტერესებული სხვა სპეციალისტებისათვის.

რეცენზენტები: სრული პროფესორი რ. სამხარაძე,  
სრული პროფესორი თ. მაჭარაძე

© საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 2013  
**ISBN 978-9941-20-374-9**

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

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



საავტორო უფლებების დარღვევა ისჯება კანონით.

## შინაარსი

შესავალი.....	7
ძირითადი ცნებები.....	16
ბრაუზერი.....	21
ტეგები და მათი ატრიბუტები.....	22
დოკუმენტის სტრუქტურა.....	24
HTML ენის ვერსია.....	25
META-ინფორმაცია.....	28
ატრიბუტი NAME.....	29
ატრიბუტი HTTP-EQUIV.....	30
HTML დოკუმენტის შენახვა და გახსნა.....	31
ცვლილებების შეტანა პროგრამაში.....	33
<Body> ტეგის ატრიბუტები.....	35
ვებ-გვერდების სტანდარტული ფერები.....	40
ფონური სურათები.....	42
სათაურის დონეები.....	44
შრიფტი.....	46
აბზაცი და სტრიქონების წყვეტა.....	50
წინასწარი დაფორმატება.....	56
შრიფტის დაფორმატება.....	56
კომენტარები.....	62

აკრონიმი.....	63
ციტატა და ტერმინები .....	64
გამოყოფილი შრიფტი.....	66
მონოსიგანის შრიფტების გამოყენება.....	67
ჰარების დასმა.....	71
კლავიატურის სიმბოლოები .....	72
სპეციალური სიმბოლოები .....	74
ინფორმაცია დოკუმენტის ავტორის შესახებ.....	75
<b>ჰორიზონტალური ხაზები .....</b>	<b>77</b>
<b>სიები.....</b>	<b>79</b>
დანომრილი სიები.....	79
მარკირებული სიები.....	82
სია განსაზღვრება.....	85
<b>ვებ-გვერდებზე გრაფიკული ელემენტის განთავსება.....</b>	<b>89</b>
გრაფიკული ფაილის ფორმატები.....	91
პიქსელები და ამოხსნადობა .....	92
ვებ-გვერდზე გრაფიკული გამოსახულების განთავსების მაგალითები .....	94
გამოსახულების სწორება.....	97
გამოსახულების მასშტაბირება.....	99
გამოსახულების რუქა (გამოსახულების დაშლა არეებად ).	103
<b>ცხრილები.....</b>	<b>111</b>

ცხრილის სტრიქონებისა და სვეტების შექმნა.....	113
ცხრილის უჯრედების შექმნა .....	114
ცხრილის სათაურისა და წარწერის შექმნა.....	114
ცხრილის სვეტების დაჯგუფება .....	115
ცხრილის სტრიქონების დაჯგუფება .....	115
ცხრილის ელემენტების ძირითადი ატრიბუტები .....	116
ცხრილების შექმნის მაგალითები.....	118
<b>ფორმები.....</b>	<b>136</b>
ფორმის შექმნა .....	136
მართვის ელემენტების ფორმაზე განთავსება.....	140
ამორჩევის სიები .....	146
მრავალსტრიქონიანი ტექსტური ველი.....	150
<b>ბმულები .....</b>	<b>155</b>
რესურსის უნივერსალური მაჩვენებელი .....	155
ბმულების დოკუმენტში ჩასმა.....	157
შიდა ბმულები.....	160
გარე ბმულები.....	161
დოკუმენტის შიგნით გადასვლა .....	161
გრაფიკული ბმულები.....	168
<b>მორბენალი სტრიქონი .....</b>	<b>170</b>
მორბენალი სტრიქონის მაგალითები .....	171
სურათი მორბენალ სტრიქონად.....	176

<b>ფრეიმები .....</b>	<b>178</b>
ფრეიმების თვისებები.....	178
ფრეიმების ნაკრები .....	179
ფრეიმის ფრეიმების ნაკრებით დოკუმენტში ჩასმა.....	183
ფრეიმის ფრეიმების ნაკრების გარეშე დოკუმენტში ჩასმა...	184
ფრეიმების ურთიერთქმედება.....	186
ფრეიმების გამოყენების მაგალითები.....	188
ფრეიმების ურთიერთქმედების მაგალითები .....	196
<b>მულტიმედია .....</b>	<b>205</b>
ფაილების ფორმატი და კოდირების ფორმატები .....	205
MIME ტიპები.....	208
აუდიო-რგოლის ჩასმა.....	209
ვიდეო-რგოლის ჩასმა .....	212
<b>დოკუმენტების პუბლიკაცია და ძებნა.....</b>	<b>214</b>
ვებ-საიტების მომზადება და პუბლიკაცია .....	217
ინტერნეტში ინფორმაციის ძებნა.....	219
ინფორმაციის ძებნა URL-მისამართის მიხედვით.....	219
საძიებო სისტემები .....	222

## შესავალი

დღეს ყველამ იცის, რა არის ინტერნეტი და ვებ-გვერდი, ეს ცნებები ყოველდღიური ცხოვრების ნაწილი გახდა. მათ შესახებ ლაპარაკობენ, ფიქრობენ, ორგანიზაციები თავის ინტერნეტ-მისამართებს სარეკლამო პროსპექტებში ათავსებენ. E-mail მისამართებს აჩვენებენ ტელევიზიით, აცხადებენ რადიოთი. ამიტომ, ვებ-გვერდების შექმნის თემა ბევრისათვის საინტერესოა.

პირველი კომპიუტერის გამოჩენასთან ერთად სამყაროს ინფორმაციის გადაცემის პრობლემაც გაჩნდა. და ამის საშუალებებიც მოიძებნა - ეს არის ინფორმაციის მატარებლები და კაბელები. დაიწერა პროგრამები, რომლებიც მონაცემების გაგზავნისა და მიღების საშუალებას იძლეოდა და თითქმის მაშინვე გაჩნდა კიდევ ერთი პრობლემა - თავსებადობის პრობლემა: თუ როგორ უნდა კაბელით დაკავშირებული ორი ან მეტი კომპიუტერის მუშაობა ლოკალურ ქსელში. ინტერნეტის ოფიციალური დაბადების დღეა 1969 წლის 2 იანვარი, როდესაც აშშ-ს თავდაცვის სამინისტროს პერსპექტიული კვლევების სამმართველომ (Advanced Research Project Agency, ARPA) თავდაცვის კომპიუტერების დაკავშირების პროექტზე დაიწყო მუშაობა.

1970 წელს ინტერნეტი საერთაშორისო ქსელად გადაიქცა. ინტერნეტის პირველი მომსახურების სამსახური ელექტრონული ფოსტა (electronic mail, e-mail) და ახალი ამბების ჯგუფი (Usenet) გახდა.

1980 წელს დამუშავებული იყო ქსელი Bitnet. ეს ქსელი, შეიქმნა ნიუ-იორკის უნივერსიტეტის სპეციალისტების მიერ (City University of New York) და მას დიდი მნიშვნელობა ჰქონდა, რადგანაც იგი არც სამხედრო, არც სამეცნიერო კვლევებისათვის არ



იყო განკუთვნილი და ამავე დროს ინტერნეტის შედგენილობაში შედიოდა.

1984 წელს შემოთავაზებული იყო სახელების დომენური სისტემა (Domain Name System, DNS), რომელიც სიმბოლურ სახელებს IP-მისამართებზე ასახავდა. ინტერნეტ DNS სამსახურის (სერვისი) მიზანი სიმბოლური და რიცხვითი სახელების დომენებს შორის შესაბამისობის დამყარება იყო.

1986 წელს, სტანდარტიზაციის საერთაშორისო ორგანიზაციამ (International Organization for Standardizations, ISO) მიიღო პირველი ISO-8879 სტანდარტი და მას უწოდა SGML (Standard Generalized Markup Language - მარკირების სტანდარტული განზოგადებული ენა). SGML - გამდიდრებული მეტა ენაა, რომელიც ნებისმიერი სახის ტექსტის ლოგიკური და სტრუქტურული მარკირების სისტემის აგების საშუალებას იძლეოდა.

*სტრუქტურული მარკირება* - ტექსტის დაფორმატება, რომელშიც მართვის კოდები გარე დოკუმენტის შესახებ არანაირ ინფორმაციას არ ფლობს, მხოლოდ საზღვრებსა და დაქვემდებარებაში მყოფ მის ნაწილებს მიუთითებს, ანუ მის ლოგიკურ სტრუქტურას იძლევა. SGML-დოკუმენტი დოკუმენტის ტიპის განსაზღვრას (Document Type Definition, DTD) მოითხოვს. DTD SGML-დოკუმენტს ერთად იგზავნება ან მასში შედის, რათა მომხმარებლების მიერ შექმნილი ტეგების ამოცნობა გახდეს შესაძლებელი. SGML ენამ ვერ მოიპოვა გავრცელება, თუმცა, მისმა იდეოლოგიამ მრავალ კომპიუტერულ დამუშავებაზე მოახდინა გავლენა.

1989 წელს ოქსფორდის უნივერსიტეტის ფიზიკის ბაკალავრიატის კურსდამთავრებულმა, ბირთვული კვლევის ევროპული ცენტრის (CERN) თანამშრომელმა ტიმ ბერნს ლიმ (Tim Berners-Lee) HTML (HyperText Markup Language - ჰიპერტექსტების მარკირების ენა) ენა შეიმუშავა, რომელიც მსოფლიო ქსელში

(World Wide Web, WWW) ვებ-დოკუმენტებისა და პროტოტიპების შექმნის ძირითადი ენა, ხოლო ტიმ ბერნს ლი კი ჰიპერბმულების იდეოლოგიის ავტორი გახდა. ფაქტია, რომ HTML ენა - ეს SGML-რეგვტების DTD სახით ჩამოყალიბებული ნაკრებია. HTML ენის შემთხვევაში დოკუმენტის ტიპის განსაზღვრა ბრაუზერში ინახება და ეს განსაზღვრა გაცილებით მარტივია, ვიდრე SGML ენის DTD.

თუმცა, არ შეიძლება ითქვას, რომ ეს ყველაფერი ტიმ ბერნს ლიმ მარტომ გამოიგონა. მთელი კაცობრიობა, დააგროვა რა საკმარისი ცოდნა, ამ აღმოჩენამდე ერთად მივიდა. ჯერ კიდევ XX საუკუნის შუაში ვანევარ ბუშმა (Vannevar Bush) მსოფლიოს განუცხადა ახალი სიტყვა - "ჰიპერტექსტი", ტედ ნელსონმა (Ted Nelson) კი შესთავაზა "დოკუმენტური სამყაროს" შექმნა, რომელიც კაცობრიობის მიერ შექმნილ ტექსტებს ერთმანეთთან ჯვარედინი მითითებით დააკავშირებდა, მაგრამ მეოცე საუკუნის სამოციანი წლების დასაწყისში ამან აკადემიურ წრეებში არავითარი რეაქცია არ გამოიწვია.

1990 წლის შემოდგომაზე CERN-ის თანამშრომლებმა ტიმ ბერნს ლის მიერ NeXTStep გარემოში შექმნილი პირველი ვებ-სერვერი და ვებ-ბრაუზერი მიიღეს.

ტექსტებთან სამუშაოდ სპეციალური ჰიპერტექსტის გადაცემის პროტოკოლი (HyperText Transfer Protocol, HTTP) იქნა დამუშავებული. ამ ტექნოლოგიამ ქსელის განვითარებას უზარმაზარი ბიძგი მისცა და ინტერნეტი ჭეშმარიტად საერთაშორისო ქსელი გახდა.

1991 წლის ზაფხულში WWW პროექტმა, რომელმაც ევროპაში მეცნიერების სამყარო დაიპყრო, ოკეანე გადაკვეთა და ამერიკულ Internet პროექტს შეუერთდა. რიგი ახალი აბრევიატურა წარმოიშვა: URL, HTTP, HTML. ახალი საინფორმაციო ტექნოლოგიების გამო ევროპასა და ამერიკას შორის მეტოქეობა წარმოიშვა, მაგრამ ეს მეტოქეობა ტიმ ბერნს ლის ამერიკის

შეერთებულ შტატებში გადასვლით დამთავრდა. 1994 წელს იგი მასაჩუსეტის ტექნოლოგიური ინსტიტუტის კათედრას, ხოლო შემდეგ - W3C (World Wide Web Consortium) საერთაშორისო კონსორციუმს ჩაუდგა სათავეში, რომელიც მსოფლიო ქსელის სტანდარტიზაციის ორგანოს როლს თამაშობდა.

ინტერნეტი მრავალი ლოკალური კომპიუტერული ქსელის გაერთიანებას წარმოადგენს. თუმცა ამბობენ, რომ ინტერნეტს მფლობელი არ ჰყავს და არავის არ ეკუთვნის.

მრავალმა კომპანიამ საკუთარი ბრაუზერის და მათთვის საკუთარი გაფართოებული HTML ენის გამოშვება დაიწყო, მაგალითად, 1993 წელს პირველი გრაფიკული ბრაუზერი Mosaic დამუშავდა. ამის შემდეგ ასზე მეტი ბრაუზერი გამოვიდა. მათ შორის ყველაზე პოპულარული Netscape Navigator და Microsoft Internet Explorer გახდა.

1991 წელს შექმნილი HTML ენა მუდმივად იხვეწება და ივსება. ყოველი ამ ცვლილების შედეგი შემდეგი ვერსიებია:

- HTML 1.0. ენის პირველი ვერსია მისი შექმნის წელი 1989 წელია.
- HTML 2.0. ეს ვერსია (RFC 1866) 1994 წელს შეიქმნა, HTML-ის IETF (Internet Engineering Task Force - ინტერნეტ საინჟინრო ჯგუფი) სამუშაო ჯგუფმა შემოიტანა შესაბამისი ცვლილებები და საბაზო ელემენტების ნაკრებმა HTML 2.0-ს სტანდარტის სტატუსი შესძინა.
- HTML 3.0. ენის ეს ვერსია 1996 წლის დასაწყისში შეიქმნა, იყო ბევრი ინოვაციები, მათ შორის ცხრილების შექმნის, მათემატიკური ფორმულების მონიშვნის, ნახატების ტექსტში ჩასმის, შენიშვნების ჩასმის და ა.შ. ტეგები. ჰიპერტექსტის მარკირების ძირითადი კონცეფციების დარღვევის გარეშე, 3.0 ვერსიის ავტორებმა შემოიტანეს

სტილების კასკადური ფურცლები - (Cascading Style Sheets, CSS).

- HTML 3.2. ენის ეს ვერსია 1996 წელს შეიქმნა. მათში გათვალისწინებული იყო დამატებითი ელემენტების: აპლეტების, სიმბოლოების რეგისტრის მითითების და ა.შ. შექმნა. HTML 3.2 გახდა კომპრომისული მცდელობა შეერიგებინათ Microsoft და Netscape, და მათი ორიგინალური გადაწყვეტილებანი გაერთიანებით.

- HTML 4.0. ენის ეს ვერსია 1997 წელს განხორციელდა, დაზუსტებული მეორე რეალიზაცია გამოვიდა 1998 წელს. HTML 4.0-ში შემოტანილი იქნა მულტიმედიური მხარდაჭერა, სკრიპტების ენა, აგრეთვე, შეზღუდული შესაძლებლობის მქონე მომხმარებლებისათვის დოკუმენტების ასახვა.

- HTML 4.01. ეს ვერსია 1999 წელს გამოვიდა. მასში გასწორებული იყო წინა ვერსიაში დაშვებული შეცდომები.

- HTML 5. ამ ვერსიის დამუშავება დაიწყო W3C სამუშაო ჯგუფში 2007 წელს და 2008 წლის 22 იანვარს ახალი სპეციფიკაციაში გამოჩნდა. სპეციფიკაცია ამჟამად დამუშავების პროცესშია და შეიძლება ამ სტადიაში კიდევ რამდენიმე წლის განმავლობაში დარჩეს.

HTML ენის მხარდასაჭერად მთელი რიგი პროგრამირების ენები შეიქმნა, როგორცაა Java, JavaScript, Perl, PHP, და ა.შ. ამ ენების საშუალებით დინამიური ვებ-გვერდების შექმნა გახდა შესაძლებელი, რომლებიც ფორმებს, ცხრილებს, ჩარჩოებს და სხვა ელემენტებს მოიცავენ.

1998 წლის თებერვალში, უკვე შემოგვთავაზეს XML ენა (Extensible Markup Language - მარკირების გაფართოებული ენა). ეს ენა HTML ენას იმით გავს, რომ დოკუმენტის სხვადასხვა ნაწილების აღწერისათვის აქაც ტეგები გამოიყენება. თუმცა, HTML

ენისაგან განსხვავებით XML დეველოპერებს საშუალებას აძლევს, რომ საკუთარი ტეგები განსაზღვროს და ინფორმაცია მათში აღქმის საკუთარი საშუალებების შესაბამისად ჩასვას. XML-ტეგები სიმბოლოების რეგისტრის მიმართაა მგრძობიარე. Internet Explorer-ის უახლესი ვერსია მხარს უჭერს XML. რაც შეეხება Netscape Navigator, Mozilla (სამუშაო ჯგუფი Netscape-ს შემადგენლობაში) მხოლოდ ექსპერიმენტების დონეზე ახდენს XML-ის მხარდაჭერას.

XHTML (Extensible HyperText Markup Language - ჰიპერტექსტის მარკირების გაფართოებული ენა) ენა - ეს HTML და XML ენების კომბინაციაა. W3C კონსორციუმი, რომელიც სხვადასხვა ქვეყნის 500-ზე მეტი ორგანიზაციის წარმომადგენელს აერთიანებს, HTML-ის ნაცვლად რეკომენდაციას XHTML ენას უწევს, მაგრამ დღეს დეველოპერების უმეტესობა უპირატესობას მაინც HTML ენას ანიჭებს.

Dynamic HTML (HTML დინამიური ენა) ენა - ეს არის DOM-ის (Document Object Model - დოკუმენტის ობიექტური მოდელი), JavaScript ენისა და სტილების კასკადური ფურცლების (CSS) ერთმანეთთან დააკავშირებისა და გაერთიანების მცდელობა. Dynamic HTML გაფართოებული მულტიმედიური მხარდაჭერით ინტერაქტიული ვებ-გვერდების შექმნის საშუალებას იძლევა. Dynamic HTML ენის მრავალი საშუალებანი HTML 4.01 სპეციფიკაციაშია აღწერილი.

Java ენა - პროგრამირების ენა Oak-იდანაა მიღებული, რომელიც ინტერნეტში მუშაობაზეა ორიენტირებული. Java ენისა და C++ ენის სინტაქსი მსგავსია. Java ენაზე დაწერილი პროგრამა შეიძლება ორ დიდ ჯგუფად დაიყოს. პირველ ჯგუფს მიეკუთვნება Java-დანართები, რომელიც Java მანქანის სპეციალური ინტერპრეტირებული მართვისათვისაა განკუთვნილი. ამ პროგრამების რეალიზაცია ყველა ძირითადი კომპიუტერული

პლატფორმისათვისაა განხორციელებული. მეორე ჯგუფი - ეს ე.წ. აპლეტებია.

*აპლეტი* - პატარა პროგრამაა, რომელიც Java ენაზეა დაწერილი და HTML-ფაილის შემადგენლობაში შედის. Java-აპლეტების შესრულების გარემოს სხვა პროგრამა უზრუნველყოფს, მაგალითად, როგორცაა ვებ-ბრაუზერი. დანართებისაგან განსხვავებით აპლეტების შესრულება უშუალოდ ოპერაციულ სისტემაში არ შეიძლება. აპლეტის კოდის ინტერპრეტირება ბრაუზერის ან სერვერის მიერ ხდება. იმისდა მიხედვით, რომელი პროგრამა გამოდის ინტერპრეტატორის როლში, აპლეტები ან კლიენტის მხარეს ან სერვერის მხარეს შეიძლება შესრულდეს. Java - არის კომპილირებადი ენა.

JavaScript - ეს სცენარების ენაა, რომელიც HTML-გვერდების შესაძლებლობების გაფართოების საშუალებას იძლევა. ეს ენა კორპორაცია Netscape Communications მიერ შექმნილ Netscape Navigator ბრაუზერში (JavaScript კლიენტისათვის) და სერვერულ პროდუქტებში (JavaScript სერვერისათვის) გამოსაყენებლად დამუშავდა. ამ ენის ბევრი ელემენტი, სტატიკური ვებ-გვერდების დამატებითი ფუნქციური შესაძლებლობები, ფირმა Sun Microsystem მიერ არის დამუშავებული.

Java-აპლეტისაგან განსხვავებით JavaScript ენაზე დაწერილი პროგრამები უშუალოდ HTML-დოკუმენტში უნდა ჩაიდგას, რისთვისაც <SCRIPT> წყვილი ტეგი გამოიყენება.

ზოგიერთი დამწყები მომხმარებელი ფიქრობს, რომ JavaScript ენა Java ენის ვერსიაა. ეს მცდარია, JavaScript-სა და Java-ს არაფერი აქვს საერთო. Java - დაპროგრამების ენაა, ხოლო JavaScript - კი ინტერპრეტატორი, სცენარების დასაწერი ენა.

PHP - სცენარების ენა, რომელიც სერვერის მიერ მუშავდება. ColdFusion მსგავსად, PHP-კოდი უშუალოდ HTML-დოკუმენტის შემადგენლობაში შედის. მისი თავდაპირველი სახელი იყო

Personal Home Page Tools (პერსონალური საშინაო გვერდი). მოგვიანებით, ამ ტექნოლოგიის, რომელიც საშუალებას იძლევა სერვერზე განთავსდეს ვებ-გვერდების სკრიპტები, რომელიც ძირითადად Unix, Linux და Solaris პლატფორმებზე მუშაობს, დასახელების გამოიფვრა სხვაგვარად მოხდა: PHP (Hypertext Preprocessor - ჰიპერტექსტის პრეპროცესორი).

PHP ენაში მრავალი ენის, როგორცაა C და Perl, საუკეთესო გადაწყვეტილებებია რეალიზებული; გარდა ამისა, PHP დეველოპერებს მონაცემთა ბაზებთან მუშაობის ძლიერი ინსტრუმენტებით უზრუნველყოფს. Perl შემდეგი გამოთქმის Practical Extraction and Report Language (პრაქტიკული ექსტრაქციისა და გამოხატვის ენა) აბრევიატურაა. Perl-ის მსგავსად PHP - ეს არის თავისუფლად გავრცელებადი ღია სისტემა, და მომხმარებელს მისი განვითარების საშუალება ეძლევა.

უფრო უკეთესი, მოხერხებული და ინტერაქტიული ვებ-დანართების შესაქმნელი ახალი ტექნოლოგია და არა ახალი დაპროგრამების ენა არის AJAX. AJAX - ეს არის შემდეგი ჩანაწერის Asynchronous JavaScript And XML (ასინქრონული JavaScript და XML) აბრევიატურა. AJAX იყენებს JavaScript-ს, რათა ვებ-ბრაუზერსა და ვებ-სერვერს შორის მონაცემების მიმოცვლა მოხდეს.

AJAX-დანართები ბრაუზერსა და დამოუკიდებელ პლატფორმას (კროს-პლატფორმა და კროს-ბრაუზერის ტექნოლოგია) წარმოადგენს. AJAX-თან ერთად ინტერნეტ-დანართები შესაძლოა უფრო მდიდარი (მცირე მოცულობის, სწრაფი და მარტივად გამოსაყენებელი) გახადოთ.

დამუშავების ცნობილ საშუალებებს ასევე მიეკუთვნება CGI (Common Gateway Interface), ASP, JSP (Java Server Pager), SSI (Server Side Includes), VRML (Virtual Reality Markup Language) და სხვები.

Apache - ეს ღია ვებ-სერვერია, ყველაზე ხშირად Unix, Linux და Solaris პლატფორმებისათვის გამოიყენება.

ASP (Active Server Pages) - ტექნოლოგია, რომელიც მძლავრი კლიენტ-სერვერის ინტერნეტ-დანართების დასაწერად გამოიყენება და რომელიც HTML დინამიური გვერდების შექმნის საშუალებას იძლევა. ვებ-გვერდების შესაქმნელად შემდეგი პროგრამების გამოყენებაა უკეთესი: Adobe GoLive, Microsoft FrontPage, Macromedia Dreamweaver. გამოსახულებათა რედაქტირებისთვის შეიძლება შემდეგი პროგრამების გამოყენება Adobe Photoshop, Macromedia Fireworks, Corel Photo-Paint, Adobe Illustrator, Macromedia Freehand, CorelDRAW, Adobe LiveMotion, Macromedia Flash. აუდიო და ვიდეო მონაცემებთან სამუშაოდ - პროგრამა Adobe Premier, Apple Final Cut Pro, Apple Quick Time, RealProducer, Windows Media.

HTML-გვერდების დასაწერი უამრავი მარტივი და რთული პროგრამები არსებობს. ყველა ისინი ორ ძირითადად კატეგორიად იყოფა: პროგრამები, რომლებიც ტეგების ჩასმის ავტომატიზებას ახდენს, და ჩვეულებრივი სტანდარტული ტექსტური რედაქტორები და პროცესორები, რომლებიც ნებისმიერი პლატფორმისათვისაა განკუთვნილი. თუმცა, ამ პროგრამების უმეტესობა, ვებ-გვერდებს ავტომატურად ქმნის, ამიტომ, HTML-ენის შესწავლა საჭიროა, რათა გაიგოთ, თუ როგორ იგება ვებ-გვერდი.



## ძირითადი ცნებები

HTML ენა (HyperText Markup Language - ჰიპერტექსტების მარკირების ენა) არის სტანდარტული ენა, WWW (World Wide Web - მსოფლიო ქსელში) გარემოში ჰიპერტექსტული დოკუმენტების შესაქმნელად. HTML-დოკუმენტების (ვებ-დოკუმენტები) დათვალიერება სხვადასხვა ტიპის ვებ-ბრაუზერებით შეიძლება. თუ დოკუმენტი HTML ენის გამოყენებითაა შექმნილი, მაშინ ვებ-ბრაუზერს, დოკუმენტის სხვადასხვა ელემენტისა და მათი პირველადი გადამუშავების გამოყოფისათვის, HTML-ის ინტერპრეტაცია შეუძლია.

HTML ენის მთავარი უპირატესობა ის არის, რომ დოკუმენტის დათვალიერება შესაძლებელია სხვადასხვა ტიპის და სხვადასხვა პლატფორმის ვებ-ბრაუზერის საშუალებით. HTML - ვებ-გვერდების შესაქმნელი ერთ-ერთი ყველაზე მარტივი ენაა.

WWW - ეს ინტერნეტში შემავალი ვებ-სერვერების (ქსელური კომპიუტერების) ერთობლიობაა, რომელშიც ტექსტური, გრაფიკული, ვიდეო-, აუდიო- და სხვა საინფორმაციო ფაილები ინახება.

WWW - ეს სერვერებზე განთავსებული უამრავი რაოდენობის დოკუმენტებზე, უნივერსალური წვდომის უზრუნველყოფაზე ორიენტირებული, ფართომასშტაბიანი ჰიპერმედია-გარემოა.

WWW - ეს არის ქსელური მულტიმედიური ჰიპერტექსტის მონაცემთა განაწილებული სტრუქტურა.

ათასობით ინფორმაციული სერვერი მთელ მსოფლიოშია გაბნეული. ნებისმიერი ახალი კომპიუტერი, რომელიც ინტერნეტში ჩაირთვება, შეიძლება თავად გახდეს სერვერი, სხვა სერვერებიდან მიიღოს საჭირო ინფორმაცია, ნებისმიერი მრავალი საფოსტო სამსახურის დახმარებით გააგზავნოს შეტყობინებები და მიიღოს პასუხები.

საძიებო სერვერებზე სხვადასხვა დოკუმენტების მილიონობით მიმართვები ინახება და მომხმარებლის მოთხოვნის შესაბამისად საჭირო ინფორმაციის ძებნა ხორციელდება. ელექტრონული გამოცემების გლობალური ძიება WWW ტექნოლოგიების გამოყენებით ხდება. დიდ სერვერებზე განთავსებული ცნობილი ინგლისურენოვანი და რუსულენოვანი საძიებო სისტემების გარდა, საკუთარი სერვერები საინფორმაციო სააგენტოებს, გამომცემლობებს, ბანკებსა და სხვა ორგანიზაციებსაც აქვთ.

ინტერნეტი - ეს ქსელების ერთობლიობაა, რომელიც ინფორმაციის გადასაცემად მიმოცვლის ერთიან პროტოკოლს იყენებს.

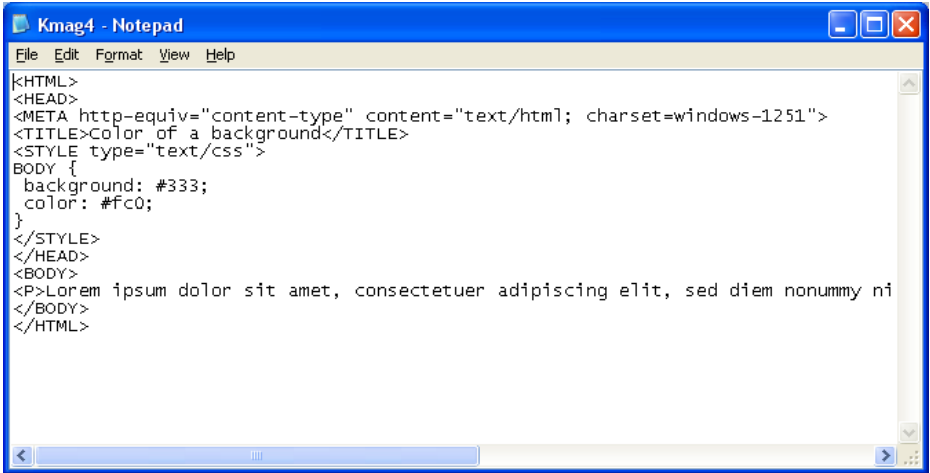
ვებ-საიტი (web site) - ეს არის სატიტულო გვერდისა და რამდენიმე მასთან დაკავშირებული ვებ-დოკუმენტების ერთობლიობა. თითოეულ ასეთ, ინტერნეტში ხელმისაწვდომ ერთობლიობას, შეიძლება ნებისმიერი ზომა და შინაარსი, ინფორმაციული და ემოციური ორიენტაცია ჰქონდეს.

ვებ-ოსტატი - პირი, რომელიც ვებ-სერვერებზე დოკუმენტების მოვლასა და მის მუდმივ განახლებაზეა პასუხისმგებელი.

როგორ იქმნება HTML-დოკუმენტი?

HTML-დოკუმენტები შეიძლება შეიქმნას ნებისმიერი ტექსტური რედაქტორის ან სპეციალიზებული HTML-რედაქტორებისა და კონვერტორების გამოყენებით.

HTML-დოკუმენტი - ASCII ფორმატში ჩაწერილი ჩვეულებრივი ფაილია. იგი ეფუძნება სპეციალურ დესკრიპტორებს - ტეგებს, რომლებიც მონაცემების დაფორმატების წესებს განსაზღვრავს. HTML ენაზე დაწერილი პროგრამის კოდი - ჩვეულებრივი ტექსტური ფაილია, ამიტომ უფრო მარტივია, უბრალოდ ჩავწეროთ ის Notepad-ში - რომლის გამოძახება შემდეგი ბრძანებით შეიძლება Start ⇒ All Programs ⇒ Accessories ⇒ Notepad. HTML ენაზე Notepad-ში ჩაწერილი პროგრამის მაგალითი ნახატ 1-ზეა მოცემული.

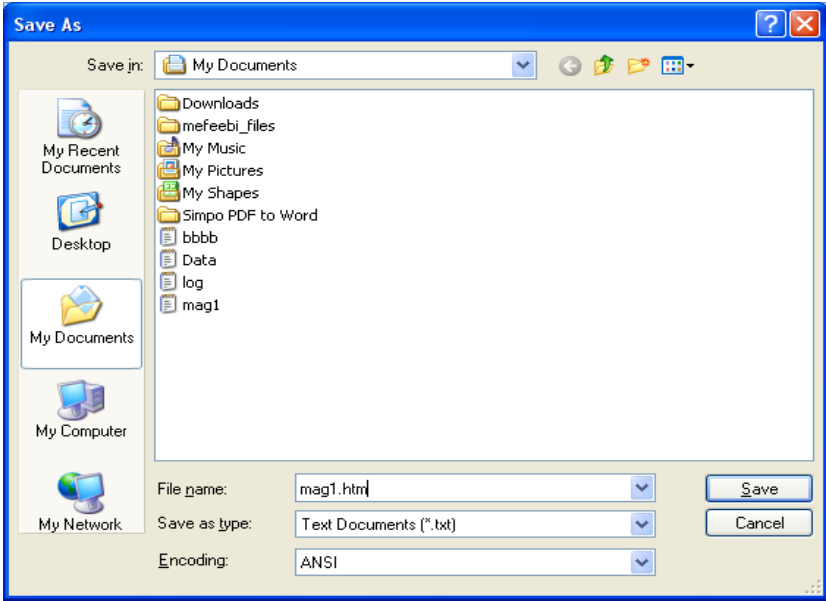


ნახ. 1. HTML ენაზე Notepad-ში ჩაწერილი პროგრამის მაგალითი

როდესაც პროგრამა ჩაწერილია, მისი შენახვა უნდა მოხდეს არა .txt, არამედ .htm (.html) გაფართოებით (ნახ. 2). HTML-ფაილის სახელი File name: ველში სასურველია ჩაწერილი იყოს პატარა ასოებით. მასში შეიძლება გამოყენებული იყოს ციფრები, ქვედა ხაზი და ტირე.

თუ HTML-დოკუმენტში ვებ-გვერდზე გამოსატანად ქართული ტექსტია გამოყენებული, მაშინ Encoding ჩამოშლად ველში უნდა ავირჩიოთ Unicode და ისე შევინახოთ მოცემული დოკუმენტი.

როგორც კი ფაილი .htm (.html) გაფართოებით ჩაიწერება, მისი ნიშნაკი დაუყოვნებლივ "ინტერნეტის" ნიშნაკით შეიცვლება, როგორც ეს ნახატ 3-ზეა ნაჩვენები. მეორე ფიგურა არის ფაილი .htm (.html) გაფართოებით Mozilla Firefox ბრაუზერისათვის, ხოლო, მესამე - Internet Explorer-სათვის.



ნახ. 2. ფაილის შენახვის დიალოგური ფანჯარა

თუ იყენებთ სხვა ოპერაციულ სისტემას, მაშინ შეგიძლიათ გამოიყენოთ მსგავსი მარტივი ტექსტური რედაქტორი. მაგალითად, Pico (Linux) ან TextEdit (Mac). HTML - ეს არის ტეგების ენა. ტეგის ქვეშ სპეციალური მართვის კოდები იგულისხმება, რომლებიც ტექსტში კუთხური ფრჩხილების (<, >) გამოყენებითაა ჩაწერილი. ტეგები, რომელიც გარს ერტყმის ტექსტურ ან გრაფიკულ ბრძანებებს, განსაზღვრავს ტექსტის დაფორმატების პარამეტრებს, ტექსტში სურათის ადგილმდებარეობას, ობიექტების ვებ-გვერდზე განთავსების პარამეტრებს.



ნახ. 3. TXT- და HTML-ფაილების ნიშნაკების მაგალითი

ყველა HTML-ტეგი იწყება მარცხენა კუთხური ფრჩხილით (<) და მთავრდება მარჯვენა კუთხური ფრჩხილით (>). როგორც წესი, არსებობს გახსნისა და დახურვის ტეგი. მოვიყვანოთ დოკუმენტის დასახელების განმსაზღვრელი ტეგების მაგალითი:

<TITLE> დოკუმენტის დასახელება </TITLE>

თითქმის ყველა ტეგი წყვილ-წყვილად არსებობს, ყოველ გახსნის ტეგს შეესაბამება დახურვის ტეგი. ამის გარდა, არსებობს არაწყვილი ტეგებიც, როგორიცაა, <BR>, <AREA>, მაგრამ ასეთი ტეგები ბევრი არ არის.

დახურვის ტეგი ისევე გამოიყურება, როგორც გახსნის ტეგი, მხოლოდ იგი კუთხური ფრჩხილის შემდეგ დახრილი ხაზით იწყება. ზემოთ მოყვანილ მაგალითში ტეგი <TITLE> ვებ-ბრაუზერს სათაურის ფორმატის გამოყენების, ხოლო ტეგი </TITLE> - სათაურის ტექსტის დასრულების შესახებ ატყობინებს .

ზოგიერთი ტეგები, როგორიცაა <BR> (ტეგი ახალ ხაზზე გადასვლას განსაზღვრავს), არ საჭიროებს დახურვის ტეგს, მაგრამ მისი გამოყენება ტექსტს ორიგინალობას აძლევს.

HTML სიმბოლოები რეგისტრის მიმართ არ არის მგრძობიარე, და ზემოთ მოყვანილი მაგალითი შეიძლება ასეც ჩაგვეწერა:

<title> დოკუმენტის დასახელება </title>

HTML-დოკუმენტში დამატებული ზედმეტი ინტერვალები, ტაბულაციისა და სტრიქონის დამთავრების სიმბოლოები, რათა იგი უფრო გასაგები გახდეს, ბრაუზერის მიერ დოკუმენტის ინტერპრეტირების დროს იქნება იგნორირებული. HTML-დოკუმენტი შეიძლება შეიცავდეს აღნიშნულ ელემენტებს, მხოლოდ იმ შემთხვევაში თუ ისინი <PRE> და </PRE> ტეგებს შორის იქნებიან განთავსებულნი. მთელი პროგრამა რიგი ტეგებისაგან შედგება. ტეგებს შეიძლება ჰქონდეთ პარამეტრები და ატრიბუტები, რომლებიც ტეგების დამატებით როლს თამაშობენ.

ზოგჯერ ტეგებს დესკრიპტორებს უწოდებენ. დესკრიპტორი (აღწერა, Descriptor - ლათ.) - საინფორმაციო-საძიებო ენის ლექსიკური ერთეულია (სიტყვა, ფრაზა), რომელიც რაიმე ტექსტის ძირითად სემანტიკურ შინაარსს გვიჩვენებს. დესკრიპტორი საინფორმაციო-საძიებო სისტემაში დოკუმენტების ინფორმაციული ძებნის პროცესში გამოიყენება.

უმეტესობა ბრაუზერისთვის არა აქვს მნიშვნელობა თუ რომელ რეგისტრშია ჩაწერილი პროგრამის კოდი და ტეგი. ჩანაწერებს <HTML>, <html> ან <HtMl> ჩვეულებრივ ერთი და იგივე შედეგი აქვს.

## ბრაუზერი

WWW-ში HTML-დოკუმენტების დათვალიერებისათვის სპეციალური პროგრამული უზრუნველყოფაა საჭირო. ასეთ პროგრამებს ბრაუზერებს უწოდებენ (ინგლ. *browse* - გადაფურცვლა, დათვალიერება, წაკითხვა). მათი მეშვეობით, შეიძლება სასურველი ვებ-გვერდი ჩავტვირთოთ და დავათვალიეროთ, ინტერნეტში ნავიგაცია განვახორციელოთ და ა.შ.

ბრაუზერი - HTML ფაილების დათვალიერების პროგრამა, HTML ენის პროგრამა თარჯიმანია. ბრაუზერი ეკრანზე ბრძანებების (ტეგები) ინტერპრეტირებით აჩვენებს ტექსტს და გრაფიკულ გამოსახულებას.

ვებ-ბრაუზერი - ეს არის გამოყენებითი პროგრამა, რომელიც ლოკალური ან დაშორებული კომპიუტერიდან მიღებული ფაილის შინაარსს, ამ ფაილში ჩაწერილი ინსტრუქციების შესაბამისად, აჩვენებს. ბრაუზერი კლიენტის პროგრამას წარმოადგენს და შეკითხვების ვებ-სერვერზე გასაგზავნად HTTP (HyperText Transfer Protocol - ჰიპერტექსტის გადაცემის პროტოკოლი) პროტოკოლს იყენებს.

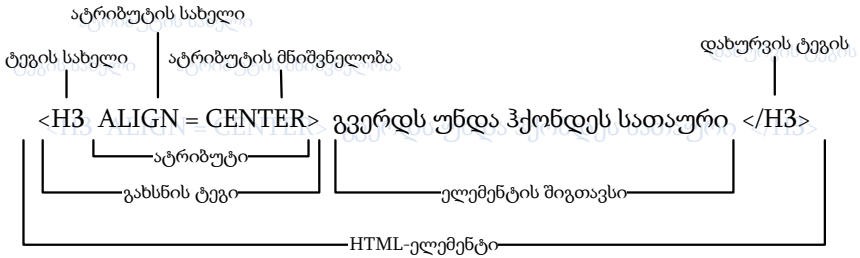
არსებობს საკმაოდ მრავალი ბრაუზერი, რომელთა შორის ყველაზე პოპულარული ბრაუზერებია Microsoft Internet Explorer, Netscape Navigator, Mozilla Firefox, Opera, Chrome, Safari და NCSA Mosaic. ბრაუზერი, HTML-ფაილის წაკითხვის შემდეგ, ტეგების გამოყენებით ახდენს დოკუმენტში განთავსებული მონაცემების ინტერპრეტირებას და შესაბამისად გამოანათებს მათ კომპიუტერის ეკრანზე. ბრაუზერებს Netscape Navigator და Internet Explorer აქვს სიახლის წამკითხველი ჩაშენებული პროგრამები (Newsreader).

## ტეგები და მათი ატრიბუტები

ტეგი - HTML ენის კოდია, რომლითაც საწყისი ტექსტის მონიშვნა ხორციელდება. ტეგი კუთხურ ფრჩხილებში იწერება. როგორც უკვე აღვნიშნეთ, ყველა ტეგი იწყება „<“ სიმბოლოთი და მთავრდება „>“ სიმბოლოთი. გახსნილი კუთხური ფრჩხილის შემდეგ იწერება ტეგის სახელი (ბრძანება). თითოეულ ტეგს შეიძლება ჰქონდეს ატრიბუტები. ყველა ატრიბუტის ჩამოთვლის შემდეგ გახსნილი კუთხური ფრჩხილი უნდა დაიხუროს. გახსნის ტეგის შემდეგ იწერება ტეგის შიგთავსი. HTML-დოკუმენტის კოდი საბოლოო, ანუ დახურვის ტეგით მთავრდება (ნახ. 4). დახურვის ტეგი დახრილი ხაზით (/) იწყება. ტეგების გარეშე ბრაუზერს ან სხვა მსგავს პროგრამას არ შეუძლია დოკუმენტის ფორმატის იდენტიფიცირება და მისი სწორად ინტერპრეტირება.

<ტეგი> </ტეგი> – მსგავსი კონსტრუქციის ტეგებს ჰქვიათ ტეგი-კონტეინერები. ისინი შეიძლება სხვა ტეგებსა და ტექსტს შეიცავდნენ. ზოგიერთი ტეგი, მაგალითად, ტეგი <BR> არ მოითხოვს დახურვის ტეგს.

ტეგების გახსნისა და დახურვის მიმდევრობა შემდეგნაირად გამოიყურება:



ნახ. 4. HTML-ელემენტის სტრუქტურა

<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.

ტეგების ჩაწერის დროს რეგისტრი მხედველობაში არ მიიღება. თუ ტეგი შეცდომითაა ჩაწერილი, მაშინ პროგრამა მის იგნორირებას აკეთებს, თუმცა ამის შესახებ ინფორმაციას არ იძლევა. თვითონ ტეგების გამოტანა არ ხდება, მაგრამ ისინი დოკუმენტის გამოტანაზე ახდენენ ზეგავლენას.

ატრიბუტები - დამატებითი მმართველი სიტყვებია, რომელიც ტეგისაგან და ერთმანეთისაგან ინტერვალითაა დაშორებული. შეიძლება ითქვას, რომ ატრიბუტები - ტეგის თვისებების სახელებია, რომლებსაც გარკვეული მნიშვნელობების მიღება შეუძლია. ატრიბუტები მხოლოდ გახსნის ტეგებს აქვთ, ხოლო დახურვის ტეგებს არ აქვთ. ატრიბუტები ბრაუზერის მიერ ტეგების ინტერპრეტაციაზე მოქმედებს.

ზოგიერთი ატრიბუტს კონკრეტული მნიშვნელობა აქვს, რომელიც ტოლობის ნიშნის შემდეგ უნდა მიეთითოს. ადრე ატრიბუტის მნიშვნელობა ბრჭყალებში უნდა ჩაგვესვა, მაგრამ ახლა უმეტეს შემთხვევაში, ეს ბრჭყალები შეიძლება გამოიტოვოს, რაც შედეგებზე არ იმოქმედებს.



## დოკუმენტის სტრუქტურა

როდესაც ვებ-ბრაუზერი მიიღებს დოკუმენტს, იგი ტეგებით განსაზღვრავს, თუ როგორ უნდა იყოს დოკუმენტი ინტერპრეტირებული.

- *<HTML>* ტეგი. პირველი ტეგი, რომელიც დოკუმენტში გვხვდება, უნდა იყოს *<HTML>* ტეგი. ეს ტეგი ბრაუზერს ატყობინებს, რომ დოკუმენტი HTML ენაზეა დაწერილი. მინიმალური HTML-დოკუმენტი შეიძლება ასე გამოიყურებოდეს:

`<html> ... დოკუმენტის ტანი ... </html>`

ამგვარად, *<HTML>* და *</HTML>* ტეგები HTML-დოკუმენტისათვის ე. წ. კონტეინერს ქმნიან.

- *<HEAD>* ტეგი. სტრუქტურულად დოკუმენტი ორ ნაწილად იყოფა: სათაურის და ძირითადი ნაწილი, ანუ დოკუმენტის ტანი. სათაურის ნაწილი მოთავსებულია *<HEAD>* და *</HEAD>* ტეგებს შორის, ხოლო ძირითადი ნაწილი *<BODY>* და *</BODY>* ტეგებს შორის.

- *<TITLE>* ტეგი. *<HEAD>* და *</HEAD>* კონტეინერის შიგნით უნდა განთავსდეს მხოლოდ ერთადერთი სავალდებულო *<TITLE>* და *</TITLE>* კონტეინერი, რომელიც სათაურის ტექსტს შეიცავს.

დოკუმენტის სათაურის ნაწილის ტეგი უნდა მიეთითოს უშუალოდ *<HTML>* ტეგის შემდეგ და სხვაგან არსად დოკუმენტის ტანის სხვა ნაწილში. მოცემული ტეგი არის დოკუმენტის განზოგადებული აღწერა. *<HEAD>* ტეგის შიგნით რაიმე ტექსტის განთავსებას თავი უნდა ავარიდოთ. *<HEAD>* გახსნის ტეგი უშუალოდ *<TITLE>* ტეგის და სხვა დოკუმენტის აღწერის ტეგების წინ უნდა მივუთითოთ, ხოლო *</HEAD>* დახურვის ტეგი - უშუალოდ დოკუმენტის აღწერის დამთავრებისთანავე.

ვებ-ბრაუზერების უმეტესობა *<TITLE>* ტეგის შიგთავსს ფანჯრის სათაურში, ხოლო დოკუმენტის შინაარსს სანიშნობელ

ფაილში, თუ ის მხარდაჭერილია ვებ-ბრაუზერის მიერ, ასახავს. ბრაუზერის ფანჯარაში დოკუმენტის გამოტანის შემთხვევაში დოკუმენტის სათაური არ ჩანს.

დოკუმენტის ტანის ტეგები ფანჯარაში გამოტანილი HTML-დოკუმენტის კომპონენტების იდენტიფიცირებას ახდენენ. დოკუმენტის ტანი შეიძლება სხვა დოკუმენტებზე მიმართვას, ტექსტს და სხვა ფორმატიზებულ ინფორმაციას შეიცავდეს.

• *<BODY>* ტეგი. დოკუმენტის ტანი *<BODY>* და *</BODY>* ტეგებს შორის უნდა იყოს მოთავსებული. ეს დოკუმენტის ის ნაწილია, რომელიც ტექსტური და გრაფიკული ინფორმაციის გამოტანას უზრუნველყოფს.

უნდა აღინიშნოს, რომ *<HTML>*, *<HEAD>* და *<BODY>* გახსნისა და დახურვის ტეგები აუცილებელი არ არის, მაგრამ დაჟინებით მოითხოვენ მის გამოყენებას, რადგან იგი ვებ-ბრაუზერს საშუალებას აძლევს დოკუმენტის სათაურისა და ტანის ნაწილი ერთმანეთისაგან მკაცრად განაცალკევოს.

*<BODY>* წყვილი ტეგია. ამ წყვილებს შორის პროგრამის ყველა ის ტეგი განთავსდება, რომლებიც დოკუმენტის ძირითად შინაარსს წარმოადგენს. *<BODY>* ტეგი დოკუმენტის იმ ნაწილს გამოყოფს, რომელიც ეკრანზე მომხმარებელმა უნდა ნახოს.

## HTML ენის ვერსია

კომპიუტერის დამწყები მომხმარებელი ვებ-გვერდის შექმნას *<HTML>* ტეგით იწყებს. მაგრამ თუ ვებ-გვერდს განმეორებით სხვა სახელით შევინახავთ, მაშინ მის დასაწყისში, პირველ ტეგად თავისი ატრიბუტებით შეიძლება *<!DOCTYPE>* ტეგი გამოჩნდეს.

*<!DOCTYPE>* ტეგი მოცემულ დოკუმენტში გამოყენებულ HTML ენის ვერსიას გვიჩვენებს. ეს ცნობა SGML-განცხადების

სახით დოკუმენტის პირველ სტრიქონში განთავსდება. <!DOCTYPE> დოკუმენტში პირველი ტეგი უნდა იყოს. იგი პირველად HTML-2.0 ვერსიაში გამოჩნდა.

თეორიულად, <!DOCTYPE> ტეგი ვებ-სერვერს დოკუმენტის დამუშავების საშუალებების არჩევაში უნდა დაეხმაროს. იგი ვებ-სერვერს აწვდის ინფორმაციას, თუ რომელი დესკრიპტორები შეიძლება აღმოჩნდეს მოცემულ გვერდზე, თუმცა ამჟამად, ბრაუზერის მიერ მისი იგნორირება ხდება, ზოგადად რომ ვთქვათ, მისი მითითება აუცილებელი არ არის. მისი სინტაქსია:

```
<!DOCTYPE ზედა_დონის_ელემენტი წვდომის_დონე  
"რეგისტრაციის_ნიშანი//ორგანიზაცია//ობიექტის_ტიპი//ჭდე  
//ენის_კოდი"URI">
```

მაგალითი:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
```

ქვემოთ <!DOCTYPE> ტეგის პარამეტრებია ჩამოთვლილი.

ზედა დონის ელემენტი დოკუმენტის ტიპის (Document Type Definition, DTD) განსაზღვრის დროს ცხადდება. ვინაიდან HTML ენა, ტექსტური დოკუმენტების მარკირების მიზნით, დოკუმენტთა აღწერის სტანდარტული ზოგადი ენისაგან (Standard General Markup Language, SGML) განვითარდა, სწორედ ამით აიხსნება დოკუმენტის ტიპის მიწოდების აუცილებლობა. დოკუმენტებში, რომლებშიც შეიძლება ესა თუ ის ელემენტი შეგვხვდეს, სხვადასხვა ტიპის დოკუმენტებს (document type) მიაკუთვნეს. იმისათვის, რომ მოხდეს დოკუმენტის მონიშვნის ელემენტის აღწერა, დოკუმენტის ტიპის სტანდარტული განსაზღვრების შერჩევაა საკმარისი. HTML-დოკუმენტისათვის HTML მნიშვნელობა გამოიყენება .

ვინაიდან, HTML-დოკუმენტი საერთო წვდომის დოკუმენტს წარმოადგენს, ამიტომ წვდომის დონისათვის PUBLIC მნიშვნელობა გამოიყენება.

რეგისტრაციის ნიშანი გვიჩვენებს, მოცემული ორგანიზაცია ISO-ში დარეგისტრირებულია (+) თუ არა (-). W3C (World Wide Web Consortium) კონსორციუმი ISO-ში დარეგისტრირებული არ არის, ამიტომ უნდა მივუთითოთ ნიშანი - მინუსი.

ორგანიზაციის სახით მიეთითება ორგანიზაცია, რომელიც DTD-ს შექმნასა და მის მხარდაჭერაზე პასუხისმგებელი. HTML-დოკუმენტის ტიპის განსაზღვრის მხარდაჭერაზე პასუხისმგებელია W3C კონსორციუმი, ამიტომ W3C მნიშვნელობა გამოიყენება.

HTML ენის შემთხვევაში ობიექტის ტიპად DTD მნიშვნელობა გამოიყენება.

ქვე HTML ენის ვერსიას განსაზღვრავს. HTML-4.0 ვერსიისათვის HTML 4.0, HTML 4.0 Transitional ან HTML 4.0 Frameset მნიშვნელობები გამოიყენება.

ენის კოდი - ISO-ში ორასოიანი აღნიშვნაა მიღებული. მაგალითად, ინგლისური ენის კოდია - EN.

URI იდენტიფიკატორი DTD მდებარეობას განსაზღვრავს. HTML-4.0 ვერსია შემდეგ URI იდენტიფიკატორებს უჭერს მხარს:

`http://www.w3.org/TR/REC-html40/strict.dtd;`

`http://www.w3.org/TR/REC-html40/loose.dtd;`

`%o http://www.w3.org/TR/REC-html40/frameset.dtd;`

`%o http://www.w3.org/TR/REC-html40/HTMLlat1.ent;`

`%o http://www.w3.org/TR/REC-html40/HTMLsymbol.ent;`

`%o http://www.w3.org/TR/REC-html40/HTMLspecial.ent .`

`<!DOCTYPE>` ტეგით დოკუმენტის ზოგადი სქემა შემდეგნაირად გამოიყურება:

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0`

`Transitional//EN">`

`<html>`

```
<head>
<title>დოკუმენტის სათაური</title>
</head>
<body>
ტეგის შემცველი დოკუმენტის ტანი, ეს პროგრამის ის ნაწილია,
რომელიც ეკრანზე გამოსატანი ინფორმაციის მართვის
საშუალებას იძლევა
</body>
</html>
```

## META-ინფორმაცია

ჩვეულებრივი ტეგების გარდა, კიდევ ეგრეთ წოდებული META-ტეგები არსებობს. ისინი ბრაუზერების მიერ არ გამონათდება, მაგრამ შეიძლება საძიებო სისტემისათვის მნიშვნელოვან, რაიმე სამოსამსახურო ინფორმაციას შეიცავდეს (მაგალითად, საგასაღებო სიტყვას, ავტორზე მიმართვას, აღწერას).

როგორც <!DOCTYPE> ტეგი, ასევე META-ინფორმაცია ზოგჯერ გვერდის ახალი სახელით პირველად შენახვის შემდეგ შეიძლება ავტომატურად წარმოიშვას და <HEAD> და </HEAD> ტეგებს შორის განთავსდება:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<title>დოკუმენტის სათაური</title>
<meta http-equiv=content-type content="text/html; charset=windows-
1251">
<meta content="mshtml 6.00.2600.0" name=generator></head>
<body>
```

ტეგის შემცველი დოკუმენტის ტანი, ეს პროგრამის ის ნაწილია, რომელიც ეკრანზე გამოსატანი ინფორმაციის მართვის საშუალებას იძლევა

</body>

</html>

<META> ტეგი ვებ-გვერდის შესახებ დამატებითი ინფორმაციის განსათავსებლად გამოიყენება, მათ შორის მონაცემები დოკუმენტზე, მის ავტორზე, შექმნის თარიღი, საძიებო სისტემისათვის საგასაღებო სიტყვების სია და ა. შ. META-ტეგის ზოგადი სინტაქსი ასეთია:

<meta name="meta-ტეგის\_სახელი" content="meta-ტეგის\_შინაარსი">

კიდევ ერთი ვარიანტი:

<meta http-equiv=" meta-ტეგის\_სახელი " content=" meta-ტეგის\_შინაარსი">

ამ ორ ატრიბუტს შორის სხვაობა გახლავთ ის, რომ NAME ატრიბუტის დამუშავება ბრაუზერის მიერ არ ხდება, ხოლო HTTP-EQUIV-ს კი ამუშავებს.

## ატრიბუტი NAME

ქვემოთ <META> ტეგის მაგალითები NAME ატრიბუტებითაა მოყვანილი.

დოკუმენტის შინაარსის აღწერა:

<meta name="description" content="რესურსის აღწერა">

რეკომენდირებულია, რომ რესურსის აღწერა მცირე ზომის გაკეთდეს (არაუმეტეს 250-300 სიმბოლოსი), მაგალითად:

<meta name = "description" content = "ეს გვერდი განკუთვნილია html ენის საცნობარო ინფორმაციის განსათავსებლად">

საძიებო სისტემისათვის საგასაღებო სიტყვების სია:

<META NAME="keywords" CONTENT="საგასაღებო სიტყვები">

სიაში საგასაღებო სიტყვები ერთმანეთისაგან ინტერვალით ან მძიმით გამოიყოფა. ეს სიტყვები საძიებო სისტემას თავის კატალოგში გვერდების ინდექსაციისათვის ესაჭიროება, მაგალითად:

```
<meta name="keywords" content="HTML html JavaScript javascript  
css CSS VBScript vbscript VRML vrml php PHP Perl perl ტეგი  
ატრიბუტი პროგრამა ბრაუზერი">;
```

დოკუმენტის ავტორი, მისი ელექტრონული ფოსტის მისამართი, სხვა მონაცემები:

```
<meta name="author" content="ინფორმაცია_ავტორის_შესახებ">  
პროგრამის დასახელება, რომლის გამოყენებითაც შეიქმნა
```

დოკუმენტი:

```
<meta name="generator" content="პროგრამის_დასახელება">  
დოკუმენტის ვარგისიანობის ვადა:
```

```
<meta name = "expires" content = "თარიღი">
```

ელექტრონული ფოსტის მისამართი:

```
<meta name="Reply-to" content="სახელი@მისამართი">
```

დოკუმენტზე საავტორო უფლების აღწერა:

```
<meta name="Copyright" content="საავტორო_უფლება">
```

## ატრიბუტი HTTP-EQUIV

ქვემოთ <META> ტეგის მაგალითებია მოყვანილი HTTP-EQUIV ატრიბუტის გამოყენებით.

შემდეგი ჩანაწერი მოცემული დროის გასვლის შემდეგ საშუალებას იძლევა გარკვეული მისამართზე დაბრუნდეს.

```
<meta http-equiv="Refresh" content="0;_URL=http://www.gtu.ge">
```

მოცემულ შემთხვევაში ბრაუზერის მიერ გვერდის ჩატვირთვის შემდეგ მაშინვე მოხდება გადასვლა მისამართზე

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

შიგთავსის MIME-ტიპი და მისი კოდირება:

```
<meta http-equiv="Content-Type" content="text/html;  
charset=windows-1251">
```

მოცემულ შემთხვევაში MIME-ტიპად მითითებულია მნიშვნელობა text/html, ანუ ჰიპერტექსტი, ხოლო კოდირებისათვის კი - მნიშვნელობა windows-1251. საერთოდ, კოდირების ზუსტად მითითება რეკომენდირებული არ არის, რადგანაც ბრაუზერები მათ შერჩევას შესანიშნავად ართმევენ თავს.

დოკუმენტის ვარგისიანობის ვადა:

```
<meta http-equiv="expires" content="Mon, 7 Nov 2011 10:00:00 GMT">
```

თარიღი RFC-850 ფორმატში მიეთითება (ელექტრონული ფოსტის სტანდარტი):

- Mon - კვირის დღე;
- 7 Nov რიცხვი და თვე;
- 2011 - წელი;
- 10:00:00 - დრო (საათი, წუთი, წამი);
- GMT - დროის სარტყელი (მოცემულ შემთხვევაში

თბილისის დრო).

მოცემული დროის გასვლის შემდეგ დოკუმენტი ლოკალური კეშიდან წაიშლება. ამ პროცედურაში მონაწილეობს როგორც ბრაუზერი, ასევე სერვერი. თუ ნულოვანი თარიღი მიეთითება, მაშინ დოკუმენტი ჩატვირთვისთანავე წაიშლება, ანუ კეში მისი შენახვა საერთოდ არ მოხდება.

## HTML დოკუმენტის შენახვა და გახსნა

უპირველეს ყოვლისა, თქვენი მომავალი Web-გვერდისათვის ცალკე საქალაღე შექმენით. შემდეგ გახსენით



Notepad (Start ⇒ Programs ⇒ Accessories ⇒ Notepad) და მასში ტექსტი აკრიფეთ. მაგალითი:

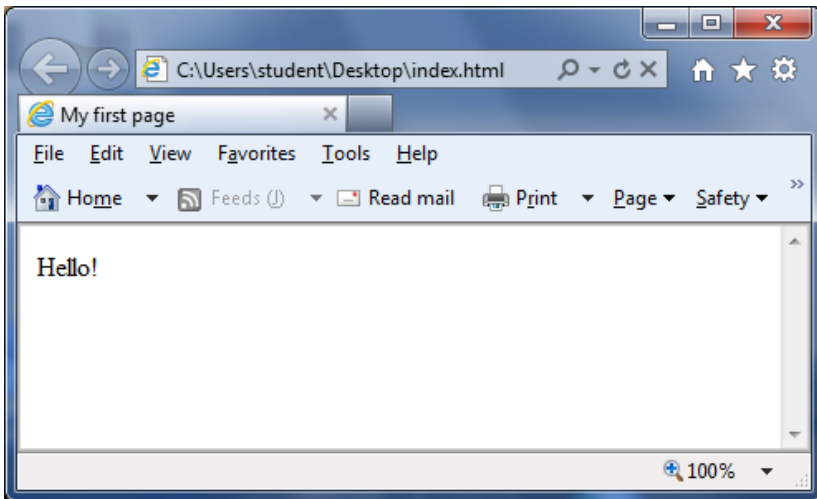
```
<html>
<head><title>My first page</title></head>
<body>
Hello!
</body>
</html>
```

შეინახეთ ფაილი შემდეგნაირად:

- File ⇒ Save as;
- შემდეგ შეიტანეთ დოკუმენტის სახელი. მაგალითად, index.html (არ აკრიფოთ მხოლოდ index; აუცილებლად მიუთითეთ გაფართოება html ან htm). გახსენით ბრაუზერი, მაგალითად Internet Explorer და დაათვალიერეთ თქვენს მიერ შექმნილი დოკუმენტი.

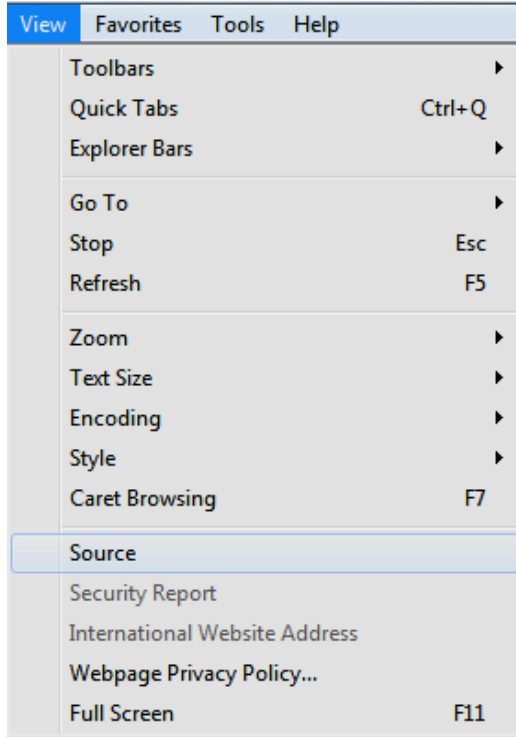
File ⇒ Open ⇒ Browse ⇒ index.html

ჩვენ მიერ შექმნილი მარტივი Web-გვერდი ასე გამოიყურება:



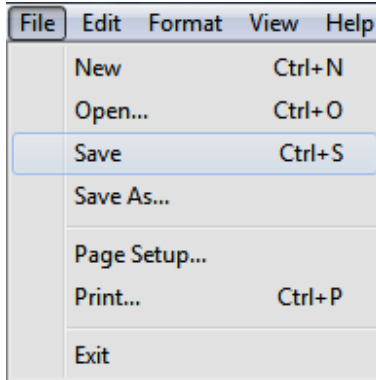
## ცვლილებების შეტანა პროგრამაში

თუ ვებ-დოკუმენტის (ფაილი .htm გაფართოებით) კოდის, რომელიც გაიხსნა Internet Explorer ბრაუზერის საშუალებით, დათვალიერება საჭირო, შეიძლება გაიხსნას Internet Explorer ბრაუზერის ფანჯარაში შემდეგი ბრძანებით View ⇒ Source (იხ. ნახ. 5).



ნახ. 5. დოკუმენტის HTML-კოდის დათვალიერება

რედაქტირებისათვის HTML-ფაილი Notepad-ში უნდა გაიხსნას. პროგრამის კოდის ცვლილებების დასრულების შემდეგ, გამოიყენეთ ფაილის შენახვის ბრძანება File ⇒ Save ან <Ctrl>+<S> კლავიშთა კომბინაცია (ნახ. 6).



ნახ. 6. პროგრამის შენახვა

ცვლილებების სანახავად Internet Explorer ბრაუზერში გამოვიყენოთ ბრძანება View ⇒ Refresh ან <F5> კლავიში. ყველა შემდგომი ცვლილებების შემთხვევაში ოპერაცია უნდა განმეორდეს. ზოგიერთ ბრაუზერში პროგრამაში განახლებები შეიძლება სხვა გზითაც განხორციელდეს.

## <Body> ტეგის ატრიბუტები

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

```
<body bgcolor="#ffffff" leftmargin=0 topmargin=4 marginwidth=0  
marginheight=4 link="#000099" vlink="#000099" alink="#cc0000">
```

- BACKGROUND ატრიბუტი გრაფიკულ გამოსახულებას განსაზღვრავს, რომელიც დოკუმენტის ფონს მოზაიკის მსგავსად შეავსებს. გამოსახულებიანი ფაილი GIF ან JPEG ფორმატით უნდა იყოს შენახული. მაგალითი:

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

თუ გამოსახულებიანი ფაილი იგივე საქალაქდემია ჩაწერილი, რომელშიც დოკუმენტია, მაშინ URL-მისამართისა და გზის მითითება არ დაგვჭირდება.

- BGCOLOR ატრიბუტი დოკუმენტში ფონის ფერს, ფერის ინტენსივობის მოდელის RGB (Red, Green, Blue - წითელი, მწვანე, ლურჯი) თექვსმეტობითი მნიშვნელობით, ან ხაზოვანი ლიტერალით ანუ შესაბამისი ფერის სახელით განსაზღვრავს, მაგალითად:

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

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

- TEXT ატრიბუტი იმ გამოყენებული ტექსტის ფერს განსაზღვრავს, რომელიც ჰიპერბმული არ არის. ჩუმათობის პრინციპით, ეს ტექსტი შავია. მაგალითი:

```
<body text="ფერი">
```

- LINK ატრიბუტი ჰიპერბმულის ფერს განსაზღვრავს, ჩუმათობის პრინციპით, უმეტეს ბრაუზერებში, ის მუქი ლურჯი ფერისაა. მაგალითი:

```
<body link="ფერი">
```

- ALINK ატრიბუტი აქტიური ჰიპერბმულის ფერს

განსაზღვრავს, რომელიც მასზე მაუსის დაწკაპუნების მომენტში იცვლება. სასურველია, იგი ფონის ფერისაგან, რომელიც LINK ატრიბუტითაა მოცემული, განსხვავდებოდეს. მაგალითი:

`<body alink="ფერი">`

• VLINK ატრიბუტი უკვე ნანახი ჰიპერბმულის ფერს განსაზღვრავს. სასურველია, იგი ფონის ფერისა (LINK ატრიბუტითაა მოცემული) და აქტიური ჰიპერბმულის ფერისაგან (ALINK ატრიბუტითაა მოცემული) განსხვავდებოდეს. მაგალითი:

`<body vlink="ფერი">`

• BGPORPERTIES ატრიბუტი ფონური გამოსახულების თვისებებს განსაზღვრავს. ამჟამად, ბრაუზერები მის მხოლოდ ერთადერთ fixed მნიშვნელობას უჭერს მხარს. თუ ეს ატრიბუტი მითითებულია, მაშინ ვებ-გვერდის ფონად გამოყენებული გამოსახულება ტექსტის გადაფურცვლის დროს უძრავად განთავსდება:

`<body bgproperties=fixed>`

• TOPMARGIN ატრიბუტი გვერდის ზედა მინდვრის საზღვარს განსაზღვრავს პიქსელებში. მაგალითი:

`<body topmargin=რიცხვი>`

• BOTTOMMARGIN ატრიბუტი გვერდის ქვედა მინდვრის საზღვარს განსაზღვრავს პიქსელებში. მაგალითი:

`<body bottommargin=რიცხვი>`

• LEFTMARGIN ატრიბუტი გვერდის მარცხენა მინდვრის საზღვარს განსაზღვრავს პიქსელებში. მაგალითი:

`<body leftmargin=რიცხვი>`

• RIGHTMARGIN ატრიბუტი გვერდის მარჯვენა მინდვრის საზღვარს განსაზღვრავს პიქსელებში. მაგალითი:

`<body rightmargin=რიცხვი>`

ქვემოთ მარტივი Web-გვერდის შექმნის მაგალითებია მოცემული:

პროგრამა 1-ის შესაბამის ფანჯარაში მოცემულია Web-გვერდი, რომლის ზედა, მარცხენა, მარჯვენა და ქვედა საზღვრები მითითებული არ არის და უდრის 0. ჩვენ ვხედავთ, ტექსტი, როგორ არის მიკრული მის გვერდებზე, ზედა და ქვედა მინდვრები არ შეინიშნება. მარჯვენა კიდეზე ტექსტი მხოლოდ იმის გამო არ არის მიკრული, რომ ტექსტის ორივე მხრიდან სწორების ბრძანება არ არის გამოყენებული.

**პროგრამა 1.** Web-გვერდის შექმნის მაგალითი, რომლისთვისაც მინდვრის ზომები მითითებული არ არის:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> ჩემი პირველი პროგრამა</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR=FFFF05 TOPMARGIN=0 BOTTOMMARGIN=0
```

```
LEFTMARGIN=0 RIGHTMARGIN=0>
```

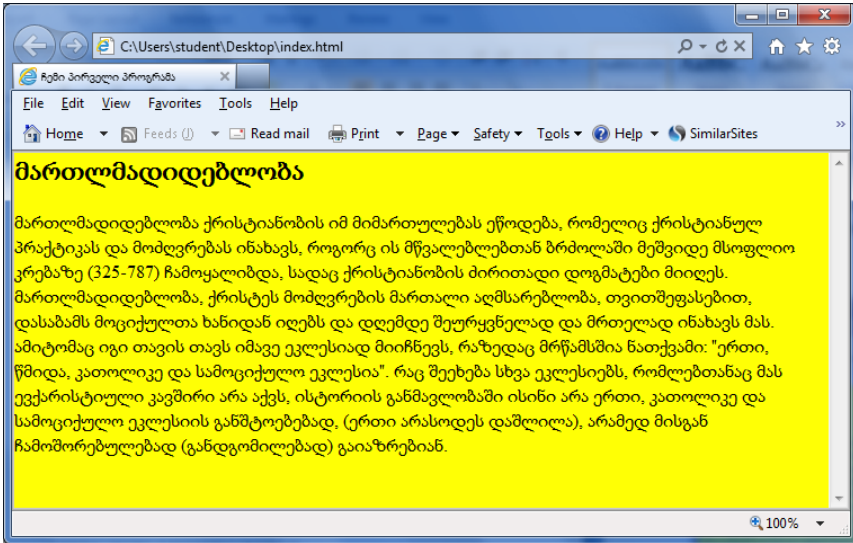
```
<H2> მართლმადიდებლობა</H2>
```

მართლმადიდებლობა ქრისტიანობის იმ მიმართულებას ეწოდება, რომელიც ქრისტიანულ პრაქტიკას და მოძღვრებას ინახავს, როგორც ის მწვალებლებთან ბრძოლაში მეშვიდე მსოფლიო კრებაზე (325-787) ჩამოყალიბდა, სადაც ქრისტიანობის ძირითადი დოგმატები მიიღეს. მართლმადიდებლობა, ქრისტეს მოძღვრების მართალი აღმსარებლობა, თვითშეფასებით, დასაბამს მოციქულთა ხანიდან იღებს და დღემდე შეურყვნელად და მრთელად ინახავს მას. ამიტომაც იგი თავის თავს იმავე ეკლესიად მიიჩნევს, რაზედაც მრწამსშია ნათქვამი: "ერთი, წმიდა, კათოლიკე და სამოციქულო ეკლესია". რაც შეეხება სხვა ეკლესიებს, რომლებთანაც მას ექვარისტიული კავშირი არა აქვს, ისტორიის განმავლობაში ისინი არა ერთი, კათოლიკე და სამოციქულო ეკლესიის განშტოებებად,

(ერთი არასოდეს დაშლილა), არამედ მისგან ჩამოშორებულებად (განდგომილებად) გაიაზრებიან.

</BODY>

</HTML>



შემდეგ ნახატზე მოცემულია Web-გვერდი, რომლისთვისაც ვველა მინდორი 20 პიქსელის ზომის არის (პროგრამა 2). ჩვენ ვხედავთ, რომ ტექსტი ზედა, მარცხენა და მარჯვენა კიდეებიდან დაცილებულია.

**პროგრამა 2.** Web-გვერდის შექმნის მაგალითი, რომლისთვისაც მინდვრის ზომები მითითებულია:

<HTML>

<HEAD>

<TITLE> ჩემი პირველი პროგრამა </TITLE>

</HEAD>

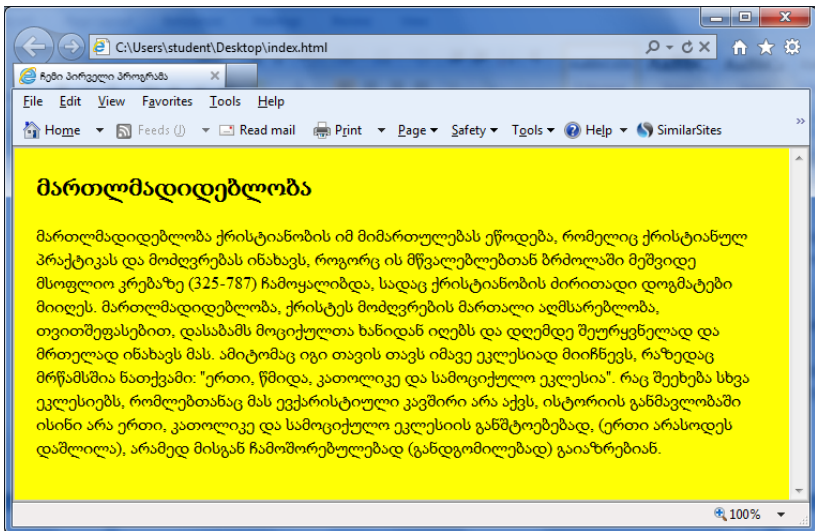
<BODY BGCOLOR=FFFF05 TOPMARGIN=20 BOTTOMMARGIN=20  
LEFTMARGIN=20 RIGHTMARGIN=20>

<H2> მართლმადიდებლობა</H2>

მართლმადიდებლობა ქრისტიანობის იმ მიმართულებას ეწოდება, რომელიც ქრისტიანულ პრაქტიკას და მოძღვრებას ინახავს, როგორც ის მწვალებლებთან ბრძოლაში მეშვიდე მსოფლიო კრებაზე (325-787) ჩამოყალიბდა, სადაც ქრისტიანობის ძირითადი დოგმატები მიიღეს. მართლმადიდებლობა, ქრისტეს მოძღვრების მართალი აღმსარებლობა, თვითშეფასებით, დასაბამს მოციქულთა ხანიდან იღებს და დღემდე შეურყვნელად და მრთელად ინახავს მას. ამიტომაც იგი თავის თავს იმავე ეკლესიად მიიჩნევს, რაზედაც მრწამსშია ნათქვამი: "ერთი, წმიდა, კათოლიკე და სამოციქულო ეკლესია". რაც შეეხება სხვა ეკლესიებს, რომლებთანაც მას ექვარისტიული კავშირი არა აქვს, ისტორიის განმავლობაში ისინი არა ერთი, კათოლიკე და სამოციქულო ეკლესიის განშტოებებად, (ერთი არასოდეს დაშლილა), არამედ მისგან ჩამოშორებულებად (განდგომილებად) გაიაზრებიან.

</BODY>

</HTML>





## ვებ-გვერდების სტანდარტული ფერები

ვებ-გვერდების მრავალი მომხმარებელი დოკუმენტის ფონისთვის, ჩვეულებრივი ტექსტისათვის და ჰიპერბმულებისათვის განსაზღვრულ ფერს იყენებს. ამ ფერების მისაცემად <BODY> ტეგში დამატებითი პარამეტრების გამოყენებაა საჭირო:

```
<BODY BGCOLOR="#XXXXXX" TEXT="#XXXXXX" LINK="#XXXXXX">
```

აქ თითოეული პარამეტრი განსაზღვრავს გვერდის ამა თუ იმ ელემენტის ფერს:

- BGCOLOR - დოკუმენტის ფონის ფერი;
- TEXT - დოკუმენტის ჩვეულებრივი ტექსტის ფერი;
- LINK - ჰიპერბმულის ფერი.

Html-ში ფერები განსაზღვრება ციფრებით თექვსმეტობით კოდში. ფერების გამა ეყრდნობა სამ ძირითად ფერს: წითელს, მწვანესა და ლურჯს და RGB-თი აღინიშნება. ფერი შეიძლება აღიწეროს როგორც მისი დასახელებით, ასევე მისი მნიშვნელობით RGB (Red, Green, Black) პალიტრაში. თითოეული ფერისათვის 00-დან FF-მდე თექვსმეტობითი მნიშვნელობა მიეთითება, რასაც ათობით სისტემაში 0-255 დიაპაზონი შეესაბამება. შემდეგ ეს მნიშვნელობები ერთ რიცხვში ერთიანდებიან, 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 ჰიპერბმულს განსაზღვრავს.

ქვემოთ თექვსმეტი სტანდარტული ფერი, თავისი შესაბამისი თექვსმეტობითი კოდებითაა მოცემული:

## 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

მოლისფერი - Lime - #00FF00

ყვითელი - Yellow - #FFFF00

ლურჯი - Blue - #0000FF

ვარდისფერი - Fuchsia - #FF00FF

ცისფერი - Aqua - #00FFFF

თეთრი - White - #FFFFFF

## პროგრამა 4.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ფერადი ფონი</TITLE>
```

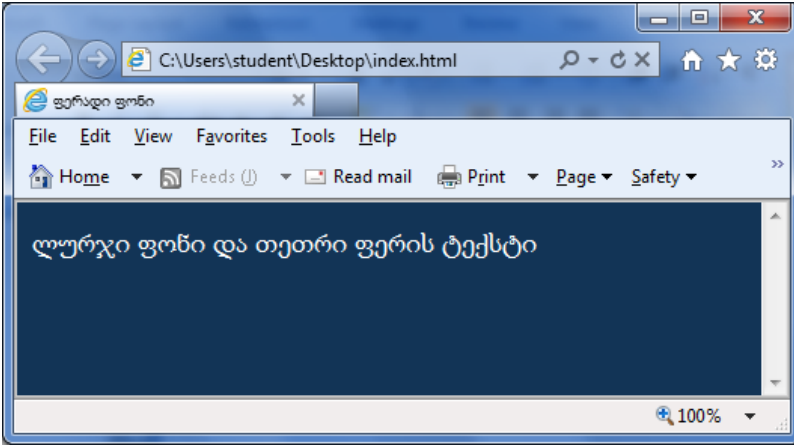
```
</HEAD>
```

```
<BODY BGCOLOR=123456 TEXT=FFFFFF>
```

```
<BIG>ლურჯი ფონი და თეთრი ფერის ტექსტი</BIG>
```

```
</BODY>
```

```
</HTML>
```



## ფონური სურათები

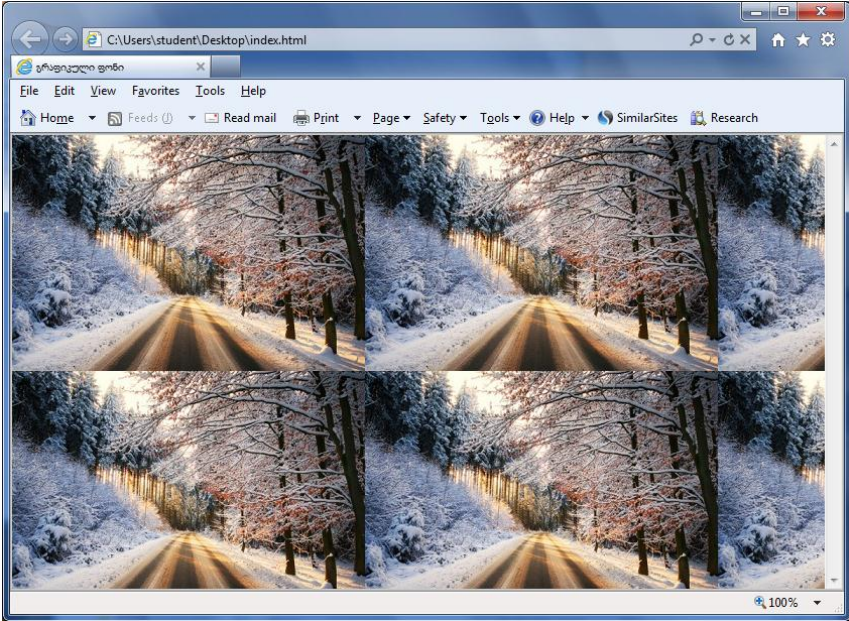
ბრაუზერების უმეტესობა დოკუმენტში ფონური სურათების განთავსების საშუალებას იძლევა, რომელიც მთელი დოკუმენტის უკანა მხარეს აისახება. ზოგ მომხმარებელს ვებ-გვერდზე განთავსებული ფონური გრაფიკა მოსწონს, ზოგს არა. მრავალ დოკუმენტში ფონის სახით გამოყენებული სურათი საკმაოდ ეფექტურია. ფონური სურათების აღწერა <BODY> ტეგში ხდება და შემდეგი სახით გამოიყურება:

```
<BODY BACKGROUND = "picture.gif">
```

### პროგრამა 3.

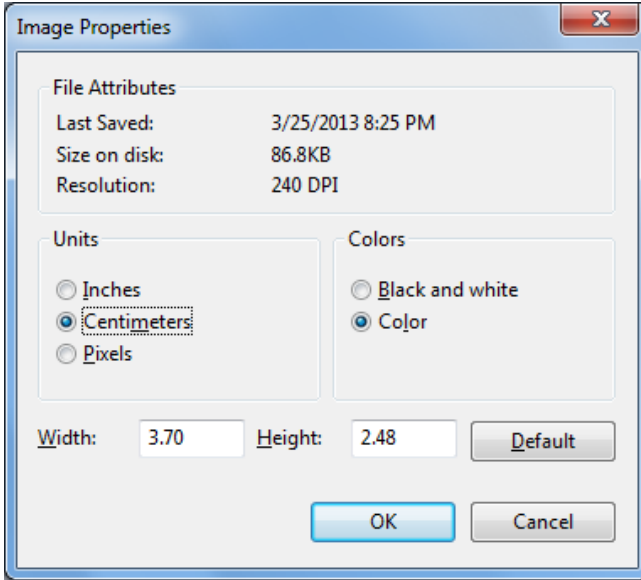
```
<HTML>  
<HEAD>  
<TITLE> გრაფიკული ფონი</TITLE>  
</HEAD>  
<BODY BACKGROUND = winter.jpg>  
</BODY>  
</HTML>
```

ქვემოთ ნაჩვენებია ამ პროგრამის შედეგი:



დოკუმენტის ფონად არჩეულია სურათი ზომით 3,70×2,48. მიუხედავად სურათის მცირე ზომისა, სურათმა მთლიანად დაფარა გვერდის მთელი ფართობი მრავალი ფილის სახით. ქვემოთ პირველ ნახატზე მოცემულია საწყისი სურათი, ხოლო მეორე ნახატზე ამ სურათის ზომის დიალოგური ფანჯარა.





## სათაურის დონეები

სიდიდით შემოუსაზღვრელი ტექსტები ფორმალურად სათაურების სხვადასხვა დონის საშუალებით შეიძლება ცალკეულ თავებად და აბზაცებად დავყოთ. სათაურის პირველი - ყველაზე მაღალი დონე აღინიშნება ციფრი 1-ით, შემდეგი დონე - 2, და ა.შ. ნაკლები ნომრის მქონე ტეგი უფრო მაღალი დონის სათაურს აღნიშნავს. ბრაუზერების უმეტესობა სათაურების ექვს დონეს უჭერს მხარს. 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 გვერდის მიმართ), CENTER (ცენტრის მიმართ), RIGHT (მარჯვენა გვერდის მიმართ) და JUSTIFY (ორივე გვერდის მიმართ) მნიშვნელობებს ღებულობს.

## პროგრამა 5.

<HTML>

<HEAD>

<TITLE> Head </TITLE>

</HEAD>

<BODY>

<H1> Head 1 – Georgian Technical University </H1>

<H2> Head 2 – Georgian Technical University </H2>

<H3> Head 3 – Georgian Technical University </H3>

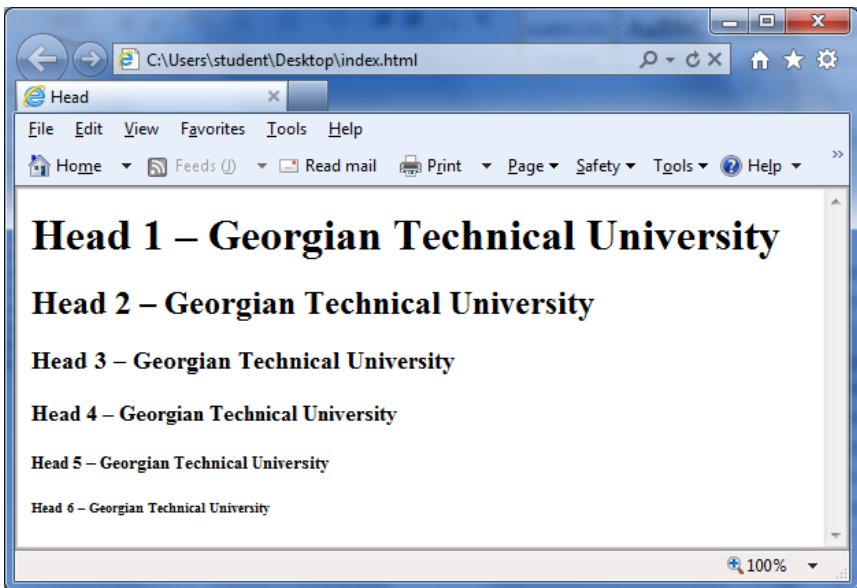
<H4> Head 4 – Georgian Technical University </H4>

<H5> Head 5 – Georgian Technical University </H5>

<H6> Head 6 – Georgian Technical University </H6>

</BODY>

</HTML>



სურათზე სხვადასხვა დონის სათაურის მაგალითია მოცემული.

მოცემული ტეგი პირველად HTML-2.0-ში გამოჩნდა.

## შრიფტი

<FONT> 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. შრიფტი შეიძლება მიეთითოს როგორც ციფრით (ჩუმათობის პრინციპით იგი 3-ის ტოლია), ასევე მისი საბაზო მნიშვნელობის მიმართ, უარყოფითი ან დადებითი მიმართულებით წანაცვლება. ქვემოთ მოყვანილ ცხრილში შრიფტის ზომებსა და ტიპურ სიდიდეებს შორის დამოკიდებულება არის მოცემული:

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

შრიფტის ზომა შემდეგი ბრძანებით შეიძლება შეიცვალოს

<FONT SIZE=+|- n>

საბაზო მნიშვნელობის შეცვლა კი შემდეგი ბრძანებითაა შესაძლებელი:

<BASEFONT SIZE=n>

### პროგრამა 1.

<HTML>

<HEAD>

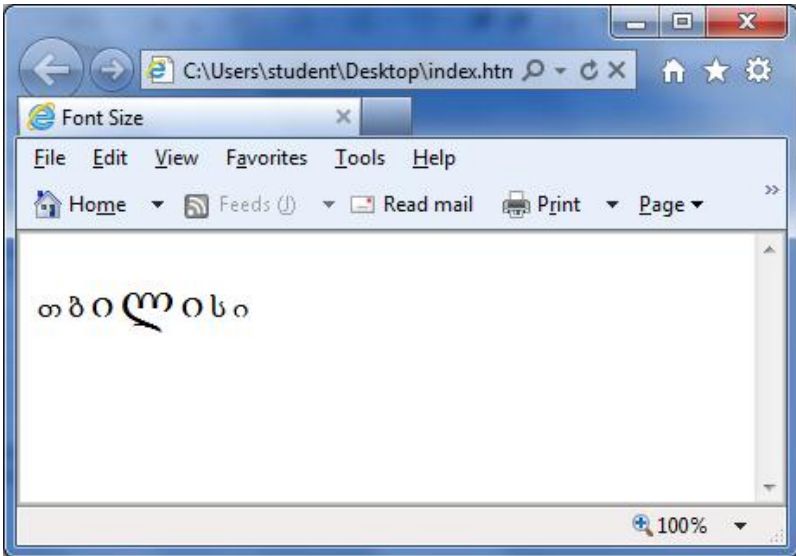
<TITLE>Font Size </TITLE>

```

</HEAD>
<BODY>
<P>თ
<FONT SIZE=+1>ბ</FONT>
<FONT SIZE=+2>ი</FONT>
<FONT SIZE=+3>ლ</FONT>
<FONT SIZE=+2>ო</FONT>
<FONT SIZE=+1>ს</FONT>
ო</P>
</BODY>
</HTML>

```

ქვემოთ მოცემულია სხვადასხვა ზომის შრიფტის გამოტანის მაგალითი:



მოცემული ტეგი პირველად HTML-3.2 ვერსიაში გამოჩნდა. ამჟამად, მისი გამოყენება არ არის რეკომენდირებული. მის მაგივრად სასურველია CSS საშუალებების გამოყენება.



• COLOR ატრიბუტი გამოიყენება შრიფტის ფერის შესაცვლელად. მისი სინტაქსია:

```
<FONT COLOR="#xxxxxx">
```

ფერი RGB მოდელით უნდა მივუთითოთ, რასაც ფერის თექვსმეტობითი ფორმატი შეესაბამება. მაგალითად, FFFFFFF თეთრი ფერია, 000000 - შავი, 0000FF - ლურჯი და ა.შ.

**პროგრამა 2.** სხვადასხვა ფერის ტექსტის გამოტანის მაგალითი

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Font Color </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

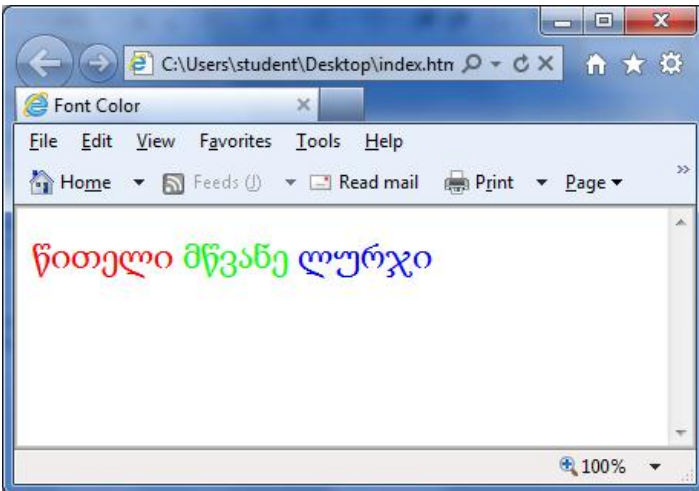
```
<FONT SIZE=5 COLOR="#FF0000"> წითელი </FONT>
```

```
<FONT SIZE=5 COLOR="#00FF00"> მწვანე </FONT>
```

```
<FONT SIZE=5 COLOR="#0000FF"> ლურჯი </FONT>
```

```
</BODY>
```

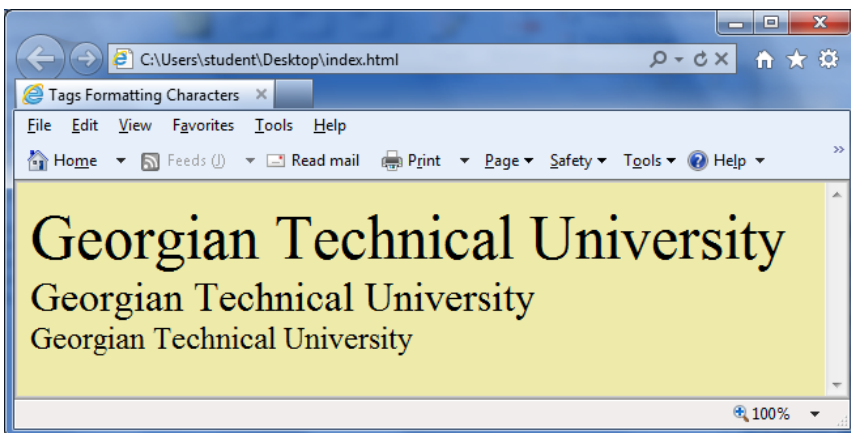
```
</HTML>
```



<BASEFONT> ტეგი შრიფტის საბაზო ზომას მთელი დოკუმენტისათვის განსაზღვრავს. მისი ატრიბუტია SIZE, რომელსაც იგივე მახასიათებლები აქვს რაც <FONT> ტეგის შემთხვევაში ჰქონდა. იგი გახსნის ტეგს წარმოადგენს და მას დახურვის ტეგი არ სჭირდება.

**პროგრამა 3.** <BASEFONT> ტეგის შემცველი ვებ-გვერდი

```
<HTML>
<HEAD>
<TITLE> Tags Formatting Characters </TITLE>
</HEAD>
<BODY BGCOLOR=#EEEEAAA>
<BASEFONT SIZE=8> Georgian Technical University <BR>
<BASEFONT SIZE=6> Georgian Technical University <BR>
<BASEFONT SIZE=5> Georgian Technical University
</BODY>
</HTML>
```



<BASEFONT> ტეგი პირველად HTML-3.2 ვერსიაში გამოჩნდა. ამჟამად, მოცემული ტეგის მაგივრად სასურველია CSS

საშუალებების, ხოლო შრიფტის ზომის შესაცვლელად <FONT> ტეგის გამოყენება.

## აბზაცი და სტრიქონების წყვეტა

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

<P> ტეგს აქვს ALIGN - ტექსტის სწორების ატრიბუტი. ამ ატრიბუტს შეუძლია შემდეგი მნიშვნელობები მიიღოს:

LEFT - ტექსტის სწორება მარცხენა კიდის მიმართ;

RIGHT - ტექსტის სწორება მარჯვენა კიდის მიმართ;

CENTER - ტექსტის სწორება ცენტრის მიმართ;

JUSTIFY - ტექსტის სწორება ორივე კიდის მიმართ.

მოცემული ტეგი პირველად HTML-2.0 ვერსიაში გამოჩნდა. მასში გახსნისა და დახურვის სხვა ტეგები არ შეიძლება იყოს გამოყენებული, ანუ <P> და </P> ტეგებს შორის შეიძლება იყოს მხოლოდ ტექსტი და ახალ სტრიქონზე გადასვლის <BR> ტეგი.

<BR> ტეგი ბრაუზერს ახალ სტრიქონზე გადასვლას ატყობინებს.

### პროგრამა 4.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> MY FIRST PROGRAMM</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFF05" TOPMARGIN=20
```

```
BOTTOMMARGIN=20 LEFTMARGIN=20 RIGHTMARGIN=20>
```

```
<H2> South Caucasus </H2>
```

<P>The South Caucasus is a geopolitical region located on the border of Eastern Europe and Southwest Asia also referred to as Transcaucasia, or The Trans-Caucasus. More specifically, the South Caucasus area spans the southern portion of the Caucasus Mountains and its lowlands, straddles the border between the two continents of Europe and Asia, extends from the southern part of the Greater Caucasus mountain range of southwestern Russia southerly to the Turkish and Armenian borders, and from the Black Sea to the Caspian Sea coast of Iran. </P>

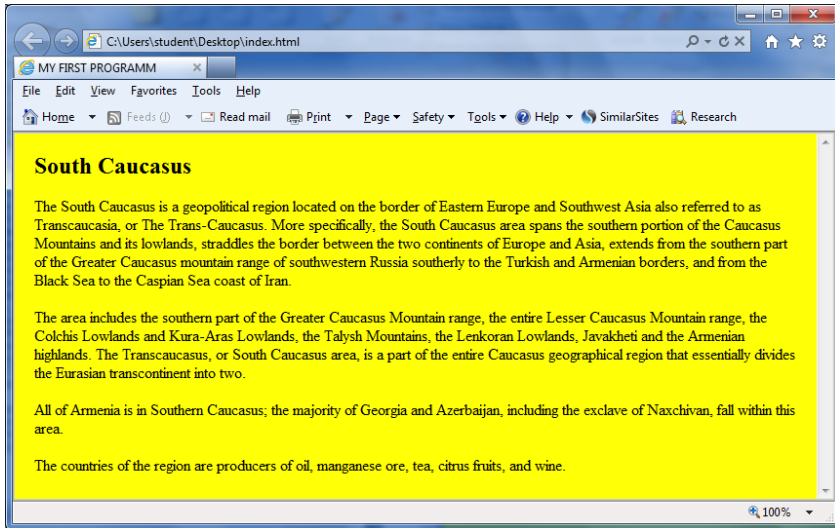
<P>The area includes the southern part of the Greater Caucasus Mountain range, the entire Lesser Caucasus Mountain range, the Colchis Lowlands and Kura-Aras Lowlands, the Talysh Mountains, the Lenkoran Lowlands, Javakheti and the Armenian highlands. The Transcaucasus, or South Caucasus area, is a part of the entire Caucasus geographical region that essentially divides the Eurasian transcontinent into two. </P>

<P>All of Armenia is in Southern Caucasus; the majority of Georgia and Azerbaijan, including the exclave of Naxchivan, fall within this area. </P>

<P>The countries of the region are producers of oil, manganese ore, tea, citrus fruits, and wine.</P>

</BODY>

</HTML>



მე-5 პროგრამაში მოყვანილია მაგალითი, სადაც ფონური სურათია გამოყენებული და ამასთან მისი ატრიბუტი არის `BGPROPERTIES=fixed`.

### პროგრამა 5.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> გრაფიკული ფონი</TITLE>
```

```
</HEAD>
```

```
<BODY TEXT=white BACKGROUND=winter.jpg
```

```
BGPROPERTIES=fixed TOPMARGIN=20 BOTTOMMARGIN=20
```

```
LEFTMARGIN=20 RIGHTMARGIN=20>
```

```
<H2> South Caucasus </H2>
```

```
<P>The South Caucasus is a geopolitical region located on the border of Eastern Europe and Southwest Asia also referred to as Transcaucasia, or The Trans-Caucasus. More specifically, the South Caucasus area spans the southern portion of the Caucasus Mountains and its lowlands, straddles the border between the two continents of Europe and Asia,
```

extends from the southern part of the Greater Caucasus mountain range of southwestern Russia southerly to the Turkish and Armenian borders,

and from the Black Sea to the Caspian Sea coast of Iran. </P>

<P>The area includes the southern part of the Greater Caucasus Mountain range, the entire Lesser Caucasus Mountain range, the Colchis Lowlands and Kura-Aras Lowlands, the Talysh Mountains, the Lenkoran Lowlands, Javakheti and the Armenian highlands. The Transcaucasus, or South Caucasus area, is a part of the entire Caucasus geographical region that essentially divides the Eurasian transcontinent into two. </P>

<P>All of Armenia is in Southern Caucasus; the majority of Georgia and Azerbaijan, including the exclave of Naxchivan, fall within this area. </P>

<P>The countries of the region are producers of oil, manganese ore, tea, citrus fruits, and wine.</P>

</BODY>

</HTML>



<DIV> (ინგლ.*division* - განყოფილება) ტეგი საშუალებას იძლევა დოკუმენტის სტრუქტურაში რამდენიმე ნაწილი გამოვყოთ. ის კონტეინერული ტეგია და <P> ტეგის მსგავსად ფუნქციონირებს. თუ დახურვის </P> ტეგი გამოტოვებულია, მაშინ <DIV> ეფექტურად ცვლის მას და ახალ აზრს იწვევს. მას შეიძლება ჰქონდეს ALIGN ატრიბუტი, რომელსაც LEFT, CENTER ან RIGHT მნიშვნელობების მიღება შეუძლია და ტექსტის სწორებას უზრუნველყოფს. ყოველი შემდეგი განყოფილება უარყოფს ALIGN ატრიბუტის წინა მნიშვნელობას. მისი სინტაქსია:

<DIV ALIGN=სწორება> განყოფილების ტექსტი </DIV>

### პროგრამა 6.

<HTML>

<HEAD>

<TITLE> MY FIRST PROGRAMM</TITLE>

</HEAD>

<BODY BGCOLOR =“#FFFF05“ TOPMARGIN=20

BOTTOMMARGIN=20 LEFTMARGIN=20 RIGHTMARGIN=20>

<H2> South Caucasus </H2>

<DIV ALIGN=CENTER>The South Caucasus is a geopolitical region located on the border of Eastern Europe and Southwest Asia also referred to as Transcaucasia, or The Trans-Caucasus. More specifically, the South Caucasus area spans the southern portion of the Caucasus Mountains and its lowlands, straddles the border between the two continents of Europe and Asia, extends from the southern part of the Greater Caucasus mountain range of southwestern Russia southerly to the Turkish and Armenian borders, and from the Black Sea to the Caspian Sea coast of Iran. </DIV><BR>

<DIV ALIGN=LEFT>The area includes the southern part of the Greater Caucasus Mountain range, the entire Lesser Caucasus Mountain range,

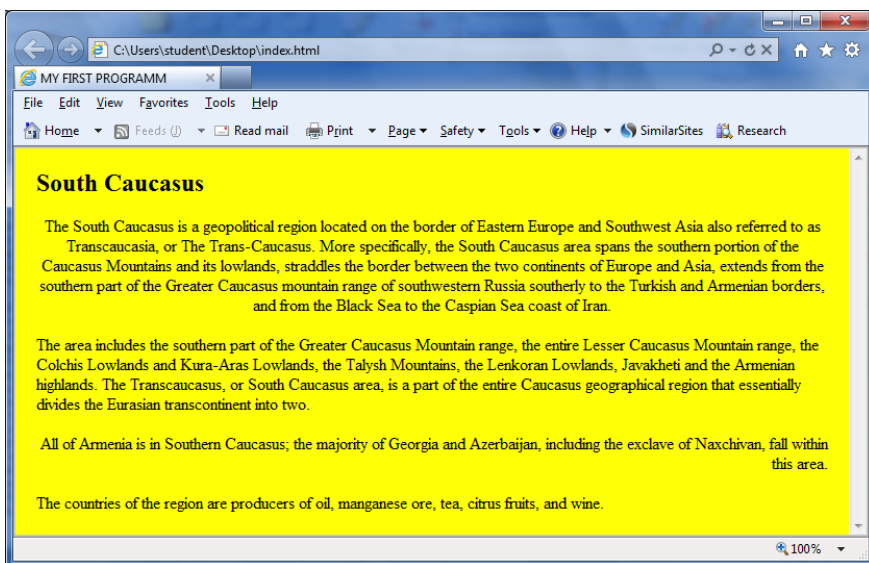
the Colchis Lowlands and Kura-Aras Lowlands, the Talysh Mountains, the Lenkoran Lowlands, Javakheti and the Armenian highlands. The Transcaucasus, or South Caucasus area, is a part of the entire Caucasus geographical region that essentially divides the Eurasian transcontinent into two. </DIV><BR>

<DIV ALIGN=RIGHT>All of Armenia is in Southern Caucasus; the majority of Georgia and Azerbaijan, including the exclave of Naxchivan, fall within this area. </DIV><BR>

<P>The countries of the region are producers of oil, manganese ore, tea, citrus fruits, and wine.</P>

</BODY>

</HTML>



<DIV> ტეგი პირველად HTML-3.2 ვერსიაში გამოჩნდა და CSS საშუალებებთან ერთობლიობაში გამოიყენება. იგი დოკუმენტის სტრუქტურის მძლავრ საშუალებას წარმოადგენს.



ტექსტის სწორების ატრიბუტი ALIGN უზრუნველყოფს ტექსტის სწორებას გვერდის მიმართ. ელემენტების სწორება ცენტრის მიმართ სხვადასხვა მეთოდით შეიძლება. მაგალითად, ეს შეიძლება განხორციელდეს აბზაცის ტეგით

<P ALIGN=CENTER>

თვით სწორების ტეგით

<ALIGN=CENTER>

და <CENTER> და </CENTER> ტეგების კონტეინერით.

## წინასწარი დაფორმატება

<PRE> ტეგი საშუალებას გვაძლევს ეკრანზე სხვადასხვანაირად დაფორმატებული ტექსტი გამოვიტანოთ, რომელიც </PRE> ტეგით უნდა დავამთავროთ. წინასწარ დაფორმატებული ტექსტის შიგნით დასაშვებია:

- სტრიქონის წყვეტის სიმბოლოების გამოყენება;
- ტაბულაციის სიმბოლოების გამოყენება (8 სიმბოლოთი მარჯვნივ გადაწევა);
- ბრაუზერის მიერ დაყენებული არაპროცესორული შრიფტის გამოყენება.

აბზაცის სხვა ტეგებით (როგორცაა, <Hx> და <ADDRESS>) დაფორმატება <PRE> და </PRE> ტეგებს შორის განთავსების შემთხვევაში იგნორირებული იქნება.

მოცემული ტეგი პირველად HTML-2.0 ვერსიაში გამოჩნდა. წინასწარ დაფორმატებულ ტექსტში სასურველია ტაბულაციის სიმბოლო არ იქნას გამოყენებული, რადგანაც ამან შეიძლება ტექსტის სწორების დარღვევა გამოიწვიოს.

## შრიფტის დაფორმატება

დოკუმენტში ტექსტური ინფორმაციის გამოსაყოფად

შრიფტების დაფორმატების სხვადასხვა საშუალებანი გამოიყენება, ეს ტეგებია:

- <B> და </B> - მუქი ტექსტი;
- <I> და </I> - დახრილი ტექსტი;
- <U> და </U> - ხაზგასმული ტექსტი;
- <TT> და </TT> - ტექსტი არაპროპორციული შრიფტით;
- <BIG> და </BIG> - ტექსტი დიდი შრიფტით;
- <SMALL> და </SMALL> - ტექსტი პატარა შრიფტით;
- <STRONG> და </STRONG> - ტექსტი გაზრდილი შრიფტით;
- <STRIKE> და </STRIKE> - ხაზგადასმული ტექსტი;
- <S> და </S> - ხაზგადასმული ტექსტი;
- <SUB> და </SUB> - ზედა ინდექსი;
- <SUP> და </SUP> - ქვედა ინდექსი.

<B>, <I>, <TT>, <STRONG> ტეგები პირველად HTML-2.0 ვერსიაში, ხოლო <U>, <BIG>, <SMALL>, <STRIKE>, <SUB>, <SUP> - HTML-3.2 ვერსიაში და <S> - HTML-4 ვერსიაში გამოჩნდა.

### პროგრამა 7.

<HTML>

<HEAD>

<TITLE> Font Style </TITLE>

</HEAD>

<BODY>

<B> ეს ტექსტი მუქია </B><BR>

<I> ეს ტექსტი დახრილია </I><BR>

<U> ეს ტექსტი ხაზგასმულია </U><BR>

<TT> ეს ტექსტი არაპროპორციული შრიფტითაა </TT><BR>

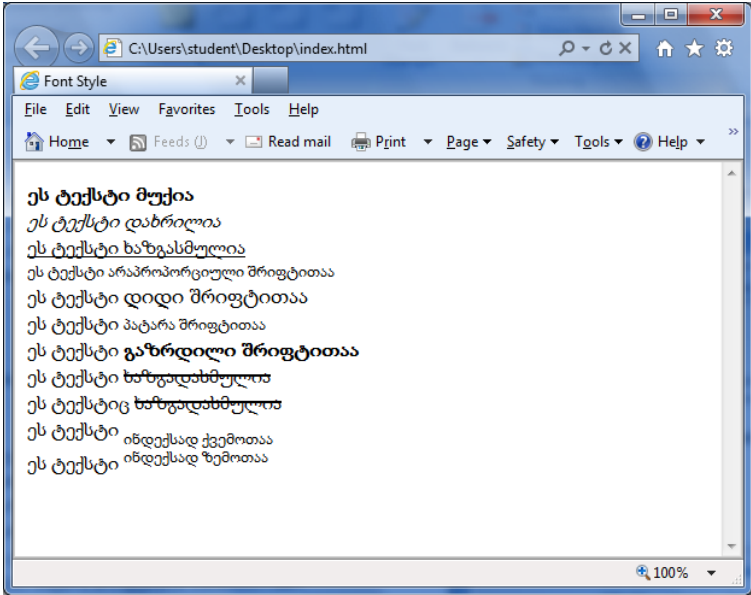
ეს ტექსტი <BIG> დიდი შრიფტითაა </BIG><BR>

ეს ტექსტი <SMALL> პატარა შრიფტითაა </SMALL><BR>

ეს ტექსტი <STRONG> გაზრდილი შრიფტითაა </STRONG><BR>

ეს ტექსტი <STRIKE> ხაზგადასმულია </STRIKE><BR>

ეს ტექსტიც <S> ხაზგადასმულია </S><BR>  
 ეს ტექსტი <SUB> ინდექსად ქვემოთაა </SUB><BR>  
 ეს ტექსტი <SUP> ინდექსად ზემოთაა </SUP><BR>  
 </BODY>  
 </HTML>



<SPAN> ტეგი ტექსტის ფრაგმენტის მისი შემდგომი დაფორმატების მიზნით გამოყოფის საშუალებას იძლევა, მაგრამ <DIV> ტეგისაგან განსხვავებით ის ახალ აბზაცს არ იწყებს. <SPAN> ტეგი მომხმარებლის მიერ განსაზღვრულ ტექსტური დონის სტრუქტურას ქმნის. შესაბამისი მაგალითი მე-8 პროგრამაშია მოყვანილი.

მოცემული ტეგი პირველად HTML-4.0-ში გამოჩნდა. ვინაიდან იგი CSS სტილებთან ერთობლიობაში გამოიყენება, ამიტომ მისი ამ მაგალითში გამოყენების უპირატესობა სრულად არ ჩანს.

## პროგრამა 8.

<HTML>

<HEAD>

<TITLE>სიმბოლოების დაფორმატების ტეგები</TITLE></HEAD>

<BODY BGCOLOR=#EAAAAC>

<H1> საქართველო </H1>

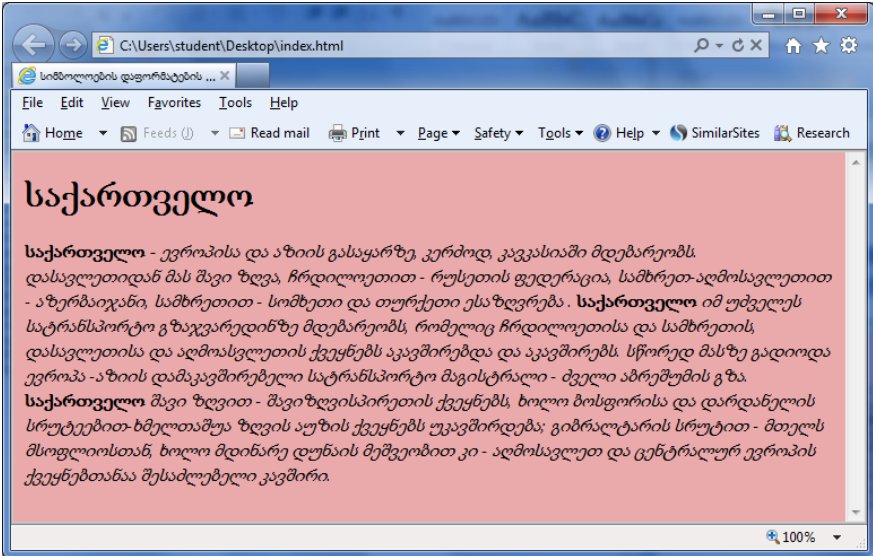
<SPAN><B> საქართველო </B><I> ევროპისა და აზიის გასაყარზე, კერძოდ, კავკასიაში მდებარეობს. დასავლეთიდან მას შავი ზღვა, ჩრდილოეთით - რუსეთის ფედერაცია, სამხრეთ-აღმოსავლეთით - აზერბაიჯანი, სამხრეთით - სომხეთი და თურქეთი ესაზღვრება.</SPAN></I>

<SPAN><B> საქართველო</B><I> 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 აბრეშუმის გზა.</SPAN></I>

<SPAN><B> საქართველო</B><I> 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.</SPAN></I>

</BODY>

</HTML>



<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 განათავსდეს, ამასთან, ეს ტექსტი არავითარ შემთხვევაში რამდენიმე სტრიქონად არ უნდა დაიყოს. ამგვარად, თავისი ფუნქციებით <NOBR> ტეგი, <BR> 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 ბილიკი გაჩნდება.

### პროგრამა 9.

<HTML>

<HEAD>

<TITLE>ტექსტი ერთ სტრიქონად </TITLE></HEAD>

<BODY BGCOLOR=#EAAAAC>

<H1> საქართველო </H1>

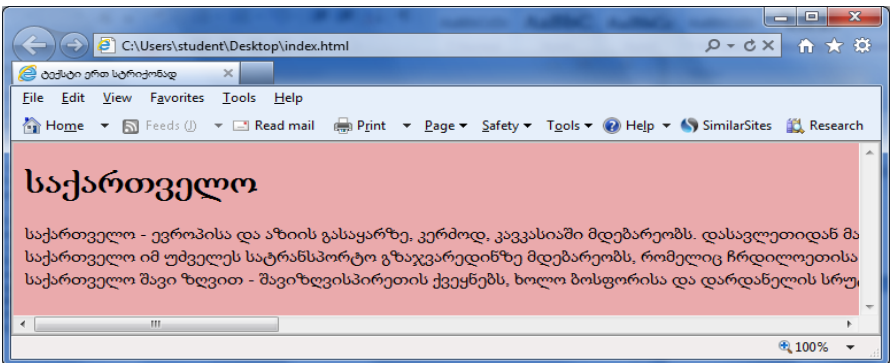
<NOBR> საქართველო - ევროპისა და აზიის გასაყარზე, კერძოდ, კავკასიაში მდებარეობს. დასავლეთიდან მას შავი ზღვა, ჩრდილოეთით - რუსეთის ფედერაცია, სამხრეთ-აღმოსავლეთით - აზერბაიჯანი, სამხრეთით - სომხეთი და თურქეთი ესაზღვრება. </NOBR>

<NOBR> 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 აბრეშუმის გზა. </NOBR>

<NOBR> საქართველო შავი ზღვით - შავიზღვისპირეთის 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>



<BLOCKQUOTE> ტეგი ტექსტს განსაზღვრავს როგორც დიდი ზომის ციტატას და მარცხენა და მარჯვენა კიდედან დაცილებულად გამოიტანს მას. ეს ტეგი საშუალებას იძლევა ტექსტი გვერდის ცენტრში კომპაქტურად განთავსდეს. ამ ტეგის რამდენიმეჯერ გამოყენების შემთხვევაში ტექსტი სულ უფრო ცენტრისკენ მჭიდროვდება. <BLOCKQUOTE> ტეგს აქვს CITE="URI" ატრიბუტი,

სადაც URI ციტირებულ დოკუმენტს ან ფრაზას იძლევა. URI (Uniform Resource Identifier) - ეს რესურსის უნიფიცირებული იდენტიფიკატორია, რომლის შემადგენლობაშიც შედის URL. <BLOCKQUOTE> ტეგით მონიშნული ტექსტის გამოტანა დოკუმენტის მარცხენა გვერდიდან 8 ჰარის (ინტერვალი) დაცილებით ხდება.

<Q> ტეგი ტექსტს მოკლე ციტატის სახით აფორმებს. მასაც CITE="URI" ატრიბუტი აქვს. <BLOCKQUOTE> ტეგთან შედარებით იგი უფრო მოკლე ციტატების გამოსატანად გამოიყენება. ჩვეულებრივ, ციტატის შემადგენლობაში სტრიქონის წყვეტის სიმბოლოები არ შედის, იგი ტექსტური დონის ელემენტად ითვლება. ბრაუზერს სიმბოლო ბრჭყალების (") გამოტანა შეუძლია.

## კომენტარები

როგორც ნებისმიერ ენაში, HTML-შიც, დოკუმენტის ტანში ნებისმიერ ადგილზე და ნებისმიერი რაოდენობით შეიძლება განთავსდეს კომენტარი, რომელიც დოკუმენტის გადაცემის დროს ინახება, მაგრამ ბრაუზერის მიერ მისი გამოტანა არ ხდება. მისი სინტაქსია:

<! - - ერთსტრიქონიანი კომენტარი - - >

<! მრავალსტრიქონიანი კომენტარი >

## აკრონიმი

ორი <ABBR> და <ACRONYM> ტეგი ტექსტის ფარგლებში აკრონიმის, ანუ აბრევიატურის გამოყოფის საშუალებას იძლევა. <ABBR> და <ACRONYM> ტეგების შიგთავსის გამოტანა "მცურავი" მოკარნახის სახით ხდება, რომლებიც მომხმარებელს აკრონიმის გაშიფვრის საშუალებას აძლევს.

### პროგრამა 10.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> აკრონიმი </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<!-- აკრონიმი, ანუ აბრევიატურა, - ცნობილი დასახელებების  
შემოკლებაა -->
```

```
<ABBR TITLE=Georgian_Technical_University> GTU </ABBR><BR>
```

```
</BODY>
```

```
</HTML>
```

ან

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> აკრონიმი </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<!-- აკრონიმი, ანუ აბრევიატურა, - ცნობილი დასახელებების  
შემოკლებაა -->
```

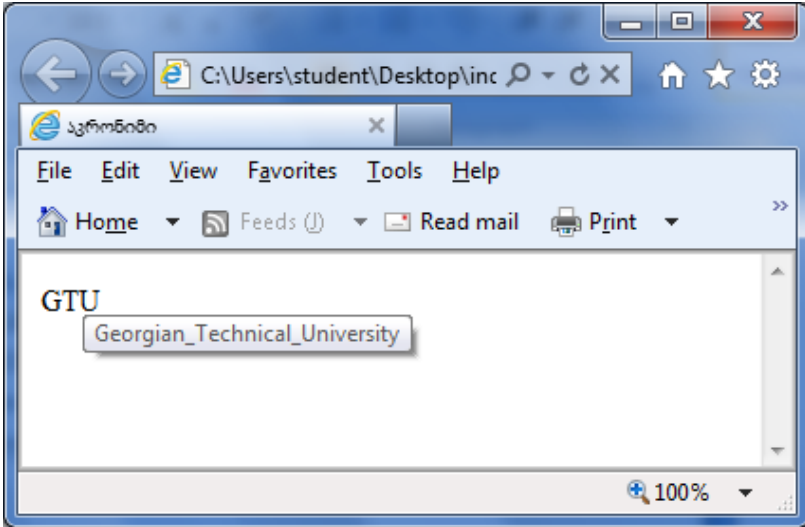
```
<ACRONYM TITLE=Georgian_Technical_University>
```

```
GTU</ACRONYM><BR>
```

```
</BODY>
```

```
</HTML>
```





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

<ACRONYM> ტეგი პირველად HTML-4.0 ვერსიაში იქნა გამოყენებული.

## ციტატა და ტერმინები

თუ დასახელებას <CITE> ტეგებს შორის მოვათავსებთ, მაშინ ეს დასახელება დახრილი შრიფტით გამოვა. იგი ციტატას ან წყაროზე მიმართვას განსაზღვრავს. ტერმინები ასევე შეიძლება <DFN> ტეგებით იყოს მოცემული. მათი გამოყენების მაგალითი ქვემოთაა მოყვანილი.

### პროგრამა 11.

```
<HTML>  
<HEAD>  
<TITLE> განსაზღვრება </TITLE>  
</HEAD>  
<BODY>
```

<CITE><B>**ბიოსფერო** </B>(ინგ. biosphere, ბერძ. bios - სიცოცხლე და sphaira - სფერო)</CITE> - დედამიწის თხელი ფენა, სადაც არსებობს და ვითარდება სიცოცხლე. ტერმინი „ბიოსფერო“ პირველად გამოჩენილმა ფრანგმა ნატურალისტმა ჟან ბატისტ ლამარკმა გამოიყენა, ხოლო ბიოსფეროს შესახებ მწყობრი სწავლება მხოლოდ მე–XX საუკუნის დასაწყისში შეიქმნა. მისი ავტორია რუსი მეცნიერი ვლადიმერ ვერნადსკი.

თანამედროვე ბიოსფერო ლითოსფეროს, ანუ დედამიწის ქერქს, ჰიდროსფეროს ანუ დედამიწის წყლიან გარსს

და ატმოსფეროს ანუ პლანეტის აიროვან გარსს მოიცავს . <BR>

<DFN><B> **ლითოსფერო**</B> ანუ <B>**დედამიწის**

**ქერქი** </B></DFN>- დედამიწის ყველაზე გარე, მაგარი გარსია.

დანალექი, გრანიტული და ბაზალტური შრეებისაგან შედგება.

განასხვავებენ ოკეანურ და კონტინენტურ ქერქს. ოკეანურ ქერქს

გრანიტის შრე არ ახასიათებს. დედამიწის ქერქის მაქსიმალური

სისქე მთიან მხარეებში - 70 კმ, ვაკეების ქვეშ - 30-40 კმ , ოკეანის

ქვეშ - 5-10 კმ-ია.<BR>

<DFN><B> **ჰიდროსფერო** </B>(ძვ. ბერძნ. – ὕδωρ – წყალი და

σφαῖρα – სფერო) </DFN>- დედამიწის წყლის გარსი, რომელიც

ატმოსფეროსა და დედამიწის ქერქს შორის წყვეტილად არის

განლაგებული ; <BR>

<DFN><B> **ატმოსფერო** </B>(ბერძ. „ἀτμός–ატმოს“- *ორთქლი,*

*ჰაერი, „σφαῖρα–სფერა“-სფერო)</DFN>-*

დედამიწის ჰაერის გარსი. იგი აზოტის (78 %), ჟანგბადის(21 %),

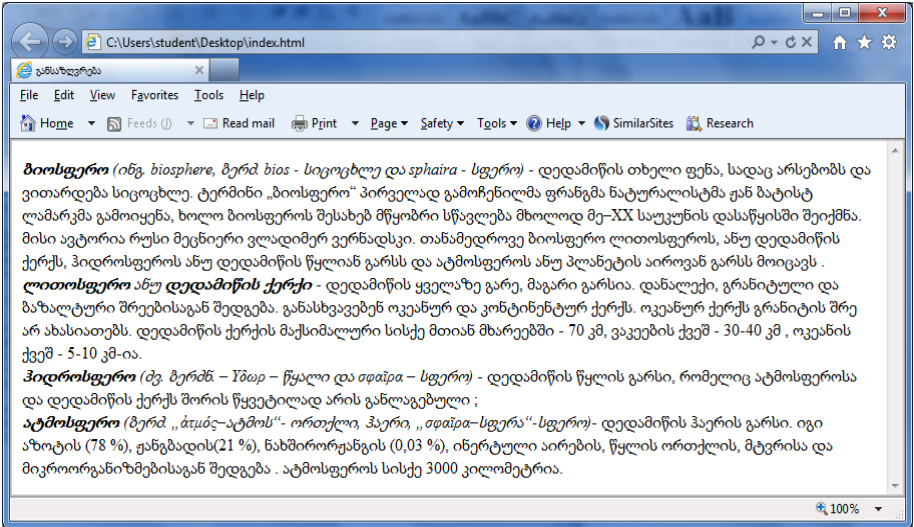
ნახშირორჟანგის (0,03 %), ინერტული აირების, წყლის

ორთქლის, მტვრისა და მიკროორგანიზმებისაგან შედგება .

ატმოსფეროს სისქე 3000 კილომეტრია. <BR>

</BODY>

</HTML>



<CITE> ტეგი პირველად HTML-2.0 ვერსიაში იქნა გამოყენებული.

## გამოყოფილი შრიფტი

<EM> ტეგი სიტყვის ან ტექსტის განსხვავებული შრიფტით (ჩვეულებრივ დახრილი შრიფტით) გამოყოფის მიზნით გამოიყენება. მისი სინტაქსია:

<EM> ტექსტი </EM>

## პროგრამა 12.

<HTML>

<HEAD>

<TITLE>დაფორმატების ტეგები</TITLE>

</HEAD>

<BODY BGCOLOR=00FFFF><B> გეოგრაფიული გარემო -

</B><EM>გეოგრაფიული გარსის ნაწილი; ადამიანთა

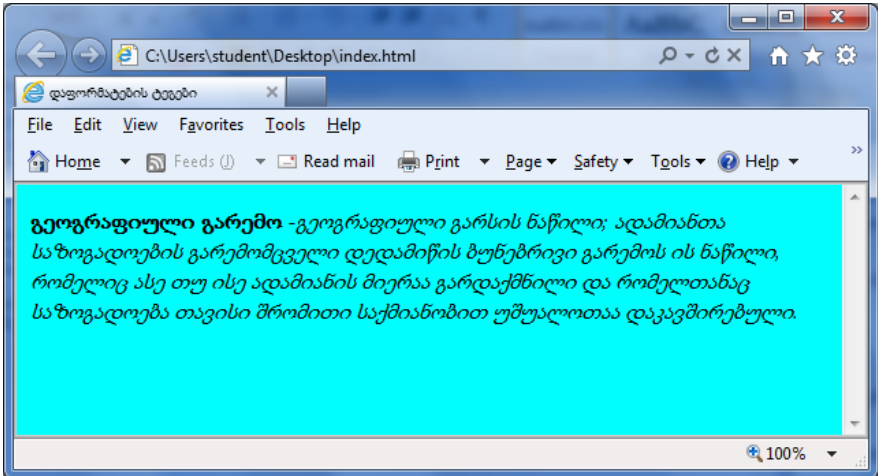
საზოგადოების გარემომცველი დედამიწის ბუნებრივი გარემოს

ის ნაწილი, რომელიც ასე თუ ისე ადამიანის მიერაა გარდაქმნილი და რომელთანაც საზოგადოება თავისი შრომითი საქმიანობით უშუალოთაა დაკავშირებული.

</EM>

</BODY>

</HTML>



<EM> ტეგი პირველად HTML-2.0 ვერსიაში იქნა გამოყენებული.

## მონოსიგანის შრიფტების გამოყენება

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

<CODE> ტეგი პროგრამული კოდის ფრაგმენტის მონოსიგანის შრიფტით გამოყოფის მიზნით გამოიყენება. მოცემული ელემენტი უფრო სრულყოფილია, ვიდრე <TT>

ელემენტი. მისი სინტაქსია:

<CODE> კოდი </CODE>

### პროგრამა 13.

<HTML>

<HEAD>

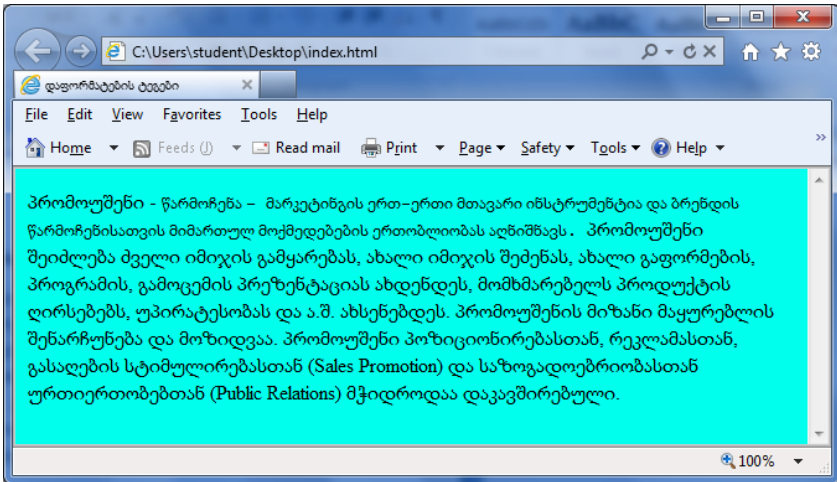
<TITLE>დაფორმატების ტეგები</TITLE>

</HEAD>

<BODY BGCOLOR=#00FFEC> პრომოუშენი - <CODE> წარმოჩენა – მარკეტინგის ერთ-ერთი მთავარი ინსტრუმენტია და ბრენდის წარმოჩენისათვის მიმართულ მოქმედებების ერთობლიობას აღნიშნავს. </CODE>პრომოუშენი შეიძლება ძველი იმიჯის გამყარებას, ახალი იმიჯის შექმნას, ახალი გაფორმების, პროგრამის, გამოცემის პრეზენტაციას ახდენდეს, მომხმარებელს პროდუქტის ღირსებებს, უპირატესობას და ა.შ. ახსენებდეს. პრომოუშენის მიზანი მაყურებლის შენარჩუნება და მოზიდვაა. პრომოუშენი პოზიციონირებასთან, რეკლამასთან, გასაღების სტიმულირებასთან (Sales Promotion) და საზოგადოებრიობასთან ურთიერთობებთან (Public Relations) მჭიდროდაა დაკავშირებული.

</BODY>

</HTML>



<KBD> ტეგი, მომხმარებლის კომპიუტერთან დიალოგის გამოყოფის მიზნით, ძირითადად კლავიატურით შეტანილი ტექსტის გამოსაყოფად გამოიყენება. მისი სინტაქსია:

<KBD> კლავიატურით შეტანა </KBD>

#### **პროგრამა 14.**

<HTML>

<HEAD>

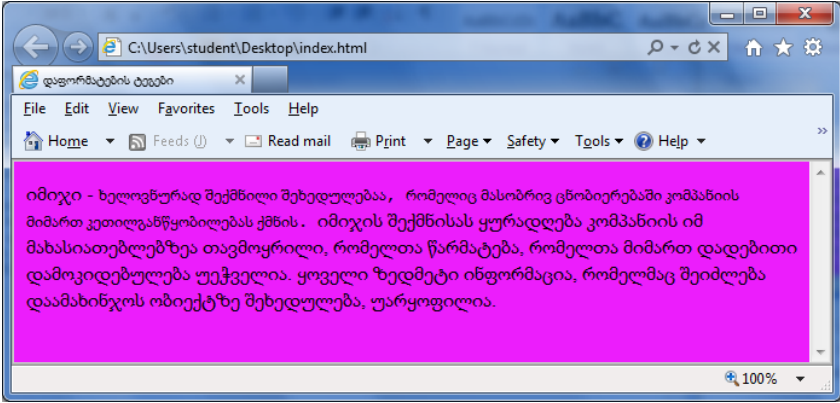
<TITLE>დაფორმატების ტეგები</TITLE>

</HEAD>

<BODY BGCOLOR=#EC1DFC> იმიჯი - <KBD> ხელოვნურად შექმნილი შეხედულებაა, რომელიც მასობრივ ცნობიერებაში კომპანიის მიმართ კეთილგანწყობილებას ქმნის. </KBD>

იმიჯის შექმნისას ყურადღება კომპანიის იმ მახასიათებლებზეა თავმოყრილი, რომელთა წარმატება, რომელთა მიმართ დადებითი დამოკიდებულება უეჭველია. ყოველი ზედმეტი ინფორმაცია, რომელმაც შეიძლება დაამახინჯოს ობიექტზე შეხედულება, უარყოფილია. </BODY>

</HTML>



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

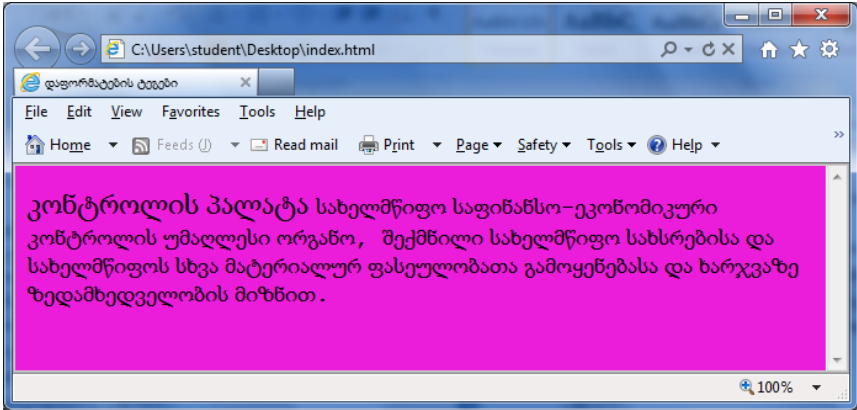
<SAMP> კომპიუტერის სისტემური შეტყობინება</SAMP>

**პროგრამა 15.**

```
<HTML>  
<HEAD>  
<TITLE>დაფორმატების ტეგები </TITLE>  
</HEAD>  
<BODY BGCOLOR=#EC1DDA><FONT size=5>კონტროლის პალატა  
<SAMP>
```

სახელმწიფო საფინანსო-ეკონომიკური კონტროლის უმაღლესი ორგანო, შექმნილი სახელმწიფო სახსრებისა და სახელმწიფოს სხვა მატერიალურ ფასეულობათა გამოყენებასა და ხარჯვაზე ზედამხედველობის მიზნით.

```
</SAMP>  
</FONT>  
</BODY>  
</HTML>
```



<CODE>, <KBD> და <SAMP> ტეგი პირველად HTML-2.0 ვერსიაში იქნა გამოყენებული.

## ჰარების დასმა

თუ ტექსტურ რედაქტორში ავკრებთ ტექსტს და მასში მრავალ ჰარს გამოვიყენებთ, ბრაუზერი ამ ჰარების იგნორირებას მოახდენს და მათ მინიმუმამდე შეამცირებს, ხოლო თუ <LISTING> ტეგს გამოვიყენებთ, მაშინ ეს ტეგი ტექსტს მონოსიგანის შრიფტით და ყველა იმ ჰარით გამოიტანს, რომლითაც იგი აიკრიბა.

### პროგრამა 16.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ციტატა</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR=BBCCFF>
```

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:

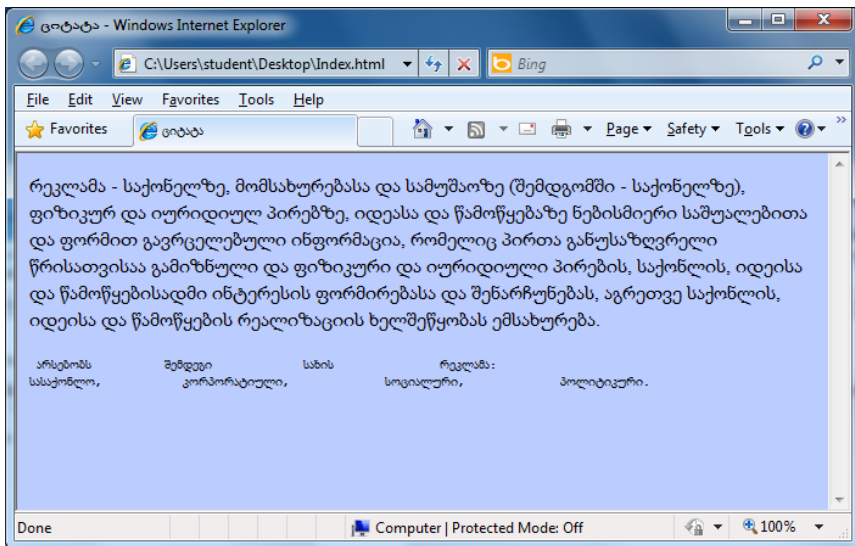
სასაქონლო, კორპორატიული, სოციალური,

პოლიტიკური.

</LISTING>

</BODY>

</HTML>



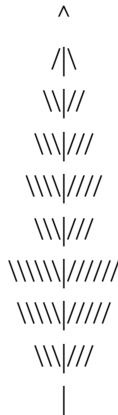
## კლავიატურის სიმბოლოები

კლავიატურის დახმარებით შეიძლება დაიხატოს კორიზონტალური ხაზი (ქვედა ხაზით \_), აგრეთვე მარცხნივ (\) და მარჯვნივ (/) დახრილი ხაზი, ვერტიკალური ხაზი (|). ასეთი

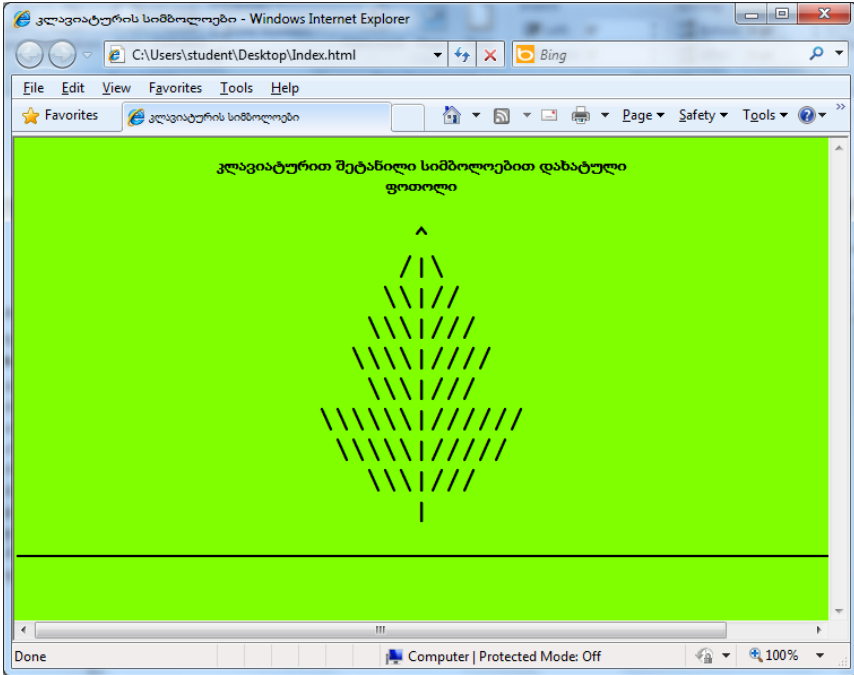
ხაზების ერთობლიობით შეიძლება საინტერესო ნახატები მივიღოთ. მაგალითი:

**პროგრამა 17.**

```
<html>
<head>
<title>კლავიატურის სიმბოლოები</title>
</head>
<body bgcolor=#7fff00 leftmargin=300>
<h5 align=center>კლავიატურით შეტანილი სიმბოლოებით
დახატული <br>
ფოთოლი <br></h5>
<h1>
<pre>
```



```
</h1></pre>
</body>
</html>
```



## სპეციალური სიმბოლოები

ვებ-გვერდებზე შეიძლება მომხმარებელს სურდეს ისეთი სიმბოლოები განათავსოს, როგორცაა ბრჭყალები, ტოლობის ნიშანი, პარაგრაფი, ჰარი და სხვა სპეციალური სიმბოლოები, რომლებსაც სპეციალური აღნიშვნები აქვთ. ეს სიმბოლოები და აღნიშვნებია:

- ამპერსანტი & - &amp;
- ნაკლებობის ნიშანი < - &lt;
- მეტობის ნიშანი > - &gt;
- სავაჭრო მარკის ნიშანი <sup>TM</sup> - &trade;
- განუყოფელი ჰარი - &nbsp;
- ცენტი - &cent;

ფუნტი - &pound;  
ვალუტის ნიშანი - &curren;  
ვერტიკალური ხაზი | - &brvbar;  
პარაგრაფის ნიშანი § - &sect;  
კოპირების ნიშანი © - &copy;  
ბრჭყალები ” - &quot;  
მარცხენა ტიპოგრაფიული ბრჭყალი « - &laquo;  
მარჯვენა ტიპოგრაფიული ბრჭყალი » - &raquo;  
დარეგისტრირების ნიშანი ® - &reg;  
გრადუსის ნიშანი ° – &deg;  
ხარისხად 1 - &sup1;  
ხარისხად 2 - &sup2;  
ხარისხად 3 - &sup3;  
ნიშანი მიკრო μ - &micro;  
აზნაგის ნიშანი ¶ - &para;  
წერტილი მარკერი · - &middot;  
1/4 - &frac14;  
1/2 - &frac12;  
გამრავლების ნიშანი × - &times;  
გაყოფის ნიშანი ÷ - &divide.

## ინფორმაცია დოკუმენტის ავტორის შესახებ

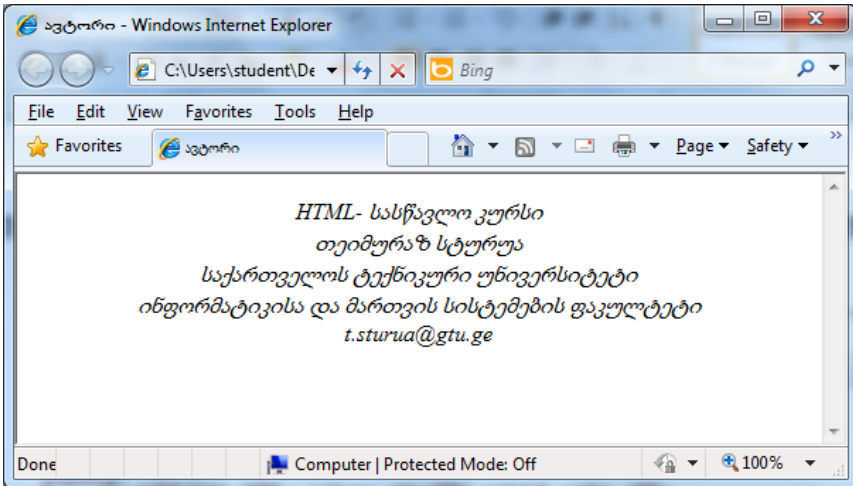
დოკუმენტის ავტორის შესახებ ინფორმაციის ვიზუალურად გამოსაყოფად გამოიყენება <ADDRESS> ტეგი. მისი სინტაქსია:

<ADDRESS> ავტორის მისამართი </ADDRESS>

ეს ტეგი ჩვეულებრივ დოკუმენტის ბოლოში მიეთითება. ტეგის შიგთავსის გამოტანა ჩვეულებრივ დახრილი სიმბოლოებით ხდება.

## პროგრამა 18.

```
<html>
<head>
<title>ავტორი</title></head>
<body background=#000001>
<center>
<address> HTML- სასწავლო კურსი<BR> თეიმურაზ სტურუა <BR>
საქართველოს ტექნიკური უნივერსიტეტი <BR>
ინფორმატიკისა და მართვის სისტემების ფაკულტეტი<BR>
t.sturua@gtu.ge
</address>
</body>
</html>
```



## 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 შეუძლია ცალკეულ ნაწილებად 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 - ევრანზე მდებარეობა (მარცხნივ, მარჯვნივ, ცენტრში);
- COLOR - ხაზის ფერი;
- NOSHADE - ხაზი ერთფერი მუქი წრფის სახით იქნება მოცემული.

პროგრამა.

```
<html>  
<head>  
<title>ლამაზი სქელი და წვრილი ხაზები</title>  
</head>  
<body bgcolor="#ff8c00" leftmargin=60><center>  
<h2>ლამაზი სქელი და წვრილი ხაზები </h2>  
<hr>
```

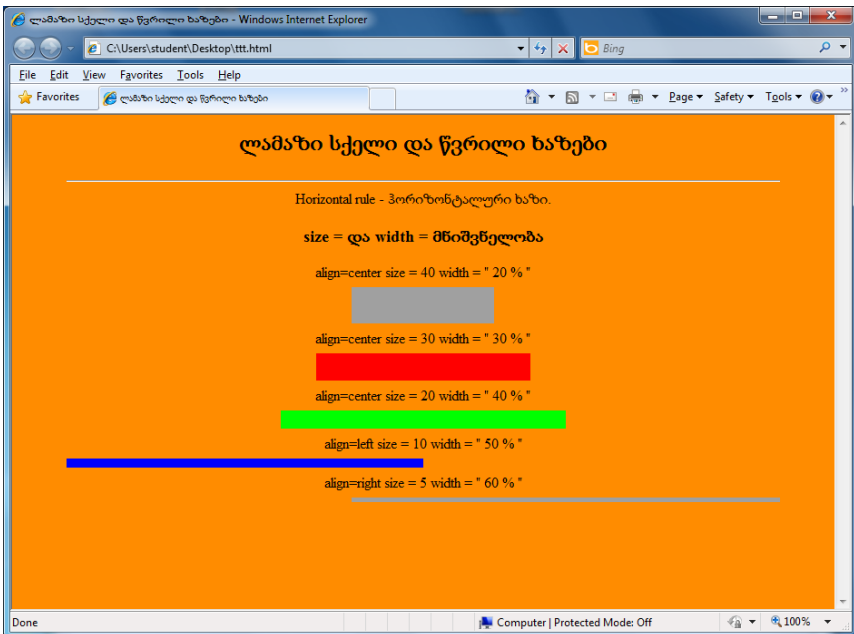
Horizontal rule - ჰორიზონტალური ხაზი.

```
<h3> size = და width = მნიშვნელობა</h3>  
align=center size = 40 width = &quot; 20 % &quot;;  
<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;;
<hr align=right width="60%" noshade size=5>
</center>
</body>
</html>

```



## სიები

HTML-დოკუმენტში ძირითადად სამი სახის სია არსებობს:

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

სიების შესაქმნელი სხვადასხვა ან ერთი და იგივე ტეგის ერთმანეთის შიგნით გამოყენებით ჩადგმული სიების შექმნა შეიძლება. ამისათვის, უბრალოდ ერთი წყვილი ტეგი მეორე წყვილის შიგნით განვათავსოთ.

### დანომრილი სიები

ზრაუხერი 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 <OL> (ინგლ. *Ordered List* - დანომრილი სია) ტეგის საშუალებით იქმნება. სიის თითოეული ელემენტი <LI> (ინგლ. *List Item* - სიის ელემენტი) ტეგით იწყება. დანომრილი სიის მაგალითი, რომელშიც გადანომვრა ავტომატურად ერთიდან იწყება ქვემოთაა მოყვანილი.

#### პროგრამა 1.

```
<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>პორტატულ (Notebook)

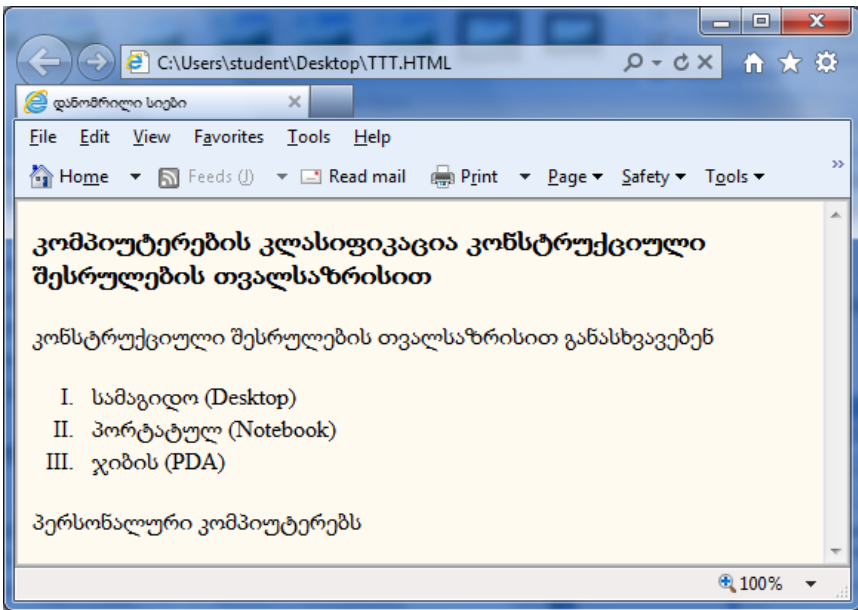
<LI>ჯიბის (PDA)

</OL>

პერსონალური კომპიუტერებს

</BODY>

</HTML>



<OL> 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 . . .);

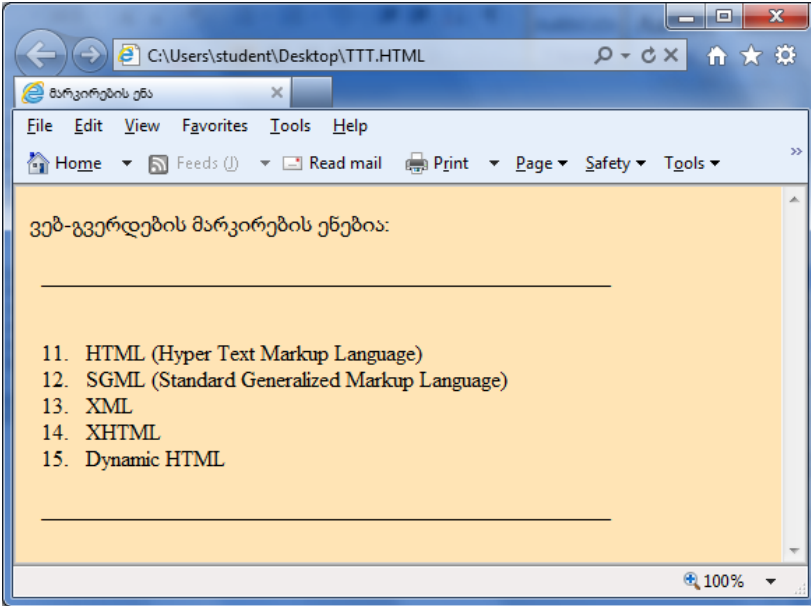
i - რომაული პატარა ციფრები (i, ii, iii . . .);

1 - არაბული ციფრები (1, 2, 3 . . .);

START=n - სიის საწყისი მნიშვნელობაა.

**პროგრამა 2.** არაბული ციფრებით გადანომრილი სია 11-ის ტოლი საწყისი მნიშვნელობით.

```
<html>
<head>
<title>მარკირების ენა</title>
</head>
<body bgcolor=moccasin>ვებ-გვერდების მარკირების ენებია:<br>
<pre> _____
</pre>
<br>
<ol type=1 start=11>
<li> HTML (Hyper Text Markup Language)
<li> SGML (Standard Generalized Markup Language)
<li> XML
<li> XHTML
<li> Dynamic HTML</ol>
<pre> _____
</pre>
</body>
</html>
```



## მარკირებული სიები

მარკირებული სიები წყვილი `<UL>` (ინგლ. *Unordered List* - არაგადანომრილი სია) ტევით იქმნება. სიის თითოეული ელემენტი, როგორც დანომრილი სიების შემთხვევაში, აქაც `<LI>` (ინგლ. *List Item* - სიის ელემენტი) ტევით იწყება. `<UL>` ტევს 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:

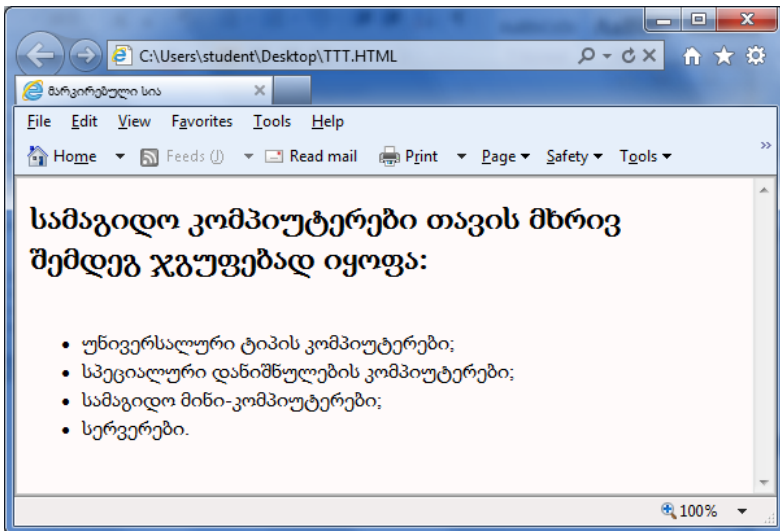
disc - გაფერადებული წრიული მარკერები;

circle - წრიული მარკერები;

square - კვადრატული მარკერები.

### პროგრამა 3.

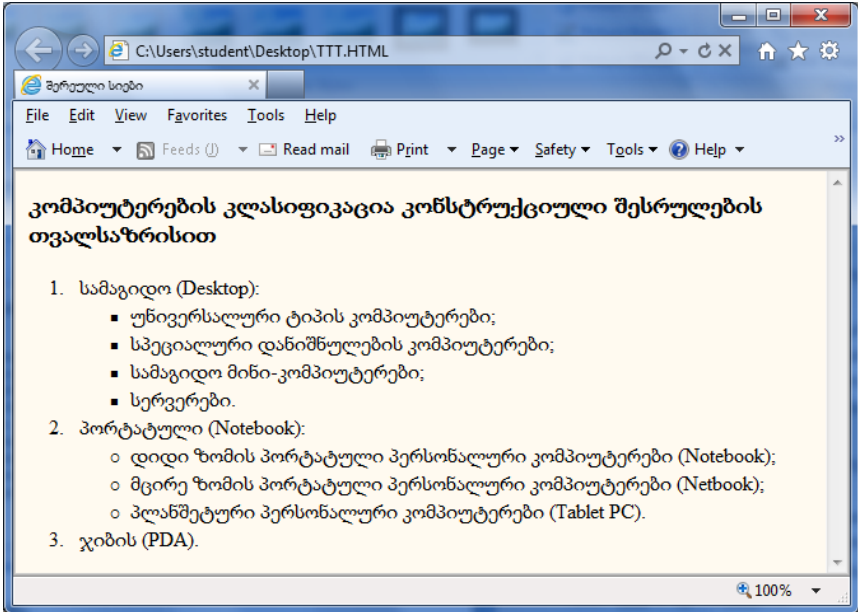
```
<html>
<head>
<title>მარკირებული სია</title>
</head>
<body bgcolor=snow>
<h2>სამაგიდო კომპიუტერები თავის მხრივ შემდეგ ჯგუფებად
იყოფა:</h2>
<br>
<ul>
<li> უნივერსალური ტიპის კომპიუტერები;
<li> სპეციალური დანიშნულების კომპიუტერები;
<li> სამაგიდო მინი-კომპიუტერები;
<li> სერვერები.
</ul>
</body>
</html>
```



ერთმანეთში ჩადგმული სიების მაგალითი მე-4 პროგრამაშია მოყვანილი:

#### **პროგრამა 4.**

```
<HTML>
<HEAD>
<TITLE>შერეული სიები</TITLE>
</HEAD>
<BODY BGCOLOR = "floralwhite">
<H3> კომპიუტერების კლასიფიკაცია კონსტრუქციული
შესრულების თვალსაზრისით </H3 >
<OL>
<LI>სამაგიდო (Desktop):
<UL TYPE=square>
<LI> უნივერსალური ტიპის კომპიუტერები;
<LI> სპეციალური დანიშნულების კომპიუტერები;
<LI> სამაგიდო მინი-კომპიუტერები;
<LI> სერვერები.
</UL>
<LI>პორტატული (Notebook):
<UL TYPE=circle>
<LI> დიდი ზომის პორტატული პერსონალური კომპიუტერები
(Notebook);
<LI> მცირე ზომის პორტატული პერსონალური კომპიუტერები
(Netbook);
<LI> პლანშეტური პერსონალური კომპიუტერები (Tablet PC).
</UL>
<LI>ჯიბის (PDA).
</OL>
</BODY>
</HTML>
```



## სია განსაზღვრება

სია განსაზღვრება სპეციალურად განლაგებული ტერმინებისა და მათი აღწერისაგან (განსაზღვრებიდან) შედგება. ვებ-გვერდებზე სია განსაზღვრების შესაქმნელად <DL>, <DT> და <DD> ტეგები გამოიყენება.

სია განსაზღვრება <DL> (ინგლ. *Definition List* - სია განსაზღვრება) ტეგით იწყება. იგი <DT> (ტერმინის ტექსტის მონიშვნა) და <DD> (ტერმინის აღწერის ტექსტის მონიშვნა) ტეგებს მოიცავს. სია განსაზღვრება ხშირად სამეცნიერო-ტექნიკურ და სასწავლო გამოცემებში გამოიყენება, რომლითაც ტერმინებს აფორმებენ. <DL> ტეგს COMPACT, CLASS, ID, LANG და STYLE ატრიბუტები აქვს.

თუ განსაზღვრელი ტერმინები მოკლეა, მაშინ გამოიყენება COMPACT ატრიბუტი, რომელიც სიის ელემენტების კომპაქტური

ფორმით გამოტანისათვის არის განკუთვნილი:

```
<dl compact>...</dl>
```

COMPACT ატრიბუტს მნიშვნელობა არა აქვს. სხვა ატრიბუტების სინტაქსი შემდეგია:

```
class="სტილის კლასი"
```

```
id="სახელი"
```

```
lang="ენის კოდი"
```

```
style="ჩაშენებული სტილის განსაზღვრა"
```

<DT> ტეგი (ინგლ. *Definition Term* - განმსაზღვრელი სიტყვა, ტერმინი) ბლოკურ არაწყვილ ტეგს წარმოადგენს, რომელიც სია განსაზღვრებაში ტერმინის ტექსტს მონიშნავს. <DT> ტეგის ატრიბუტებია:

```
class="სტილის კლასი"
```

```
id="სახელი"
```

```
lang="ენის კოდი"
```

<DD> ტეგი (ინგლ. *Definition Description* - განმსაზღვრელი ტერმინის აღწერა) ბლოკურ არაწყვილ ტეგს წარმოადგენს, რომელიც სია განსაზღვრებაში განსაზღვრების ტექსტს მონიშნავს.

<DD> ტეგის ატრიბუტებია:

```
class="სტილის კლასი"
```

```
id="სახელი"
```

```
lang="ენის კოდი"
```

```
style="ჩაშენებული სტილის განსაზღვრა"
```

## პროგრამა 5.

```
<html>
```

```
<head>
```

```
<title>ინტერნეტის საფუძვლები</title>
```

```
</head>
```

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

```
<dl>
```

<dt>ინტერნეტი

<dd>- (Internet) არის "მსოფლიო-ქსელი", ერთმანეთზე მიერთებული კომპიუტერების საჯაროდ ხელმისაწვდომი ქსელი, სადაც მომხმარებლებს, თუ მათ აქვთ უფლებები, ინფორმაციის მიღება ნებისმიერი კომპიუტერიდან შეუძლიათ. </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>

<dl>

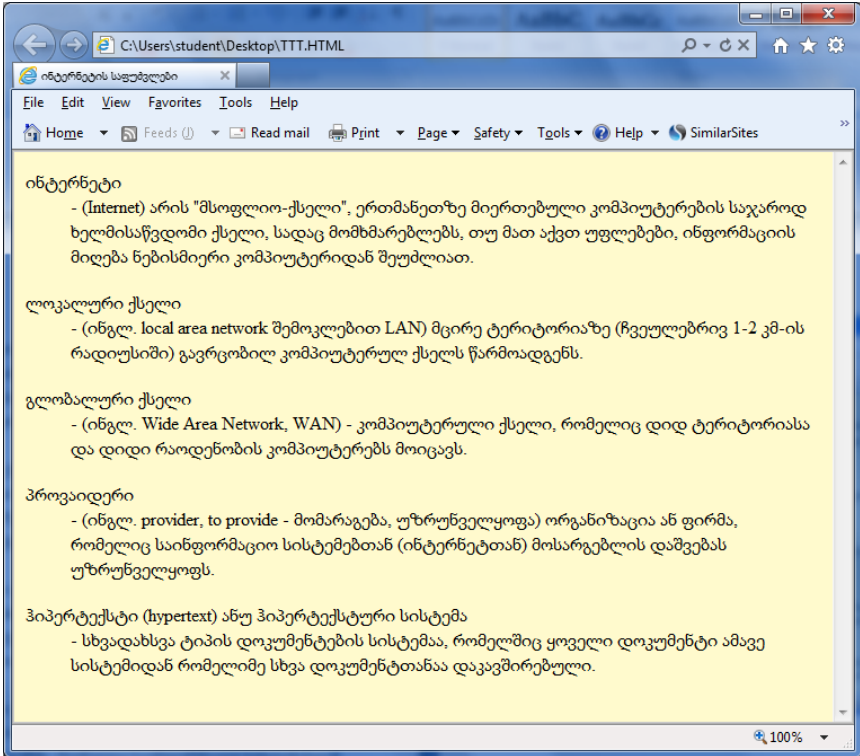
<dt>ჰიპერტექსტი (hypertext) ანუ ჰიპერტექსტური სისტემა

<dd>- სხვადასხვა ტიპის დოკუმენტების სისტემაა, რომელშიც ყოველი დოკუმენტი ამავე სისტემიდან რომელიმე სხვა დოკუმენტთანაა დაკავშირებული. </dl>

</body>

</html>





## ვებ-გვერდებზე გრაფიკული ელემენტის განთავსება

ვებ-გვერდებზე გრაფიკული ელემენტის განთავსებისათვის <IMG> ტეგი გამოიყენება. გრაფიკული ელემენტი დოკუმენტის შესახედაობას და ფუნქციურობას აუმჯობესებს. <IMG> ტეგი (ინგლ. *image* - გამოსახულება) პირველად HTML-2.0 ვერსიაში გამოჩნდა. 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 სურათის ადგილმდებარეობას უჩვენებს. სურათი გრაფიკულ ფორმატში უნდა იყოს შენახული, მაგალითად, GIF ან JPG ფორმატში. თუ გრაფიკული ფაილი იმავე საქალაქში მდებარეობს, სადაც HTML-დოკუმენტი, მაშინ მხოლოდ ფაილის სახელის მითითებაა საკმარისი. თუ ფაილი იმავე სერვერზე მდებარეობს, სადაც HTML-დოკუმენტი, მაშინ საქალაქის სახელიც უნდა მივუთითოთ, ხოლო წინააღმდეგ შემთხვევაში სრული მისამართის მითითებაა საჭირო. მაგალითად,

```
.
```

- *ALT* ატრიბუტი აუცილებელი არ არის და ვიდრე გამოსახულების ჩატვირთვა მიმდინარეობს, 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. ხშირად მონიტორის ამოხსნადობის მაჩვენებელი სხვადასხვა კომპიუტერზე 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 ევრანზე სურათის ზუსტი პოზიციონერობისათვის გამოიყენება. მისი მნიშვნელობებია:

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

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

- *BORDER* ატრიბუტი გამოსახულების ჩარჩოს სისქეს გვიჩვენებს;

- *VSPACE* ატრიბუტი გამოსახულების ზემოთ და ქვემოთ ცარიელი არის ზომას უთითებს პიქსელებში, რათა ტექსტი

გამოსახულებაზე არ გადავიდეს;

- *HSPACE* ატრიბუტი გამოსახულების მარცხნივ და მარჯვნივ ცარიელი არის ზომას უთითებს პიქსელებში, რათა ტექსტი გამოსახულებაზე არ გადავიდეს.

## გრაფიკული ფაილის ფორმატები

კომპიუტერული გრაფიკის ფაილების უმეტესობა ფორმატისა საშუალებას იძლევა, გამოსახულების შესახებ ინფორმაცია შენახულ იქნეს როგორც წერტილების ნაკრები. ზუსტად ასევე, გამოსახულება გამოდის ეკრანზეც.

ჩვეულებრივ, ინტერნეტში, ორი რასტრული ფორმატი - JPEG და GIF გამოიყენება. ამის გარდა, ინტერნეტში საკმაოდ მრავალ გამოსახულებას BMP და PCX ფორმატი აქვს. სპეციალისტებს მიაჩნიათ, რომ PNG ფორმატსაც - ყველაზე თანამედროვე ფორმატს, მომავალში საკმაოდ დიდი პერსპექტივა აქვს.

*GIF (Graphic Interchange Format - გრაფიკული მონაცემების მიმოცვლის ფორმატი)* ფორმატი რასტრული გრაფიკული გამოსახულებების ჩასაწერად და შესანახად გამოიყენება. ეს ფორმატი Unisys ტექნოლოგიის გამოყენებით Compuserve კორპორაციის მიერ იყო დამუშავებული. GIF ფორმატი 256-ფერიანი (ანუ მცირე ფერიანი) გამოსახულებების შესანახად გამოიყენება.

GIF ფორმატის მქონე გამოსახულებები ფაილებში .gif გაფართოებით ინახება. ასეთი გამოსახულებების უპირატესობას ის არის, რომ გამოსახულების სახე არც ბრაუზერზე და არც მის პლატფორმაზე არ არის დამოკიდებული.

*JPEG (Joint Photographic Expert Group - ფოტოგრაფიის სფეროს ექსპერტების გაერთიანებული ჯგუფი)* ფორმატი ისეთი სურათებისათვის გამოიყენება, სადაც 24-ბიტისანი პალიტრაა

გამოყენებული, რაც 16,7 მილიონი ფერის გამოსახვის საშუალებას იძლევა.

JPEG - ერთ-ერთი ყველაზე მძლავრი ალგორითმია. პრაქტიკულად იგი მრავალფერიანი გამოსახულებებისათვის დე-ფაქტო სტანდარტს წარმოადგენს. JPEG ფორმატი GIF ფორმატის ნაკლოვანი მხარის შესავსებად იყო შექმნილი. ეს ფორმატი HTML-დოკუმენტებისა და გრაფიკული გამოსახულებების ქსელის საშუალებით გადასაცემად ფართოდ გამოიყენება. გრაფიკის პარამეტრებს ფერთა RGB მოდელში ინახავს. JPEG ფორმატის მქონე გამოსახულებები ფაილებში .jpg გაფართოებით ინახება.

## **პიქსელები და ამოხსნადობა**

ინტერნეტში გამოსახულების გადასაცემად ძირითადად რასტრული გრაფიკა გამოიყენება.

*რასტრული გრაფიკა* - პირველ რიგში ეს არის გრაფიკა, რომელიც კომპიუტერში მრავალი წერტილის (პიქსელის) სახითაა წარმოდგენილი, მეორე რიგში, განსაზღვრული ტიპის ფაილია.

რასტრული გრაფიკული გამოსახულებისათვის ამ გამოსახულების ამოხსნადობის მაჩვენებელი მნიშვნელოვანი მახასიათებელია. ერთი და იგივე სურათი შეიძლება კარგი ან ცუდი ხარისხით იყოს წარმოდგენილი, იმისდა მიხედვით, სიგრძის ერთეულში, თუ მას რამდენი წერტილი შეესაბამება.

*ამოხსნადობის მაჩვენებელი* - ზომის ერთეულში წერტილების რაოდენობაა:

- dpi (dots per inch) - წერტილების რაოდენობა დიუიმში;
- ppi (points/pixels per inch) - პიქსელების რაოდენობა დიუიმში;
- spi (samples per inch) - სკანერის ოპტიკური, საწყისი ამოხსნადობის მაჩვენებელი, რაც დიუიმზე ცდების რაოდენობას

ნიშნავს ანუ სკანირების დროს სკანერის მიერ გამოსახულების „დათვალიერების“ რაოდენობა.

dpi – ყველაზე ხშირად გამოყენებული ტერმინია, რომელიც ინფორმაციის გამომტანი მოწყობილობების მიერ გამოტანილ პატარა წერტილს შეესაბამება. ამ წერტილს სურათის ამოხსნადობის მაჩვენებელთან საერთო არაფერი არ აქვს.

ილუსტრაციის ამოხსნადობის მაჩვენებელი ჩვეულებრივ dpi-ში იზომება. რაც მეტია ამოხსნადობის მაჩვენებელი, მით უფრო მაღალი ხარისხისაა გამოსახულება, მაგრამ ამასთან, იმ ფაილის მოცულობა, რომელშიც მოცემული გამოსახულება ინახება, დიდია.

არსებობს გამოსახულების, ეკრანისა და პრინტერის ამოხსნადობის მაჩვენებელი. სხვადასხვა ფორმის მონიტორები გამოდის, მაგრამ როგორც წესი, მათ სტანდარტული ამოხსნადობის მაჩვენებლები აქვთ: 640×480, 800×600, 1024×768, 1280×960, 1280×1024, 1600×1280, 1920×1600 პიქსელი. კარგი მონიტორისათვის ლუმინოფორებს (წერტილებს) შორის მანძილი 0,2-0,25 მმ-ს შეადგენს.

ხარისხიანი დიზაინისათვის მაღალი ამოხსნადობის მაჩვენებელის მქონე მონიტორია საჭირო. ზოგჯერ დიზაინერს გამოსახულების რედაქტირება შეიძლება წერტილების მიხედვით დასჭირდეს. მონიტორის ამოხსნადობის მაჩვენებელის მნიშვნელობა მონიტორზე და კომპიუტერის ვიდეო ადაპტერზე დამოკიდებულია.

სტატიკური რასტრული გამოსახულება ეს არის ორგანზომილებიანი რიცხვითი მასივი. ამ მასივის ელემენტებს პიქსელები (ინგლ. *pixel - picture element*, გამოსახულების ელემენტი) ეწოდება. პიქსელი - ეს არის გამოსახულების ერთი წერტილი.

## ვებ-გვერდზე გრაფიკული გამოსახულების განთავსების მაგალითები

ინტერნეტით გავრცელებული გამოსახულებებისათვის ჩვეულებრივ მიღებულია ამოხსნადობის მაჩვენებელი 72 dpi, ხოლო ამოხსნადობის მაჩვენებელი 300 dpi კი ძირითადად ბეჭდვითი გამოსახულებისათვის არის დამახასიათებელი.

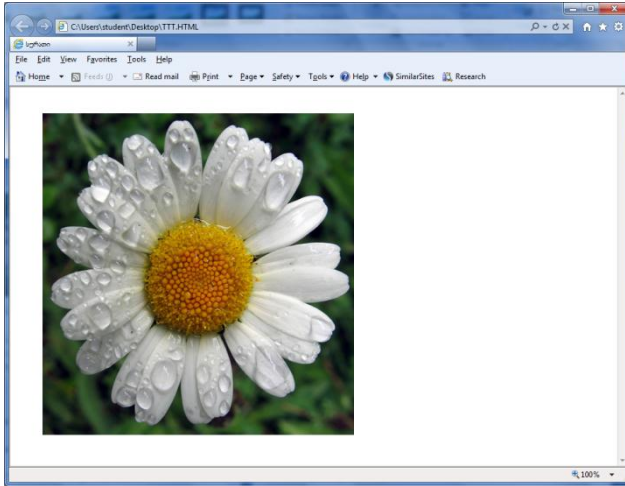
გამოსახულების ზომები შეიძლება დაზუსტდეს და აუცილებლობის შემთხვევაში Adobe Photoshop პროგრამის საშუალებით შეიცვალოს.

### პროგრამა 1.

```
<html>
<head>
<title>სურათი</title>
</head>
<body topmargin=40 leftmargin=50>

</body>
</html>
```

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



ქვემოთ მოყვანილია მაგალითი, სადაც უკვე სურათის განთავსების ადგილი (ცენტრი) და სურათის ზომები (სიგანე 281, სიმაღლე 300 პიქსელი) მითითებულია და აგრეთვე, სურათის ქვეშ წარწერა გაკეთებულია.

## პროგრამა 2.

```
<html>
```

```
<head>
```

```
<title>სურათი</title>
```

```
</head>
```

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

```
<center>
```

```

```

```
<br><font size=6>
```

```
დაცვარული გვირილა
```

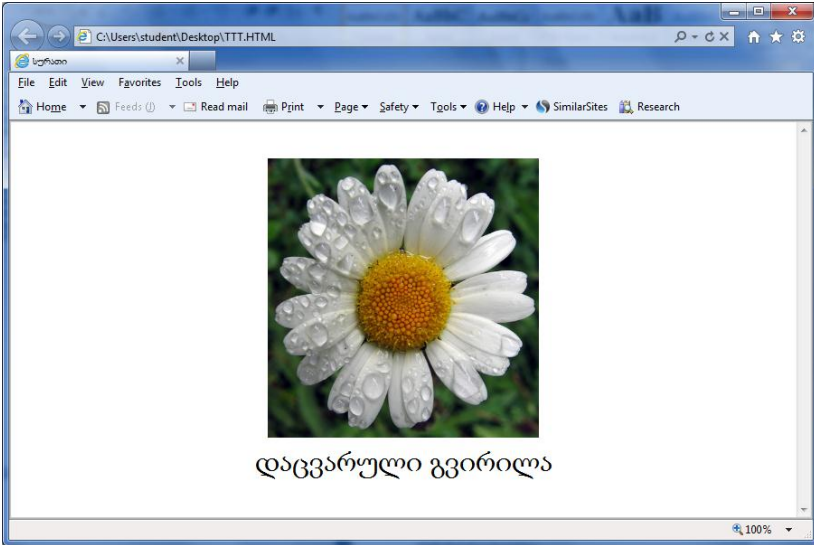
```
</center>
```

```
</font>
```

```
</body>
```

```
</html>
```





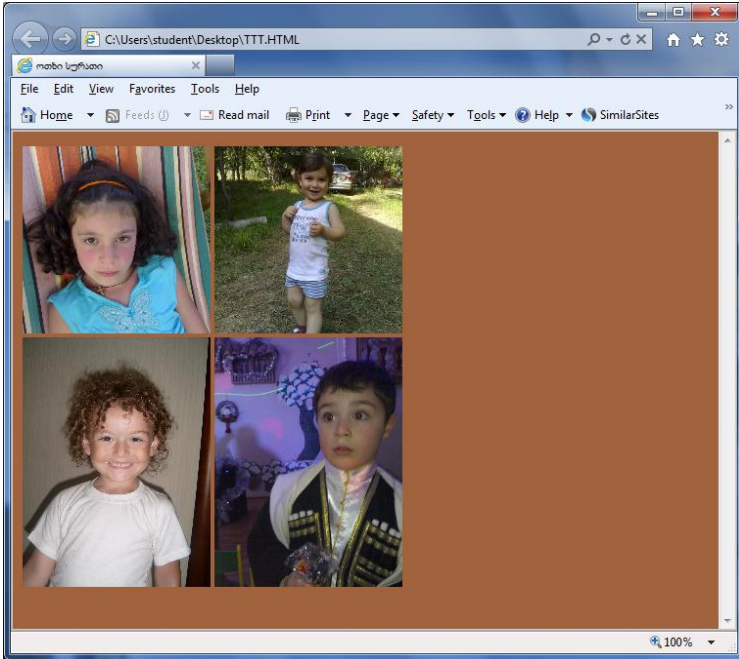
ქვემოთ მოყვანილია მაგალითი, სადაც 192 პიქსელი სიმაღლისა და 214 პიქსელი სიგანის ოთხი სურათია განთავსებული.

### პროგრამა 3.

```
<html>
<head>
<title>ოთხი სურათი</title></head>
<body bgcolor=#a1633d>

 <br>


</body>
</html>
```



## გამოსახულების სწორება

ALIGN ატრიბუტის დახმარებით ვებ-გვერდის მიმართ სურათის სწორება შეიძლება. ქვემოთ მაგალითში პირველი სურათი გასწორებულია ვებ-გვერდის მარჯვენა კიდის მიმართ, ხოლო მეორე სურათი მარცხენა კიდის მიმართ.

### პროგრამა 4.

```
<html>
```

```
<head>
```

```
<title>გრაფიკა ტექსტით</title>
```

```
</head>
```

```
<body>
```

```
<h1>გვირილა</h1>
```

```

```

<font size=4> **გვირილა**, რთულყვავილოვანთა ოჯახის რამდენიმე გვარის მცენარეთა კრებითი სახელწოდებაა. უფრო მეტად ამ სახელწოდებით Pyrethrum-ის გვარია ცნობილი, რომლის 100-მდე სახეობა ევრაზიის ზომიერ ზონაში და ნაწილობრივ ჩრდილოეთ ამერიკაში, უმრავლესობა კი კავკასიაში, ხმელთაშუაზღვისპირეთსა და წინა აზიის ქვეყნებშია გავრცელებული. საქართველოში 24 სახეობაა, მათგან 10 სახეობა კავკასიის ენდემია, 3 - საქართველოსი. იგი მთის ქვემო სარტყლიდან მოკიდებული ალპურ სარტყლამდე მდელოებსა და მაღალბალახეულში, ტყისპირებსა და ბუჩქნარებში, ტენიან და დაჭაობებულ ადგილებში, ქვიან და კლდოვან კალთებზე, ჩამონახვავებზე და სხვა ადგილებში იზრდება. <br>



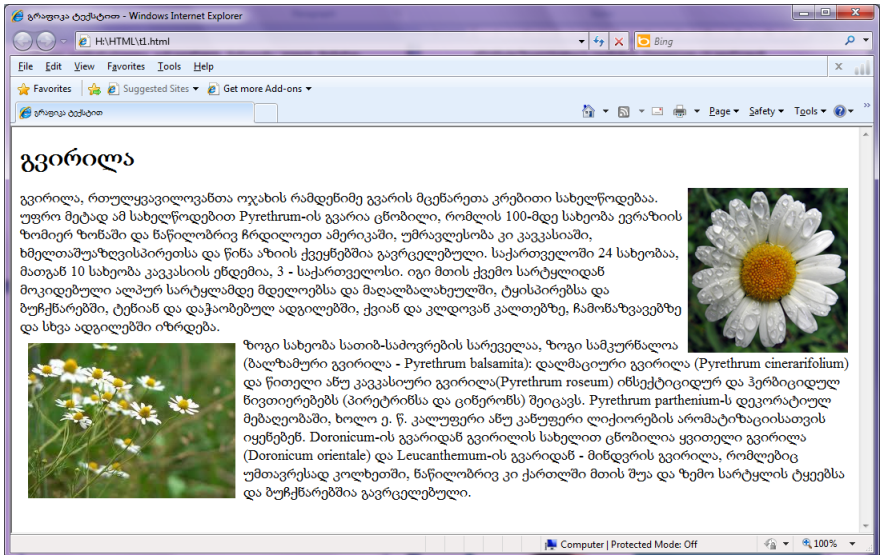
ზოგი სახეობა სათიბ-სამოვრების სარეველაა, ზოგი სამკურნალოა (ბალზამური გვირილა - Pyrethrum balsamita): დაღმაცვიური გვირილა (Pyrethrum cinerarifolium) და წითელი ანუ კავკასიური გვირილა(Pyrethrum roseum) ინსექტიციდურ და ჰერბიციდულ ნივთიერებებს (პირეტრინსა და ცინერონს) შეიცავს. Pyrethrum parthenium-ს დეკორატიულ მებაღეობაში, ხოლო ე. წ. კალუფერი ანუ კანუფერი ლიქიორების არომატიზაციისათვის იყენებენ. Doronicum-ის გვარიდან გვირილის სახელით ცნობილია ყვითელი გვირილა (Doronicum orientale) და Leucanthemum-ის გვარიდან - მინდვრის გვირილა, რომლებიც უმთავრესად კოლხეთში, ნაწილობრივ კი ქართლში მთის შუა და ზემო სარტყლის ტყეებსა და ბუჩქნარებშია გავრცელებული.

<br>

</font>

</body>

</html>



## გამოსახულების მასშტაბირება

ქვემოთ მოყვანილია ვებ-გვერდის მაგალითი, რომელიც ერთი და იგივე სხვადასხვა ზომის სურათს შეიცავს. სურათს აქვს ზომები 244 პიქსელი სიგრძეში და 321 პიქსელი სიგანეში. პირველი გამოსახულების ზომა ორჯერ არის გაზრდილი, მეორეს აქვს რეალური ზომა, ხოლო მესამე გამოსახულება ორჯერაა შემცირებული.

### პროგრამა 5.

```
<html>
```

```
<head>
```

```
<title>სურათის ზომა</title>
```

```
</head>
```

```
<body bgcolor="#008080">
```

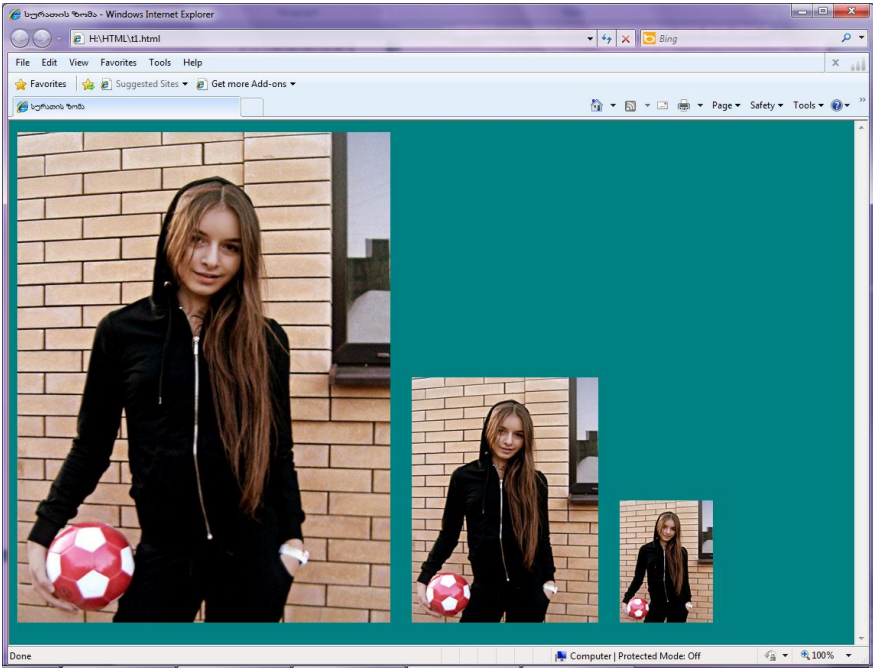
```
 &nbsp; &nbsp; &nbsp; &nbsp;
```

```
 &nbsp; &nbsp; &nbsp; &nbsp;
```

```

</body>
</html>
```

გამოსახულებები ერთმანეთისაგან სამი განუყოფელი ჰარით არის გამოყოფილი (რომელიც სიმბოლოებითა მოცემული).



თუ გამოსახულების ზომებს სწორად არ ავიღებთ, მაშინ ის დამახინჯდება. ასეთი მაგალითი მე-6 პროგრამითაა მოცემული.  
**პროგრამა 6.**

```
<html>
<head>
<title>დამახინჯებული სურათები</title>
</head>
```



- მეშვიდე - 50 პიქსელი სიმაღლეში და 100 პიქსელი სიგანეში.

### პროგრამა 7.

```

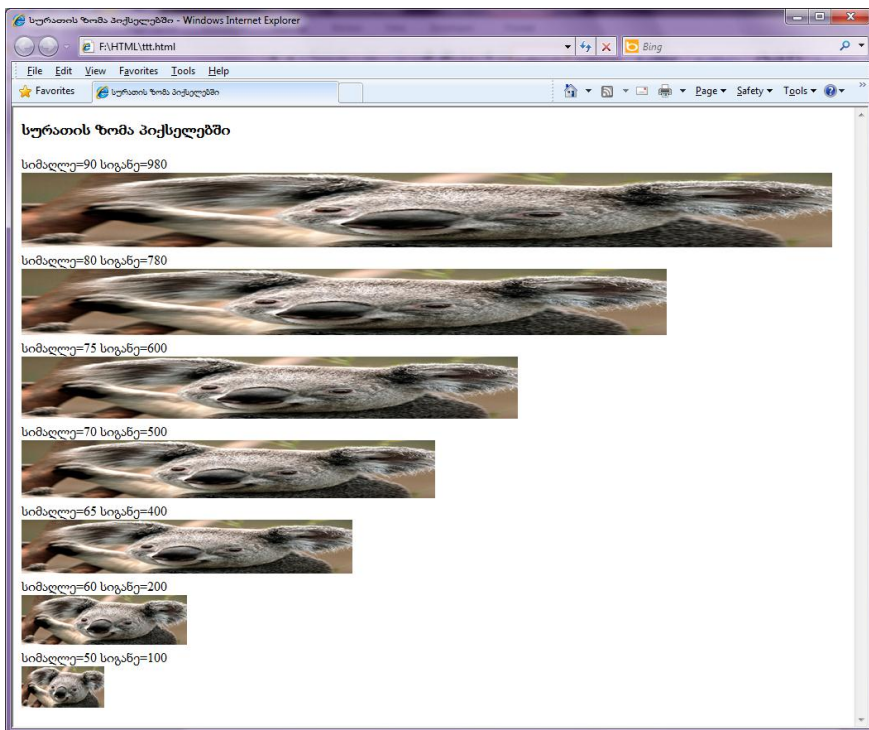
<html>
<head>
<title>სურათის ზომა პიქსელებში</title>
</head>
<body>
<h3>სურათის ზომა პიქსელებში</h3>
სიმაღლე=90 სიგანე=980<br>
 <br>
სიმაღლე=80 სიგანე=780<br>
 <br>
სიმაღლე=75 სიგანე=600<br>
 <br>
სიმაღლე=70 სიგანე=500<br>
 <br>
სიმაღლე=65 სიგანე=400<br>
 <br>
სიმაღლე=60 სიგანე=200<br>
 <br>
სიმაღლე=50 სიგანე=100<br>

</body>
</html>

```

თუ გამოსახულების ზომა სიგანეში 980 პიქსელს აღემატება, მაშინ იგი ეკრანზე სრულად ვერ მოთავსდება და ფანჯარას ჰორიზონტალური გადაფურცვლის ზოლი გაუჩნდება. თუ გამოსახულების ზომა სიმაღლეში 570 პიქსელს აღემატება, მაშინ იგი ეკრანზე სრულად ვერ მოთავსდება და ფანჯარას ვერტიკალური გადაფურცვლის ზოლი გაუჩნდება. მონიტორებისათვის,

რომელთა ზომებიც პიქსელში 1024×768 ზომისაგან განსხვავებულია, ეკრანზე სრულად განთავსებული გამოსახულების ზომები სხვა იქნება, მაგრამ არსი იგივე რჩება.



## გამოსახულების რუქა (გამოსახულების დაშლა არეებად )

გრაფიკული გამოსახულების რუქა (image maps) საშუალებას იძლევა გამოსახულების ცალკეული ნაწილები სხვა დოკუმენტებს ბმულით დავუკავშიროთ (იხ. თავი ბმულები გვ. 159). გამოსახულების ცალკეულ ნაწილებზე, ე.წ. აქტიურ არეებზე (hot spots) მაუსით დაწკაპუნების შემთხვევაში მომხმარებელს ამა თუ იმ მოქმედების შესრულება, სხვა დოკუმენტებზე გადასვლა და სხვა შეუძლია.



თუ მომხმარებელი გამოსახულებაზე, რომელიც <IMG> ტეგში ISMAP ატრიბუტის საშუალებით განსაზღვრულია როგორც გამოსახულების რუქა, მაუსის მაჩვენებლით პოზიციონირებს და მასზე მარცხენა კლავიშით დააწკაპუნებს, ვებ-სერვერს გადაეცემა დაწკაპუნების წერტილის კოორდინატები. ამ კოორდინატების მიღების შემდეგ სერვერი რუქაზე 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 რუქა 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 არეებად დასაშლელად (საიდანაც სხვა დოკუმენტზე ან WEB-გვერდზე გადასვლა იქნება შესაძლებელი) <MAP> და <AREA> ტეგები გამოიყენება. გამოსახულება, რომელზეც ვაპირებთ მუშაობას, <IMG> ტეგის საშუალებით უნდა იყოს რეალიზებული, სადაც USEMAP და SRC პარამეტრები იქნება გამოყენებული. მაგალითად,

<IMG SRC=სურათის\_სახელი USEMAP=#რუქის\_სახელი>

ტეგი <MAP>. <MAP> ტეგი გრაფიკული გამოსახულების რუქის კონფიგურაციის აღწერისათვის გამოიყენება და მას ერთადერთი NAME პარამეტრი აქვს. აქტიურ არეებზე მაუსის მაჩვენებლის მიყვანის შემდეგ ამ არის იდენტიფიცირება მცურავი

მოკარნახით შეიძლება. რუქის აღწერისათვის <MAP> და </MAP> კონტეინერული ტეგი გამოიყენება. ამ კონტეინერულ ტეგებს შორის გამოსახულების აქტიური არეები უნდა იყოს აღწერილი, რისთვისაც სპეციალური ტეგი <AREA> გამოიყენება. <MAP> ტეგი <IMG> 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 რუქის მხოლოდ ერთ-ერთი აქტიური არის შესახებ გვაძლევს ინფორმაციას. მას არ ჭირდება დახურვის ტეგი. <MAP> და </MAP> ტეგებს შორის თითოეული არის აღწერისთვის ცალკ-ცალკე <AREA> ტეგი უნდა იქნას 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="რუქის_სახელი"><AREA ატრიბუტები></MAP>
```

<AREA> ტეგს შემდეგი ატრიბუტები აქვს: SHAPE, COORDS, HREF, NOHREF, TITLE, TARGET და ALT. 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. რუქის არის განმსაზღვრელი ატრიბუტის მითითება აუცილებელია.

**შენიშვნა:** სერვერზე განთავსებული რუქისაგან განსხვავებით, სადაც DEFAULT-ი გამოსახულების ყველა იმ

წერტილს განსაზღვრავს, რომელიც რომელიმე აქტიურ არეს არ ეკუთვნის, კლიენტის სერვერზე განთავსებული რუქის შემთხვევაში DEFAULT-ი გამოსახულების ყველა წერტილს განსაზღვრავს. ამიტომ, DEFAULT-ი აქტიური არეების ჩამონათვალის ბოლოს უნდა იყოს გამოყენებული. მაგალითად, თუ ამ ჩამონათვალიდან 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 წერტილების კოორდინატა ჩამონათვალია, 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 ვერტიკალურს კოორდინატს განსაზღვრავს.

მართკუთხა ფორმის არისათვის ანუ როდესაც 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.

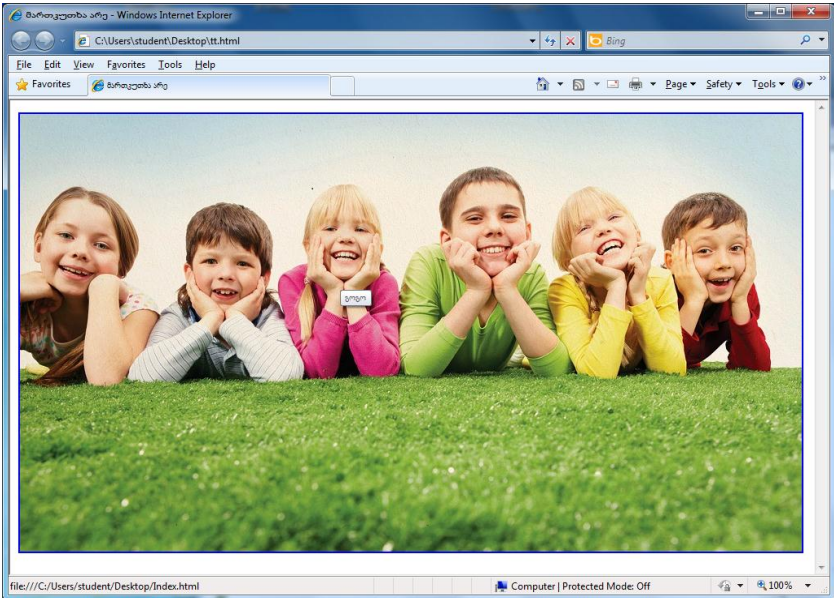
```
<html>
<head>
<title>მართკუთხა არე</title>
</head>
<body>

<map name="girl">
```

```

<area shape=rect coords="350,85,450,200" href=Index.html
title="გოგო" >
</map>
</body>
</html>

```



წრიული ფორმის არისათვის ანუ როდესაც 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.

```

<html>
<head>
<title>წრიული არე</title>

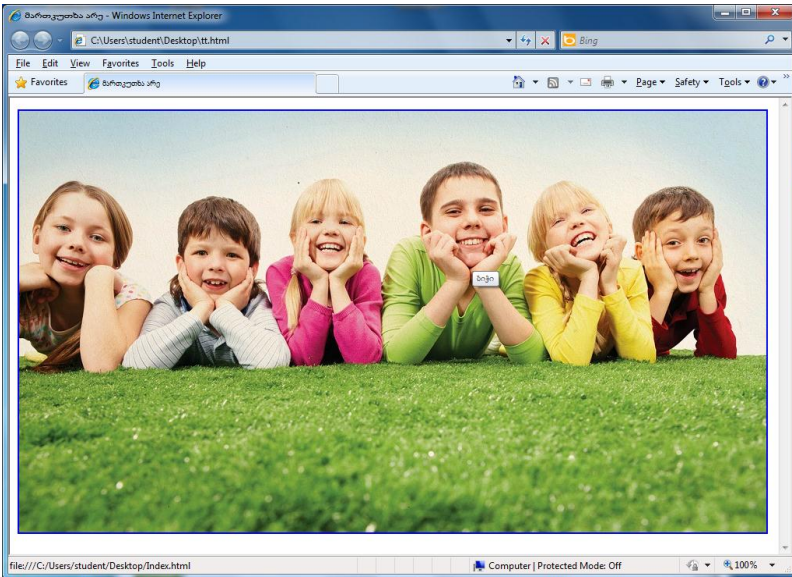
```

```

</head>
<body>

<map name="boy">
<area shape=circle coords="580,135,65" href=Index.html
title="ბიჭი" >
</map>
</body>
</html>

```



მრავალკუთხა ფორმის არისათვის ანუ როდესაც SHAPE=POLY შემთხვევაში, პირველად იწერება ზედა წერტილის კოორდინატი, შემდეგ თანმიმდევრობით სხვა წერტილის კოორდინატები, ხოლო ბოლო კოორდინატი არ არის აუცილებელი ემთხვეოდეს პირველს. ინტერნეტ ბრაუზერი 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 რუქის 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 *TARGET*. *TARGET* ატრიბუტი ფრეიმებთან მუშაობის დროს გამოიყენება და მისი მნიშვნელობა იმ ფრეიმის სახელია, რომელშიც მითითებული ბმული უნდა გაიხსნას. მისი სინტაქსია:

TARGET="name|\_BLANK|\_SELF|\_PARENT|\_TOP"

TARGET ატრიბუტის პარამეტრებია:

- name - დოკუმენტი მოცემული დასახელების ფრეიმში გაიხსნება, რომელიც <FRAME> ტეგში NAME ატრიბუტით არის მოცემული (უნდა გვახსოვდეს, რომ ფრეიმის სახელი აუცილებლად ასოთი იწყება);
- \_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 მშობლიურ ფრეიმს წარმოადგენს (თუ მშობლიური ფრეიმი არ არსებობს, მაშინ დოკუმენტი მიმდინარე ფრეიმში გაიხსნება);
- \_TOP - დოკუმენტი მიმდინარე ფანჯარაში გაიხსნება და ყველა ფანჯარას დაიჭერს.

- 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 თამაშობს.

- ატრიბუტი *ACCESSKEY*. ACCESSKEY ატრიბუტი გრაფიკული გამოსახულების აქტიურ არეებზე წვდომისათვის (URL-ის მიხედვით გადასასვლელად) კლავიშთა კომბინაციას განსაზღვრავს. მაგალითად,

```
<AREA ACCESSKEY=Ctrl+M>
```

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

```
<AREA TITLE = "კარნახის ტექსტი">
```

პროგრამა:

```
<html>
<head>
<title>მართკუთხა არე</title>
</head>
<body>

<map name="Child">
<area shape=rect coords="350,85,450,200" href=Index1.html
title="გოგო" ACCESSKEY=Ctrl+G >
<area shape=circle coords="580,135,65" href=Index2.html
title="ბიჭი" ACCESSKEY=Ctrl+B >
</map>
</body>
</html>
```

## ცხრილები

HTML-დოკუმენტებში ცხრილები არამარტო დამოუკიდებლად, არამედ ტექსტის ფრაგმენტებისა და სურათების ურთიერთ ზუსტი პოზიციონერებისათვის გამოიყენება. HTML ენაში ცხრილის უჯრედები შეიძლება ნებისმიერ ელემენტს, მათ შორის სათაურს, სიას, ტექსტურ აბზაცს, გრაფიკას და აგრეთვე ფორმის ელემენტებსაც შეიცავდეს.

ცხრილის შესაქმნელად ძირითადად <TABLE> წყვილი ტეგი გამოიყენება. ყველა ელემენტი <TABLE> და </TABLE> ტეგებს შორის უნდა განთავსდეს. ჩუმათობის პრინციპით ცხრილს ჩარჩო და გამყოფები არა აქვს. ჩარჩო BORDER ატრიბუტით ემატება. ჩარჩოს ზომა შეიძლება ფიქსირებული იყოს ან მისი ზომა ავტომატურად იცვლებოდეს ბრაუზერის ფანჯრის ზომის, ა უჯრედში ტექსტისა და სურათის ზომის მიხედვით. თავისი ბუნებრივი დანიშნულების (ტექსტური ინფორმაციის მოწესრიგება) გარდა, ცხრილები სუფთა დიზაინერული საკითხების გადაწყვეტის საშუალებას გვაძლევს: გვერდის ცალკეული ფრაგმენტები ერთმანეთს გაუთანაბროთ, სურათი და ტექსტი ერთმანეთის გვერდით განვათავსოთ, ფერებით გავაფორმოთ გვერდი, ტექსტი სვეტებად დავყოთ და სხვა.

<TABLE> ტეგს შემდეგი ატრიბუტები აქვს: ALIGN, BORDER, BACKGROUND, BGCOLOR, BORDERCOLOR, CELLPADDING, CELLSPACING, HSPACE, VSPACE, COLSPEC, WIDTH .

- ატრიბუტი *ALIGN* - თუ <CAPTION> და </CAPTION> ტეგებს შორისაა განთავსებული, მაშინ იგი ცხრილის წარწერას განსაზღვრავს და შეიძლება შემდეგი მნიშვნელობები მიიღოს TOP (ზემოთ, მნიშვნელობა ჩუმათობის პრინციპით) და BOTTOM (ქვემოთ). თუ ეს ატრიბუტი გვხვდება <TR>...</TR>, <TH>...</TH> ან <TD>...</TD> ტეგებს შორის, მაშინ იგი ცხრილის უჯრედში



მდგომი ტექსტის სწორებას უზენებს და შეუძლია შემდეგი მნიშვნელობები მიიღოს 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* - მთელი ცხრილისათვის ფონურ გამოსახულებას ქმნის;

- ატრიბუტი *BGCOLOR* - მთელი ცხრილისათვის ფონურ ფერს იძლევა;

- ატრიბუტი *BORDERCOLOR* - ჩარჩოს ფერს იძლევა და მხოლოდ *BORDER* ატრიბუტთან ერთად გამოიყენება, მაგალითად  
<table border="რიცხვი" bordercolor="ფერი">

- ატრიბუტი *CELLPADDING* - განსაზღვრავს უჯრედის საზღვარსა და მის შიგთავსს შორის მანძილს პიქსელებში;

- ატრიბუტი *CELLSPACING* - განსაზღვრავს უჯრედებს შორის მანძილს პიქსელებში;

- ატრიბუტი *HSPACE* - განსაზღვრავს ცხრილის მარჯვნივ და მარცხნივ ცარიელ ადგილს პიქსელებში;

- ატრიბუტი *VSPACE* - განსაზღვრავს ცხრილის ზემოთ და ქვემოთ ცარიელ ადგილს პიქსელებში;

- ატრიბუტი *COLSPEC* - ფიქსირებული სიგანის სვეტების განსაზღვრის საშუალებას იძლევა. სიგანის მნიშვნელობა როგორც სიმბოლოების რაოდენობით, ასევე პროცენტებით შეიძლება იყოს მოცემული;

- ატრიბუტი *WIDTH* - განსაზღვრავს ცხრილის სიგანეს. სიგანის მნიშვნელობა შეიძლება როგორც პიქსელებით, ასევე პროცენტებით იყოს მოცემული.

თუ შეტანილი ტექსტი მოცემულ უჯრედში არ ეტევა, მაშინ &nbsp; სიმბოლოს საშუალებით საჭიროა მათში განუყოფელი ჰარი შევიტანოთ. ეს ცხრილის ბადის დახატვისათვისაა აუცილებელი.

## ცხრილის სტრიქონებისა და სვეტების შექმნა

ცხრილის სტრიქონებისა და სვეტების შესაქმნელად <TR> და <COL> წყვილი ტეგები გამოიყენება.

<TR> ტეგი (ინგლ. *Table Row* - ცხრილის სტრიქონი) ცხრილის სტრიქონს ქმნის. სტრიქონების რაოდენობა <TR> და </TR> წყვილი ტეგების რაოდენობით განისაზღვრება. სტრიქონს შეიძლება ALIGN (ჰორიზონტალურად სწორება) და VALIGN (ვერტიკალურად სწორება) ატრიბუტი ჰქონდეს.

<COL> (ინგლ. *Column* - სვეტი) ტეგი ცხრილის სვეტის შესაქმნელად გამოიყენება და შეიძლება WIDTH, BGCOLOR, ALIGN, VALIGN და ID ატრიბუტები ჰქონდეს.

- ატრიბუტი *WIDTH* - სვეტში უჯრედის სიგანეს განსაზღვრავს:

<col width=რიცხვი>

- ატრიბუტი *BGCOLOR* - სვეტში უჯრედის ფონს განსაზღვრავს:

<col bgcolor=ფერი>

- ატრიბუტი *ALIGN* - უჯრედის შიგთავსის ჰორიზონტალურად სწორების რეჟიმს განსაზღვრავს. მან შეიძლება LEFT (მარცხნივ), CENTER (ცენტრში) და RIGHT (მარჯვნივ) მნიშვნელობები მიიღოს;

- ატრიბუტი *VALIGN* - უჯრედის შიგთავსის ვერტიკალურად სწორების რეჟიმს განსაზღვრავს. მან შეიძლება MIDDLE (შუაში) და TOP (ზემოთ) მნიშვნელობები მიიღოს;

- ატრიბუტი *ID* - ჰიპერმიმართვის სახელს იძლევა.

## ცხრილის უჯრედების შექმნა

ცხრილის უჯრედის შესაქმნელად გამოიყენება <TD> (ინგლ. *Table Data* - ცხრილური მონაცემი) წყვილი ტეგი. ცხრილის უჯრედის აღწერა მხოლოდ ცხრილის სტრიქონის შიგნით შეიძლება. თითოეული უჯრედი იმ სვეტის ნომრით უნდა დაინომროს, რომელსაც იგი ეკუთვნის. თუ რომელიმე სტრიქონში ერთი ან რამდენიმე უჯრედი არ არის მითითებული, მაშინ ბრაუზერი მათ ადგილზე ცარიელ უჯრებს გამოიტანს. მონაცემთა განთავსებისათვის ჩუმათობის პრინციპით შემდეგი ატრიბუტები ALIGN=LEFT და VALIGN=MIDDLE გამოიყენება. მოცემული განთავსება შეიძლება როგორც სტრიქონის, ასევე უჯრედის აღწერის დონეზე შეიცვალოს.

## ცხრილის სათაურისა და წარწერის შექმნა

ცხრილის სათაურის შესაქმნელად გამოიყენება <TH> (ინგლ. *Table Head* - ცხრილის სათაური) წყვილი ტეგი. ცხრილის სათაურს მთელი ცხრილის სიგანე აქვს. მოცემულ უჯრედში ტექსტს BOLD და ALIGN=CENTER ატრიბუტები აქვს.

ცხრილის წარწერის შესაქმნელად გამოიყენება <CAPTION> (ინგლ. *caption* - წარწერა) წყვილი ტეგი. <CAPTION> ტეგი <TABLE> და </TABLE> ტეგებს შორის და სტრიქონისა და უჯრედის აღწერის გარეთ უნდა იყოს განთავსებული. ALIGN ატრიბუტი განსაზღვრავს სად უნდა იყოს წარწერა. ჩუმათობის პრინციპით ALIGN ატრიბუტი TOP (ზემოთ) მნიშვნელობას ღებულობს, მაგრამ მან შეიძლება BOTTOM (ქვემოთ) მნიშვნელობაც მიიღოს. წარწერა ყოველთვის ცხრილის ფარგლებში ცენტრშია განთავსებული.

## ცხრილის სვეტების დაჯგუფება

<COLGROUP> ტეგი ცხრილის სვეტების დასაჯგუფებლად გამოიყენება და შეიძლება შემდეგი ატრიბუტები ჰქონდეს BGCOLOR, ALIGN, VALIGN, SPAN, WIDTH, ID.

- ატრიბუტი *BGCOLOR* - უჯრედის ფონის ფერს განსაზღვრავს;
- ატრიბუტი *ALIGN* - უჯრედის შიგთავსის ჰორიზონტალურად სწორებას იწვევს. მან შეიძლება შემდეგი მნიშვნელობები მიიღოს LEFT (მარცხნივ), CENTER (ცენტრში) და RIGHT (მარჯვნივ);
- ატრიბუტი *VALIGN* - უჯრედის შიგთავსის ვერტიკალურად სწორებას იწვევს. მან შეიძლება შემდეგი მნიშვნელობები მიიღოს MIDDLE (შუაში) და TOP (ზემოთ);
- ატრიბუტი *SPAN* - ჯგუფში სვეტების რაოდენობას გვამღევეს;
- ატრიბუტი *WIDTH* - სვეტის უჯრედის სიგანის ზომა;
- ატრიბუტი *ID* - ჰიპერმიმართვის სახელს იძლევა.

## ცხრილის სტრიქონების დაჯგუფება

ცხრილის ტანში ლოგიკურად ურთიერთდაკავშირებული სტრიქონების ჯგუფის შესაქმნელად <TBODY> ტეგი გამოიყენება, მაგალითად:

```
<tbody><tr><td>...<td>...</tbody>
```

<TBODY> ტეგს BGCOLOR, ALIGN, VALIGN ატრიბუტები აქვს.

ცხრილის ზედა ნაწილში (ცხრილის ზედა ქუდი) ლოგიკურად ურთიერთდაკავშირებული სტრიქონების ჯგუფის შესაქმნელად <THEAD> ტეგი გამოიყენება, მაგალითად:

```
<thead><tr><td>...<td>...</thead>
```

<THEAD> ტეგს BGCOLOR, ALIGN, VALIGN, ID ატრიბუტები აქვს.

ცხრილის ქვედა ნაწილში (ცხრილის ქვედა ქუდი) ლოგიკურად ურთიერთდაკავშირებული სტრიქონების ჯგუფის შესაქმნელად <TFOOT> ტეგი გამოიყენება, მაგალითად:

```
<tfoot><tr><td>...<td>...</tfoot>
```

<TFOOT> ტეგს BGCOLOR, ALIGN, VALIGN, ID ატრიბუტები აქვს. ყველა ზემოთ ჩამოთვლილ ატრიბუტებს იგივე აზრი აქვს, რაც <COLGROUP> ტეგის იმავე დასახელების ატრიბუტებს ჰქონდათ.

## ცხრილის ელემენტების ძირითადი ატრიბუტები

ქვემოთ ჩამოთვლილია ცხრილის ელემენტების ძირითადი ატრიბუტები. ზოგიერთ მათგანს ჩვენ ტეგების შესწავლის დროს გავეცანით.

- ატრიბუტი *ALIGN*;
- ატრიბუტი *BACKGROUND*;
- ატრიბუტი *BGCOLOR*;
- ატრიბუტი *BORDER*;
- ატრიბუტი *VALIGN*;
- ატრიბუტი *NOWRAP* - მიუთითებს, რომ მოცემული

უჯრედი ლოგიკურად არ შეიძლება სტრიქონებად დაიყოს და ერთი სტრიქონით უნდა იყოს წარმოდგენილი;

- ატრიბუტი *COLSPAN* - მიუთითებს, თუ რამდენი უჯრედი უნდა გაერთიანდეს ჰორიზონტალურად მონიშნულ უჯრედთან ერთად. ჩუმათობის პრინციპით ეს მნიშვნელობა ერთის ტოლია;

- ატრიბუტი *ROWSPAN* - მიუთითებს, თუ რამდენი უჯრედი უნდა გაერთიანდეს ვერტიკალურად მონიშნულ უჯრედთან ერთად. ჩუმათობის პრინციპით ეს მნიშვნელობა ერთის ტოლია;

• ატრიბუტი *RULES* - უჯრედებს შორის ხაზების გამოტანის წესს განსაზღვრავს, მაგალითად:

<table rules=all>

მისი დასაშვები მნიშვნელობებია:

- NONE - ხაზი არ არის (ეს მნიშვნელობა ჩუმათობის პრინციპით მიენიჭება);
- GROUPS - ხაზები მხოლოდ სტრიქონების ჯგუფს (რომლებიც <THEAD>, <TFOOT> და <TBODY> ტეგებით მიენიჭება) ან სვეტების ჯგუფს (რომლებიც <COLGROUP> და <COL> ტეგებით მიენიჭება) შორის გამონათდება;
- ROWS - ხაზები მხოლოდ სტრიქონებს შორის გამონათდება;
- COLS - ხაზები მხოლოდ სვეტებს შორის გამონათდება;
- ALL - ხაზები სტრიქონებსა და სვეტებს შორის გამონათდება;

• ატრიბუტი *FRAME* - მიუთითებს ცხრილის ჩარჩოს რომელი გვერდი უნდა ჩანდეს, მაგალითად:

<table frame=void>

მისი დასაშვები მნიშვნელობებია:

- VOID - გვერდები არ ჩანს (ეს მნიშვნელობა ჩუმათობის პრინციპით მიენიჭება);
- ABOVE - მხოლოდ ზედა ნაწილი ჩანს;
- BELOW - მხოლოდ ქვედა ნაწილი ჩანს;
- HSIDES - მხოლოდ ზედა და ქვედა ნაწილები ჩანს;
- VSIDES - მხოლოდ მარცხენა და მარჯვენა ნაწილები ჩანს;
- LHS - მხოლოდ მარცხენა ნაწილი ჩანს;
- RHS - მხოლოდ მარჯვენა ნაწილი ჩანს;
- BOX - ოთხივე ნაწილი ჩანს;

- BORDER - როგორც წინა შემთხვევაში ოთხივე ნაწილი ჩანს (ვიზუალურად მათ შორის სხვაობა არ არის).
- ატრიბუტი ID.

## ცხრილების შექმნის მაგალითები

ქვემოთ მოყვანილია პირველი პროგრამა, რომელიც ორი სტრიქონისა და შვიდი სვეტისაგან შემდგარ ცხრილს ქმნის. ამ მაგალითში უჯრედებში ჩანაწერები <TH> და </TH> ტეგების საშუალებითაა გაკეთებული, ამიტომ მონაცემები ავტომატურად უჯრედების ცენტრშია განთავსებული.

### პროგრამა 1.

```

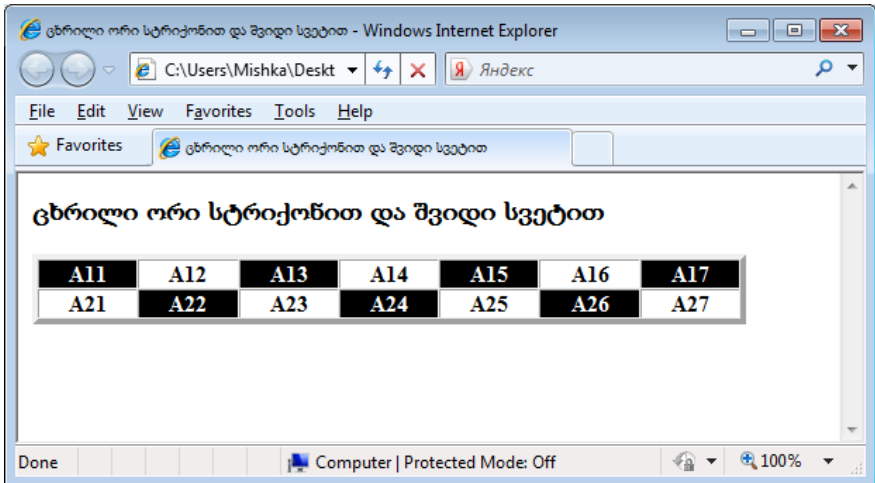
<html>
<head>
<title>ცხრილი ორი სტრიქონით და შვიდი სვეტით </title>
</head>
<body>
<h3>ცხრილი ორი სტრიქონით და შვიდი სვეტით </h3>
<table cellspacing=0 cellpadding=0 width="90%" border=4>
<tr>
<th bgcolor=black><font color=white> A11</font>
<th bgcolor=white> A12
<th bgcolor=black><font color=white> A13</font>
<th bgcolor=white> A14
<th bgcolor=black><font color=white> A15</font>
<th bgcolor=white> A16
<th bgcolor=black><font color=white> A17</font>
</tr>
<tr>
<th bgcolor=white> A21

```

```

<th bgcolor=black><font color=white> A22</font>
<th bgcolor=white> A23
<th bgcolor=black><font color=white> A24</font>
<th bgcolor=white> A25
<th bgcolor=black><font color=white> A26</font>
<th bgcolor=white> A27</tr>
</table>
</body>
</html>

```



მე-2 პროგრამაში უჯრედებში ჩანაწერები <TD> და </TD> ტეგების საშუალებითაა გაკეთებული, რის გამოც ტექსტი უჯრედებში ავტომატურად მარცხენა კუთხეში განთავსდება.  
**პროგრამა 2 .**

```

<html>
<head>
<title>ცხრილი ორი სტრიქონით და შვიდი სვეტით </title>
</head>

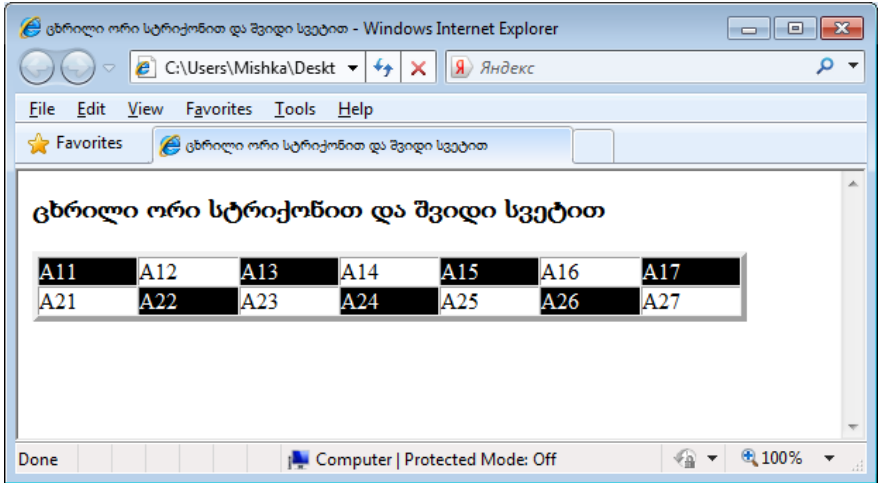
```



```

<body>
<h3>ცხრილი ორი სტრიქონით და შვიდი სვეტით </h3>
<table cellpadding=0 cellspacing=0 width="90%" border=4>
<tr>
<td bgcolor=black><font color=white> A11</font>
<td bgcolor=white> A12
<td bgcolor=black><font color=white> A13</font>
<td bgcolor=white> A14
<td bgcolor=black><font color=white> A15</font>
<td bgcolor=white> A16
<td bgcolor=black><font color=white> A17</font>
</tr>
<tr>
<td bgcolor=white> A21
<td bgcolor=black><font color=white> A22</font>
<td bgcolor=white> A23
<td bgcolor=black><font color=white> A24</font>
<td bgcolor=white> A25
<td bgcolor=black><font color=white> A26</font>
<td bgcolor=white> A27</tr>
</table>
</body>
</html>

```



მე-3 პროგრამაში გამრავლების ტაბულაა მოცემული, რომელიც ცხრა სვეტისა და ცხრა სტრიქონისაგან შედგება. ამ ცხრილში წარწერა <CAPTION> ტეგითაა გაკეთებული და ამიტომ იგი განთავსებულია ცხრილის ზემოთ.

### პროგრამა 3.

```

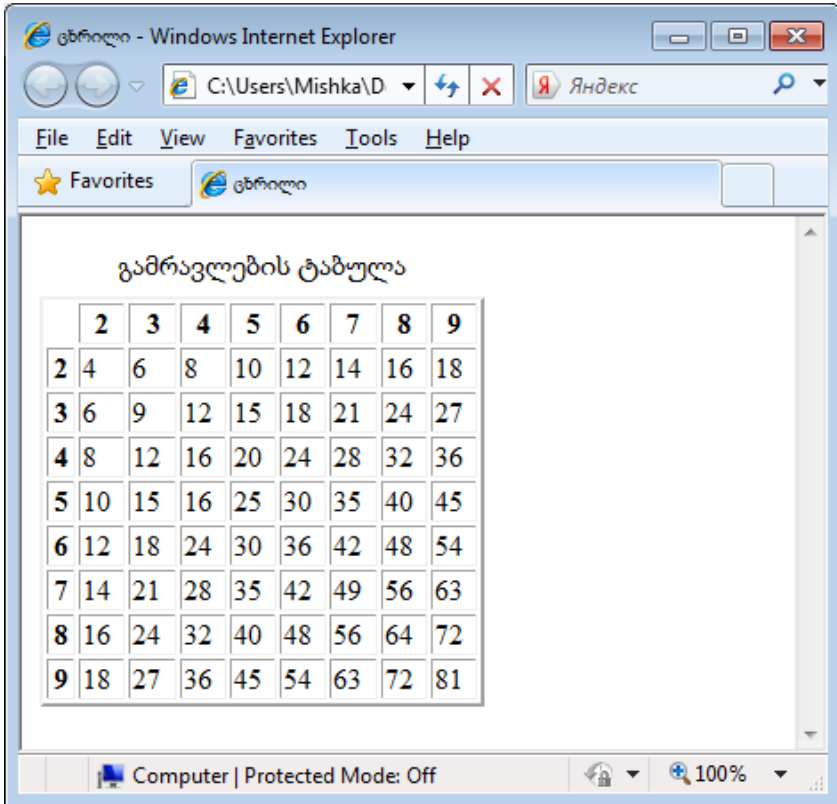
<html>
<head>
<title>ცხრილი</title>
</head>
<body>
<table border=2 width=250>
<caption>გამრავლების ტაბულა</caption>
<tr><th>2<th>3<th>4<th>5<th>6<th>7<th>8<th>9</tr>
<tr><th>2<td>4<td>6<td>8<td>10<td>12<td>14<td>16<td>18</tr>
<tr><th>3<td>6<td>9<td>12<td>15<td>18<td>21<td>24<td>27</tr>
<tr><th>4<td>8<td>12<td>16<td>20<td>24<td>28<td>32<td>36</tr>
<tr><th>5<td>10<td>15<td>16<td>25<td>30<td>35<td>40<td>45</tr>
<tr><th>6<td>12<td>18<td>24<td>30<td>36<td>42<td>48<td>54</tr>

```

```

<tr><th>7<td>14<td>21<td>28<td>35<td>42<td>49<td>56<td>63</tr>
<tr><th>8<td>16<td>24<td>32<td>40<td>48<td>56<td>64<td>72</tr>
<tr><th>9<td>18<td>27<td>36<td>45<td>54<td>63<td>72<td>81</tr>
</table>
</body>
</html>

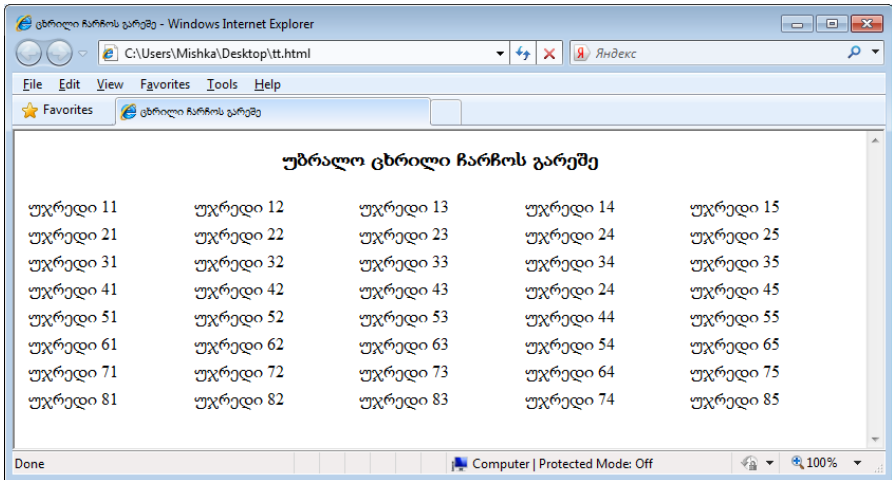
```



მე-4 პროგრამა ქმნის ცხრილს ჩარჩოს გარეშე რვა სტრიქონითა და ხუთი სვეტით. მასში გამოყენებულია WIDTH=100% ატრიბუტი, რომლის შესაბამისი ვებ-გვერდი პროგრამის ქვემოთაა მოცემული.

#### პროგრამა 4.

```
<html>
<head>
<title>ცხრილი ჩარჩოს გარეშე</title>
</head>
<body>
<h3 align=center>უბრალო ცხრილი ჩარჩოს გარეშე</h3>
<table width=100%>
<tr><td>უჯრედი 11 </td><td>უჯრედი 12 </td><td>უჯრედი 13
</td><td>უჯრედი 14</td><td>უჯრედი 15 </td></tr>
<tr><td>უჯრედი 21 </td><td>უჯრედი 22 </td><td>უჯრედი 23
</td><td>უჯრედი 24</td><td>უჯრედი 25 </td></tr>
<tr><td>უჯრედი 31 </td><td>უჯრედი 32 </td><td>უჯრედი 33
</td><td>უჯრედი 34</td><td>უჯრედი 35 </td></tr>
<tr><td>უჯრედი 41 </td><td>უჯრედი 42 </td><td>უჯრედი 43
</td><td>უჯრედი 24</td><td>უჯრედი 45 </td></tr>
<tr><td>უჯრედი 51 </td><td>უჯრედი 52 </td><td>უჯრედი 53
</td><td>უჯრედი 44</td><td>უჯრედი 55 </td></tr>
<tr><td>უჯრედი 61 </td><td>უჯრედი 62 </td><td>უჯრედი 63
</td><td>უჯრედი 54</td><td>უჯრედი 65 </td></tr>
<tr><td>უჯრედი 71 </td><td>უჯრედი 72 </td><td>უჯრედი 73
</td><td>უჯრედი 64</td><td>უჯრედი 75 </td></tr>
<tr><td>უჯრედი 81 </td><td>უჯრედი 82 </td><td>უჯრედი 83
</td><td>უჯრედი 74</td><td>უჯრედი 85 </td></tr>
</table>
</body>
</html>
```



თუ ზედა პროგრამაში WIDTH=100% ატრიბუტი არ იქნებოდა, მაშინ ცხრილი ვებ-გვერდის მარცხენა მხარეს განთავსდებოდა.

თუ მე-4 პროგრამაში <table> ტეგში cellpadding=10 ატრიბუტს ჩავამატებთ, მაშინ მიღებული პროგრამა იგივე ცხრილს შექმნის, ოღონდ უჯრედების შემცველობა მისი საზღვრიდან CELLPADDING პარამეტრით მოცემულ მანძილზე, ანუ 10 პიქსელით იქნება დაცილებული.

მე-5 პროგრამაში ROWSPAN и COLSPAN ატრიბუტების გამოყენებით ცხრილის უჯრედების გაერთიანებაა განხილული.

### პროგრამა 5.

```

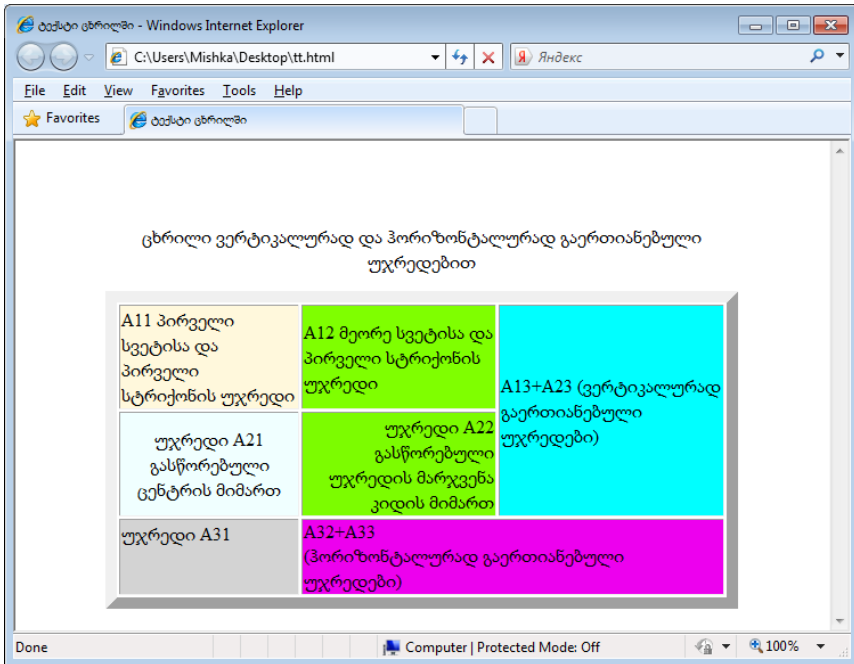
<html>
<head>
<title>ტექსტი ცხრილში</title>
</head>
<body>
<table border=10 width=80% align=center>

```

```

<caption>ცხრილი ვერტიკალურად და ჰორიზონტალურად
გაერთიანებული უჯრედებით </caption>
<br><br><br>
<tr><td width=30% bgcolor=cornsilk> A11 პირველი სვეტისა და
პირველი სტრიქონის უჯრედი </td>
<td width=200 bgcolor=chartreuse> A12 მეორე სვეტისა და პირველი
სტრიქონის უჯრედი </td>
<td rowspan=2 bgcolor=aqua> A13+A23 (ვერტიკალურად
გაერთიანებული უჯრედები) </td></tr>
<tr><td align=center bgcolor=azure> უჯრედი A21 <br>
გასწორებული ცენტრის მიმართ</td>
<td align=right bgcolor=lawngreen> უჯრედი A22 <br>
გასწორებული უჯრედის მარჯვენა კიდის მიმართ </td></tr>
<tr><td height=60 valign=top bgcolor=lightgrey> უჯრედი A31 </td>
<td colspan=2 valign=bottom bgcolor=mediumspringgreen> A32+A33
<br> (ჰორიზონტალურად გაერთიანებული უჯრედები)</td></tr>
</table>
</body>
</html>

```



მე-6 პროგრამაში შექმნილია პროგრამა, სადაც წარწერა ცხრილის ქვეშ არის განთავსებული (ALIGN=BOTTOM).

**პროგრამა 6.**

```

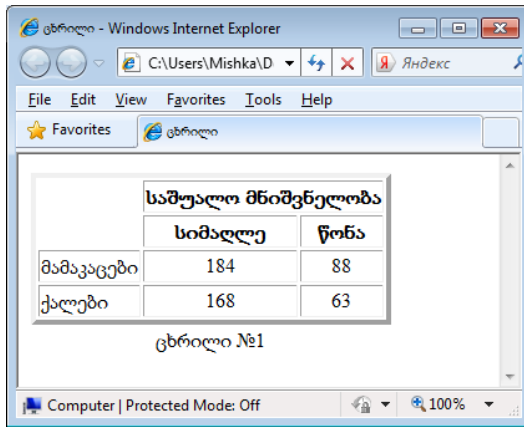
<html>
<head>
<title>ცხრილი</title>
</head>
<body>
<table border=4>
<caption align=bottom> ცხრილი №1 </caption>
<tr><td rowspan=2></td>
<th colspan=2>საშუალო მნიშვნელობა</th></tr>
<tr><th>სიმაღლე</th><th>წონა</th></tr>

```

```

<tr><td>მამაკაცები</td><td align=center>184</td>
<td align=center>88</td></tr>
<tr><td>ქალები</td><td align=center>168</td>
<td align=center>63</td></tr>
</table>
</body>
</html>

```



განვიხილოთ ცხრილის შექმნის კოდეზ ერთი მაგალითი, სადაც ნაჩვენებია, რომ ცხრილის სქელ ჩარჩოებს შეუძლია სხვადასხვა ვიზუალური ეფექტები შექმნას. საწინააღმდეგო შედეგი მიიღება ჩარჩოს გარეშე ცხრილებში. ერთმანეთში ჩადგმული ცხრილების შემთხვევაში სხვადასხვა სიგანის ჩარჩოებს აკეთებენ, რათა ამ ცხრილების ერთმანეთისაგან გარჩევა მარტივად მოხდეს.

### პროგრამა 7.

```

<html>
<head>
<title>ცხრილები სხვადასხვა სისქის ჩარჩოთი</title>
</head>

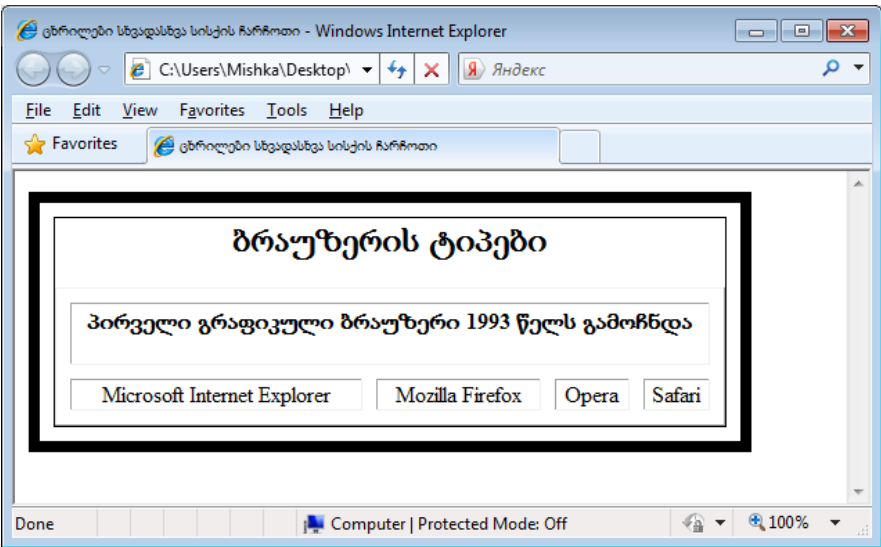
```



```

<body>
<table cellspacing=10 cellpadding=0 width="90%" border=8
bordercolor=black>
<tr><td align=center><h2>ბრაუზერის ტიპები</h2>
<table border cellspacing=10 width="100%">
<tr><td align=center colspan=4><h4>პირველი გრაფიკული
ბრაუზერი 1993 წელს გამოჩნდა</h4>
</td></tr>
<tr><td align=center>Microsoft Internet Explorer </td>
<td align=center>Mozilla Firefox</td>
<td align=center>Opera </td>
<td align=center>Safari</td></tr>
</table></td></tr>
</table>
</body>
</html>

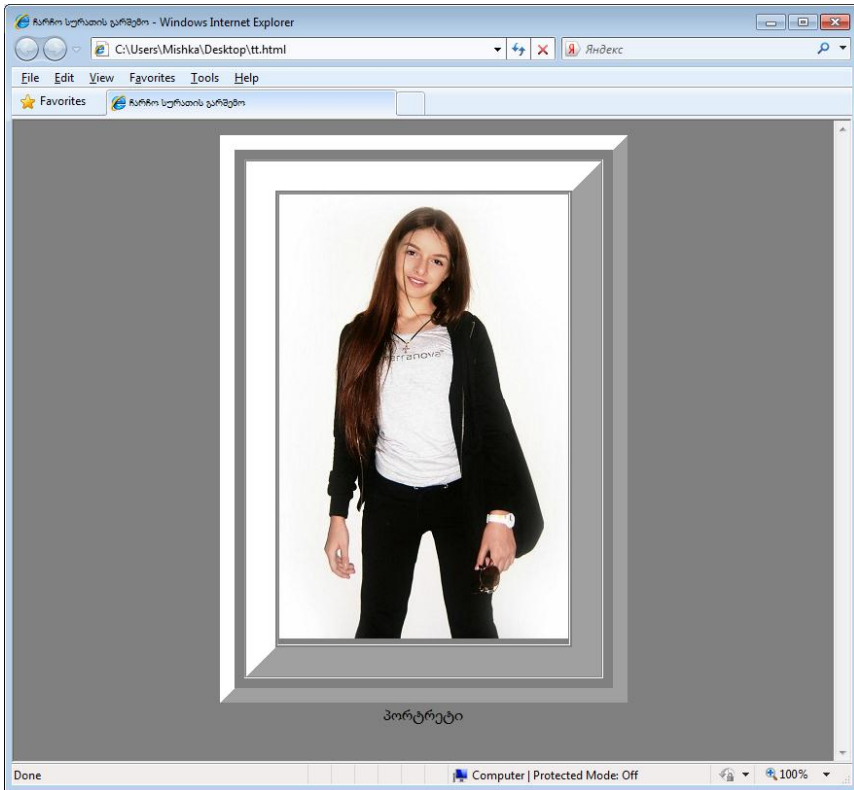
```



მე-8 პროგრამაში მოცემულია ცხრილის მაგალითი, რომელშიც CELSPACING ატრიბუტის დახმარებით გამოსახულების ირგვლივ ჩარჩოს იმიტირება ხდება.

**პროგრამა 8.**

```
<html>
<head>
<title>ჩარჩო სურათის გარშემო</title>
</head>
<body bgcolor=#808080>
<center>
<table cellspacing=10 border=15>
<caption align=bottom>პორტრეტი</caption>
<tr><td>
<table border=30>
<tr><td> </td></tr>
</table></td></tr>
</table>
</center>
</body>
</html>
```



ცხრილები ტექსტის სვეტებად განთავსების საშუალებას იძლევა. ასეთი მაგალითი მე-9 პროგრამაშია მოცემული.

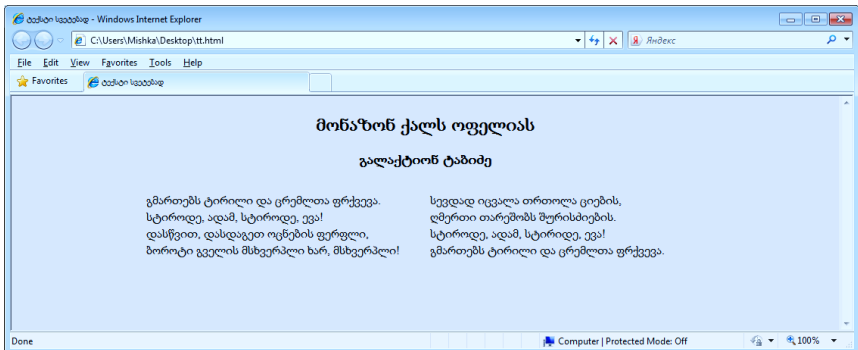
### პროგრამა 9.

```
<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>
ზოროტი 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>

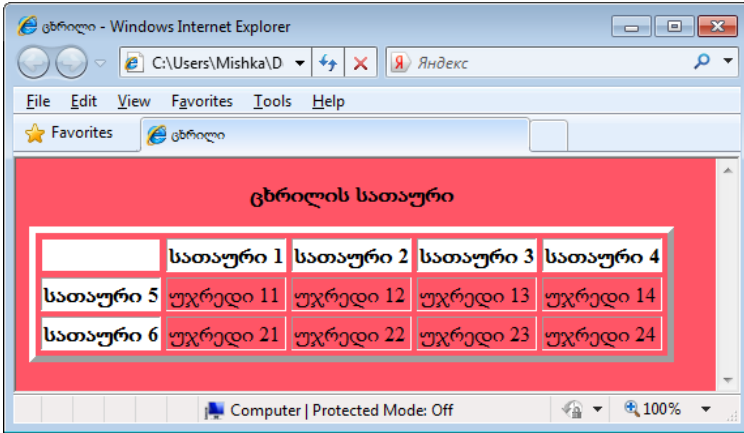
```



მე-10 პროგრამაში განხილულია საერთო სათაურისა და ცალკე სტრიქონებისა და სვეტებისათვის სათაურის მქონე ცხრილის მაგალითი.

**პროგრამა 10.**

```
<html>
<head>
<title>ცხრილი</title>
</head>
<body bgcolor=#FF5566>
<table border=5 cellspacing=4>
<caption><B>ცხრილის სათაური </B></caption>
<tr><th bgcolor="#ffffff">
<th bgcolor="#ffffff">სათაური 1
<th bgcolor="#ffffff">სათაური 2
<th bgcolor="#ffffff">სათაური 3
<th bgcolor="#ffffff">სათაური 4 </tr>
<tr> <th bgcolor="#ffffff">სათაური 5
<td>უჯრედი 11<td>უჯრედი 12<td>უჯრედი 13<td>უჯრედი 14
</tr>
<tr><th bgcolor="#ffffff">სათაური 6
<td>უჯრედი 21<td>უჯრედი 22<td>უჯრედი 23<td>უჯრედი 24
</tr>
</table>
</body>
</html>
```



ცხრილის სტრიქონებისა და სვეტების უჯრედების რთული გაერთიანების მაგალითი მე-11 პროგრამაშია მოცემული.

**პროგრამა 11.**

```

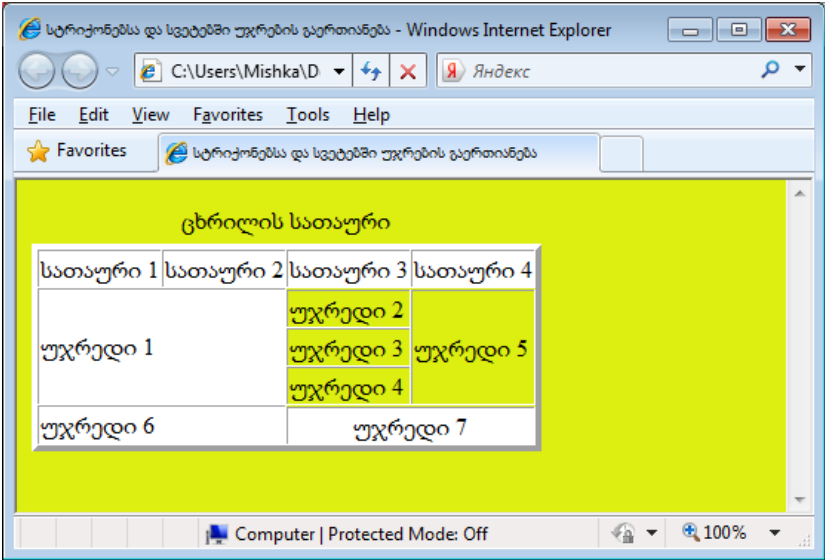
<html>
<head>
<title>სტრიქონებსა და სვეტებში უჯრედების გაერთიანება </title>
</head>
<body bgcolor=#ddee11>
<table border=4 cellspacing=0 >
<caption>ცხრილის სათაური </caption>
<tr><td bgcolor="#ffffff">სათაური 1
<td bgcolor="#ffffff">სათაური 2
<td bgcolor="#ffffff">სათაური 3
<td bgcolor="#ffffff">სათაური 4
<tr><td rowspan=4 colspan=2 bgcolor="#ffffff">უჯრედი 1
<tr><td>უჯრედი 2<td rowspan=3> უჯრედი 5
<tr><td>უჯრედი 3
<tr><td>უჯრედი 4
<tr><td rowspan=4 colspan=2 bgcolor="#ffffff">უჯრედი 6

```

```

<tr><td colspan=3 bgcolor="white" align="center">უჯრედი 7
</table>
</table>
</body>
</html>

```



მე-12 პროგრამაში მოყვანილია ცხრილის მაგალითი, რომელიც სხვა ცხრილის უჯრედშია განთავსებული და თანაც სხვადასხვა სიგანის ჩარჩო აქვთ.

**პროგრამა 12.**

```

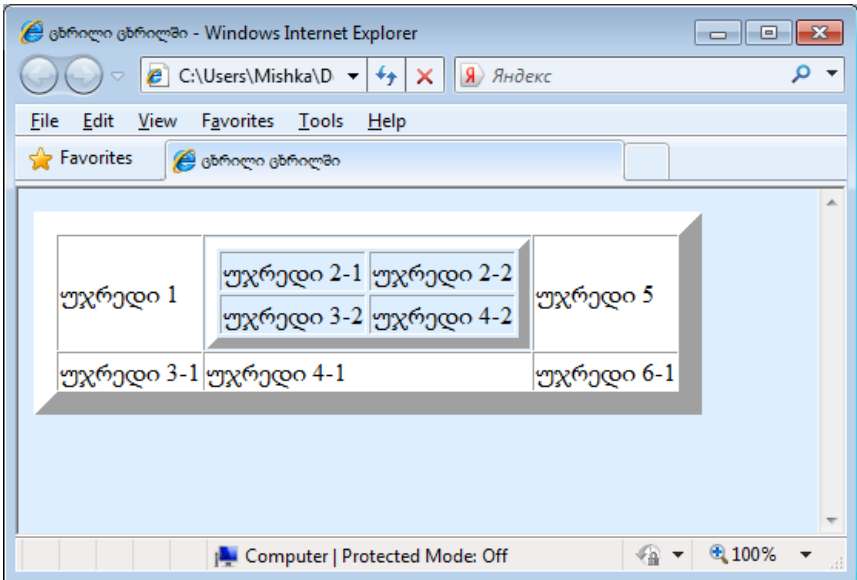
<html>
<head>
<title>ცხრილი ცხრილში</title>
</head>
<body bgcolor=#ddeeff>
<table border=15 cellspacing=0>
<tr><td bgcolor="#ffffff">უჯრედი 1

```

```

<td>
<table border=7>
<tr><td>უჯრედი 2-1
<td>უჯრედი 2-2
<tr><td>უჯრედი 3-2
<td>უჯრედი 4-2
</table>
<td bgcolor="#ffffff">უჯრედი 5
<tr><td bgcolor="#ffffff">უჯრედი 3-1
<td bgcolor="#ffffff">უჯრედი 4-1
<td bgcolor="#ffffff">უჯრედი 6-1
</table>
</body>
</html>

```





## ფორმები

ფორმები HTML-გვერდების მნიშვნელოვან ელემენტს წარმოადგენს, რომელიც ვებ-გვერდის დამპროექტებელს საშუალებას აძლევს ინტერაქტიული ურთიერთობა დაამყაროს მომხმარებელთან. მისი საშუალებით მომხმარებელს შეუძლია ამა თუ იმ ნაწილის მონახულების შესახებ გაგზავნოს კომენტარი, გადააგზავნოს შეკითხვა, დარეგისტრირდეს.

შექმნის რა ფორმას დამპროექტებელი სვამს კითხვას, ხოლო მომხმარებელი ამ ფორმის შევსებით პასუხობს მას. ფორმა სხვადასხვა ტეგისა და ატრიბუტის საშუალებით იქმნება, რომლებიც <FORM> და </FORM> ტეგებს შორის უნდა იყოს მოთავსებული.

### ფორმის შექმნა

ფორმები ინფორმაციის შეტანისათვისაა განკუთვნილი. ყველა ფორმა იწყება <FORM> (ინგლ. form - ფორმა) ტეგით და მთავრდება </FORM> ტეგით. ფორმას შეიძლება შემდეგი ატრიბუტები ჰქონდეს: NAME, ACTION, METHOD, TARGET.

- ატრიბუტი *NAME* - ფორმის სახელს განსაზღვრავს და ჩვეულებრივ არ მიეთითება, თუ დოკუმენტში რამდენიმე ფორმა არ არის გამოყენებული;

- ატრიბუტი *ACTION* - ფორმის აუცილებელი ატრიბუტია, რომელიც ფორმის დასამუშავებელი პროგრამის URL-მისამართს უთითებს;

- ატრიბუტი *METHOD* - ფორმის პარამეტრების გადაცემის მეთოდს განსაზღვრავს. იგი GET ან POST მნიშვნელობებს ღებულობს. ამჟამად, GET მეთოდის გამოყენება რეკომენდირებული აღარ არის;

- ატრიბუტი *TARGET* - განსაზღვრავს ფანჯარას, რომელშიც

გაგზავნილი ფორმის დამუშავების შედეგები დაბრუნდება. მისი შესაძლო პარამეტრებია SELF, PARENT, TOP, BLANK ან ფანჯრის სახელი ცხადად უნდა იყოს მითითებული.

TARGET ატრიბუტის პარამეტრები აღნიშნავს, რომ:

- 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 მშობლიურ ფრეიმს წარმოადგენს (თუ მშობლიური ფრეიმი არ არსებობს, მაშინ დოკუმენტი მიმდინარე ფრეიმში გაიხსნება);
- TOP - დოკუმენტი მიმდინარე ფანჯარაში გაიხსნება და ყველა ფანჯარას დაიჭერს.

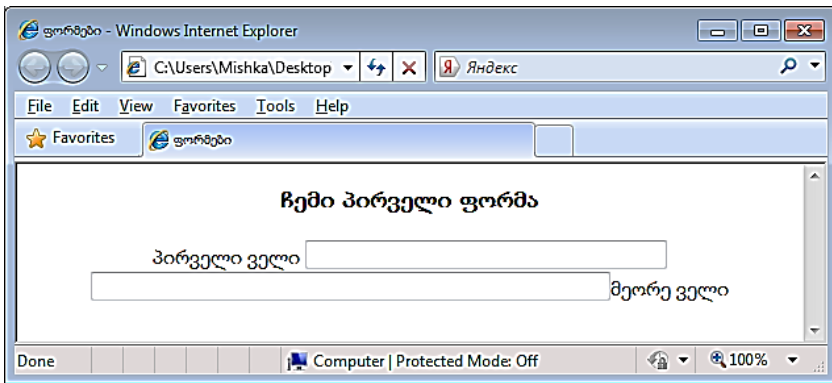
ქვემოთ მოყვანილია ფორმის შექმნის მაგალითი, სადაც ფორმაში გამოყენებული მონაცემთა შესატანი ველი <INPUT> ტეგითაა მოცემული.

### პროგრამა 1.

```
<html>
<head>
<title> ფორმები</title>
</head>
<body>
<form>
<center><h3>ჩემი პირველი ფორმა</h3>
პირველი ველი <input name="პირველი ველი"size="40"><br>
<input name="მეორე ველი"size="60">მეორე ველი <br>
</center>
</form>
```

</body>

</html>



ფორმის შექმნა ადვილია, მაგრამ ძნელია ყველა ელემენტის ლამაზად დაწყოება. ამ პრობლემის გადაწყვეტა ადვილი არ არის. შეიძლება მონაცემთა შესატანი ველები და ჩამოშლადი სიები მარცხენა მხარეს განთავსდეს და ყველა თანმხლები ტექსტი კი - მარჯვენა მხარეს. ვინაიდან მონაცემთა შესატანი ველის სიგრძის განსაზღვრა SIZE ატრიბუტით ხდება, შეიძლება იგი მარცხენა კიდეს გაუსწორდეს, ხოლო ტექსტი მის მარჯვნივ ჩაიწეროს. ეს ფორმა უფრო ლამაზად გამოიყურება. ასეთი ფორმის მაგალითი მე-2 პროგრამაშია ნაჩვენები.

## პროგრამა 2.

<html>

<head>

<title>ფორმები</title>

</head>

<body>

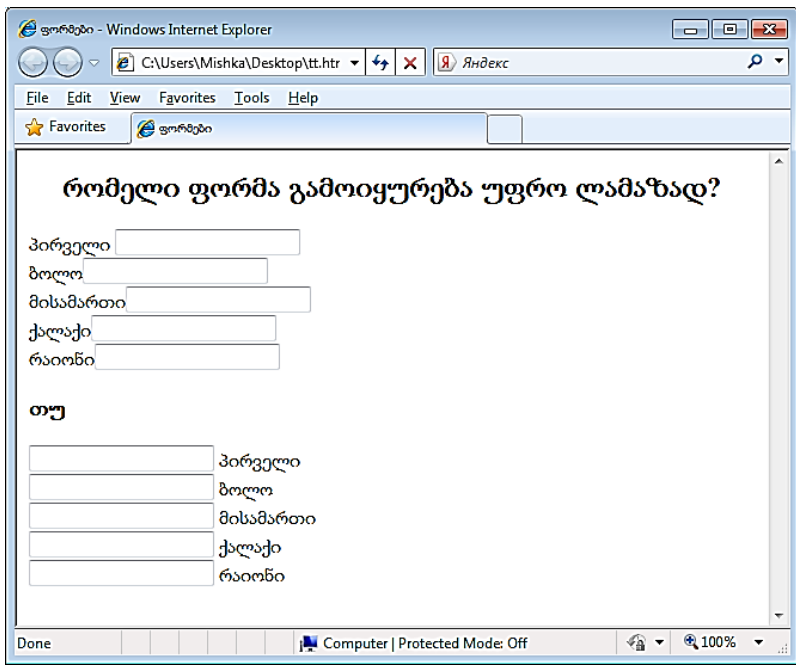
<form>

<center><h2>რომელი ფორმა გამოიყურება უფრო ლამაზად?</h2></center>

```

პირველი <input name="პირველი"size="20"><br>
ბოლო<input name="ბოლო"size="20"><br>
მისამართი<input name="მისამართი"size="20"><br>
<br>
ქლაქი<input name="ქლაქი"size="20"><br>
რაიონი<input name="რაიონი"size="20"><br>
<br>
<h3>თუ</h3>
<br>
<input name="პირველი"size="20"> პირველი <br>
<input name="ბოლო"size="20"> ბოლო<br>
<input name="მისამართი"size="20"> მისამართი<br>
<input name="ქლაქი"size="20"> ქლაქი<br>
<input name="რაიონი"size="20"> რაიონი
</form>
</body>
</html>

```



## მართვის ელემენტების ფორმაზე განთავსება

მონაცემთა შესატანი ველის, რომელსაც ჩვენ ზემოთ გავვეცანით, და მართვის სხვა ელემენტების შესაქმნელად <INPUT> (ინგლ. *input* - შეტანა) ტეგი გამოიყენება. ამ ტეგს NAME, SIZE, CHECKED, MAXLENGTH, SRC, TYPE ატრიბუტები აქვს.

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

- ატრიბუტი *CHECKED* - ალმებით მოსანიშნად და გადამრთველებისათვის გამოიყენება. იგი უჩვენებს, რომ ვებ-გვერდის გახსნის დროს ამ ატრიბუტის მქონე ალამი და გადამრთველი იქნება მონიშნული;

- ატრიბუტი *MAXLENGTH* - ეს არის მომხმარებლის მიერ მონაცემთა შესატან ველში შესატანი სიმბოლოების დასაშვები რაოდენობა. ამ რაოდენობაზე მეტი სიმბოლოს შეტანის მცდელობის დროს ბრაუზერი გამოსცემს ხმოვან სიგნალს და ზედმეტი სიმბოლოს შეტანის საშუალებას არ მოგვცემს. თუ მოცემული ატრიბუტი არ იქნა მითითებული, მაშინ მისი მნიშვნელობა ზოგადად უსასრულობის ტოლია;

- ატრიბუტი *SRC* - გამოსახულების URL-მისამართს მიუთითებს (გამოიყენება *IMAGE* ატრიბუტთან ერთად);

- 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:

- მნიშვნელობა CHECKBOX მონაცემთა შესატან ველს ალამით მოსანიშნ ველად გარდაქმნის, რომელიც მომხმარებლის მიერ მიცემული მარტივი ლოგიკური მნიშვნელობების (ON ან OFF) გამოსაძახებელი პროგრამებისათვის გადასაცემად გამოიყენება;
- მნიშვნელობა HIDDEN მართვის ელემენტს დამალავს - იგი ბრაუზერის მიერ არ აისახება და მომხმარებელს ჩუმათობის პრინციპით მინიჭებული მნიშვნელობის შეცვლის უფლებას არ აძლევს. მართვის ასეთი ელემენტი პროგრამისათვის უცვლელი ინფორმაციის გადასაცემად გამოიყენება, მაგალითად, მომხმარებლის იდენტიფიკატორი ან პაროლი;
- მნიშვნელობა IMAGE საშუალებას იძლევა მართვის ელემენტის სახელს დაუკავშიროთ გამოსახულება. ასეთ გამოსახულებაზე მაუსით დაწკაპუნების შემდეგ ფორმის ასოცირებული პროგრამის გამოძახება მოხდება;
- მნიშვნელობა PASSWORD მონაცემთა შესატან ველს გარეგნულად არ ცვლის, მაგრამ ბრაუზერი მომხმარებლის მიერ შეტანილ მნიშვნელობებს ეკრანზე არ ასახავს;
- მნიშვნელობა RADIO მონაცემთა შესატან ველს ისეთ გადამრთველად გადააქცევს, რომელიც საშუალებას იძლევა NAME ატრიბუტის ერთი და იგივე მნიშვნელობის და VALUE ატრიბუტით მოცემულ გადამრთველის სხვადასხვა მნიშვნელობებს შორის მხოლოდ ერთი მნიშვნელობა აირჩეს. გამოსაძახებელ პროგრამას გადაეცემა NAME=VALUE მნიშვნელობა, ამასთან, VALUE

გადამრთველით მინიჭებულ მნიშვნელობას მიიღებს. ერთი გადამრთველისათვის მინიჭებული მნიშვნელობა სხვა მნიშვნელობების ავტომატურ გაუქმებას იწვევს;

- მნიშვნელობა RESET მონაცემთა შესატან ველს ღილაკად გარდაქმნის, რომელზეც მაუსის დაწკაპუნებით ფორმის მართვის ყველა ელემენტი მათთვის ჩუმათობის პრინციპით მინიჭებულ მნიშვნელობას მიიღებს;
- მნიშვნელობა SUBMIT მონაცემთა შესატან ველს ღილაკად გარდაქმნის, რომელზეც მაუსის დაწკაპუნებით ფორმასთან ასოცირებულ პროგრამაზე ან URL-მისამართზე გადასვლა მოხდება;
- მნიშვნელობა TEXT ერთსტრიქონიან მონაცემთა შესატან ველს აღწერს. მაქსიმალური სიგრძის მნიშვნელობისა და მონაცემთა შესატანი ველის (ჩუმათობის პრინციპით 20 სიმბოლო გამოიყოფა) განსაზღვრისათვის MAXLENGTH და SIZE ატრიბუტები გამოიყენება.

• ატრიბუტი *VALUE* - საშუალებას იძლევა მართვის ელემენტს მიენიჭოს მნიშვნელობა ჩუმათობის პრინციპით ან მნიშვნელობა, რომელიც *NAME* ატრიბუტს შესაბამისი გადამრთველის დაყენების საშუალებით მიენიჭება (გადამრთველის ტიპისათვის მოცემული ატრიბუტი აუცილებელია).

ქვემოთ მოცემულია ფორმის შექმნის მაგალითი გადამრთველით (TYPE=RADIO).

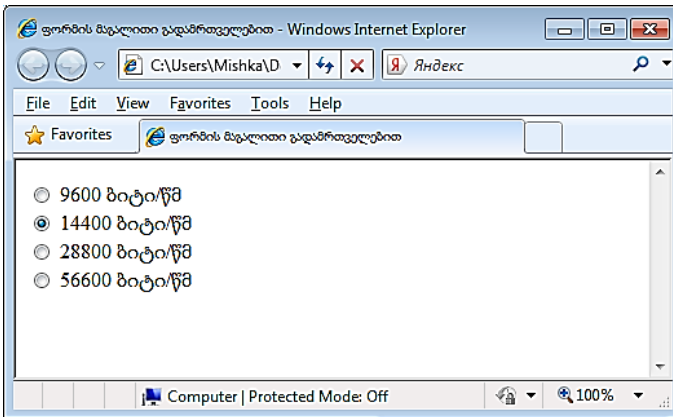
### პროგრამა 3.

```
<html>
<head>
<title>ფორმის მაგალითი გადამრთველებით</title>
</head>
<body>
<input type=radio name=modem value="9600" checked> 9600 ბიტი/წმ
```

```

<br>
<input type=radio name=modem value="14400" checked> 14400
ბიტი/წმ <br>
<input type=radio name=modem value="28800" checked> 28800
ბიტი/წმ <br>
<input type=radio name=modem value="56600" checked> 56600
ბიტი/წმ <br>
</body>
</html>

```



ქვემოთ მოცემულია ფორმის შექმნის მაგალითი, რომელიც გაგზავნის ღილავს შეიცავს (TYPE=SUBMIT).

#### პროგრამა 4.

```

<html>
<head>
<title>ატრიბუტი submit</title>
</head>
<body>
<form action=www.google.ge method=post>
<input size=40 name=text value="მნიშვნელობა ჩუმათობის

```



პრინციპით">

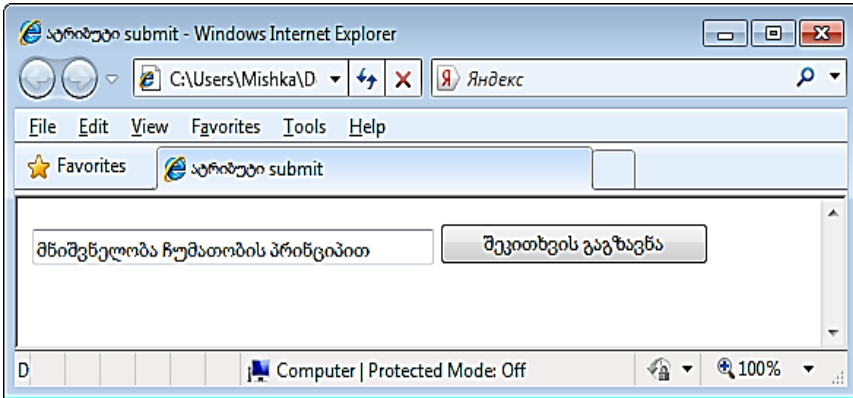
<input type=submit value="შეკითხვის გაგზავნა"

value="გამოქვეყნდეს">

</form>

</body>

</html>



გადამრთველებისაგან განსხვავებით ერთნაირ სახელიანი ალმეებით მოსანიშნ ჯგუფში (TYPE=CHECKBOX) შეიძლება რამდენიმე ალამით იყოს მონიშნული. ქვემოთ პროგრამაში მოყვანილია ალმეებიანი ფორმის მაგალითი, სადაც მესამე ფორმა ალმითაა მონიშნული, რადგან მისთვის CHECKED ატრიბუტია მითითებული.

### პროგრამა 5.

<html>

<head>

<title>ალამი</title>

</head>

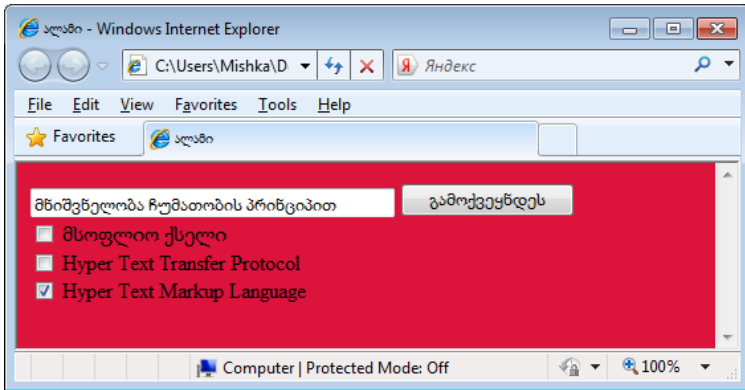
<body bgcolor="#dc143c">

<form method=post action=www.google.ge>

```

<input type=text name=text value="მნიშვნელობა ჩუმათობის
პრინციპით" size=40>
<input type=submit value=გამოქვეყნდეს> <br>
<input type=checkbox name=comp value="www"> მსოფლიო ქსელი
<br>
<input type=checkbox name=comp value="http"> Hyper Text Transfer
Protocol <br>
<input type=checkbox name=comp value="html" checked> Hyper Text
Markup Language
</form>
</body>
</html>

```



ქვემოთ მე-6 პროგრამაში მოყვანილია პაროლის შესაყვანი (TYPE=PASSWORD) ფორმის შექმნის პროგრამა. ასეთ ველში პაროლის შეტანის დროს მასში წერტილები აისახება.

#### პროგრამა 6.

```

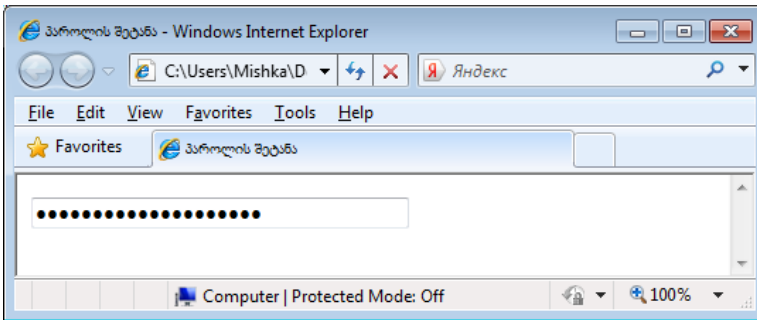
<html>
<head>
<title>პაროლის შეტანა </title>
</head>

```

```

<body>
<form method=post action=www.google.ge>
<input type=password name=text value="შიტანეთ მნიშვნელობა"
size=40 maxlength=40>
<br>
</form>
</body>
</html>

```



## ამორჩევის სიები

ფორმაში სიების შესაქმნელად გამოიყენება წყვილი <SELECT> ტეგი, რომელიც საშუალებას იძლევა მრავალი შესაძლო მნიშვნელობიდან ერთი ან რამდენიმე მნიშვნელობა ამოირჩეს. ჩუმათობით სიის ველში მისი პირველი ელემენტი აისახება. სიის ელემენტი <OPTION> ტეგის საშუალებით იქმნება. ქვემოთ შესაბამისი მაგალითია მოცემული:

### პროგრამა 7.

```

<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

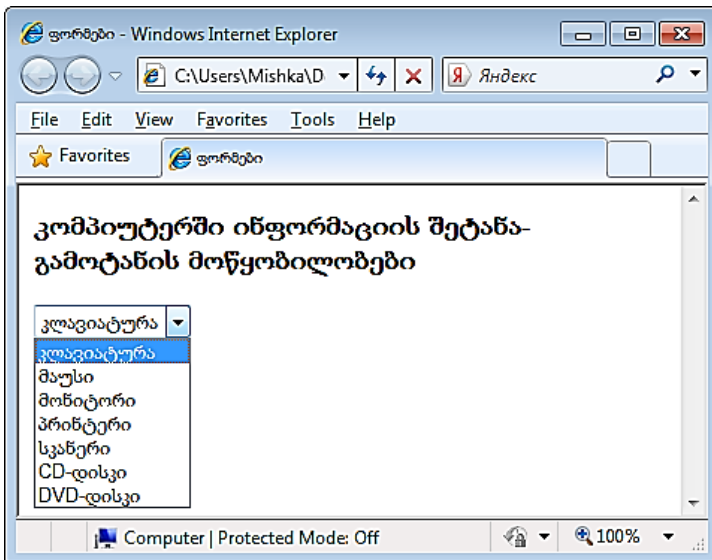
<option> DVD-დისკი

</select>

</form>

</body>

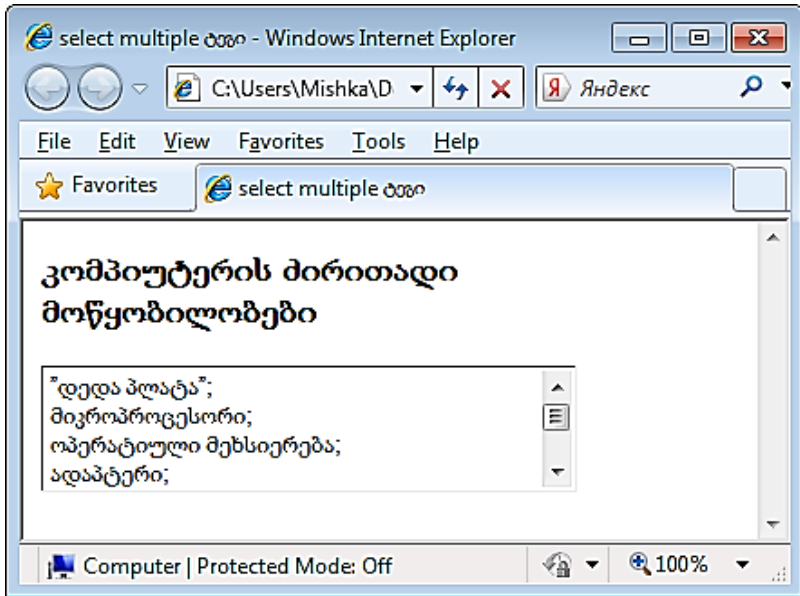
</html>



<SELECT> ტეგით არა მარტო ერთსტრიქონიანი, არამედ მრავალსტრიქონიანი ამორჩევის სიის შექმნა შესაძლებელი. SIZE ატრიბუტით ეკრანზე ერთდროულად გამოსაჩენი სიის ელემენტების, ხოლო MULTIPLE ატრიბუტით ერთდროულად ამოსარჩევი ელემენტების (Shift ან Ctrl კლავიშების დახმარებით) რაოდენობის მითითება ხდება. თუ SIZE ატრიბუტით მითითებული რაოდენობა სიის გამოჩენას სრულად ვერ უზრუნველყოფს, მაშინ მარჯვენა მხარეს გადაფურცვლის ზოლი გაჩნდება. შესაბამისი მაგალითი მე-8 პროგრამაშია მოყვანილი.

**პროგრამა 8.**

```
<html>
<head><title>select multiple ტეგი</title></head>
<body>
<h3 align=left>კომპიუტერის ძირითადი მოწყობილობები</h3>
<form>
<select name=group size=4 multiple=2>
<option> ”დედა პლატა“;
<option> მიკროპროცესორი;
<option> ოპერატიული მეხსიერება;
<option> ადაპტერი;
<option> მყარ დისკზე დამხსომებელი მოწყობილობა;
<option> მონიტორი;
<option> კლავიატურა;
</select>
</form>
</body>
</html>
```



<OPTION> ტეგში აუცილებლად უნდა მივუთითოთ ატრიბუტი VALUE. მხოლოდ ამ შემთხვევაში, ფორმასთან ასოცირებული პროგრამა მიხვდება თუ სიის რომელი ელემენტია ამორჩეული. შესაბამისი მაგალითი მე-9 პროგრამაშია მოყვანილი.

### პროგრამა 9.

```
<html>
<head>
<title> select.htm </title>
</head>
<body>
<form>
```

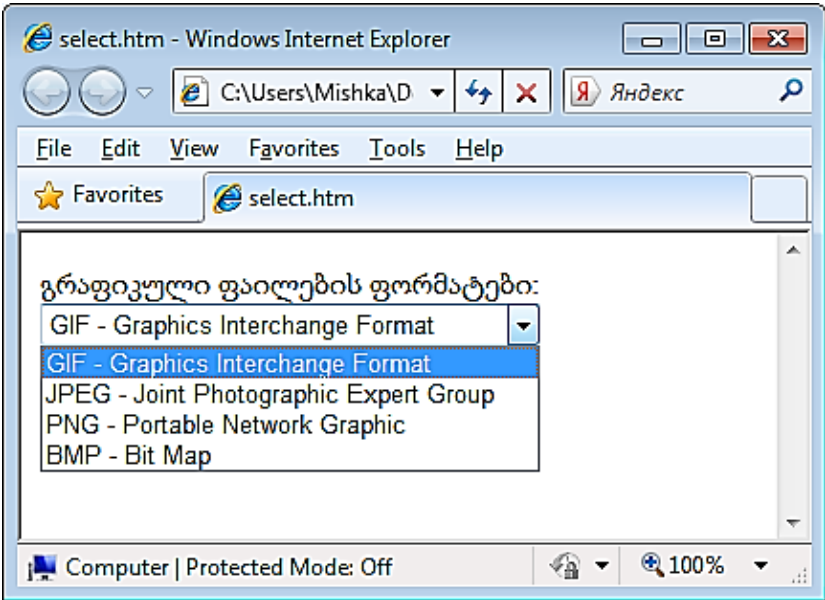
გრაფიკული ფაილების ფორმატები:

```
<select name="GIF - Graphics Interchange Format">
<option selected value="Graphics Interchange Format"> GIF - Graphics
Interchange Format
```

```

<option value="JPEG">JPEG - Joint Photographic Expert Group
<option value="PNG">PNG - Portable Network Graphic
<option value="BMP">BMP - Bit Map
</select>
</form>
</body>
</html>

```



### მრავალსტრიქონიანი ტექსტური ველი

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 გახსნის დროს ჩუმათობის პრინციპით რაიმე ტექსტი გამოვიდეს, საჭიროა იგი <TEXTAREA> და </TEXTAREA> ტეგებს შორის ჩავსვათ.

<TEXTAREA> ტეგის გამოყენების მაგალითი მე-10 პროგრამაშია მოცემული.

### **პროგრამა 10.**

```
<html>
```

```
<head>
```

```
<title>ტეგი textarea </title></head>
```

```
<body>
```

```
<form>
```

```
<textarea name="რესურსი და ფორმატი" rows=10 cols=50>URL
```

Uniform Resource Locator - რესურსის უნიფიცირებული  
მაჩვენებელი

WWW-ში ფაილების დამისამართების სტანდარტული  
საშუალებანი

EPS

Encapsulated PostScript

ფორმატი, რომელიც გრაფიკული გამოსახულების შენახვის  
საშუალებას იძლევა და ერთი დანართიდან სხვა დანართში  
გადააქვს ისინი.

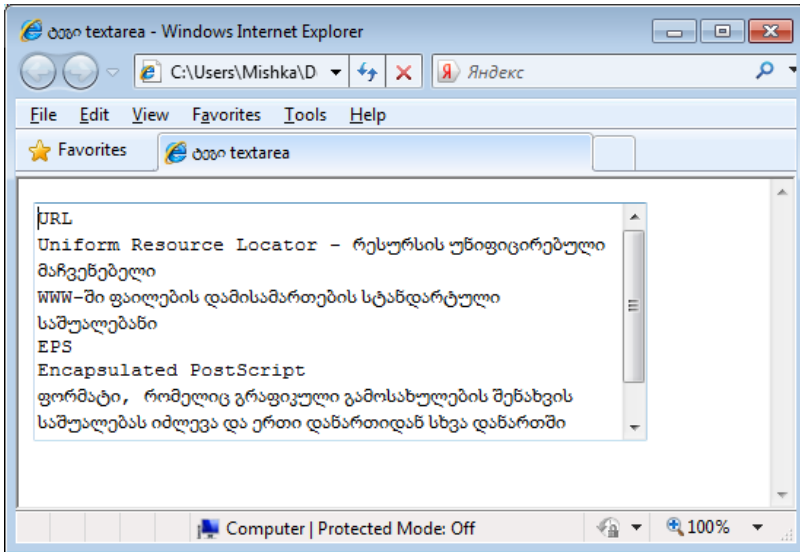
```
</textarea>
```

```
</form>
```

```
</body>
```

```
</html>
```





მე-11 პროგრამაში მოყვანილია მაგალითი სადაც ფორმის სხვადასხვა ატრიბუტები და მათი სხვადასხვა მნიშვნელობები კომპლექსურადაა გამოყენებული:

### პროგრამა 11.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> New Document </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<center> <H2 >დეტექტივების მოყვარულთა კლუბის  
სარეგისტრაციო გვერდი </H2>
```

```
</center>
```

ანკეტის შევსების შემდეგ შეგეძლებათ ისარგებლოთ ელექტრონული ბიბლიოთეკით

```
<FORM METHOD="get" ACTION=" ">
```

```
<pre>
```

<input type="text" name="sax" maxlength=8 size=20> შეიყვანეთ  
რეგისტრაციის სახელი

<input type="password" name="pas1" maxlength=8 size=22>

შეიყვანეთ პაროლი

<input type="password" name="pas2" maxlength=8 size=22>

დაადასტურეთ პაროლი

</pre>

თქვენი ასაკი:<br>

<input type="radio" name="age" value="20">20-მდე <br>

<input type="radio" name="age" value="20-30">20-30<br>

<input type="radio" name="age" value="30-50">30-50<br>

<input type="radio" name="age" value="50z">50-ის ზემოთ<br>

რომელ ენაზე გსურთ ლიტერატურის წაკითხვა:

<pre>

<input type="checkbox" name="language"

value="georgian">ქართული

<input type="checkbox" name="language"

value="english">ინგლისური

<input type="checkbox" name="language"

value="german">გერმანული

<input type="checkbox" name="language" value="french">ფრანგული

<input type="checkbox" name="language" value="russian">რუსული

</pre>

წიგნის რომელი ფორმატია თქვენთვის მისაღები:<br>

<select name="format" size="1">

<option value="html">html

<option value="plain text">plain text

<option value="postscript">postscript

<option selected value="PDF">PDF

<option value="djvu">DJVU

</select>

<br><br>

თქვენი საყვარელი ავტორი:<br>

<textarea name="wish" cols=40 rows=3>

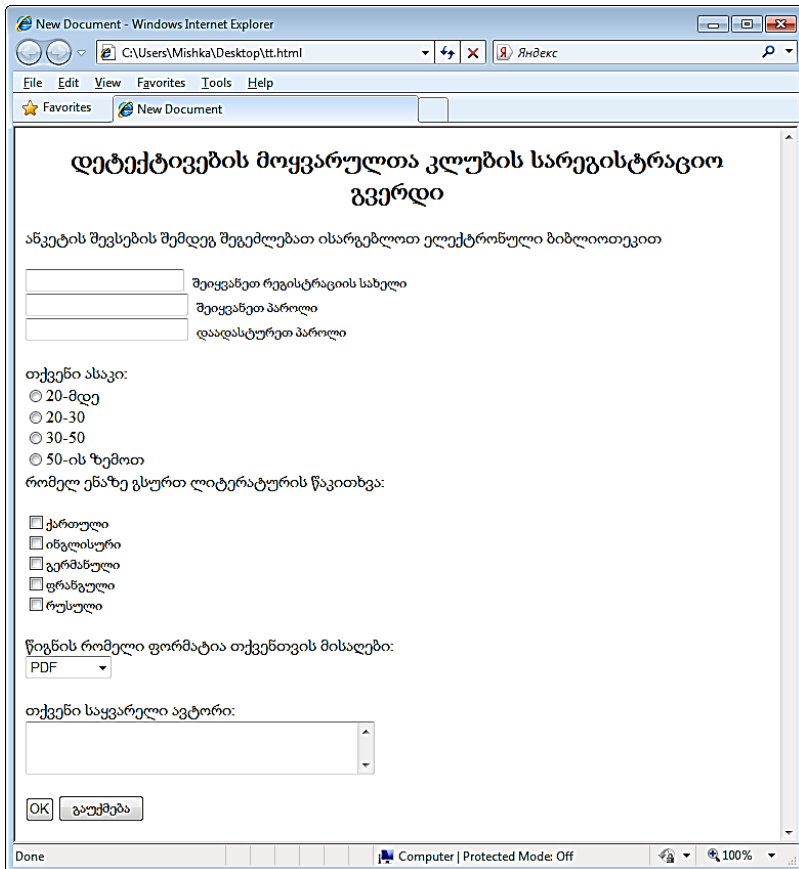
</textarea><br><br>

<input type="submit" value="OK">

<input type="reset" value="გაუქმება">

</BODY>

</HTML>



## ბმულები

ბმულები ერთ-ერთი მნიშვნელოვანი კომპონენტია, ვინაიდან იგი ვებ-გვერდებს მომხმარებლისათვის უფრო მომხიბვლელს ხდის. ბმულების, ანუ როგორც ზოგჯერ მას უწოდებენ ჰიპერბმულების დახმარებით ვებ-გვერდის 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, მაუსის უბრალო დაწკაპუნებით ერთი გვერდიდან მეორეზე გადასვლა შეუძლია.

## რესურსის უნივერსალური მაჩვენებელი

HTML ენაში ბმულების ორგანიზებისათვის რესურსის უნივერსალური მაჩვენებელი (Uniform Resource Locator, URL) გამოიყენება, რომლის ფორმატია:

მეთოდი://სერვერის\_სახელი:პორტი/გზა#ღუზა

URL-ის პირველი ნაწილი (მეთოდი) წვდომის მეთოდს ან ქსელის სამსახურს აღწერს, მეორე ნაწილი კი წვდომის მეთოდზეა დამოკიდებული. ჩვეულებრივ ორი წერტილის შემდეგ მდგომი ორი დახრილი ხაზი კომპიუტერის სახელს აღნიშნავს.

უფრო ხშირად გამოყენებული მეთოდებია file, http, ftp, mailto, telnet და news.

*file* მეთოდი ლოკალური დისკიდან ფაილის წაკითხვას უზრუნველყოფს. მაგალითად, შემდეგი ჩანაწერი kat41/spb საქაღალდიდან document.html HTML-ფაილს მომხმარებლის კომპიუტერში გახსნის: □

file:/kat41/spb/ document.html

*http* მეთოდი ვებ-გვერდთან წვდომას HTTP პროტოკოლით უზრუნველყოფს. ეს ქსელში HTML-დოკუმენტზე წვდომის ყველაზე გავრცელებული მეთოდია. მაგალითად, □

`http://www.gtu.ge/`

*ftp* მეთოდის მიერ FTP-სერვერზე გაგზავნილ შეკითხვაზე პასუხის მიღება სრულდება. მაგალითად:□

`ftp://myname/directory/name`

*mailto* მეთოდი მითითებულ მომხმარებელსა და ჰოსტს შორის საფოსტო კავშირის სენსის გააქტიურებას ახდენს. მაგალითად, შემდეგი ჩანაწერი gmail.com კომპიუტერზე temuri.sturua მომხმარებლისათვის შეტყობინების გაგზავნის გააქტიურებას ახდენს, რა თქმა უნდა თუ ბრაუზერი ელექტრონული ფოსტის გაგზავნას უჭერს მხარს:

`mailto:temuri.sturua@gmail.com`

*mailto* მეთოდის გამოყენების დროს ორი წერტილის შემდეგ ორი დახრილი ხაზის ჩაწერა საჭირო არ არის. როგორც წესი მას მომხმარებლის საფოსტო მისამართი მოსდევს.

*telnet* მეთოდი telnet სამსახურზე მიმართვისათვის გამოიყენება.

*news* მეთოდი ახალი ამბების სამსახურის გამოძახებას უზრუნველყოფს, თუ ბრაუზერი ასეთ გამოძახებას მხარს უჭერს. მაგალითად:

`news:relcom.www.support`

როგორც *mailto* მეთოდის დროს, აქაც ორი წერტილის შემდეგ ორი დახრილი ხაზის ჩაწერა საჭირო არ არის.

სერვერის სახელი - ქსელში კომპიუტერის სრულ სახელს აღწერს და URL-ის აუცილებელი პარამეტრი არ არის. მაგალითად, `www.gtu.ge` – ეს არის საქართველოს ტექნიკური უნივერსიტეტის სერვერის სრული ქსელური სახელი. თუ სერვერის სახელი არ არის მითითებული, მაშინ ბმული ითვლება ლოკალურად და

URL-ში მითითებული სრული სახელი იმ კომპიუტერს მიეკუთვნება, რომელშიც ამ ბმულის შემცველი HTML-დოკუმენტია განთავსებული.

URL-ში სერვერის სახელის შემდეგ იმ TCP პორტის ნომერი მითითდება, რომელზეც ვებ-სერვერი ფუნქციონირებს. თუ პორტი არ არის მითითებული, მაშინ ჩუმათობის პრინციპით გამოიყენება 80 პორტი (რაც უმეტესობა URL-მისამართებისათვისაა დამახასიათებელი).

შემდეგ URL-ში მითითება დოკუმენტის კერძო ან ზუსტი მისამართი, რომლის გამოძახებაც ამ ბმულზე გადასვლის შედეგად უნდა მოხდეს.

ლუზა - ეს არის მიმდინარე HTML-დოკუმენტის შიგნით განთავსებული ბმული. როგორც შემდგომში იქნება ნაჩვენები, დოკუმენტში ადგილი, რასაც ლუზა გვიჩვენებს, NAME ატრიბუტის საშუალებით მითითდება.

URL-მისამართის ტექსტი ბრაუზერის მიერ გამონათებული არ იქნება და იგი მხოლოდ ბმულის გააქტიურების დროს იქნება შესრულებული.

## ბმულების დოკუმენტში ჩასმა

ბმულების დოკუმენტში ჩასასმელად <A> წყვილი ტეგი გამოიყენება. იგივე ტეგი საშუალებას იძლევა დოკუმენტში ეგრეთ წოდებული ლუზა ჩავსვათ. <A> ტეგს შემდეგი ატრიბუტები აქვს HREF, NAME, ID, METHODS, FRAME, TITLE, ACCESSKEY .

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

<a href="url">

ამ ატრიბუტის გამოყენების დროს ნებისმიერი ტექსტი,

რომელიც <A> და </A> ტეგებს შორის არის მოთავსებული ვებ-ბრაუზერის მიერ სპეციალურად გამოიყოფა (ჩვეულებრივ სტანდარტულად ხაზგასმული გამოდის და ლურჯი ფერითაა გამოყოფილი.

- ატრიბუტი *NAME* – HREF ატრიბუტის მაგივრად დოკუმენტში ღუზის ჩადგმის დროს მიეთითება. □

- ატრიბუტი *ID* – NAME ატრიბუტის მსგავსია, იმ განსხვავებით, რომ იგი არა მხოლოდ <A> ტეგთან, არამედ სხვა ტეგებშიც გამოიყენება. მაგალითად: □

```
<p id="აზზაცის სახელი">
```

ასეთი მონიშვნის არსებობის შემთხვევაში მიმდინარე დოკუმენტში ბმული შეიძლება შემდეგნაირად შეიქმნას:

```
<a href="#აზზაცის სახელი"> ბმული</a>
```

- ატრიბუტი *METHODS* - დოკუმენტის ამოღების მეთოდს განსაზღვრავს, მაგალითად, FTP, Gopher და ა. შ. მისი სინტაქსია:

```
<A METHODS="დასახელება"> არა აუცილებელი ტექსტი </A>
```

- ატრიბუტი *FRAME* - მიუთითებს ფრეიმს, რომელშიც საჭირო დოკუმენტის გამოტანა უნდა მოხდეს. მისი სინტაქსია:

```
<a frame="ფრეიმის სახელი"> არა აუცილებელი ტექსტი </a>
```

- ატრიბუტი *TITLE* - მოკარნახის გამონათებისათვის გამოიყენება. მისი სინტაქსია: □

```
<a title="ბმულის სახელი"> ბმული</a>
```

- ატრიბუტი *ACCESSKEY* - საშუალებას გვაძლევს ე. წ. "ცხელი კლავიში" მივუთითოთ, რომელზეც ხელის დაჭერით ბმულზე გადასვლა მოხდება. მისი სინტაქსია:

```
<a accesskey="კლავიშის სახელი"> ბმული</a>
```

ქვემოთ ბმულის რამოდენიმე მაგალითია მოყვანილი:

მიმართვა HTML-ფაილზე HTTP პროტოკოლის გამოყენებით:

```
<a href="http://www.gtu.ge/page1.html"> ბმული</a>
```

მიმართვა ZIP-ფაილზე HTTP პროტოკოლის გამოყენებით: □  
<a href="http://www.gtu.ge/test.zip"> ბმული</a>

მიმართვა TXT-ფაილზე HTTP პროტოკოლის გამოყენებით: □  
<a href="http://www.gtu.ge/text.txt"> ბმული</a>

მიმართვა MPG-ფაილზე HTTP პროტოკოლის გამოყენებით:  
<a href="http://www.gtu.ge/name.mpg">ბმული</a>

მიმართვა EXE-ფაილზე FTP პროტოკოლის გამოყენებით: □  
<a href="ftp://ftp.spbszip.com/pub/priem/priem/install.exe">ბმული</a>  
ბმული, რომელიც ელექტრონული ფოსტით შეტყობინების  
ინიცირებას ახდენს:

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

ბმული, რომელიც ელექტრონული ფოსტით ავტომატურად  
შეყვანილი შეტყობინების თემის შეტყობინების ინიცირებას  
ახდენს:

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

ბმული, რომელიც ელექტრონული ფოსტით ავტომატურად  
შეყვანილი შეტყობინების თემისა და ტექსტის შეტყობინების  
ინიცირებას ახდენს:

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

ბმული, რომელიც ელექტრონული ფოსტით ავტომატურად  
შეყვანილი შეტყობინების თემის, ტექსტის და შეტყობინების  
ინიცირებას ახდენს, აგრეთვე, ასლებსა და დაფარულ ასლებს  
მოცემულ მისამართებზე გაგზავნის:

<a href="mailto:name@gmail.com, SUBJECT=თემა &body=ტექსტი&cc=  
name1@gmail.com&bcc=name2@gmail.com"> ბმული</a>

ბმული ახალი ამბების ჯგუფზე:

<a href="news:comp.infosystems.www.announce"> ბმული</a>

როდესაც მაუსის მაჩვენებელს მიუჯახლოვებთ რომელიმე  
ბმულს, იგი შეიცვლება და ხელის ფორმას მიიღებს.



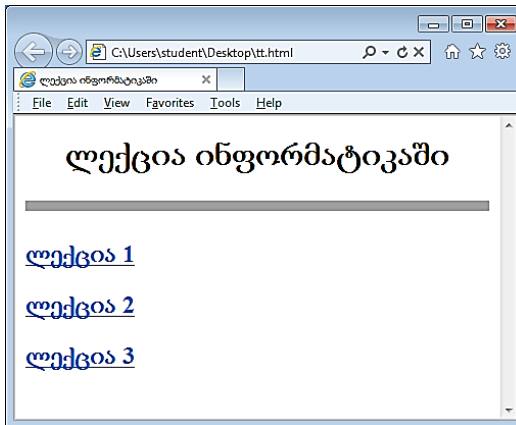
## შიდა ბმულები

შიდა ბმულები მიმდინარე ვებ-კვანძზე განთავსებულ სხვა ვებ-გვერდებზე მიუთითებენ. ამიტომ, შიდა ბმულების შექმნის დროს <A> ტეგის HREF ატრიბუტში ძებნის ფარდობითი გზის მითითებაა საკმარისი.

დოკუმენტის მაგალითი, რომელიც შიდა ბმულს შეიცავს პირველ პროგრამაშია მოყვანილი.

### პროგრამა 1.

```
<html>
<head>
<title>ლექცია ინფორმატიკაში</title></head>
<body background=ffffff>
<center><h1>ლექცია ინფორმატიკაში</h1></center>
<hr noshade size=10 width="100%" align=center>
<p><h2><a href="ლექცია1.htm">ლექცია 1</a></p>
<p><a href="ლექცია2.htm">ლექცია 2</a></p>
<p><a href="ლექცია3.htm">ლექცია 3</a></p></h2>
</body>
</html>
```



## გარე ბმულები

გარე ბმულები იმ ვებ-გვერდებზე მიუთითებენ, რომლებიც მიმდინარე ვებ-კვანძზე არ არის განთავსებული. ამიტომ, გარე ბმულების შექმნის დროს <A> ტეგის HREF ატრიბუტში URL-მისამართის სრულად (ვებ-კვანძის პროტოკოლისა და მისამართის ჩათვლით) მითითებაა აუცილებელი.

თუ ფაილებს ლექცია1.htm, ლექცია2.htm და ლექცია3.htm C:/GTU/HTML/ საქაღალდეში განვთავსებთ, მაშინ ვებ-გვერდის კოდს მე-2 პროგრამის სახე ექნება.

### პროგრამა 2.

```
<html>
<head>
<title>ლექცია ინფორმატიკაში</title>
</head>
<body background="">
<center>
<h1>ლექცია ინფორმატიკაში </h1></center>
<hr align=center width="100%" noshade size=10>
<h2><a href="file:///C:/GTU/HTML/ლექცია1.htm">ლექცია 1</a>
<p><a href="file:///C:/GTU/HTML/ლექცია2.htm">ლექცია 2</a></p>
<a href="file:///C:/GTU/HTML/ლექცია3.htm">ლექცია 3</a></h2>
</body>
</html>
```

## 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 გადასვლების ორგანიზებისათვის სპეციალური ღუზები გამოიყენება, რომლებიც ჩვეულებრივ ფრაგმენტის პირველ სტრიქონში ან სათაურში განთავსდება.

ღუზის ჩასმისათვის ასევე <A> ტეგი გამოიყენება, მაგრამ HREF ატრიბუტის ნაცვლად NAME ატრიბუტი უნდა მიეთითოს. ამ ატრიბუტის მნიშვნელობა ღუზის სახელია. ის მხოლოდ ლათინური ასოებისაგან და ციფრებისაგან უნდა შედგებოდეს და ჰარს არ უნდა შეიცავდეს, მაგალითად:

```
<a name="ღუზის_სახელი"> დოკუმენტის შიგნით გადასვლა </a>
```

დაყენებულ ღუზაზე გადასასვლელად URL-მისამართის ბოლოს ღუზის სახელის მითითებაა აუცილებელია, რომლის წინ # სიმბოლო იქნება ჩაწერილი:

```
<a href="#ღუზის სახელი"> ბმული </a>
```

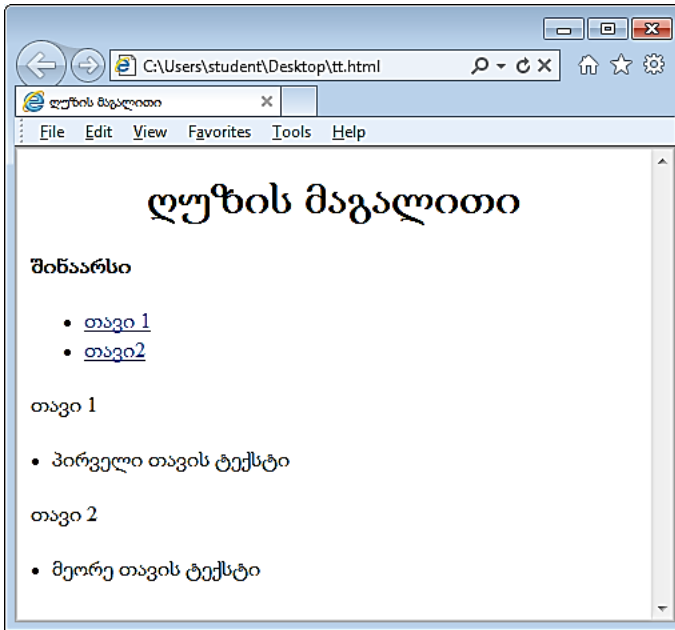
ღუზის გამოყენების მაგალითი მე-3 პროგრამაშია განხილული:

### პროგრამა 3.

```
<html>
<head>
<title>ღუზის მაგალითი</title>
</head>
<body background="#fff8dc">
<center>
<h1>ღუზის მაგალითი </h1></center>
<p><b>შინაარსი</b></p>
<ul> <li><a href="#chapter1">თავი 1</a></li>
<li><a href="#chapter2">თავი2</a></li> </ul>
<p><a name="chapter1"></a>თავი 1</p>
<li> <p>პირველი თავის ტექსტი</p> </ul>
<p><a name="chapter2"></a>თავი 2</p>
<li> <p>მეორე თავის ტექსტი<br></p>
```

</body>

</html>



როდესაც მომხმარებელი მაუსით თავი 1-ზე დააწკაპუნებს, ბრაუზერი მაშინვე პირველ თავზე გადავა, ლუზა შეიძლება როგორც მიმდინარე, ასევე სხვა დოკუმენტში დადგეს. მეორე შემთხვევაში ბრაუზერი ჯერ ახალ დოკუმენტს ჩატვირთავს და შემდეგ მითითებულ თავზე გადავა.

ლუზის გამოყენების კიდეც ერთი მაგალითი მე-4 პროგრამაშია განხილული:

#### პროგრამა 4.

```
<html>
```

```
<head>
```

```
<title>დიდ დოკუმენტში ლუზების შექმნის
```

```
მაგალითი</title></head>
```

```

<body bgcolor="#fffacd" topmargin=4 marginheight=4 leftmargin=100>
<center>
<h1>პერსონალური კომპიუტერის აპარატურული
უზრუნველყოფა </h1>
</center>
<p align="justify" >პერსონალური კომპიუტერების ვარიანტებისა
და მოდიფიკაციების დიდი მრავალფეროვნების მიუხედავად
კომპიუტერის აპარატურული ნაწილი (Hardware) ერთიდაიმავე
სახის მოწყობილობებისაგან შედგება. პირობითად ისინი
შეიძლება ძირითად და დამატებით მოწყობილობებად დაიყოს.
კომპიუტერის ძირითადი მოწყობილობებია: <a href="#chapter1">
სისტემური პლატა</a>,
<a href="#chapter2">მიკროპროცესორი</a>, მეხსიერება
(კომპიუტერის მეხსიერება ორ ჯგუფად იყოფა: ოპერატიული და
გარე მეხსიერება. კომპიუტერის გარე მეხსიერებას მიეკუთვნება
მყარ, დრეკად და ლაზერულ დისკოზე დამხსომებელი
მოწყობილობები), <a href="#chapter3">ოპერატიული
მეხსიერება</a>, <a href="#chapter4">მყარ (ვინჩესტერი) მაგნიტურ
დისკზე დამხსომებელი მოწყობილობა</a>,
<a href="#chapter5">კლავიატურა</a> და
<a href="#chapter6">მონიტორი.</a> <br>
დამატებითი მოწყობილობებია: <a href="#chapter7">მაუსი</a>,
<a href="#chapter8">ლაზერულ (კომპაქტ-დისკები) მაგნიტურ
დისკზე დამხსომებელი მოწყობილობა</a>,
<a href="#chapter9">პრინტერი</a>,
<a href="#chapter10">სკანერი</a>,
<a href="#chapter11">მოდემი</a>, <a href="#chapter12">ქსელური
ადაპტერი</a> და სხვა. ყველა ეს მოწყობილობა სპეციალური
პორტების (შესაერთებლების) საშუალებით უკავშირდება
სისტემურ ბლოკს. </p>

```

<p align="justify" ><a name="chapter1"><b> სისტემური პლატა  
</b></a>

ძალზე რთული "ორგანიზმია", რომლის თითოეული ნაწილის მუშაობაზეა დამოკიდებული კომპიუტერის სწრაფი და სტაბილური მუშაობა, რის გამოც სისტემურ პლატას ხშირად დედა პლატის სახელწოდებითაც მოიხსენიებენ.</p><br>

<p align="justify" ><a name="chapter2"><b>მიკროპროცესორი (პროცესორი)</b></a> კომპიუტერში ყველაზე ძირითად ელემენტს, მის ტვინს წარმოადგენს. იგი მცირე ზომის ელექტრონული სქემას, რომელიც გამოთვლებსა და პროგრამებს ასრულებს, ახდენს ინფორმაციის დამუშავებას, მართავს კომპიუტერის სხვადასხვა მოწყობილობების მუშაობას. </p><br>

<p align="justify" ><a name="chapter3"><b>ოპერატიული მეხსიერება </b></a> ერთ-ერთი ძირითადი მოწყობილობაა. სახელწოდება "ოპერატიული" მისი სწრაფქმედების გამო უწოდეს. თანამედროვე ოპერატიული მეხსიერებისათვის მონაცემებზე მიმართვის დრო 6-7 ნანოწამით (1 ნწმ=10-9წმ) განისაზღვრება. მასში მონაცემები დროებით ინახება, ვიდრე კომპიუტერი ჩართულია, ხოლო გამორთვის შემდეგ ეს ინფორმაცია იკარგება. </p><br>

<p align="justify" ><a name="chapter4"><b>მყარ მაგნიტურ დისკოზე დამსხომებული მოწყობილობა (ვინჩესტერი)</b></a> კომპიუტერის მუშაობისათვის საჭირო ინფორმაციის მუდმივად შესანახადაა განკუთვნილი. ესენია: ოპერაციული სისტემის პროგრამები, ხშირად გამოსაყენებელი პროგრამათა პაკეტი, დოკუმენტთა რედაქტორები, დაპროგრამების ენათა ტრანსლატორები, მომხმარებელთა მონაცემები და მრავალი სხვა. </p><br>

<p align="justify" ><a name="chapter5"><b>კლავიატურა</b></a> მოწყობილობაა, რომლის მეშვეობითაც კომპიუტერს

მომხმარებლის მიერ დასამუშავებელი ინფორმაცია და ურთიერთობისათვის აუცილებელი ბრძანებები მიეწოდება.

</p><br>

<p align="justify" ><a name="chapter6"><b>მონიტორი </b></a> მოწყობილობა, რომელიც ციფრული, ტექსტური და გრაფიკული ინფორმაციის ეკრანზე გამოსატანადაა განკუთვნილი. იგი, აგრეთვე საშუალებას გვაძლევს ჩვენს მიერ კლავიატურაზე აკრეფილი ინფორმაციის სისწორე ვაკონტროლოთ.

აგრეთვე, მნიშვნელოვანი ფაქტორია მონიტორის ადაპტერის მეხსიერება – ვიდეომეხსიერება. მასში ხდება ეკრანზე გამოსატანი კადრების დაგროვება, რაც კადრების მონიტორზე დიდი სისწრაფითა და მაღალი ხარისხით გამოტანას უზრუნველყოფს. </p><br>

<p align="justify" ><a name="chapter7"><b>მაუსი </b></a> სპეციალური მანიპულატორია, რომელიც საშუალებას გვაძლევს, კლავიატურასთან შედარებით უფრო სწრაფად და მოხერხებულად მივაწოდოთ კომპიუტერს ბრძანებები. </p><br>

<p align="justify" ><a name="chapter8"><b>ლაზერულ (კომპაქტ-დისკები) მაგნიტურ დისკოზე დამხსომებელი მოწყობილობა </b></a> საშუალებას გვაძლევს დოკუმენტები და პროგრამები ერთი კომპიუტერიდან მეორეზე გადავიტანოთ, შევინახოთ კომპიუტერზე იშვიათად გამოსაყენებელი ინფორმაცია, გავაკეთოთ ინფორმაციის არქივული ასლები.</p><br>

<p align="justify" ><a name="chapter9"><b>პრინტერი </b></a> საბეჭდი მოწყობილობაა, რომლის მეშვეობითაც ხდება ინფორმაციის გადატანა ქაღალდზე. მოქმედების პრინციპის მიხედვით მატრიცულ, ჭავლურ და ლაზერულ პრინტერებს განასხვავებენ.</p><br>

<p align="justify" ><a name="chapter10"><b>სკანერი </b></a> ინფორმაციის ვიზუალურად წამკითხავი მოწყობილობაა,

რომლის მეშვეობითაც კომპიუტერის მესხიერებაში ტექსტური, გრაფიკული და ფოტო მასალა შეგვიძლია სწრაფად შევიტანოთ.

</p><br>

<p align="justify" ><a name="chapter11"><b>მოდემი </b></a>

მოწყობილობაა, რომელიც ინფორმაციის მიმოცვლას სხვა

კომპიუტერებთან სატელეფონო ქსელის მეშვეობით

უზრუნველყოფს. </p><br>

<p align="justify" ><a name="chapter12"><b>ქსელური ადაპტერი

</b></a> საშუალებას გვაძლევს, კომპიუტერი ლოკალურ ქსელში

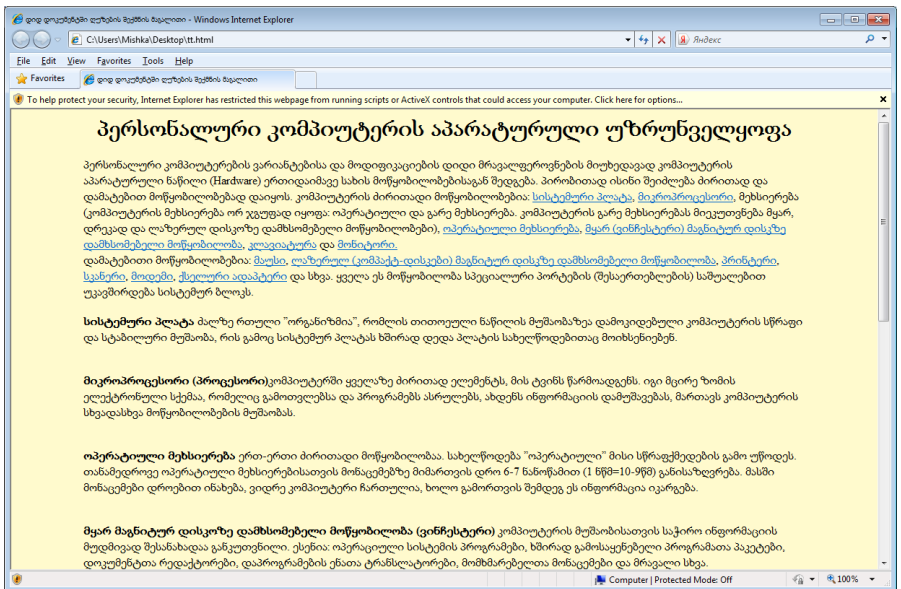
ჩავერთოთ, ამასთან, მომხმარებელს საშუალება აქვს

დაუბრკოლებლად მიიღოს და გამოიყენოს სხვა კომპიუტერში

არსებული ინფორმაცია.</p><br>

</body>

</html>





## გრაფიკული ბმულები

ბმული შეიძლება იყოს როგორც ტექსტი, ასევე გამოსახულება. ამასთან, HTML ენაზე პროგრამის კოდი და გამოსახულება ერთ საქალაქდებში უნდა იყოს განთავსებული.

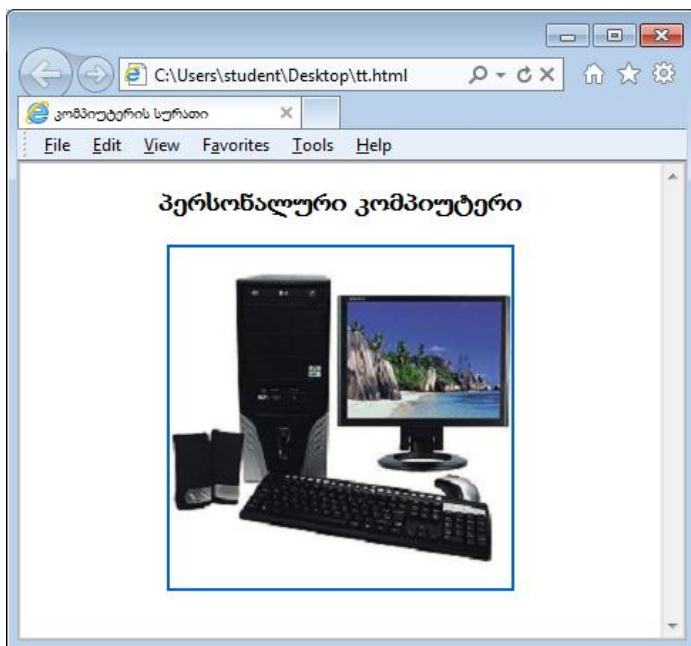
გრაფიკული ბმულის მაგალითი მე-5 პროგრამაშია მოცემული.

### პროგრამა 5.

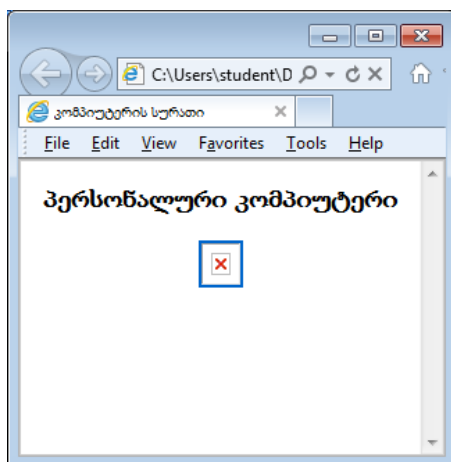
```
<html>
<head>
<title>კომპიუტერის სურათი</title>
</head>
<body bgcolor=#ffffff>
<h3 align=center>პერსონალური კომპიუტერი</h3>
<p align=center><a href="programm4.htm">
</font> </a></p>
</body>
</html>
```

გამოსახულებაზე მაუსის დაწკაპუნებით programm4.htm ფაილი გაიხსნება, ანუ გაიხსნება მე-4 პროგრამის შესაბამისი ვებ-გვერდი (იხ. ნახ. 7).

თუ სურათი მოცემულ საქალაქდებში არ არის, მაშინ გამოსახულების ნაცვლად ეკრანზე გამონათდება კვადრატი შუაში წითელი ჯვრით. ამასთან, თვითონ ბმული ჩვეულებრივად იმუშავებს, ანუ მასზე მაუსის დაწკაპუნებით მითითებულ URL-მისამართზე გადასვლა განხორციელდება (იხ. ნახ. 8).



ნახ. 7.



ნახ. 8.

## მორბენალი სტრიქონი

განვიხილოთ მორბენალი სტრიქონის - ვებ-გვერდის ერთ-ერთი ყველაზე ეფექტური საშუალების შექმნა.

მორბენალი სტრიქონის შესაქმნელი <MARQUEE> წყვილი ტეგია (ინგლ. *marquee* - სტრიქონი), რომელსაც შემდეგი ატრიბუტები აქვს ALIGN, BEHAVIOR, DIRECTION, HEIGHT, WIDTH, HSPACE, VSPACE, LOOP, SCROLLAMOUNT, SCROLLDELAY.

- ატრიბუტი *ALIGN* - საშუალებას იძლევა ტექსტი განვათავსოთ მორბენალი სტრიქონის ზედა (TOP), შუა (MIDDLE) ან ქვედა (BOTTON) ნაწილში.

- ატრიბუტი *BEHAVIOR* - ეკრანზე მორბენალ სტრიქონს მართავს. მისი შესაძლო მნიშვნელობებია:

- SCROLL - ტექსტი მთელ ეკრანს კვეთავს (ეს მნიშვნელობა ჩუმათობის პრინციპით ენიჭება);
- SLIDE - ტექსტი გამოვა ეკრანზე და ჩერდება;
- ALTERNATE - ტექსტი გამოვა ეკრანზე და შემდეგ წინ და უკან მოძრაობს.

- ატრიბუტი *BGCOLOR* - მორბენალ სტრიქონს ფონის ფერს აძლევს;

- ატრიბუტი *DIRECTION* - მორბენალი სტრიქონის ტექსტის მოძრაობის მიმართულებას განსაზღვრავს: LEFT - მარჯვნიდან მარცხნივ (ეს მნიშვნელობა ჩუმათობის პრინციპით ენიჭება), RIGHT - მარცხნიდან მარჯვნივ, DOWN - ზემოდან ქვემოთ და UP - ქვემოდან ზემოთ;

- ატრიბუტი *HEIGHT* - სტრიქონის სიმაღლეს განსაზღვრავს პიქსელებში ან ეკრანის სიმაღლის პროცენტებში (ამ შემთხვევაში რიცხვს ემატება პროცენტის ნიშანი);

- ატრიბუტი *WIDTH* - სტრიქონის სიგრძეს განსაზღვრავს პიქსელებში ან ეკრანის სიგანის პროცენტებში;

- ატრიბუტი *HSPACE* - საშუალებას იძლევა მორბენალი სტრიქონსა და მის გარშემო განთავსებული ტექსტის ან გრაფიკული გამოსახულების არეებს შორის განისაზღვროს მარცხენა და მარჯვენა მინდვრის სიგანე პიქსელებში;

- ატრიბუტი *VSPACE* - საშუალებას იძლევა მორბენალი სტრიქონსა და მის გარშემო განთავსებული ტექსტის ან გრაფიკული გამოსახულების არეებს შორის განისაზღვროს ზედა და ქვედა მინდვრის სიგანე პიქსელებში;

- ატრიბუტი *LOOP* - მორბენალი სტრიქონის ეკრანზე გასვლის რაოდენობა. თუ ეს ატრიბუტი გამოტოვებულია, მაშინ მორბენალი სტრიქონი იმოდრავებს მანამ, ვიდრე მოცემული გვერდი ეკრანზე იქნება გახსნილი;

- ატრიბუტი *SCROLLAMOUNT* - მორბენალი სტრიქონის ეკრანზე გადაადგილების სიჩქარეს მართავს. თუ ეს პარამეტრი 1-ის ტოლია - ეს ყველაზე დაბალ სიჩქარეს ნიშნავს, ხოლო ყველაზე მაღალი სიჩქარეა 2000 (ამ შემთხვევაში მორბენალი სტრიქონის უბრალოდ ჩაიქროლებს);

- ატრიბუტი *SCROLLDELAY* - იგი თავისი ფუნქციებით წინა ატრიბუტს მოგვაგონებს, და მორბენალი სტრიქონის ტექსტის ერთი წერტილიდან მეორეში გადასვლის დროს განსაზღვრავს მილიწამებში.

ამ ბოლო ორი ატრიბუტის კომბინაცია მორბენალი სტრიქონის გადაადგილების ოპტიმალური სიჩქარის განსაზღვრის საშუალებას გვაძლევს.

## მორბენალი სტრიქონის მაგალითები

მორბენალი სტრიქონის მოძრაობის მაგალითი პირველ პროგრამაში, ხოლო მისი შედეგი მის ქვემოთ სურათზეა მოყვანილი.

## პროგრამა 1.

```
<html>
<head>
<title> მორბენალი სტრიქონი</title>
</head>
<body>
<font color=blue>
მორბენალი სტრიქონის შესაქმნელი ტეგია <MARQUEE>
(ინგლ. marquee - სტრიქონი) <br>
<marquee bgcolor=red vspace=5 width=80% height=50 align=middle
behavior=scroll> ტეგები და მისი ატრიბუტები </marquee>
</font><br>
ატრიბუტი ALIGN - საშუალებას იძლევა ტექსტი განვითარდეს
მორბენალი სტრიქონის ზედა (TOP), შუა (MIDDLE) ან ქვედა
(BOTTOM) ნაწილში.<br>
<marquee bgcolor=red vspace=30 width=30% align=bottom
behavior=slide> ტეგები და მისი ატრიბუტები </marquee>
</font><br>
<font color=#ff44ff>
ატრიბუტი BEHAVIOR - ეკრანზე მორბენალ სტრიქონს მართავს.
მისი შესაძლო მნიშვნელობებია:<br>
SCROLL - ტექსტი მთელ ეკრანს კვეთავს (ეს მნიშვნელობა
ჩუმათობის პრინციპით ენიჭება);<br>
SLIDE - ტექსტი გამოვა ეკრანზე და ჩერდება; <br>
ALTERNATE - ტექსტი გამოვა ეკრანზე და შემდეგ წინ და უკან
მოძრაობს. </font><br>
<font color=green>
<marquee bgcolor=#aaaaaa vspace=20 width=50% align=bottom
behavior=alternate> ტეგები და მისი ატრიბუტები </marquee>
</font><br>
```

<font color=#000000>

ატრიბუტი *BGCOLOR* - მორბენალ სტრიქონს ფონის ფერს აძლევს; <br>

<marquee bgcolor=magenta vspace=20 width=90% align=top direction=right height=15%> ტეგები და მისი ატრიბუტები </marquee> </font><br>

<p> ატრიბუტი *DIRECTION* - მორბენალი სტრიქონის ტექსტის მოძრაობის მიმართულებას განსაზღვრავს: <br>

LEFT - LEFT - მარჯვნიდან მარცხნივ (ეს მნიშვნელობა ჩუმათობის პრინციპით ენიჭება), RIGHT - მარცხნიდან მარჯვნივ, DOWN - ზემოდან ქვემოთ და UP - ქვემოდან ზემოთ;<br>

ატრიბუტი *HEIGHT* - სტრიქონის სიმაღლეს განსაზღვრავს პიქსელებში ან ეკრანის სიმაღლის პროცენტებში (ამ შემთხვევაში რიცხვს ემატება პროცენტის ნიშანი). <br>

ატრიბუტი *WIDTH* - სტრიქონის სიგრძეს განსაზღვრავს პიქსელებში ან ეკრანის სიგანის პროცენტებში. <br>

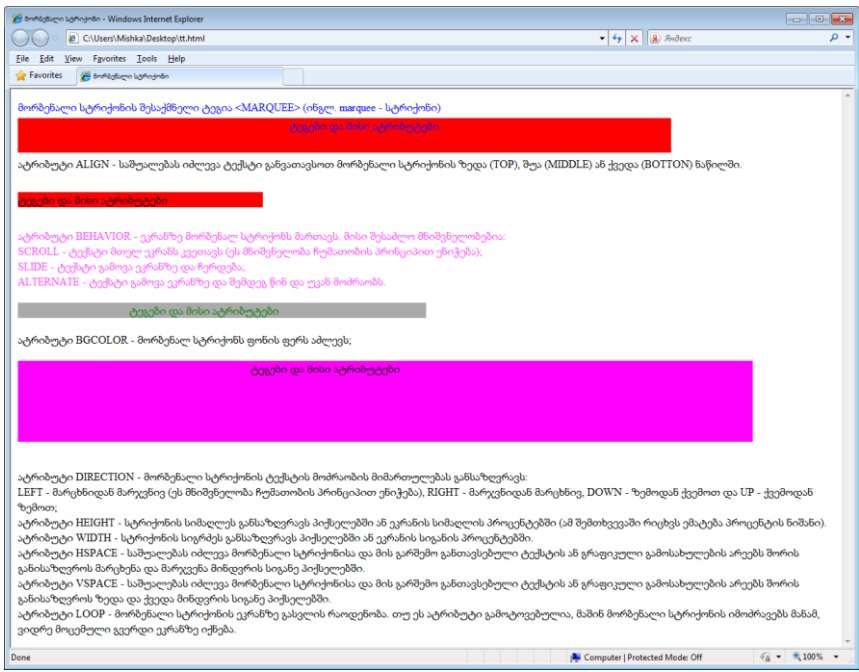
ატრიბუტი *HSPACE* - საშუალებას იძლევა მორბენალი სტრიქონისა და მის გარშემო განთავსებული ტექსტის ან გრაფიკული გამოსახულების არეებს შორის განისაზღვროს მარცხენა და მარჯვენა მინდვრის სიგანე პიქსელებში. <br>

ატრიბუტი *VSPACE* - საშუალებას იძლევა მორბენალი სტრიქონისა და მის გარშემო განთავსებული ტექსტის ან გრაფიკული გამოსახულების არეებს შორის განისაზღვროს ზედა და ქვედა მინდვრის სიგანე პიქსელებში. <br>

ატრიბუტი *LOOP* - მორბენალი სტრიქონის ეკრანზე გასვლის რაოდენობა. თუ ეს ატრიბუტი გამოტოვებულია, მაშინ მორბენალი სტრიქონის იმოდრავებს მანამ, ვიდრე მოცემული გვერდი ეკრანზე იქნება. <br>

</body>

</html>



**პროგრამა 2.** ამ პროგრამაში მორბენალი სტრიქონის ტეგის პარამეტრების ყველა შესაძლო მნიშვნელობებია გამოყენებული.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
<META NAME="Generator" CONTENT="EditPlus">
<META NAME="Author" CONTENT="">
<META NAME="Keywords" CONTENT="">
<META NAME="Description" CONTENT="">
</HEAD>
<BODY>

```

```

<marquee bgcolor="#ffcc00" loop="5">მოგესალმებით(გაივლის 5-
ჯერ)</marquee>
<BR><BR>
<!-- scrolldelay="10" რაც მეტია რიგხვითი მნიშვნელობა, მით
უფრო ანელებს სიჩქარეს-->
<marquee bgcolor="#ffcc00" scrolldelay="200"
behavior="alternate">ნახე რა სწრაფი ვარ??? </marquee><BR><BR>
<!-- scrollamount="10" რაც მეტია რიგხვითი მნიშვნელობა, მეტია
სიჩქარე-->
<marquee bgcolor="#ffcc00" scrollamount="100"
behavior="alternate">ნახე რა სწრაფი ვარ??? </marquee><BR><BR>
<!--behavior=scroll/slide/alternate -->
<marquee bgcolor="#c0c0c0"
behavior="scroll"><H1>მოგესალმებით</H1></marquee><BR><BR>
<marquee bgcolor="#c0c0c0"
behavior="alternate"><H1>მოგესალმებით</H1></marquee>
<BR><BR>
<marquee bgcolor="#c0c0c0"
behavior="slide"><H1>მოგესალმებით</H1></marquee> <BR><BR>
<!-- direction=down /left/right /up -->
<marquee direction="right"
bgcolor="#ffcc00"><H1>მოგესალმებით</H1></marquee> <BR><BR>
<marquee direction="left"
bgcolor="#ffcc00"><H1>მოგესალმებით</H1></marquee> <BR><BR>
<center><TABLE border=1>
<tr height="200" >
<td width="200">
<marquee direction="down"
bgcolor="#ffcc00"><H1>მოგესალმებით</H1></marquee></td>
<td width="200">

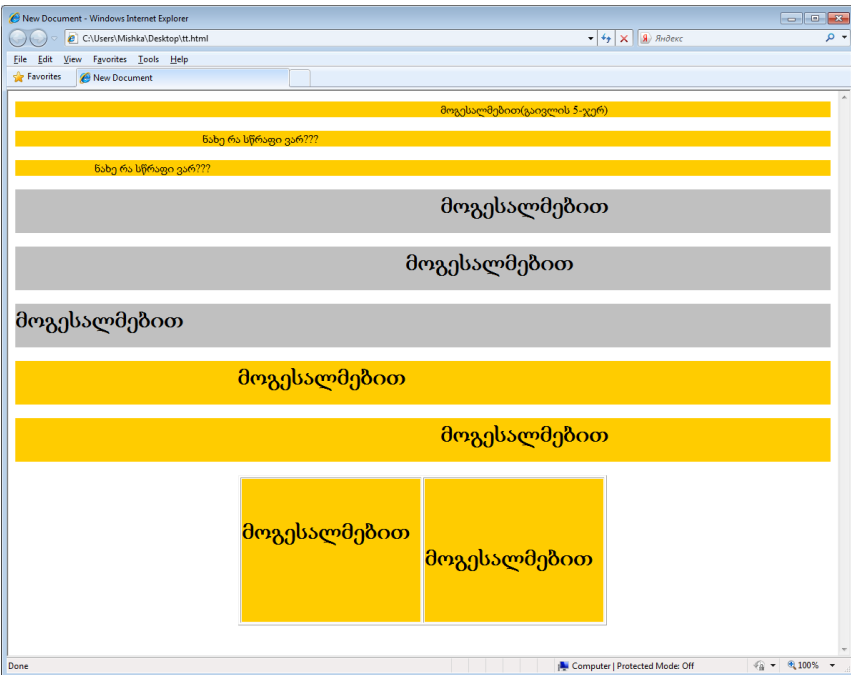
```



```

<marquee direction="up"
bgcolor="#ffcc00"><H1>მოგესალმებით</H1></marquee></td>
</tr>
</TABLE>
</BODY>
</HTML>

```

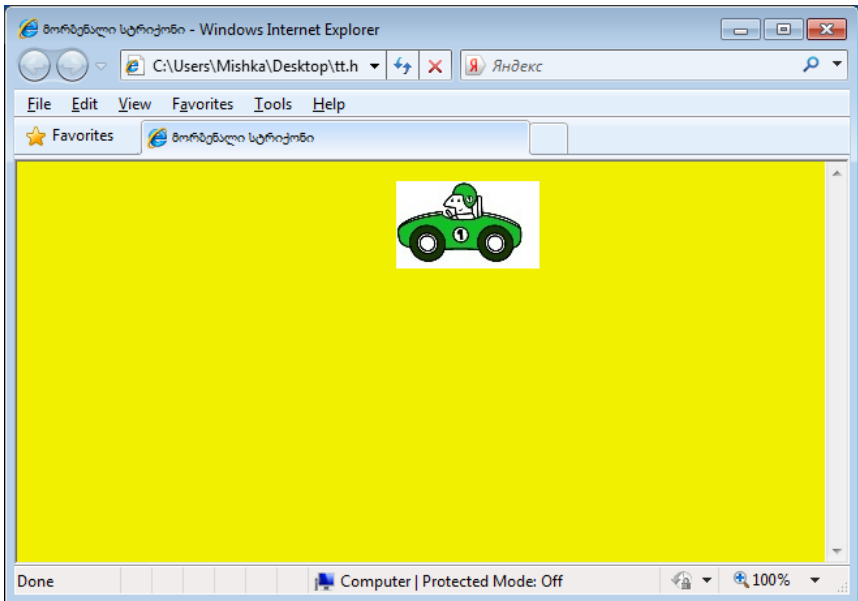


## სურათი მორბენალ სტრიქონად

მორბენალი სტრიქონი არამარტო ტექსტი, არამედ სურათიც შეიძლება იყოს. პროგრამა გრაფიკული მორბენალი სტრიქონით ქვემოთაა მოცემული. მორბენალი სტრიქონისათვის სურათი (მოცემულ შემთხვევაში ეს არის ფაილი car.jpg) და პროგრამის კოდი ერთ საქაღალდეში უნდა იყოს.

### პროგრამა 3.

```
<html>  
<head>  
<title>მორბენალი სტრიქონი</title>  
</head>  
<body bgcolor="f0f000">  
<marquee scrollamount=10 border=0>  
<img src=car.jpg>  
</marquee>  
</body>  
</html>
```



## ფრეიმები

ფრეიმები საშუალებას იძლევა ვებ-გვერდი რამოდენიმე ფანჯარად დავყოთ, ამასთან თითოეულ ფანჯარას (ფრეიმს) შეიძლება საკუთარი გადაფურცვლის ზოლები ჰქონდეს და ყველა ფანჯარა დამოუკიდებლად ფუნქციონირებდეს ან პირიქით, მათ შინაარსს მართავდეს. ფრეიმები ვებ-გვერდის სახეს და მის ფუნქციონირებას მნიშვნელოვნად აუმჯობესებენ.

## ფრეიმების თვისებები

ქვემოთ ფრეიმების ზოგიერთი თვისებაა ჩამოთვლილი:

ყოველ ფრეიმს თავისი URL-მისამართი აქვს, რაც საშუალებას იძლევა მივმართოთ მას და სხვა ფრეიმებისაგან დამოუკიდებლად ჩატვირთოთ იგი.

თითოეულ ფრეიმს აქვს თავისი სახელი (რომელსაც მას NAME ატრიბუტი აძლევს) და რომელიც საშუალებას იძლევა სხვა ფრეიმებიდან მივმართოთ მას.

მომხმარებელს შეუძლია ფრეიმის ზომა პირდაპირ ეკრანზე მაუსის დახმარებით შეცვალოს (თუ ეს სპეციალური პარამეტრით აკრძალული არ არის)

ფრეიმების ზემოთ ჩამოთვლილი თვისებები საშუალებას გვაძლევს სხვადასხვა ინტერფეისული გადაწყვეტა შევქმნათ.

ცალკე ფრეიმში, რომელიც მუდმივად იქნება ეკრანზე, შეიძლება ის სტატიკური ინფორმაცია გათავსდეს, რომელიც, თუ ავტორი საჭიროდ ჩათვლის, მუდმივად მიეწოდოს მომხმარებელს. ეს შეიძლება იყოს ფირმის გრაფიკული ლოგოტიპი, ავტორთა უფლებების შეთანხმება, მმართველი დილაკების ნაკრები, ვებ-სერვერზე განთავსებული ვებ-გვერდების სარჩევი და სხვა.

მომხმარებლის მიერ ვებ-სერვერზე ინტენსიური

მიმართვის შემთხვევაში ეკრანზე მუდმივად შეიძლება ორი ფრეიმი იყოს გამონათებული - ერთი ფრეიმიდან მომხმარებელი სერვერზე თავის შეკითხვებს გაგზავნის, ხოლო მეორეში სერვერიდან მიღებული შესაბამისი დამუშავებული ინფორმაცია გამონათდება.

## ფრეიმების ნაკრები

ფრეიმების გამოყენებული დოკუმენტის ფორმატი გარეგნულად ჩვეულებრივი დოკუმენტის ფორმატს ძალიან გავს, ოღონდ <BODY> ტეგის მაგივრად <FRAMESET> წყვილი ტეგია გამოყენებული:

```
<html>
<head>...</head>
<frameset>...</frameset>
</html>
```

ანუ ვიდრე ვებ-გვერდზე ფრეიმებს განვათავსებთ, <FRAMESET> (ინგლ. *frame set* ფრეიმების ნაკრები) ტეგის დახმარებით ე. წ. ფრეიმების ნაკრებს ქმნიან, რომელიც შეიცავს ინფორმაციას ფრეიმების რაოდენობის შესახებ, მათი ფარდობითი და აბსოლუტური ზომების შესახებ. <FRAMESET> ტეგს რაიმე ინფორმაციული დატვირთვა არა აქვს, ის მხოლოდ ფრეიმებს აღწერს. <FRAMESET> ტეგები შეიძლება ერთმანეთშიც იყოს ჩადგმული. ფრეიმების ნაკრების საერთო სინტაქსია:

```
<frameset cols="value" | rows="value">
<frame src="url1">
<frame ...>...
</frameset>
```

<FRAMESET> საერთო კონტეინერი ყველა იმ ფრეიმს აღწერს, რომლებსაც ეკრანი იყოფა. ეკრანი შეიძლება რამდენიმე

ვერტიკალურ ან ჰორიზონტალურ ფრეიმად დაიყოს. თითოეული ფრეიმის აღწერისათვის ცალკეული <FRAME> ტეგი გამოიყენება. <FRAMESET> და </FRAMESET> ტეგებს შორის შეიძლება განთავსდეს <FRAME> ტეგი, ჩადგმული <FRAMESET> და </FRAMESET> ტეგები, აგრეთვე <NOFRAME> და </NOFRAME> ტეგები.

<FRAMESET> ტეგს შეიძლება COLS, ROWS, BORDER, FRAMEBORDER ატრიბუტები ჰქონდეს, ამასთან ROWS და COLS ურთიერთგამომრიცხავი ატრიბუტებია.

- ატრიბუტი *COLS* - შესაქმნელი ფრეიმების ნაკრების სვეტების რაოდენობასა და ზომებს გვიჩვენებს. ცალკეული ფრეიმების სიგანის მნიშვნელობების ჩამონათვალი ბრჭყალებში იწერება, ერთმანეთისაგან მძიმით გამოიყოფა და მნიშვნელობა მიეთითება პიქსელებში, პროცენტებში (%) ან ერთეულოვანი მასშტაბის კოეფიციენტებით (\*). მაგალითად:

<frameset cols="რიცხვი, რიცხვი%, რიცხვი\*">

- ატრიბუტი *ROWS* - შესაქმნელი ფრეიმების ნაკრების სტრიქონების რაოდენობასა და ზომებს გვიჩვენებს პიქსელებში, პროცენტებში (%) ან ერთეულოვანი მასშტაბის კოეფიციენტებით (\*). მაგალითად:

<frameset rows="რიცხვი, რიცხვი%, რიცხვი\*">

ფრეიმების რაოდენობა სიაში მათი მნიშვნელობათა რაოდენობით განისაზღვრება. ყველა ფრეიმის სიმაღლეთა (ატრიბუტის მნიშვნელობის ნებისმიერი სახით მოცემის შემთხვევაში) საერთო ჯამმა მთელი ფანჯრის სიმაღლე უნდა შეადგინოს. თუ ROWS ატრიბუტი არ არის მითითებული, ეს ნიშნავს, რომ ბრაუზერის ფანჯრის სიმაღლის ერთი ფრეიმი გვაქვს.

- ატრიბუტი *BORDER* - მოცემულ ფრეიმების ნაკრებში

შემაჯალი ყველა ფრეიმისათვის იძლევა ჩარჩოს სიგანეს პიქსელებში. მაგალითად:

```
<frameset border="რიცხვი">
```

- ატრიბუტი *FRAMEBORDER* - BORDER ატრიბუტის მნიშვნელობის მიუხედავად შეიძლება გააუქმოს ფრეიმების ჩარჩოს გამოტანა. ერთის ტოლი მნიშვნელობა ჩარჩოს არსებობაზე, ხოლო 0 - მის არ არსებობაზე მიუთითებს. მაგალითად:

```
<frameset frameborder="1">
```

```
<frameset frameborder="0">
```

ROWS ან COLS ატრიბუტის გამოყენება ფრეიმისათვის მისი სიმაღლისა და სიგანის განსაზღვრის საუკეთესო ფორმას არ წარმოადგენს, ვინაიდან მომხმარებელს მათი ცვლილება მაუსის დახმარებითაც თავისუფლად შეუძლია. თუ ამ ატრიბუტებს მაინც გამოვიყენებთ, მაშინ უნდა ვეცადოთ, რომ შედეგად ბრაუზერის ფანჯრის 100-პროცენტთან შევსება მივიღოთ.

თუ ROWS ან COLS ატრიბუტის მნიშვნელობას პროცენტებში (1-დან 100-მდე) მივუთითებთ, და ამასთან, ფრეიმების აღწერის საერთო ზომა 100-ს აღემატება, მაშინ ყველა ფრეიმის ზომა პროპორციულად შემცირდება, რათა შედეგად 100% იყოს მიღებული. თუ შესაბამისად, ჯამი 100%-ზე ნაკლებია, მაშინ ფრეიმების ზომა პროპორციულად გაიზრდება.

თუ ROWS ან COLS ატრიბუტის ერთ-ერთი მნიშვნელობისათვის გამოყენებულია ერთეულოვანი მასშტაბის კოეფიციენტი, ეს ნიშნავს, რომ ეკრანის მთელი დარჩენილი ნაწილი მოცემულ ფრეიმს მიეკუთვნება. თუ ერთეულოვანი მასშტაბის კოეფიციენტი გამოყენებულია ორი ან მეტი ფრეიმისათვის, მაშინ ეკრანის მთელი დარჩენილი ნაწილი მოცემულ ფრეიმებს შორის გადანაწილდება. თუ ვარსკვლავის წინ რაიმე რიცხვი დგას, მაშინ იგი ბრაუზერს მოცემული ფრეიმის მასშტაბის კოეფიციენტს აძლევს, ანუ მოცემული ფრეიმის ზომა რამდენჯერ მეტი უნდა

იყოს იმ ფრეიმზე, რომლის აღწერაში ვარსკვლავის წინ რიცხვი არ დგას.

განვიხილოთ მაგალითები:

```
cols="3*,*,*"
```

ეს ჩანაწერი გვიჩვენებს, რომ შექმნილი იქნება სამი ფრეიმი, რომელთაგან პირველი ჰორიზონტალურად დაიკავებს ეკრანის 3/5 ნაწილს, ხოლო დანარჩენი ორი თითოეული ეკრანის 1/5 ნაწილს.

```
cols="50%,50%"
```

ასეთი ჩანაწერი გვიჩვენებს, რომ ჩვენების არე ორ ტოლ ნაწილადაა გაყოფილი.

```
cols="20%,80%"
```

ეს კი ორ არათანაბარ ნაწილად (მარცხენა 20 და მარჯვენა 80%).

```
rows="100,20%,"
```

მოცემული ჩანაწერი გვიჩვენებს, რომ ეკრანი გაყოფილია სამ არათანაბარ ნაწილად. ზედა ნაწილისათვის გამოყოფილია 100 პიქსელი, მეორესათვის (შუა ნაწილი) დარჩენილი ნაწილის 20% და დანარჩენი მესამესათვის.

```
cols="*,3*"
```

ეს ჩანაწერი გვიჩვენებს, რომ მარცხენა ფრეიმისათვის სამჯერ ნაკლები არეა გამოყოფილი, ვიდრე მარჯვენასათვის. იგივე ეფექტს გამოიწვევდა შემდეგი ჩანაწერი:

```
cols="25%,75%"
```

ერთ <FRAMESET> ტეგში COLS და ROWS ატრიბუტების ერთდროულმა გამოყენებამ შეიძლება გაუთვალისწინებელ შედეგამდე მიგვიყვანოს.

ზოგიერთი ბრაუზერი ფრეიმების გამოყენებას მხარს არ უჭერს. იმისათვის, რომ ასეთი ბრაუზერების მომხმარებელმა ფრეიმებიანი ვებ-გვერდის ნახვა მოახერხონ, <NOFRAMES> წყვილი ტეგის გამოყენებაა საჭირო. ეს ტეგი <FRAMESET>

კონტეინერის შიგნით თავსდება და ის რაც <NOFRAMES> და </NOFRAMES> ტეგებს შორისაა განთავსებული იმ ბრაუზერების მიერ, რომლებიც ფრეიმებს მხარს უჭერენ, იქნება იგნორირებული. ანუ ვებ-გვერდის დამპროექტებელმა უნდა დაწეროს კოდი, რომელიც ფრეიმების შინაარსის სხვა საშუალებებით დუბლირებას ახდენს, და იგი <NOFRAMES> და </NOFRAMES> ტეგებს შორის განათავსოს. ამ შემთხვევაში ვებ-გვერდის ნახვას ყველა მომხმარებელი შეძლებს.

## ფრეიმის ფრეიმების ნაკრებით დოკუმენტში ჩასმა

როგორც უკვე აღვნიშნეთ, დოკუმენტში ფრეიმების ჩასადგმელად <FRAME> ტეგი NAME, SRC, MARGINWIDTH, MARGINHEIGHT, SCROLLING, NORESIZE, FRAMEBORDER ატრიბუტებით გამოიყენება.

- ატრიბუტი *NAME* - პროგრამას ფრეიმის სახელს აწვდის. ამ სახელით ფრეიმზე მიმართვა ხდება (ეს TARGET ატრიბუტის საშუალებით კეთდება), მაგალითად, მასში ახალი HTML-ფაილი გვინდა ჩავტვირთოთ. მისი სინტაქსია:ⓘ

name="სახელი"

- ატრიბუტი *SRC* - მოცემულ ფრეიმში ჩასატვირთი დოკუმენტის მისამართს აწვდის. თუ ეს ატრიბუტი არ არის მითითებული, მაშინ ცარიელი ფრეიმი გამოვა. მაგალითად:

<frame src="10.htm">

- ატრიბუტი *MARGINWIDTH* - ფრეიმებს შორის ვერტიკალური გამყოფი ზოლის სიდიდეს უჩვენებს. მისი მნიშვნელობა პიქსელებში მიეთითება და არ შეიძლება ერთზე ნაკლები იყოს:

marginwidth="value"

- ატრიბუტი *MARGINHEIGHT* - ფრეიმებს შორის ჰორიზონტალური გამყოფი ზოლის სიდიდეს უჩვენებს. მისი მნიშვნელობა



პიქსელებში მიეთითება და არ შეიძლება ერთზე ნაკლები იყოს:

marginheight="value"

- ატრიბუტი *SCROLLING* - ფრეიმის გადაფურცვლის ზოლის გამოტანას მართავს:

scrolling="yes|no|auto"

yes - მნიშვნელობა მიუთითებს, რომ ფრეიმს გადაფურცვლის ზოლი ნებისმიერ მომენტში ექნება, no - ფრეიმს გადაფურცვლის ზოლი ნებისმიერ მომენტში არ ექნება, ხოლო auto - ჩუმათობის პრინციპით მიეთითება და უჩვენებს, რომ ფრეიმს გადაფურცვლის ზოლი მხოლოდ აუცილებლობის შემთხვევაში ექნება.

- ატრიბუტი *NORESIZE* - საშუალებას გვაძლევს შევქმნათ ფრეიმი მისი ზომის შეცვლის შესაძლებლობის გარეშე. ჩუმათობის პრინციპით ფრეიმის ზომები მაუსის საშუალებით შეიძლება უპრობლემოდ შეიცვალოს. თუ ეს ატრიბუტი ერთ ფრეიმს მაინც აქვს მითითებული, მაშინ მის მოსაზღვრე სხვა ფრეიმებსაც ზომებს ვერ შევუცვლით.

- ატრიბუტი *FRAMEBORDER* - უჩვენებს აქვს თუ არა ფრეიმებს ჩარჩო. ერთის ტოლი მნიშვნელობა ჩარჩოს არსებობაზე, ხოლო 0 - მის არ არსებობაზე მიუთითებს.

## ფრეიმის ფრეიმების ნაკრების გარეშე დოკუმენტში ჩასმა

ვებ-გვერდის სხვა დოკუმენტში ვიზუალიზაციისათვის, აუცილებელი არ არის მასში ფრეიმების ნაკრები წინასწარ შეიქმნას. ფრეიმი შეიძლება არა მარტო ფრეიმების ნაკრებიან დოკუმენტში, არამედ ჩვეულებრივ დოკუმენტშიც ჩაიდგას. ამისათვის, გამოიყენება <IFRAME> ტეგი, რომელიც <FRAME> ტეგისაგან განსხვავებით <FRAMESET> და </FRAMESET> ტეგებს შორის კი არ, არამედ <BODY> და </BODY> ტეგებს შორის იდგმება. <IFRAME> ტევს შემდეგი ატრიბუტები აქვს: SRC, FRAMEBOARD,

MARGINHEIGHT, MARGINWIDTH, NAME, ALIGN, SCROLLING, WIDTH, HEIGHT .

- *ატრიბუტი SRC* - მოცემულ ფრეიმში ჩასატვირთი დოკუმენტის მისამართს აწვდის. მაგალითად:

```
<iframe src="url">
```

- *ატრიბუტი FRAMEBORDER* - უჩვენებს აქვს თუ არა ფრეიმებს ჩარჩო. ერთის ტოლი მნიშვნელობა ჩარჩოს არსებობაზე, ხოლო 0 - მის არ არსებობაზე მიუთითებს.

- *ატრიბუტი MARGINHEIGHT* - ფრეიმებს შორის ზედა და ქვედა გამყოფი ზოლის სიდიდეს უჩვენებს პიქსელებში. მაგალითად:

```
<iframe marginheight ="რიცხვი">
```

- *ატრიბუტი MARGINWIDTH* - ფრეიმებს შორის მარცხენა და მარჯვენა გამყოფი ზოლის სიდიდეს უჩვენებს პიქსელებში. მაგალითად:

```
<iframe marginwidth="რიცხვი">
```

- *ატრიბუტი NAME* - პროგრამას ფრეიმის სახელს აწვდის. ამ სახელით ფრეიმზე მიმართვა ბმულით ხდება.

- *ატრიბუტი ALIGN* - ფრეიმის პოზიციის შერჩევას ახდენს ტექსტის მიმართ და შემდეგ მნიშვნელობებს ღებულობს: LEFT (მარცხნივ), RIGHT (მარჯვნივ), MIDDLE (შუაში), TOP (ზემოთ) და BOTTOM (ქვემოთ), მაგალითად:

```
<iframe align="left">
```

- *ატრიბუტი SCROLLING* - ფრეიმის გადაფურცვლის ზოლის გამოტანას მართავს და შეიძლება მიიღოს მნიშვნელობები yes (გადაფურცვლის ზოლი აქვს), no (გადაფურცვლის ზოლი არა აქვს) და auto (გადაფურცვლის ზოლი მხოლოდ აუცილებლობის შემთხვევაში ექნება).

- *ატრიბუტი WIDTH* - ფრეიმის სიგანეს განსაზღვრავს პიქსელებში. მაგალითად:

<iframe width ="რიცხვი">

- ატრიბუტი *HEIGHT* - ფრეიმის სიმაღლეს განსაზღვრავს პიქსელებში. მაგალითად:

<iframe height="რიცხვი">

## ფრეიმების ურთიერთქმედება

იმისათვის, რომ ერთ ფრეიმში მაუსით ბმულზე დავაწკაპუნოთ და შესაბამისი ინფორმაცია სხვა ფრეიმში მივიღოთ ფრეიმების ურთიერთქმედების ორგანიზება უნდა მოხდეს, რასაც ქვემოთ განვიხილავთ.

ყოველ ფრეიმს შეიძლება საკუთარი სახელი ჰქონდეს, რომელიც მას შექმნის მომენტში NAME ატრიბუტით ენიჭება. ასევე არსებობს სპეციალური ატრიბუტი TARGET, რომელიც მიზნობრივი ფრეიმის მისამართს განსაზღვრავს:

target="სახელი"

მოცემული ატრიბუტი შეიძლება სხვადასხვა ტეგის შიგნით მივუთითოთ, მაგალითად, ბმულებში:

<a href="100.htm" target="frame1"> frame1 ფრეიმზე გადასვლა </a>

თუ TARGET ატრიბუტს <BASE> ტეგში განვათავსებთ, მაშინ დოკუმენტის მიმდები ფრეიმის ბმულის აღწერაში ყოველი გამოსაძახებელი ბმულის მითითება არ არის აუცილებელი. ეს განსაკუთრებით მაშინ არის მოსახერხებელი, როდესაც ერთ ფრეიმში, მაგალითად, არის მენიუ განთავსებული, ხოლო მეორეში ამ მენიუს არჩეული პუნქტის შესაბამისი ინფორმაცია გამოდის:

<html>

<head>

<base target="frame1">

</head>

<body>

```
<a href="url1"> პირველი ნაწილი </a>|<a href="url2"> მეორე  
ნაწილი </a>  
</body>  
</html>
```

TARGET ატრიბუტი შეიძლება ფორმის აღწერაში ჩავსვათ:

```
<form action = "url" target="frame2">
```

ამ შემთხვევაში, ფორმის გადაცემული პარამეტრების დამუშავების შემდეგ, საბოლოო დოკუმენტი მითითებულ ფრეიმში გამოჩნდება.

TARGET ატრიბუტში ფრეიმის სახელი უნდა დაიწყოს ლათინური ასოთი ან ციფრით. ასევე, უნდა ვიცოდეთ, რომ არსებობს დარეზერვირებული სახელები, რომლებიც ქვედა ხაზით იწყება:

\_BLANK - ბმულით მიღებული დოკუმენტი ბრაუზერის ახალ ფანჯარაში გამოისახება;

\_SELF - ბმულით მიღებული დოკუმენტი იმავე ფრეიმში გამოისახება, რომელშიც ბმულია. ეს სახელი ადრე <BASE> ტეგში მითითებული დანიშნულების ფანჯრის შესაცვლელად გამოიყენება;

\_PARENT - ბმულით მიღებული დოკუმენტი, <FRAMESET> ტეგში მითითებული ატრიბუტის მიუხედავად, მშობლიურ ფანჯარაში გამოისახება. თუ მშობლიური ფანჯარა არ არის, მაშინ ეს სახელი \_SELF სახელის ანალოგიური იმოქმედებს;

\_TOP - ბმულით მიღებული დოკუმენტი, ფრეიმებისაგან დამოუკიდებლად ნებისმიერ ფანჯარას დაიკავებს. ამ სახელის გამოყენება ჩადგმული ფრეიმების არსებობის შემთხვევაშია მოხერხებული.

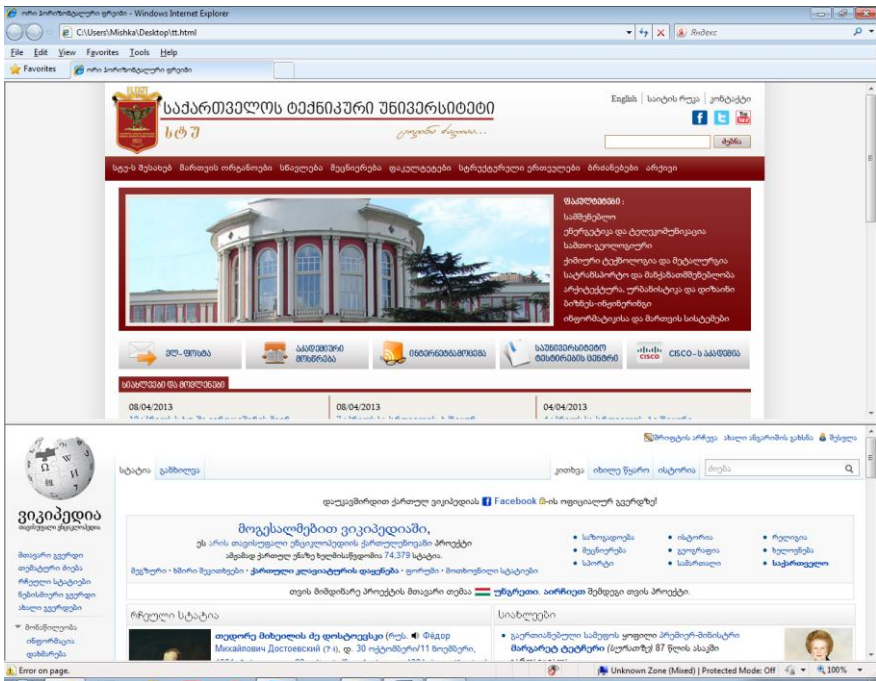
ნებისმიერი სხვა სახელი, რომელიც ქვედა ხაზით დაიწყება, ბრაუზერის მიერ იგნორირებული იქნება.

# ფრეიმების გამოყენების მაგალითები

პირველ მაგალითში მოცემულია ვებ-გვერდი, რომელიც ორი ჰორიზონტალური (გამოიყენება ROWS ატრიბუტი) ფრეიმისაგან შედგება. ფრეიმების ზომების ფარდობა ტოლია ათი შვიდან.

## პროგრამა 1.

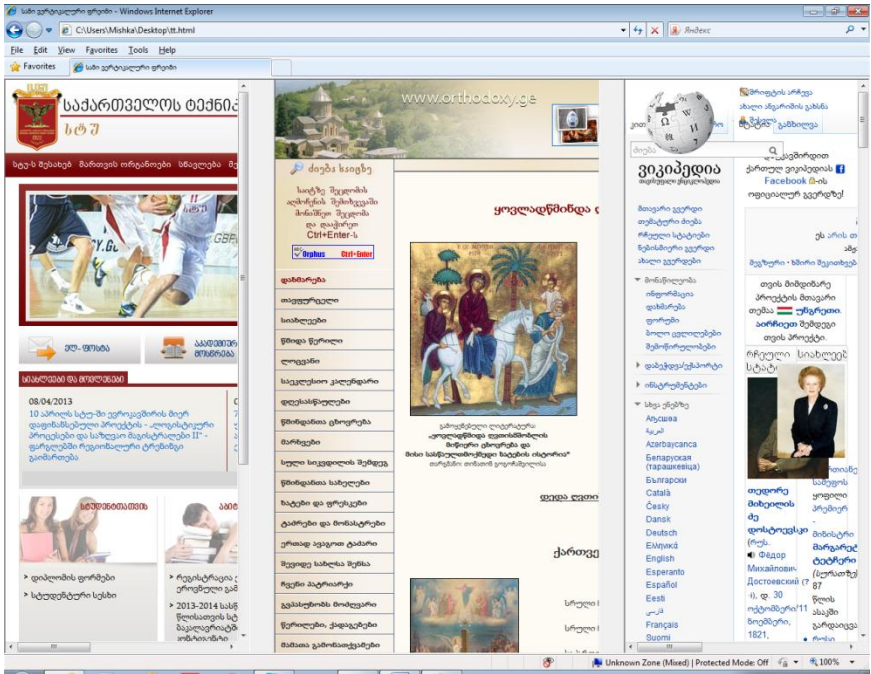
```
<html>
<head><title> ორი ჰორიზონტალური ფრეიმი</title>
<frameset frameborder=1 border="5" rows="400,280">
<frame src="http://www.gtu.ge">
<frame src="http://ka.wikipedia.org">
</frameset>
</html>
```



შემდეგ მაგალითში მოყვანილია ვებ-გვერდი სამი ვერტიკალური (გამოიყენება COLS ატრიბუტი) ფრეიმით. მე-2 პროგრამაში მოცემულ მარცხენა და მარჯვენა ფრეიმებს მთელი ეკრანის 30%, ხოლო შუა ფრეიმს 40% უჭირავს. ამასთან, მათ შორის გამავალი საზღვარი საკმაოდ განიერია, რადგანაც BORDER=35 და მეორე ფრეიმს არა აქვს გადაფურცვლის ზოლი ვინაიდან მითითებულია SCROLLING=no.

## პროგრამა 2.

```
<html>
<head>
<title>სამი ვერტიკალური ფრეიმი</title>
</head>
<frameset border="35" cols="3*,4*,3*">
<frame src="http://www.gtu.ge">
<frame scrolling="no" src="http://www.orthodoxy.ge">
<frame src="http://ka.wikipedia.org">
</frameset>
</html>
```



მე-3 პროგრამაში განხილულია მაგალითი, სადაც კონსტრუქცია ორი სვეტისაგან შედგება - 45% და 55%. მარცხენა ნაწილში მდგომ ვებ-გვერდს მთელი ფანჯარა უჭირავს, ხოლო მარჯვენა ნაწილი სამ ნაწილადაა გაყოფილი, რომელთა სიმაღლე შესაბამისად 25, 35 და 40%-ია.

**პროგრამა 3.**

```

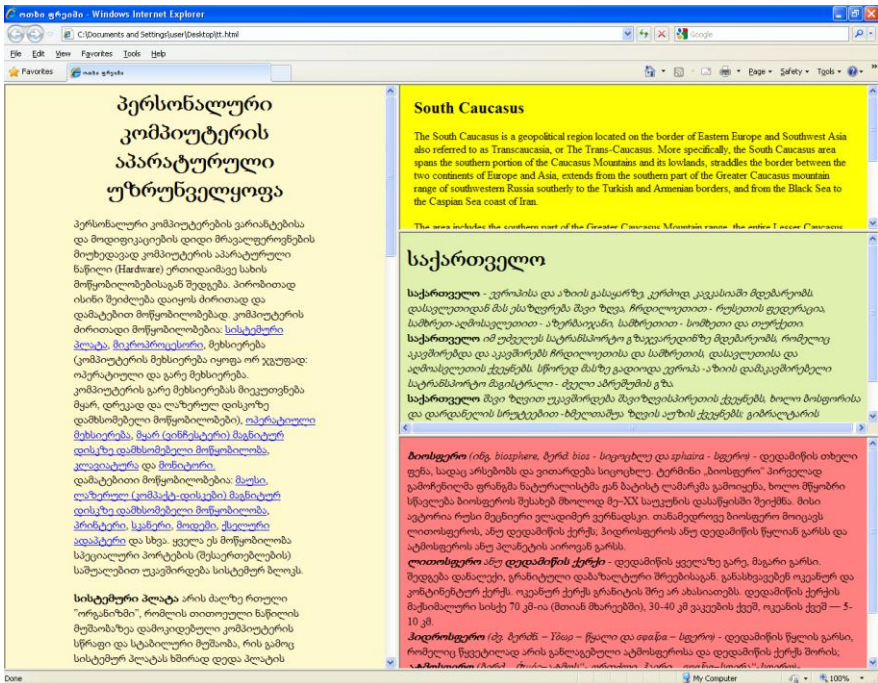
<html>
<head>
<title>ოთხი ფრეიმი</title></head>
<frameset cols="*,55%">
<frame src="1.htm">
<frameset rows="25%,35%,40%">
<frame src="2.htm">

```

```

<frame src="3.htm">
<frame src="4.htm">
</frameset>
</frameset>
</html>

```



ქვემოთ მოყვანილ მაგალითში ნაჩვენებია ფანჯარა რომელიც ოთხ ტოლ ნაწილადაა გაყოფილი.

**პროგრამა 4.**

```

<html>
<head>
<title>ოთხი ტოლი ფანჯარა</title>
</head>
<frameset cols="50%,50%">

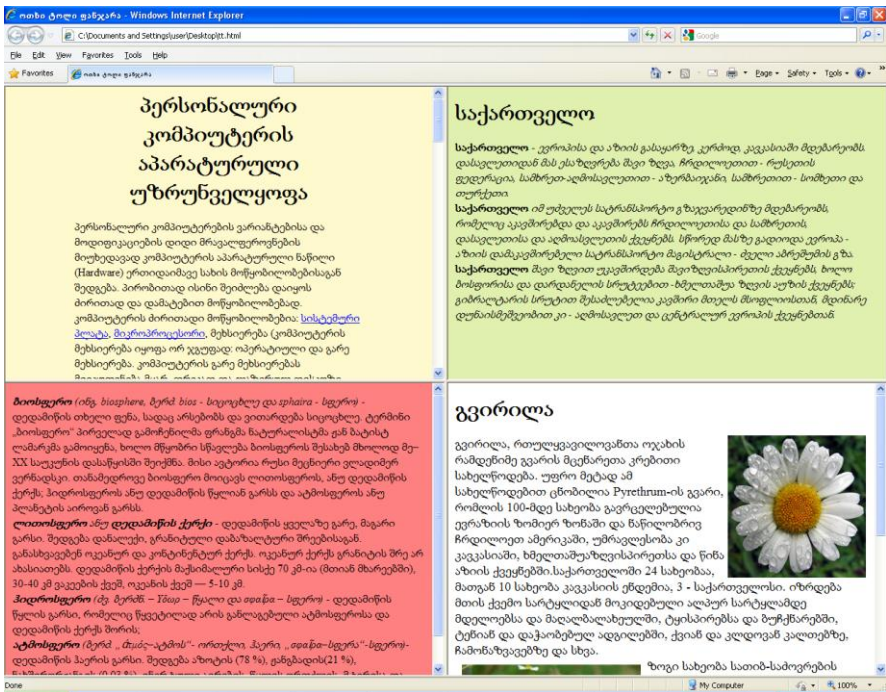
```



```

<frameset rows="50%,50%">
<frame src="1.htm">
<frame src="4.htm">
</frameset>
<frameset rows="50%,50%">
<frame src="3.htm">
<frame src="6.htm">
</frameset>
</frameset>
</html>

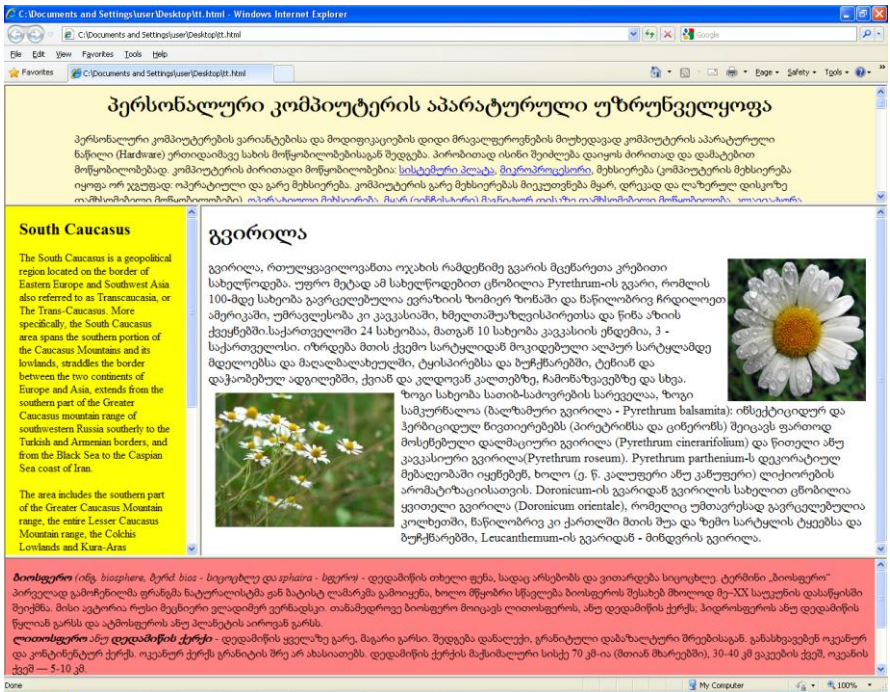
```



მე-5 პროგრამაში განხილულია მაგალითი, სადაც სხვადასხვა ზომის ფრეიმი გამოდის.

## პროგრამა 5.

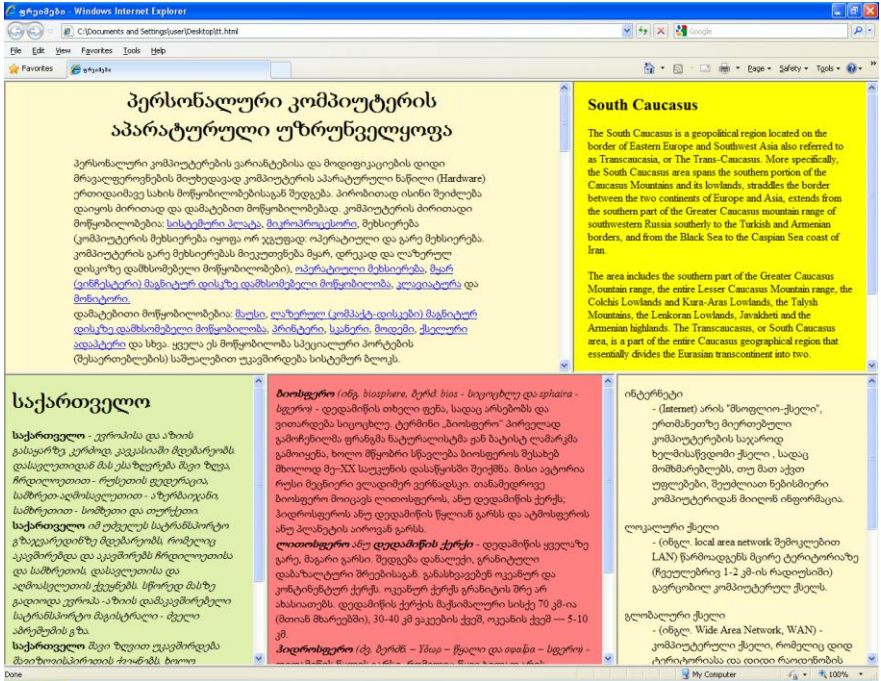
```
<html>  
<frameset rows="20%,60%,20%">  
<frame src="1.htm">  
<frameset cols="22%,78%">  
<frame src="2.htm">  
<frame src="3.htm">  
</frameset>  
<frame src="4.htm">  
</frameset>  
</html>
```



მე-6 პროგრამაში <NOFRAMES> ტეგის გამოყენების მაგალითია განხილული. გამოტანილი ფანჯრიდან ჩანს, რომ მოცემული ბრაუზერი ფრეიმებს მხარს უჭერს, ვინაიდან მან <NOFRAMES> და </NOFRAMES> ტეგებს შორის მოთავსებულ ინფორმაციას იგნორირება გაუკეთა.

### **პროგრამა 6.**

```
<html>
<head>
<title> ფრეიმები </title>
</head>
<frameset rows="*,*">
<noframes>
<h1>თქვენი ბრაუზერი ფრეიმებს მხარს არ უჭერს!</h1>
</noframes>
<frameset cols="65%,35%">
<frame src="1.htm">
<frame src="2.htm">
</frameset>
<frameset cols="*,40%,*">
<frame src="3.htm">
<frame src="4.htm">
<frame src="5.htm">
</frameset>
</frameset>
</html>
```



ფრეიმების გამოყენების კიდევ ერთი მაგალითია განხილული მე-7 პროგრამაში, სადაც ერთ-ერთ ფრეიმში მარტო სურათია გამოტანილი.

**პროგრამა 7.**

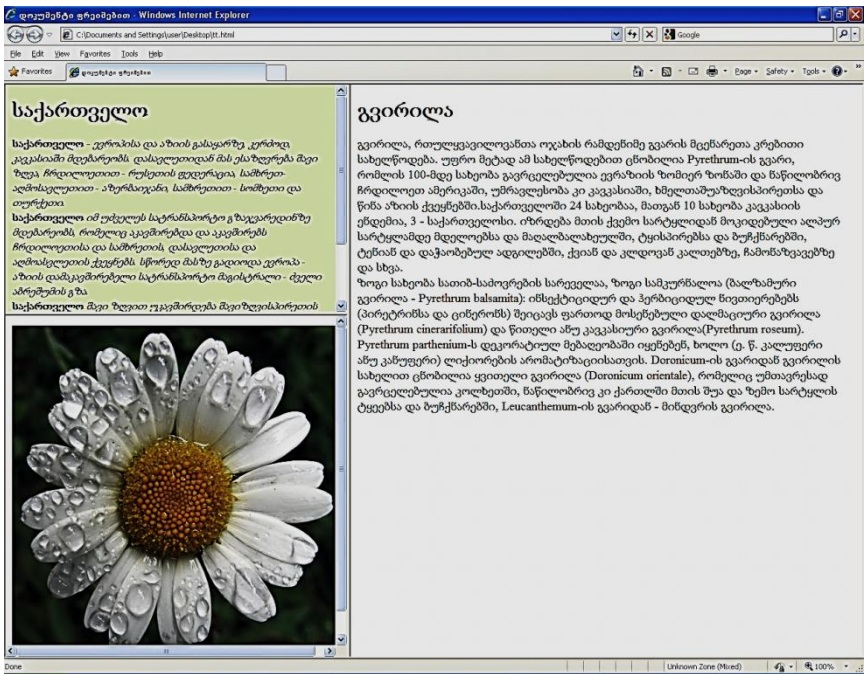
```

<html>
<head>
<title>დოკუმენტი ფრეიმებით</title></head>
<frameset cols="20%, 80%">
<frameset rows="100, 200">
<frame src="2.htm">
<frame src="გვირილა.jpg">
</frameset>
<frame src="6.htm">

```

</frameset>

</html>



The screenshot shows a web browser window with two columns of text. The left column, titled "საქართველო" (Georgia), describes the country's location and history. Below the text is a photograph of a white daisy flower with water droplets. The right column, titled "გვირილა" (Chrysanthemum), describes the plant's medicinal properties and varieties.

## ფრეიმების ურთიერთქმედების მაგალითები

ქვემოთ განვიხილოთ ფრეიმების ურთიერთქმედების მაგალითები. ასეთი ურთიერთდამოკიდებული სტრუქტურა შეიძლება სახელმძღვანელოს, ვირტუალური მუზეუმის და უბრალოდ ფირმის საიტების ვებ-გვერდებს ჰქონდეს.

თავდაპირველად განვიხილოთ პროგრამის მაგალითი, სადაც არ არის ფრეიმები, მაგრამ სხვა გვერდზე მიმართვა ხდება.

### პროგრამა 8.

<html>

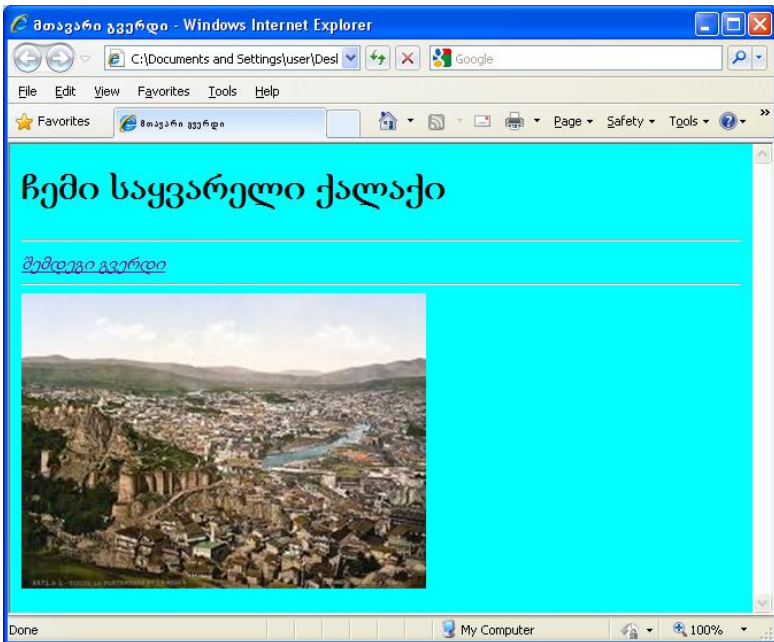
<head>

```

<title>მთავარი გვერდი</title>
</head>
<body bgcolor=aqua>
<h1>ჩემი საყვარელი ქალაქი </h1>
<hr>
<a href="ფრეიმი.htm" target="ფრეიმი"><i> შემდეგი
გვერდი</i></a>
<hr>

</body>
</html>

```



მოცემული პროგრამა მთავარი პროგრამაა და მისგან მიმართვა ისეთ ვებ-გვერდზე ხდება, რომელშიც ფრეიმების ურთიერთქმედებაა განხილული.

## პროგრამა 9.

```
<html>
<head>
<title>ფრეიმების ურთიერთქმედება</title>
</head>
<frameset frameborder=1 framespacing=5 border=5 cols="360,700">
<frame name="სია" src="სია.htm">
<frame name="ხედები" src="თბილისის ხედები.htm">
</frameset>
</html>
```

მოცემული პროგრამა ორ ფრეიმში გახსნის ორ გვერდს. მარცხენა ფრეიმში მოცემული იქნება ჰიპერბმულების სია (პროგრამა 10), ხოლო მარჯვენა მხარეს თბილისის ხედების გვერდი (პროგრამა 11) გაიხსნება.

## პროგრამა 10.

```
<html>
<head>
<title>სია</title>
<base target="ხედები">
</head>
<body>
<body text=black link=green bgcolor=f0f0a0>
<h3>თბილისი </h3></center>
<ul>
<li><a href="ძველი თბილისი.htm"><i>ძველი
თბილისი</i></a></li>
<li><a href="მთაწმინდა.htm"><i>მთაწმინდა</i></a></li>
<li><a href="რუსთაველის გამზირი.htm"><i>რუსთაველის
გამზირი</i></a></li>
<li><a href="მეტეხი.htm"><i>მეტეხი</i></a></li>
```

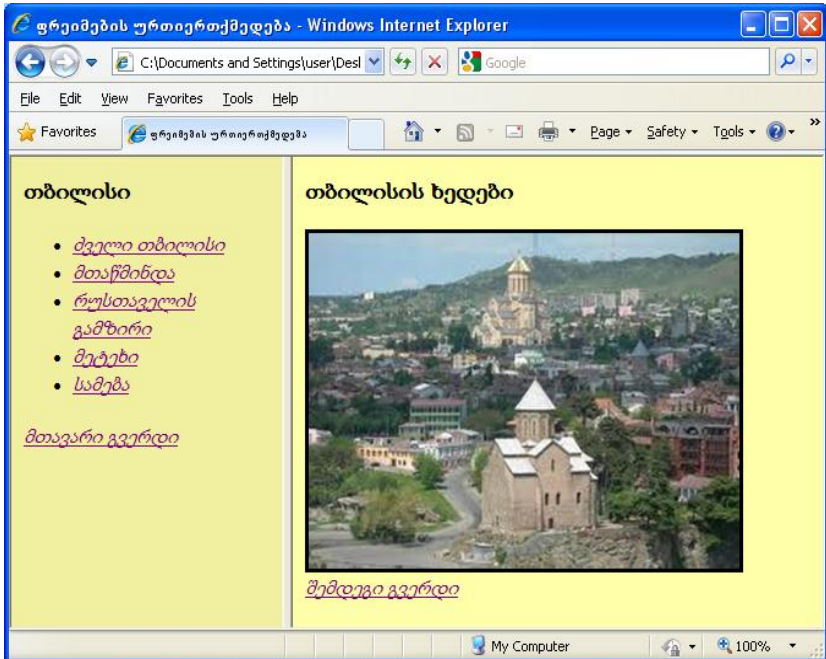
<li><a href="სამება.htm"><i>სამება</i></a></li>

</ul>

<a href="მთავარი გვერდი.htm"><i>მთავარი გვერდი</i></a>

</body>

</html>



## პროგრამა 11.

<html>

<head>

<title>თბილისის ხედები</title>

</head>

<body text=black link=green bgcolor=ffffaa>

<h3>თბილისის ხედები</h3>

<img height=117 width=352 align=center border= 3

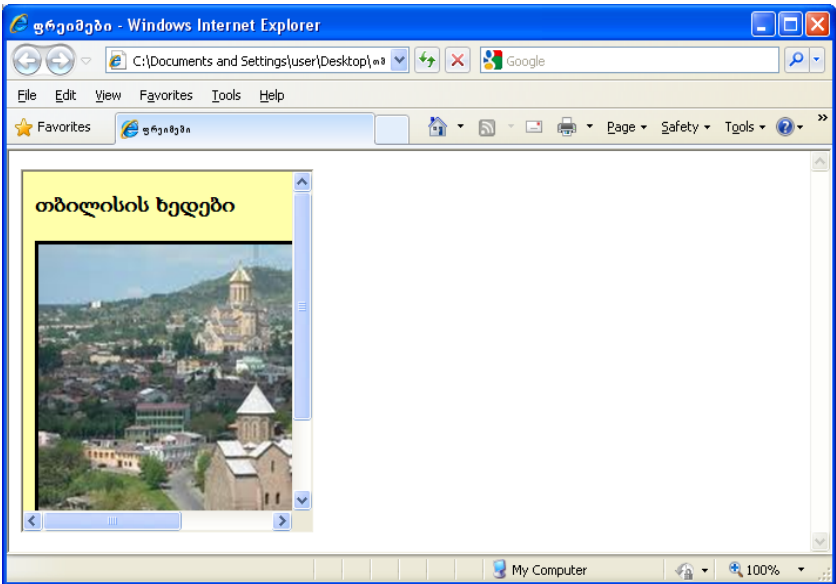




როგორც უკვე აღვნიშნეთ, <IFRAME> ტეგი საშუალებას იძლევა დოკუმენტში, რომელშიც ფრეიმების ნაკრები არ არის შექმნილი, ფრეიმის ჩადგმა მოხდეს.

**პროგრამა 12.** <IFRAME> ტეგის გამოყენება

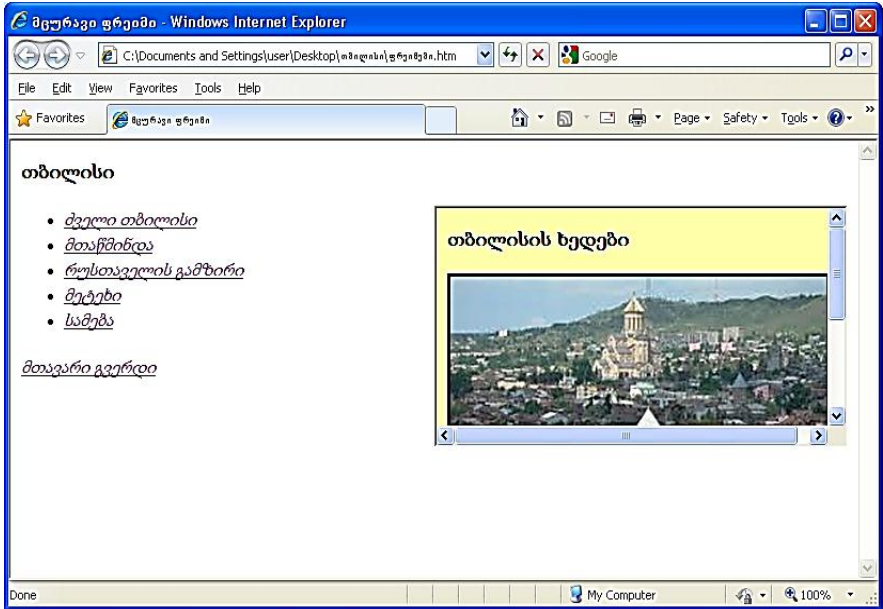
```
<html>
<head>
<title>ფრეიმები</title>
</head>
<iframe src="თბილისის ხედები.htm" width="250" height="300"
scrolling="auto" frameborder="1">
</iframe>
</html>
```



ფრეიმებს, რომლებიც დოკუმენტში <IFRAME> ტეგის გამოყენებით არის ჩასმული მცურავი ფრეიმი ეწოდება. ქვემოთ მცურავი ფრეიმის კიდეც ერთი მაგალითია მოყვანილი.

### პროგრამა 13.

```
<html>
<head>
<title>მცურავი ფრეიმი</title>
</head>
<body>
<h3>თბილისი </h3></center>
<iframe src="თბილისის ხედები.htm" align=right scrolling=auto
frameborder=1 width=50% height=60%>
</iframe>
<ul>
<li><a href="ძველი თბილისი.htm" target="ძველი
თბილისი"><i>ძველი თბილისი</i></a></li>
<li><a href="მთაწმინდა.htm"
target="მთაწმინდა"><i>მთაწმინდა</i></a></li>
<li><a href="რუსთაველის გამზირი.htm" target="რუსთაველის
გამზირი"><i>რუსთაველის გამზირი</i></a></li>
<li><a href="მეტეხი.htm" target="მეტეხი"><i>მეტეხი</i></a></li>
<li><a href="სამება.htm" target="სამება"><i>სამება</i></a></li>
</ul>
<a href="ფრეიმი.htm" target="მთავარი გვერდი" ><i>მთავარი
გვერდი</i></a>
</body>
</html>
```



**პროგრამა 14.** მცურავი ფრეიმის შექმნის მაგალითი, სადაც ფრეიმში სურათი მორბენალი სტრიქონის სახით არის წარმოდგენილი.

```
<html>
```

```
<head>
```

```
<title>მორბენალი სტრიქონი მცურავ ფრეიმში</title></head>
```

```
<body background="#806095"><font size=4>
```

```
<p>რა არის მცურავი ფრეიმი:<br>
```

```
აი ის: <iframe align=left hspace=10 vspace=10 src="მორბენალი  
სტრიქონი.htm" width=400 scrolling=no height=150></iframe>
```

```
<p>მცურავი ფრეიმი არსად არ ცურავს. <br>
```

ის, როგორც ყველა ფრეიმი, ისე მუშაობს, მაგრამ ამავე დროს იგი პირდაპირ ვებ-გვერდზე iframe წყვილი ტეგის გამოყენებით იქმნება.<br>

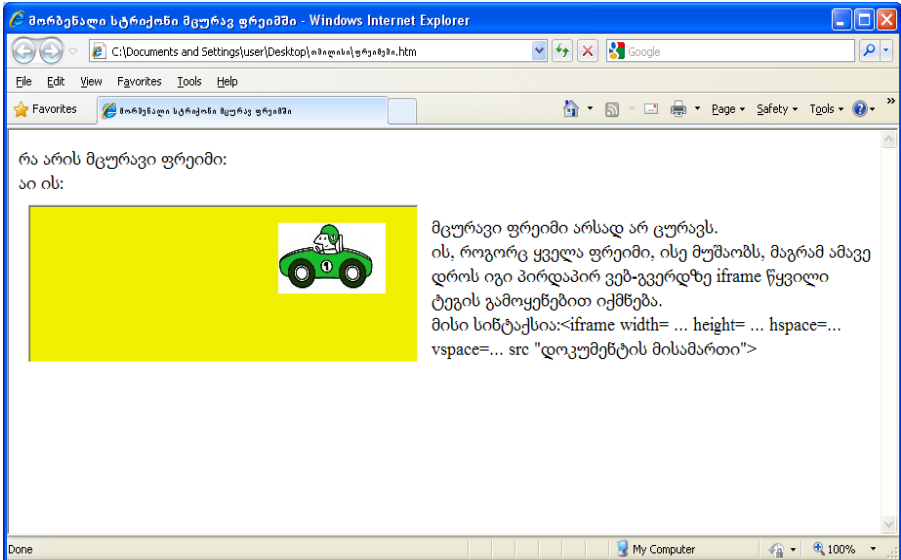
მისი სინტაქსია:&lt;iframe width= ... height= ... hspace=...

vspace=... src="დოკუმენტის მისამართი"&gt;<br></p>

</font>

</body>

</html>



## მულტიმედია

მულტიმედია - ეს პირველ რიგში აუდიო და ვიდეო მასალაა. მულტიმედია ვებ-დოკუმენტის დანართში - ეს ვებ-გვერდებზე განთავსებული აუდიო- და ვიდეო-რგოლებია.

უკანასკნელ პერიოდამდე აუდიო- და ვიდეო-რგოლების ვებ-გვერდებზე განთავსება მხოლოდ საკმაოდ დიდი HTML-კოდის ჩაწერით ან დამატებითი პროგრამით იყო შესაძლებელი. მაგრამ, ამჟამად, HTML-5 გამოსვლის შემდეგ ეს პროცესი გამარტივდა და ამისთვის ჩვენთვის ნაცნობი <IMG> 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 და აღწერილი ვებ-დოკუმენტში გამოყენებული და ბრაუზერების მიერ მხარდაჭერილი მულტიმედიური ფაილების ფორმატები.

- WAV - მულტიმედიურ ფორმატებს შორის ყველაზე ძველი. გასული საუკუნის 90-იანი წლების დასაწყისში Microsoft-ის მიერ აუდიო მონაცემების შესანახად იყო დამუშავებული და ამჟამად გამოიყენება. ასეთი ფორმატის ფაილებს wav გაფართოება აქვთ;

- *OGG* - შედარებით ახალი ფორმატი. იგი Xiph.org არაკომერციული ორგანიზაციის მიერ აუდიო- და ვიდეო-ინფორმაციის შესანახად დაახლოებით თორმეტი წლის წინ დამუშავდა. ამ ტიპის ფაილებს ogg (უნივერსალური ფორმატი), oga (აუდიო-ფაილების) და ogv (ვიდეო-ფაილების) გაფართოება აქვთ. ბოლო ორი გაფართოება ძალზე იშვიათად გვხვდება;
- *MP4* - ესეც შედარებით ახალი ფორმატია. ეს ფორმატი Motion Picture Expert Group (მოძრავი გამოსახულების საკითხების ექსპერტთა ჯგუფი) ორგანიზაციის მიერ 1998 წელს აუდიო- და ვიდეო-მონაცემთა შენახვის მიზნით იყო დამუშავებული. ამ ფორმატის ფაილებს mp4 გაფართოება აქვთ;
- *QuickTime* - ეს ფორმატი ძალიან ძველია და Apple ფირმის მიერ 1989 წელს აუდიო- და ვიდეო-მონაცემების შესანახად იყო დამუშავებული. ასეთი ფორმატის ფაილებს mov გაფართოება აქვთ;
- *PCM* (Pulse-Coded Modulation - იმპულსურ-კოდური მოდულაცია) - კოდირების ყველაზე მარტივი და ყველაზე ძველი ფორმატი. ის შემჭიდროებასაც კი არ უწევს მხარდაჭერას. აუდიო-მონაცემების კოდირებისათვის გამოიყენება.
- *Vorbis* - კოდირების თანამედროვე ფორმატი. დამუშავებულია 2002 წელს Xiph.org ორგანიზაციის მიერ. გამოიყენება აუდიო-მონაცემების კოდირებისათვის.
- *AAC* (Advanced Audio Coding - აუდიოს განვითარებული კოდირება) - კოდირების არც ისე ახალი ფორმატი. დამუშავებული იყო 1997 წელს Motion Picture Expert Group ორგანიზაციის მიერ. გამოიყენება აუდიო-მონაცემების კოდირებისათვის.

- *Theora* - კოდირების თითქმის ყველაზე „ახალგაზრდა“ ფორმატი. ისიც რამოდენიმე წლის წინ Xiph.org ორგანიზაციის მიერ იყო დამუშავებული. გამოიყენება ვიდეო-მონაცემების კოდირებისათვის.
- *H.264* - ესეც ძალიან „ახალგაზრდა“ ფორმატია. 2003 წელს Motion Picture Expert Group და Video Coding Experts Group (ვიდეოს კოდირების ექსპერტთა ჯგუფი) ორგანიზაციების მიერ იყო წარმოდგენილი. ვიდეო მონაცემების კოდირებისათვისაა განკუთვნილი.

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

პირველ ცხრილში მოყვანილია ბრაუზერებისა და კოდირების ფორმატების ურთიერთმხარდაჭერა.

ცხრილი 1

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

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



## MIME ტიპები

ქსელის საშუალებით სულ სხვადასხვა მონაცემები გადაიცემა: ვებ-გვერდები, გრაფიკული გამოსახულება, აუდიო- და ვიდეო-ფაილები, არქივი, შესრულებადი ფაილები და სხვა. ეს მონაცემები სხვადასხვა პროგრამებისათვისაა გათვალისწინებული. ამასთან, მიმღები პროგრამა შეიძლება სხვადასხვანაირად მოიქცეს. ასე მაგალითად, ვებ-გვერდის 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 მიენიჭება, 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, ვებ-სერვერი. ხოლო მიმღები პროგრამა MIME ტიპის მიხედვით განსაზღვრავს, მოცემული ბრაუზერი მხარს უჭერს თუ არა მიღებულ მონაცემებს, და თუ უჭერს, მაშინ რა უყოს მას.

ვებ-გვერდის 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.

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

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

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

### აუდიო-რგოლის ჩასმა

ვებ-გვერდზე აუდიო-რგოლის ჩასასმელად HTML-5 ენა წყვილ <AUDIO> ტეგს იყენებს. ფაილის ინტერნეტ-მისამართს, რომელშიც მოცემული აუდიო-რგოლი 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 აუდიო-ფაილი გაუშვას, შეაჩეროს იგი, წინ და უკან გადაახვიოს, არეგულიროს ხმა. ყოველივე ეს <AUDIO> ტეგის ატრიბუტების საშუალებით რეგულირდება.

<AUDIO> ტეგი ვებ-გვერდზე ბლოკურ ელემენტს ქმნის. ასე, რომ აუდიო-რგოლის აბზაცში ჩასმას ვერ შევძლებთ. სამაგიეროდ, ის რომ ცალკე აბზაცად იყოს გამოყოფილი, ამისათვის არავითარი დამატებითი მოქმედება არ არის ჩასატარებელი.

ბრაუზერი აუდიო-რგოლის ჩუმათობის პრინციპით გაშვებას ვერ უზრუნველყოფს, ამისათვის, <AUDIO> ტეგში განსაკუთრებული ატრიბუტი - AUTOPLAY უნდა მივუთითოთ. მას არ გააჩნია მნიშვნელობა - მთავარია იგი ეწეროს ტეგში და აუდიო-რგოლი 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 აუდიო-რგოლის გაშვების მართვის ელემენტებს გამოიტანს. მასში შედის გაშვებისა და გაჩერების ღილაკები, დროის შკალა 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 ვებ-გვერდის ჩატვირთვის შემდეგ, მაშინვე აუდიო-რგოლსაც ჩატვირთავს, რათა შეყოვნება არ მოხდეს. ამასთან, უნდა გვახსოვდეს, რომ ამ ატრიბუტის გამოყენება მხოლოდ იქ შეიძლება, სადაც ატრიბუტი AUTOPLAY არ არის გამოყენებული.

ქვემოთ მოყვანილია აუდიო-რგოლის 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 ვებ-გვერდზე აუდიო რგოლის ჩასასმელად
გამოიყენება.</P>
<H6>მაგალითი:</H6>
<PRE>&lt;AUDIO SRC=&quot;Kalimba.wav&quot;
CONTROLS&gt;&lt;/AUDIO&gt; </PRE>
<H6>შედეგი:</H6>
<AUDIO SRC="Kalimba.wav" CONTROLS></AUDIO>
</BODY>
</HTML>

```



შექმნილი ვებ-გვერდი და აუდიო-რგოლი ერთიდაიმავე საქალაქში უნდა შეინახოს.

## ვიდეო-რგოლის ჩასმა

ვებ-გვერდზე ვიდეო-რგოლის ჩასასმელად წყვილი `<VIDEO>` ტეგი გამოიყენება. ვიდეო-ფაილის ინტერნეტ-მისამართი ამ ტეგის SRC ატრიბუტით მიეთითება:

```
<VIDEO SRC="film.ogg"></VIDEO>
```

ამ ტეგის შეხვედრის შემთხვევაში, ბრაუზერი ვებ-გვერდის ამ ადგილზე ვიდეო-რგოლის ჩვენების პანელს გამოიტანს. `<VIDEO>` ტეგში მითითებული ატრიბუტების მიხედვით მას შეუძლია მაშინვე ჩატვირთოს ვიდეო-რგოლი და გაუშვას, მხოლოდ ჩატვირთოს ან საერთოდ არაფერი არ გააკეთოს. ასევე, მას შეუძლია ვებ-გვერდზე ვიდეო-რგოლის გაშვების მართვის ელემენტები გამოიტანოს.

როგორც `<AUDIO>` ტეგი, ასევე `<VIDEO>` ტეგი ვებ-გვერდის ბლოკურ ელემენტს ქმნის და `AUTOPLAY`, `CONTROLS` და `AUTOBUFFER` ატრიბუტი გააჩნია. მაგალითად,

```
<VIDEO SRC="film.ogg" AUTOPLAY CONTROLS></VIDEO>
```

თუ ვიდეო-რგოლის 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 ვიდეო-რგოლს ვერ მოძებნით, მაშინ შეიძლება თქვენ შექმნათ ის, სხვა ფორმატში შენახული ვიდეო-რგოლის კოდირების შეცვლის საშუალებით. ამისათვის SUPER © უტილიტა გამოიყენება, რომელიც შემდეგ ინტერნეტ მისამართზე შეიძლება იპოვოთ:

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

ქვემოთ ვებ-გვერდზე ვიდეო რგოლის 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 ვებ-გვერდზე ვიდეო რგოლის ჩასასმელად
გამოიყენება.</P>
<H6>მაგალითი:</H6>
<PRE>&lt;VIDEO SRC="Wildlife.wmv"&gt;
CONTROLS&gt;&lt;/VIDEO&gt; </PRE>
<H6>შედეგი:</H6>
<VIDEO SRC="Wildlife.wmv" CONTROLS></VIDEO>
</BODY>
</HTML>
```

## დოკუმენტების პუბლიკაცია და ძებნა

დოკუმენტების ინტერნეტში განთავსების საკითხი ორი ეტაპისაგან შედგება:

- ვებ-საიტის მომზადება;
- ვებ-საიტის პუბლიკაცია.

ვებ-საიტის პუბლიკაცია მისი გვერდების მომხმარებლისათვის ხელმისაწვდომ სპეციალურ ვებ-საიტების დისკურ სივრცეში, რომელსაც ჰოსტი (ინგლ. *host* - პატრონი) ეწოდება, შენახვით ხორციელდება, ხოლო ამ მომსახურებას ჰოსტინგი ეწოდება.

ჰოსტინგი - ეს სერვერზე, რომელიც ქსელში (უმეტეს შემთხვევაში ინტერნეტი) მუდმივად არის ჩართული, ინფორმაციის ფიზიკური განთავსებისათვის გამოთვლითი სიმძლავრეების გამოყოფის მომსახურებაა. ჰოსტინგს ხშირად უფასო და ფასიან ჰოსტინგებად ყოფენ. ჩვეულებრივ, როდესაც გვაქვს უფასო ჰოსტინგი, მაშინ ამ კომპანიის შემოსავლის წყაროს მასში ვებ-გვერდებზე განთავსებული რეკლამები წარმოადგენს. ჰოსტინგის შერჩევის ერთ-ერთი მნიშვნელოვანი გარემო ასევე, გამოყენებული ოპერაციული სისტემაა, ვინაიდან მასზეა დამოკიდებული პროგრამული უზრუნველყოფა, რომელიც ამა თუ იმ სერვისის ფუნქციონირებას დაუჭერს მხარს. ჰოსტინგის აღწერის მნიშვნელოვან ასპექტს ამა თუ იმ სამსახურისა და შესაძლებლობების (Perl, PHP, ASP, Ruby, მონაცემთა ბაზები და სხვა) არსებობა წარმოადგენს.

ქვემოთ მოყვანილია ზოგიერთი ჰოსტინგი, მათ შორის ქართულენოვანი, ინგლისურენოვანი და სხვა:

<http://www.servage.net>,  
[www.avangardhosting.com](http://www.avangardhosting.com),  
<http://domenebi.com>,

<http://www.myhosting.ge>,

<http://www.ucoz.com>,

<http://www.save.ge>.

ვებ-ჰოსტს შეუძლია განავითაროს ან პირიქით დაანგრის თქვენი ბიზნესი. ნებისმიერი ბიზნესის ონლაინ წარმატება, იგივე ზომითაა დამოკიდებული ვებ-ჰოსტინგის პროვაიდერზე, როგორც იგი დამოკიდებულია საერთოდ ბიზნეს-გეგმაზე.

დააკვირდით ჰოსტინგის დირექტორებს, და თქვენი საიტისათვის ვებ-ჰოსტის არჩევისას გაითვალისწინეთ: ტექნიკური ასპექტები, როგორც არის ტრაფიკი, სიჩქარე, საიმედოობა, გამოყოფილი სერვერები, საოპერაციო სისტემები და დაპროგრამების ხელსაწყოები. ეცადეთ ვებ-ჰოსტის სისტემებზე და გარემოებებზე რაც შეიძლება მეტი შეისწავლოთ. ფასიც მნიშვნელოვან როლს თამაშობს, ამიტომ, ეცადეთ აირჩიოთ ჰოსტი, რომელიც ყველა თქვენ მოთხოვნებს დააკმაყოფილებს.

იდეალურ ჰოსტს უნდა გააჩნდეს: თანამედროვე სერვერების ტექნოლოგიები, სისტემური ადმინისტრატორები, ქსელის და კავშირგაბმულობის სათადარიგო საშუალებები, ძლიერი მხარდაჭერის გუნდი და შეღავათიანი პაკეტები. ჰოსტინგი უნდა იზრდებოდეს და ვითარდებოდეს თქვენთან ერთად. მხოლოდ მაშინ, როდესაც თქვენ და თქვენი ჰოსტინგის პროვაიდერი სინქრონულად ხართ, თქვენი საიტიც წარმატებული იქნება. მოკლედ, ყველაზე მთავარი რაც უნდა გაითვალისწინოთ არის: ტრაფიკი, სიჩქარე, სერვერების ტექნოლოგია, სერვერის შიდა (server-side) სკრიპტები, მონაცემთა ბაზების მხარდაჭერა და მულტიმედიური საშუალებები.

ვებ-ჰოსტი უნდა უყოს:

1. ეთიკურად და სოციალურად პასუხისმგებელი;
2. გაუწიოს მაქსიმალური მხარდაჭერა თავის კლიენტებს;



3. განავითაროს და შესთავაზოს სრულყოფილი მომსახურეობა. ამავე დროს აღებული ვალდებულებები და მოცემული დაპირებები 100%-ით შეასრულოს;
4. იყოს პრაქტიკული და კლიენტების მოთხოვნები დაპირებების მიცემამდე შეისწავლოს. მან უნდა გააცნობიეროს, რა შეუძლია, და რა არა;
5. ვებ-ჰოსტი უნდა იყოს ტექნოლოგიების განვითარების სიახლეების საქმის კურსში და კლიენტების საჭიროებების თანახმად საკუთარი ტექნოლოგიები განავითაროს;
6. უნდა ცდილობდეს, კლიენტთან მყარი და ხანგძლივი ურთიერთობის დამყარებას, ხოლო ჰოსტინგის სერვისის დონე კი უმაღლესი და საუკეთესო უნდა იყოს;
7. გაუწიოს კლიენტებს ტექნიკური მხარდაჭერა და ზედმეტი, მათთვის არამნიშვნელოვანი, ინფორმაციითა და საგნებით არ შეაწუხოს;
8. უნდა იმართებოდეს მცოდნე ხალხის მიერ, რომელთაც ვებ-სერვერების ტექნოლოგია და სირთულეები ესმით. ვებ-ჰოსტს საჭესთან უნდა ჰყავდეს გუნდი, რომელიც არა მხოლოდ სწრაფ მოგებაზე, არამედ ბიზნესის განვითარებაზე ზრუნავს.

როგორც დაინტერესებული მომხმარებელი, თქვენ უნდა იყოთ დარწმუნებული რომ, ყველა თქვენი მონაცემები და ფაილები სათანადოდ არის შენახული და ნებისმიერ დროსაა ხელმისაწვდომი. ყოველთვის გამოიძიეთ ვებ-ჰოსტინგის პროვაიდერების სია, და წაიკითხეთ განხილვები. დარწმუნდით, რომ ყველა მოცემული დაპირება და გარანტიები ნახსენებია ხელშეკრილებაში ან მომსახურეობის განმარტებაში. ასევე გაითვალისწინეთ, თქვენი ვალდებულებები, იმიტომ რომ, ვებ-ჰოსტსაც სურს ჰყავდეს იდეალური კლიენტი, ისევე როგორც თქვენ გასურთ გყავდეთ იდეალური ვებ-ჰოსტი.

ვებ-ჰოსტინგი არის კონკურენტუნარიანი და შემოსავლიანი ბიზნესი, მაგრამ წარმატების მისაღწევად, ვებ-ჰოსტი სხვებისგან კონკურენტუნარიანობით, საიმედოობით და პატიოსნებით უნდა გამოირჩეოდეს.

## **ვებ-საიტების მომზადება და პუბლიკაცია**

საიტი - ეს ინტერნეტის რესურსია, სერვერის ერთ-ერთი განყოფილება, რომელიც მთლიანად ერთ თემას ეძღვნება. სერვერისაგან განსხვავებით, საიტის მომსახურებისათვის, როგორც წესი, ცალკე სერვერული პროგრამა არ გამოიყოფა, და სერვერზე განლაგებულ ყველა საიტს ერთი ასეთი პროგრამა ემსახურება. საიტი სერვერის ნაწილია, თუმცა ყველა საიტს საკუთარი დომენის სახელი აქვს. საიტებს შეიძლება ჰქონდეს სხვადასხვა ქვეთავები, მაგრამ მიუხედავად ამისა, ყველა ქვეთავი უნდა დაექვემდებაროს საერთო იდეას, შესრულების ერთ სტილს. საიტი - დამოუკიდებელი ლოგიკური ერთეულია, რომელიც რიგი აზრობრივად ურთიერთდაკავშირებული ტექსტური დოკუმენტებისა და გრაფიკული ილუსტრაციებისაგან შედგება.

მაგალითის სახით მოვიყვანოთ <http://www.myhosting.ge> ჰოსტინგი. გაიხსნება ქვემოთ მოყვანილი გვერდი, სადაც უნდა ავირჩიოთ ჩვენთვის სასურველი პაკეტი.



პაკეტის შერჩევის შემდეგ უნდა შეირჩეს საიტის სახელი და მივყვეთ შემდგომ ეტაპებს. აუცილებელია, საიტის სახელისა და პაროლის მოფიქრება, რომლის დახმარებითაც საიტზე შესვლა და მისი კორექტირება იქნება შესაძლებელი.

## ინტერნეტში ინფორმაციის ძებნა

ინტერნეტში ინფორმაციის ძებნის სამი ძირითადი საშუალება არსებობს: თემატური ბმულების, URL-მისამართებისა და საძიებო სისტემების საშუალებით. ინფორმაციის ძებნა ყველაზე მარტივი თემატური ბმულების საშუალებითაა - საკმარისია შემოთავაზებულ ბმულზე მაუსით დავაწკაპუნოთ, რითაც საძიებო არიალს ეტაპობრივად ვამცირებთ, მაგალითად, ყიდვა ⇒ სპორტული საქონელი ⇒ ტრენაჟორი და ა. შ. სხვა ტიპის ძებნა შედარებით რთულია.

## ინფორმაციის ძებნა URL-მისამართის მიხედვით

იმისათვის, რომ ინფორმაცია URL-მისამართის მიხედვით ვეძებოთ, სახელთა დომენურ სისტემაზე უნდა გვქონდეს წარმოდგენა. ასეთი სისტემის შესაბამისად კომპიუტერების მისამართები ე. წ. დომენებად იყოფა (ინგლ. *domain* - არე), ამასთან, URL-მისამართის ბოლო ნაწილის სიმბოლოები თემატურ ან გეოგრაფიულ დომენებს წარმოადგენს. ქვემოთ ზოგიერთი, განსაკუთრებულად ცნობილი მსგავსი დომენია ჩამოთვლილი.

თემატური დომენები:

- com - კორპორაცია;
- edu - სასწავლო დაწესებულება;
- gov - არასამხედრო სახელმწიფო დაწესებულება;
- mil - სამხედრო დაწესებულება;
- net - ქსელური ორგანიზაცია;
- org - სხვადასხვა ორგანიზაციები;
- int - საერთაშორისო ორგანიზაცია;
- firm, biz - ბიზნესი;
- store - ვირტუალური მაღაზიები;
- web - ვებ-ორგანიზაციები;

- arts - კულტურის სფეროს ორგანიზაციები;
- rec - გართობის სფეროს ორგანიზაციები;
- info - ინფორმაციის მიმწოდებელი ორგანიზაციები;
- nom - ინდივიდუალური მომხმარებელი;
- arpa - ისტორიულად წარმოშობილი ზედა დონის დომენი, რომელიც ARPANET ქსელს (ინტერნეტის წინამორბედი) შეესაბამება.

გეოგრაფიული დომენები:

- at - ავსტრია (Austria);
- an - ავსტრალია (Australia);
- be - ბელგია (Belgium);
- bg - ბულგარეთი (Bolgaria);
- ca - კანადა (Canada);
- ch - შვეიცარია (Switzerland);
- cy - კვიპროსი (Cypr);
- de - გერმანია (Germany);
- es - ესპანეთი (Espania);
- fi - ფინეთი (Finland);
- ge - საქართველო (Georgia);
- gr - საბერძნეთი (Greece);
- it - იტალია (Italy);
- jp - იაპონია (Japan);
- hl - ჰოლანდია (Holland);
- no - ნორვეგია (Norway);
- pl - პოლონეთი (Poland);
- ru - რუსეთი (Russia);
- se - შვეცია (Sweden);
- tr - თურქეთი (Turkey);
- uk - დიდი ბრიტანეთი (United Kingdom);
- us - აშშ (United State).

თუ მომხმარებელს ზედა დონის დომენების შესახებ აქვს ინფორმაცია, მაშინ მან საიტის ძებნის დროს დაახლოებით იცის ბრაუზერის სამისამართო ველში რა უნდა ჩაწეროს. ქვემოთ მოყვანილია რამოდენიმე მაგალითი:

კორპორატიული საიტების ძებნისათვის საჭიროა ფირმის, საწარმოს ან ორგანიზაციის სახელს .com დომენი დავუმატოთ, ხოლო წინ ჩავსვათ წვდომის მეთოდის მაჩვენებელი სიმბოლოები (www). ამ ჩანაწერის შემდეგ, უმეტეს შემთხვევაში, სასურველ შედეგს მივიღებთ, მაგალითად:

www.adobe.com - ცნობილი გრაფიკული რედაქტორის მწარმოებელი კომპანიის Adobe-ს მისამართია;

www.microsoft.com - კორპორაციის Microsoft-ის მისამართი;

www.europages.com – ევროპის ყველა ქვეყნის ტერიტორიაზე განლაგებული 150 ათასზე მეტი კომპანიის, ინგლისურ, ფრანგულ, გერმანულ, ესპანურ და იტალიურ ენებზე, ცნობარი;

www.intellimatch.com - მონაცემთა უფასო ბანკი, მათთვის ვინც ეძებს მუშახელს ან ვინც ეძებს სამუშაოს აშშ-ში;

საქართველოს სერვერებზე ინფორმაციის მოსაძებნად ფირმის, ორგანიზაციის, საწარმოს სახელს უნდა დავუმატოთ დომენი .ge, ხოლო წინ სიმბოლოები - www, მაგალითად:

www.translate.ge - ტექსტების ელექტრონული მთარგმნელი;

www.1tv.ge - საზოგადოებრივი მაუწყებლის პირველი არხი;

www.gtu.ge - საქართველოს ტექნიკური უნივერსიტეტი;

ამერიკის ან ინგლისის მსხვილი სასწავლო დაწესებულების სახელს ხშირად დომენი .edu ემატება, მაგალითად:

www.oxford.edu - ოქსფორდის უნივერსიტეტის საიტი.

ყოველ ქვეყანაში თავისი რეგიონალური სერვერი არის, მაგალითად:

www.usa.net - აშშ;

www.uk.net - დიდი ბრიტანეთი

## სადიებო სისტემები

სადიებო სისტემები, რომლებსაც კიდეც სადიებო პროგრამებს, სადიებო სერვერებს ან სადიებო მანქანებსაც უწოდებენ, ინტერნეტში მრავლადაა.

სადიებო სისტემის გამოსაძახებლად საჭიროა ბრაუზერის სამისამართო ველში მისი მისამართი შევიტანოთ. ქვემოთ მოყვანილია ზოგიერთი ცნობილი სადიებო სისტემის მისამართები:

<http://www.google.com>,

<http://www.yahoo.com>,

<http://www.bing.com>,

<http://www.lycos.com>,

<http://www.altavista.com>,

<http://www.go.com>,

<http://www.excite.com>,

<http://www.nigma.com>,

<http://www.hotbol.com>.

სადიებო სისტემის ჩატვირთვის შემდეგ მომხმარებელმა სადიებო ველში უნდა შეიტანოს შეკითხვა (ტექსტის საძებნი სტრიქონი) ნებისმიერ ენაზე (ქართული, ინგლისური და სხვა) და ძებნის ღილაკზე მაუსით დავაწკაპუნოთ.

შეკითხვის შესრულების შედეგად იმ დოკუმენტების ვებგვერდების მისამართების სია გამოვა, რომლებშიც შეკითხვაში შეტანილი სიტყვა გვხვდება. ამ სიაში გვერდის ვებ-მისამართი, მისი დასახელება, ავტორის გვარი და გვერდის მოკლე აღწერა იქნება მითითებული. საჭირო მისამართის შერჩევის შემდეგ ნაპოვნი დოკუმენტის გახსნა შეიძლება. ჩვეულებრივ, ძიების შედეგი უამრავი დოკუმენტია (ზოგჯერ ათასობით), რომლებიც გვერდებადაა დაჯგუფებული. ასეთ შემთხვევაში, მარტივად რომ მოვახდინოთ ორიენტირება, სადიებო სისტემა მასალებს

ახარისხებს - პირველ რიგში იმ გვერდების მისამართებს გვთავაზობს, რომლებიც მისი აზრით ჩვენ შეკითხვას უფრო ზუსტად აკმაყოფილებს.

ნებისმიერი საძიებო სისტემის საფუძველს ქსელური რობოტი წარმოადგენს, რომელსაც ხშირად ობობას (spider), ჭიას (worm), ან მხოხავს (crawler) უწოდებენ ხოლმე. საძიებო სისტემა ასეთ ობობას ქსელში გააგზავნის, რათა შემღებობისდაგვარად მაქსიმალური რაოდენობის ვებ-გვერდები დაათვალიეროს და მათი მისამართები თავის მონაცემთა ბაზაში შეინახოს. მას შემდეგ, რაც მომხმარებელი შეიტანს შეკითხვას და მაუსზე დააწკაპუნებს, საძიებო სისტემა თავის მონაცემთა ბაზას გადაათვალიერებს და ეკრანზე ძებნის შედეგებს გამოიტანს.

თუ რესურსის ელექტრონული მისამართის (URL) ჩანაწერი გვაქვს, მაგალითად, <http://www.gladiators.ge/Istoria.html>, მაშინ:

- Istoria.html – ეს ფარდობითი ქსელური გზაა, რომელზეც მითითებული ფაილია განთავსებული;
- <http://www.gladiators.ge/Istoria.html> - რესურსის სრული უნუვერსალური ლოკატორია, რომელზეც მოთხოვნილი ფაილია განთავსებული;
- http:// - ქსელური პროტოკოლია, რომელზეც მონაცემთა ძებნა მიმდინარეობს. ამ შემთხვევაში მონაცემთა გადაცემის ჰიპერტექსტური პროტოკოლი;
- [www.gladiators.ge](http://www.gladiators.ge) - სერვერის მისამართია.





## იბეჭდება ავტორთა მიერ წარმოდგენილი სახით

გადაეცა წარმოებას 25.06.2013. ხელმოწერილია დასაბეჭდად  
17.07.2013. ქაღალდის ზომა 60X84 1/16. პირობითი ნაბეჭდი თაბახი 14.

საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, თბილისი,  
კოსტავას 77



Verba volant,  
scripta manent

ი.მ. „გოჩა დალაქიშვილი“,  
ქ. თბილისი, ვარკეთილი 3, კორპ. 333, ბინა 38