

საქართველოს ტექნიკური უნივერსიტეტი

თეა თოდუა, ლაშა ვერულავა

**WEB-ტექნოლოგიები  
HTML**

I ნაწილი

დამტკიცებულია სტუ-ს  
სასწავლო-მეთოდური  
საბჭოს მიერ

თბილისი  
2006

უპკ 681.142

სახელმძღვანელოში განხილულია HTML-ის, web-სივრცეში ინფორმაციის წარმოდგენის ყველაზე მძლავრი და უნივერსალური საშუალების, პრაქტიკული გამოყენების საკითხები.

სახელმძღვანელო გათვალისწინებულია ბაკალავრიატის სტუდენტებისა და მაგისტრანტებისათვის, აგრეთვე ის გამოადგება მასში განხილული საკითხებით დაინტერესებულ ყველა პირს.

რეცენზენტები: ტ.მ.დ. პროფ. ო. ვერულავა  
პროფ. ზ. ბაიაშვილი

© გამომცემლობა „ტექნიკური უნივერსიტეტი“, 2006

ISBN 99940-48-93-7

## სარჩევი

1. შესავალი .....	4
2. HTML დოკუმენტის შენახვა და გახსნა .....	5
3. HTML დოკუმენტის სტრუქტურა .....	5
4. ფერები .....	6
5. აბზაცები და სათაურები HTML დოკუმენტში .....	9
6. ჰორიზონტალური ხაზი .....	11
7. წინასწარ დაფორმატებული ტექსტის გამოყენება .....	12
8. შრიფტის დაფორმატება .....	13
9. ტეგი Basefont .....	15
10. Font .....	15
11. სიები .....	18
12. გამოსახულება HTML დოკუმენტში .....	22
13. ბმული .....	25
14. ცხრილები .....	29
15. ცხრილების მაგალითები .....	35
16. სპეციალური სიმბოლოები .....	47
17. საზღვრები ბრაუზერის ფანჯარაში .....	48
18. მორბენალი სტრიქონი .....	49
19. ფრეიმები .....	50
20. კონტეინერი Frameset .....	51
21. ფრეიმების პარამეტრების განსაზღვრა .....	51
22. ტეგი <Frame> .....	53
23. ფრეიმების გამოყენების რამდენიმე მარტივი მაგალითი .....	54
24. ფრეიმების გარეგნული სახის მოდიფიკაცია .....	58
25. ფრეიმის სახელის განსაზღვრა .....	60
26. ობიექტები HTML დოკუმენტში .....	64
27. ფორმები .....	67
28. ფორმის ტეგებთან მუშაობა .....	68

## 1. შესავალი

World Wide Web (ობობას მსოფლიო ქსელი, აბლაბუდა, www) – ინტერნეტის ყველაზე პოპულარული სამსახურია.

World Wide Web – ერთიანი საინფორმაციო სივრცეა, რომელიც web-სერვერებზე დაცული ასეული მილიონი ერთმანეთთან დაკავშირებული ელექტრონული დოკუმენტისაგან შედგება. Web – სივრცის შემადგენელ ცალკეულ დოკუმენტს web – გვერდი ეწოდება. Web – გვერდების დასათვალიერებელ სპეციალურ პროგრამებს ბრაუზერები (ინგლისური სიტყვიდან browse – გადათვალიერება, გადახედვა, გადაფურცვლა) ეწოდებათ. პირველი ბრაუზერი – Mosaic, 1993 წელს შეიქმნა ილინოისის უნივერსიტეტის სუპერკომპიუტერების ეროვნულ ცენტრში (NSCA) უნივერსიტეტის სტუდენტებისა და პროგრამისტების ჯგუფის მიერ, რომელსაც ხელმძღვანელობდა მარკ ანდრისენი. Mosaic ქსელში საოცარი სისწრაფით ვრცელდებოდა, ერთი წლის განმავლობაში დაახლოებით ორი მილიონი ადამიანი სარგებლობდა ამ ბრაუზერით. მათი უმრავლესობა ინტერნეტ-გვერდების დათვალიერებით იყო დაკავებული, მხოლოდ მცირე ნაწილი შეუდგა web-გვერდების შექმნას. Mosaic – ის შემდეგ სხვა ახალი ბრაუზერებიც გამოჩნდნენ. თითოეული მათგანი თავის წინამორბედს აღემატებოდა ახალი Html ბრძანებების რაოდენობით.

მარკ ანდრისენმა დატოვა NSCA და ჯიმ კლარკთან ერთად დააფუძნა Netscape Communications Corporation. უმრავლესობა პროგრამისტებისა, რომლებიც მუშაობდნენ Mosaic-ზე, შეუერთდნენ მათ, რის შედეგადაც ძალიან სწრაფად გამოჩნდა ბრაუზერ Netscape Navigator (NN)-ის პირველი ვერსია. ახალი თვისებების დიდი რაოდენობისა და საცდელი ვერსიის უფასო გავრცელების გამო NN მალე გახდა პოპულარული. ეს ყველაფერი უყურადღებოდ არ დარჩენია კომპანია Microsoft-ს. 1995 წელს მან გამოუშვა ბრაუზერი Internet Explorer (IE).

იმისათვის, რომ თავად შექმნათ web – გვერდები, საჭიროა HTML (Hypertext Markup Language)-ის შესწავლა. ეს სახელმძღვანელო დაგეხმარებათ საფუძვლიანად დაეუფლოთ ამ ტექნოლოგიას.

## 2. HTML დოკუმენტის შენახვა და გახსნა

უპირველეს ყოვლისა, შექმენით ცალკე საქალაქო თქვენი მომავალი web გვერდისათვის. შემდეგ გახსენით Notepad (start→programs→accessories→notepad) და აკრიფეთ მასში ტექსტი:

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)

გახსენით ბრაუზერი, მაგალითად, Internet Explorer და დაათვალიერეთ თქვენს მიერ შექმნილი დოკუმენტი.

File → open→browse→index.html

HTML დოკუმენტში ცვლილების შეტანის შემთხვევაში, იმისათვის, რომ დაინახოთ როგორ გამოიყურება სახეშეცვლილი ვარიანტი, ბრაუზერის ფანჯარაში დააჭირეთ ღილაკს F5 ან Internet Explorer-ის მენიუში View→Refresh.

## 3. HTML დოკუმენტის სტრუქტურა

HTML დოკუმენტის ყველაზე მნიშვნელოვანი ელემენტია <html>. ის იწყებს HTML დოკუმენტს, ხოლო </html> წარმოადგენს HTML დოკუმენტის უკანასკნელ სტრიქონს. ამგვარი ტიპის ჩანაწერებს, რომლებიც იწყება ნიშნით „<“ და მთავრდება „>“ ნიშნით, ეწოდებათ HTML ტეგები. ტეგები <html> და </html> აღნიშნავენ, რომ მათ შორის მდებარე სტრიქონები წარმოადგენენ ერთიან HTML დოკუმენტს. ეს მნიშვნელოვანია, რადგან დოკუმენტი წარმოადგენს ჩვეულებრივ ტექსტურ ფაილს ASCII ფორმატში. ამ ტეგების გარეშე ბრაუზერს ან სხვა მსგავს პროგრამას არ შეუძლია დოკუმენტის ფორმატის იდენტიფიცირება და მისი სწორად ინტერპრეტირება.

<head> ...</head> ტეგი მიუთითებს დოკუმენტის სათაურის დასაწყისსა და დასასრულზე. ყველაფერი, რაც მოთავსებულია <title> და </title>-ს შორის, წარმოადგენს დოკუმენტის სახელწოდებას, რომელიც ჩნდება ბრაუზერის ფანჯრის სათაურის ზოლში.

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

კომენტარები არ აისახებიან ეკრანზე ბრაუზერის მიერ.

<! - - ერთსტრიქონიანი კომენტარი - - >

<! მრავალსტრიქონიანი კომენტარი>

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

მიაქციეთ ყურადღება ტეგების გახსნისა და დახურვის მიმდევრობას:

<ტეგი1> <ტეგი2> <ტეგი3>. . . </ტეგი3> </ტეგი2> </ტეგი1>. სხვა მიმდევრობით ტეგების განლაგებამ შეიძლება მოგცეთ შეცდომა.

ტეგები შეგიძლიათ დაწეროთ როგორც დიდი, ასევე პატარა ასოებით, ბრაუზერისთვის ამას მნიშვნელობა არა აქვს.

## 4. ფერები

Html-ში ფერები განისაზღვრება ციფრებით თექვსმეტობით კოდში. ფერების გამა ეყრდნობა სამ ძირითად ფერს: წითელს, მწვანესა და ლურჯს და აღინიშნება RGB-თი. ფერი შეიძლება აღიწეროს როგორც მისი დასახელებით, ასევე მისი მნიშვნელობით RGB (Red, Green, Black) პალიტრაში. თითოეული ფერისათვის მიეთითება თექვსმეტობითი მნიშვნელობა 00-დან FF-მდე, რასაც ათობით სისტემაში შეესაბამება 0-255 დიაპაზონი. შემდეგ ეს მნიშვნელობები ერთიანდებიან ერთ რიცხვში, რომელთა წინ იწერება # სიმბოლო. მაგალითად: რიცხვი #800080 აღნიშნავს იისფერს.

განვიხილოთ რამდენიმე მაგალითი: text=#FFFFFF. წითლით, მწვანითა და ლურჯით გაჯერებულობა ერთიდაიგივეა, (FF არის რიცხვ 255-ის თექვსმეტობითი წარმოდგენა), შედეგი - თეთრი ფერი. text = #000000, წითლით მწვანითა და ლურჯით გაჯერებულობა ამ შემთხვევაშიც ერთიდაიგივეა (00). შედეგი - შავი

ფერი. text = #FF0000. წითლით გაჯერებულობა - FF, მწვანით და ლურჯით - 00. შედეგად მიიღება წითელი ფერი.

ქვემოთ მოცემულია თექვსმეტი სტანდარტული ფერი, თავისი შესაბამისი თექვსმეტობითი კოდებით:

ფერი	კოდი
Black	#000000
Maroon	#800000
Green	#008000
Olive	#808000
Navy	#000080
Purple	#800080
Teal	#008080
Gray	#808080
Silver	#C0C0C0
Red	#FF0000
Lime	#00FF00
yellow	#FFFF00
blue	#0000FF
Fuchsia	#FF00FF
Aqua	#00FFFF
White	#FFFFFF

მაგალითი: 

```
<html>
<head>
<title> My html Document </title>
</head>
<body>
Hello! This is my first page
<br>
<font color =“#000080”> Welcome! </font>
</body>
</html>
```

ტეგი <font></font> მრავალფუნქციური ტეგია. შემდგომში მას უფრო დაწვრილებით განვიხილავთ.

ტექსტის ფერის მითითება სხვაგვარადაც შეიძლება: <body text="red">. ეს ნიშნავს, რომ მთელი ტექსტი იქნება წითელი ფერის, გარდა იმ ტექსტისა, რომელიც მოთავსებულია <font></font> ტეგებს შორის. შესაბამისი ფერის მიუთითებლობის შემთხვევაში ტექსტი იქნება შავი ფერის.

დოკუმენტის ფონი განისაზღვრება შემდეგნაირად: <body bgcolor="C0C0C0">

საბოლოოდ, დოკუმენტი მიიღებს სახეს:

მაგალითი:

```
<html>
  <head>
    <title>My html Document </title>
  </head>
  <body text="#FF0000" bgcolor="#C0C0C0">
    Hello! This is my first page
    <br>
    <font color="#000080"> Welcome!</font> My name is Tea
  </body>
</html>
```

ამ მაგალითში ტეგ <body>-ს ორი ატრიბუტი აქვს: text და bgcolor. ტეგ <body>-ს სხვა ატრიბუტებიც გააჩნია:

Link – ბმულის ფერი;

Alink – ბმულის ფერი მასზე მაუსით დაწკაპუნების დროს;

Vlink – უკვე ნანახი ბმულის ფერი (მაგ.: <body link="black" vlink="blue">);

Background – ფონის სურათი (მაგ.: <body background="book.gif">);

Topmargin, bottommargin – დოკუმენტის ზედა და ქვედა მინდვრების სიგრძე (Internet Explorer);

Rightmargin, Leftmargin – დოკუმენტის მარჯვენა და მარცხენა მინდვრების სიგრძე (Internet Explorer);

Marginheight-დოკუმენტის ზედა და ქვედა მინდვრების სიგრძე (Netscape Navigator);

Marginwidth - დოკუმენტის მარჯვენა და მარცხენა მინდვრების სიგრძე (Netscape Navigator);

body-ს ბოლო ექვს პარამეტრს დაწვრილებით განვიხილავთ ქვემოთ.

## 5. აბზაცები და სათაურები HTML დოკუმენტში

Html-ს აქვს ექვსი სხვადასხვა ზომის სათაურის შექმნის შესაძლებლობა. მათ მინიჭებული აქვთ ნომრები 1-დან 6-ის ჩათვლით. პირველი დონის სათაური H1 ყველაზე მსხვილია. შესაბამისად H6 ყველაზე წვრილია. <H1> ტეგს აუცილებლად უნდა ჰქონდეს მისი შესაბამისი დახურვის ტეგი </H1>.

<p>...</p> წყვილი აღწერს აბზაცს. ყველაფერი რაც მოთავსებულია <p> და </p>-ს შორის, აღიქმება როგორც ერთი აბზაცი.

სათაურებისა და აბზაცის ტეგების მეშვეობით შესაძლებელია ტექსტის გასწორება მარცხენა კიდიდან, მარჯვნიდან, მარჯვნიდან და მარცხნიდან ერთდროულად და ცენტრირება. მაგალითად:

LEFT - ტექსტი გასწორდება მარცხენა მხრიდან.

CENTER - ტექსტი მოთავსებული იქნება ცენტრში

RIGHT - ტექსტი გასწორდება მარჯვენა მხრიდან

JUSTIFY - ტექსტი გასწორდება ორივე მხრიდან.

იმისათვის, რომ გადახვიდეთ მომდევნო სტრიქონზე მიმდინარე სტრიქონის ნებისმიერ ადგილას, Html-ში არსებობს სტრიქონის გაწყვეტის ტეგი <Br>. აბზაცის ტეგისგან განსხვავებით <Br> სტრიქონსა და სტრიქონს შორის არ ტოვებს ადგილს. <Br>-ს არა აქვს შესაბამისი დახურვის ტეგი.

მაგალითი:

```
<html>
```

```
<head><title>html document's formatting </title></head>
```

```
<body>
```

```
<H1>Europe </H1>
```

```
<H2 ALIGN = "center"> England </h2>
```

```
<p align = "justify">
```

London is the capital of England and of Great Britains too. It is an ancient city.

When the Romans come to conquer the country, there already was a settlement

which they called LYindin, Later Londonium. Finally it got its present name. The

birthplace of London is the City, today it is the business heart of London. Most

banks, large offices and the stock are situated in the city. Millions of people come

here to do business, but not much people live here. The Political centre of London is,

of course, the House of Parliament on the bank of the Thames and the street called

Downing street. </p>

London consists of East End and West End. The West End is older. <br>

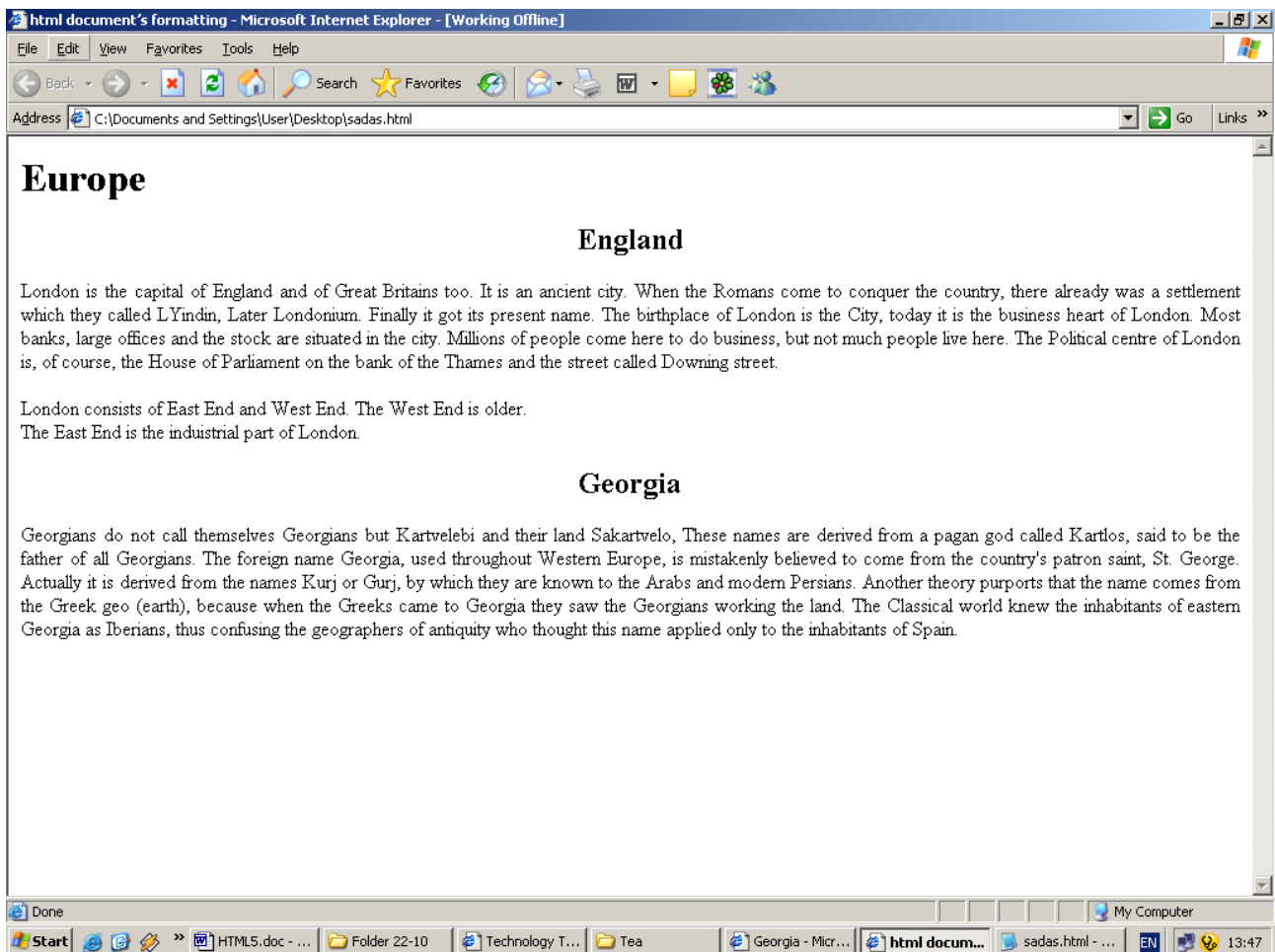
The East End is the industrial part of London.

<H2 ALIGN = "center"> Georgia </h2>

<p align="justify">Georgians do not call themselves Georgians but Kartvelebi and their land Sakartvelo, These names are derived from a pagan god called Kartlos, said to be the father of all Georgians. The foreign name Georgia, used throughout Western Europe, is mistakenly believed to come from the country's patron saint, St. George. Actually it is derived from the names Kurj or Gurj, by which they are known to the Arabs and modern Persians. Another theory purports that the name comes from the Greek geo (earth), because when the Greeks came to Georgia they saw the Georgians working the land. The Classical world knew the inhabitants of eastern Georgia as Iberians, thus confusing the geographers of antiquity who thought this name applied only to the inhabitants of Spain.</p>

</body>

</html>



## 6. ჰორიზონტალური ხაზი

დოკუმენტის ნაწილებად დაყოფის ერთ-ერთ მეთოდს წარმოადგენს ჰორიზონტალური ხაზების გაკვება. ის ვიზუალურად ხაზს უსვამს გვერდის ამა თუ იმ არის დასრულებულობას. ელემენტი <HR> რელიეფური ჰორიზონტალური ხაზის გაკვების საშუალებას იძლევა. <HR>-ს არა აქვს მისი დახურვის შესაბამისი </HR>. ჰორიზონტალურ ხაზამდე და მის შემდეგ, ავტომატურად ჩნდება ცარიელი სტრიქონი. <HR>-ის ატრიბუტებია:

**ALIGN** – ხაზს ასწორებს მარცხენა ან მარჯვენა მხრიდან ან ცენტრში. შესაბამისად, იღებს მნიშვნელობებს left, right, center.

**Width** – ხაზის სიგრძეს აყენებს პიქსელებში ან პროცენტებში ბრაუზერის ფანჯრის სიგანისგან დამოკიდებულებით. უკანასკნელ შემთხვევაში რიცხვითი მნიშვნელობის შემდეგ იწერება %-ის ნიშანი.

**SIZE** – აყენებს ხაზის სისქეს პიქსელებში.

**Noshade** – ცვლის ხაზის რელიეფურობას.

**Color** – უთითებს ხაზის ფერს. გამოიყენება RGB ფორმატი ან სტანდარტული სახელი.

მაგალითი:

```
<html>
<head>
<Title> My students </Title></head>
<Body>
<H1 ALIGN = left>My Students' names </H1>
<Br>
<H2 ALIGN = LEFT> Girls </H2>
Nino <Br>
Maiko <Br>
Shorena <Br>
<HR width = 50% size = 6 ALIGN = left color = RED>
<H2 ALIGN = LEFT> Boys </H2>
Levani <br>
Dato <br>
Rezo <br>
<HR width = 50% size = 6 ALIGN = left color =Navy>
<Body>
</html>
```

მიღებულ დოკუმენტს ექნება შემდეგი სახე:

# My Students' names

## Girls

Nino  
Maiko  
Shorena

---

## Boys

Levani  
Dato  
Rezo

---

### 7. წინასწარ დაფორმატებული ტექსტის გამოყენება

ყოველთვის არ არის აუცილებელი აბზაცისა და სტრიქონის გაწყვეტის ტეგების გამოყენება. HTML-ს აქვს სპეციალური ტეგ-კონტეინერი <PRE>, სადაც შეიძლება ჩაისვას ტექსტი, რომელსაც წინასწარ აქვს მიცემული სასურველი ფორმა. ამ კონტეინერის გამოყენების მნიშვნელოვანი მხარეა პროგრამული კოდის (C, C++ და ა.შ.) დიდი ბლოკების ეკრანზე მარტივად გამოტანის საშუალება. ეს ტეგი ასევე დაგეხმარებათ ლექსების პუბლიკაციის დროსაც.

დავუშვათ, ბრაუზერის ფანჯარაში უნდა მივიღოთ:

```
tea( etRastr, hh, ww, jjj);  
}  
while(StartAbsc != 0 || StartOrd != 0);  
  StartofLine = memEndLine + 15;  
  EndofLine = height_pix;  
  start = 0;  
  memmemEndLine=memEndLine;  
  if(k) break;  
}
```

ეს ამოცანა ძალიან მარტივად წყდება:

```

<html>
<head>
<title> code of program </title>
</head>
<pre>
tea (etRastr, hh, ww, jjj);
    }
    while(StartAbsc != 0 || StartOrd != 0);
    StartofLine = memEndLine + 15;
    EndofLine = height_pix;
    start = 0;
    memmemEndLine=memEndLine;
    if(k) break;
    }
</pre>
</body>
</html>

```

## 8. შრიფტის დაფორმატება

განვიხილოთ შრიფტის დაფორმატებისთვის საჭირო ტეგები:

<B> მუქი შრიფტი </B>;

<I> დახრილი შრიფტი </I>;

<S> გადახაზული ტექსტი </S>;

<U> ხაზგასმული ტექსტი </U>;

<TT> საბეჭდი მანქანის შრიფტის მსგავსი შრიფტი </TT>;

<SUB>ქვედა ინდექსი </SUB>;

მაგ.:  $2\text{H}_{2} + \text{O}_{2} = 2\text{H}_{2}\text{O}$ ; ბრაუზერის ფანჯარაში  
მიიღება  $2\text{H}_2 + \text{O}_2 = 2\text{H}_2\text{O}$ .

<Sup> ზედა ინდექსი </Sup>; მაგალითად, პითაგორას თეორემა:

$a^2 + b^2 = c^2$

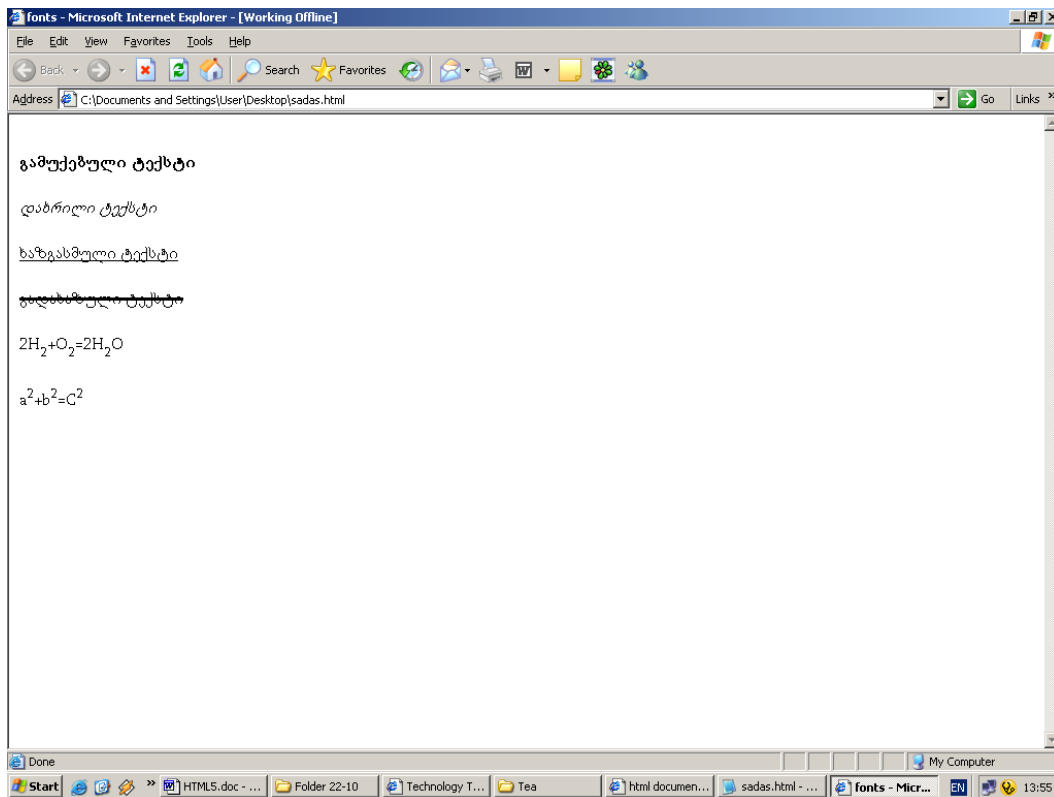
<BIG> დიდი ზომის შრიფტი </BIG>;

<SMALL> მცირე ზომის შრიფტი </SMALL>;

მაგალითი:

```
<html>
<head>
<title>fonts</title>
</head>
<body bgcolor=#FFFFFF>
<br><font face="Sylfaen"><b>გამუქებული ტექსტი</b><br>
<br><font face="Sylfaen"><i>დახრილი ტექსტი</i><br>
<br><font face="Sylfaen"><u>ხაზგასმული ტექსტი</u><br>
<br><font face="Sylfaen"><s>გადახაზული ტექსტი</s><br>
<br> 2H<Sub>2</Sub>+O<Sub>2</sub>=2H<Sub>2</Sub>O</br>
<br>a<Sup>2</Sup>+b<Sup>2</Sup>=C<Sup>2</Sup></br>
</body>
</html>
```

მივიღებთ:



## 9. ტეგი Basefont

ტეგი <Basefont> განსაზღვრავს იმ ძირითად შრიფტს, რომლითაც უნდა აისახოს ტექსტი ბრაუზერის ფანჯარაში. შემდგომში თქვენ ადვილად შეგიძლიათ დოკუმენტის ნებისმიერ ნაწილში შეცვალოთ შრიფტი ტეგ <font>-ის გამოყენებით. დახურვის ტეგ </font>-ის შემდეგ <Basefont>-ის მოქმედება აღდგება. ტეგ Basefont-ის ატრიბუტების მნიშვნელობა შეიძლება შეიცვალოს დოკუმენტის ნებისმიერ ადგილას ახალი Basefont ტეგით. <Basefont>-ის მოქმედება არ ვრცელდება ტექსტზე, რომელიც ცხრილებში ან html-ის სხვა განცალკევებულ ელემენტებშია მოქცეული. <Basefont>-ს არა აქვს შესაბამისი </Basefont> ტეგი.

პარამეტრები	აღწერა
size	განსაზღვრავს შრიფტის ბაზურ ზომას, შესაძლო მნიშვნელობებია მთელი რიცხვები 1-დან 7-ის ჩათვლით.
face	განსაზღვრავს შრიფტის სახეობას.

## 10. Font

ტეგი <Font> საშუალებას გვაძლევს შევცვალოთ შრიფტის ფერი, ზომა ან ტიპი. <Font> და </Font> ტეგების გარეთ გამოიყენება ფონტი, რომელიც მიეთითება ელემენტ <Basefont>-ით.

პარამეტრები	აღწერა
size	განსაზღვრავს შრიფტის ზომას. შესაძლო მნიშვნელობები: ა) მთელი რიცხვები 1-დან 7-ის ჩათვლით. ბ) ფარდობითი ზომა განისაზღვრება ბაზურ ზომასთან მიმატების გზით, რომელიც განისაზღვრება Basefont size-ით.
face	განსაზღვრავს შრიფტის სახეობას.
Color	განსაზღვრავს ტექსტის ფერს.

Face პარამეტრი ემსახურება იმ შრიფტის ტიპის მითითებას, რომელიც უნდა გამოჩნდეს ბრაუზერის ფანჯარაში (თუ კომპიუტერში არსებობს ასეთი სახის შრიფტი). შესაძლებელია როგორც ერთი, ასევე რამდენიმე შრიფტის ერთდროულად მითითება. ეს ძალიან მნიშვნელოვანი თვისებაა. რადგან სხვადასხვა სისტემაში

შეიძლება იყვნენ თითქმის იდენტური შრიფტები განსხვავებული სახელწოდებებით. სხვა მნიშვნელოვან თვისებას წარმოადგენს რომელიმე შრიფტის გამოყენებისადმი უპირატესობის მინიჭება. შრიფტების სია განიხილება მარცხნიდან მარჯვნივ. თუ მომხმარებლის კომპიუტერზე არ არის სიაში მითითებული პირველი შრიფტი, მაშინ იწყება მოძებნა შრიფტის ძებნა და ა. შ. თუ ასეთი შრიფტი არ მოიძებნა, მაშინ მოცემული ბრძანება იგნორირებული იქნება და გამოიყენება შრიფტი, რომელიც დაყენებულია კომპიუტერში სტანდარტულად. მითითებული შრიფტების რაოდენობა არ უნდა აჭარბებდეს სამს.

მაგალითი:

```
<html>
<head><title> fonts </title>
</head>
<body>
There is a text
<Br>
<font face = "Verdana", "Arial", "Helvetica">
Sample text
</font>
</body>
</html>
```

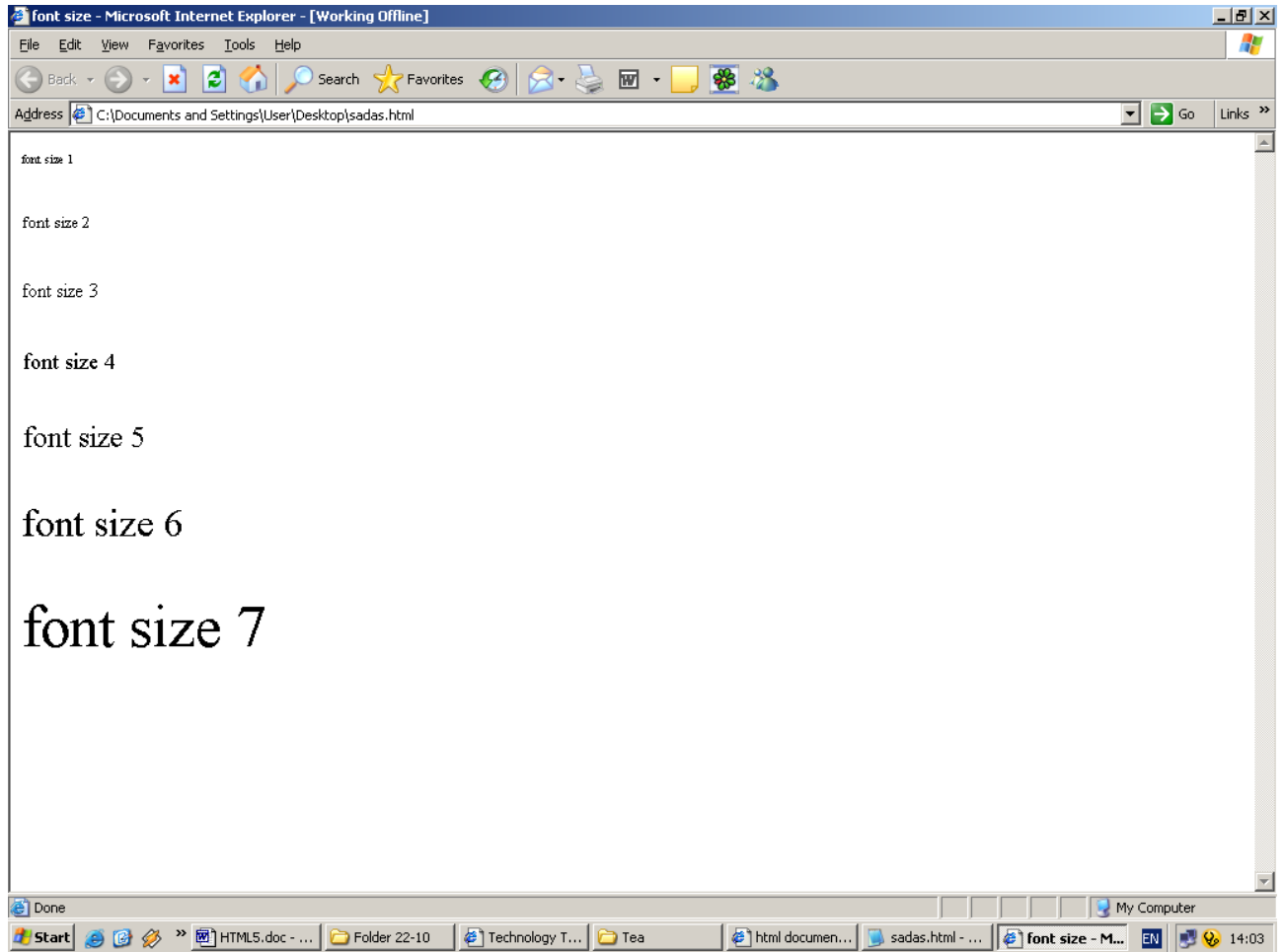
Size პარამეტრი გამოიყენება შრიფტის ზომების მისათითებლად, პირობით ერთეულებში 1-დან 7-ის ჩათვლით. ითვლება, რომ შრიფტის ზომის ნორმალური მნიშვნელობაა 3. შრიფტის ზომა მიეთითება როგორც აბსოლუტური (size = 2) ასევე ფარდობითი სიდიდის სახით (size = + 1)

მაგალითი:

```
<html>
<head><title> font size </title>
</head>
<body>
<font size = 1> font size 1 </font><br><br><br>
<font size = 2> font size 2 </font><br><br><br>
<font size = 3> font size 3 </font><br><br><br>
<font size = 4> font size 4 </font><br><br><br>
```

```
<font size = 5> font size 5 </font><br><br><br>
<font size = 6> font size 6 </font><br><br><br>
<font size = 7> font size 7 </font><br><br><br>
</body>
</html>
```

მიღებულ შედეგს აქვს შემდეგი სახე:



## 11. სიები

დანომრილი სია იქმნება ტეგ <ol>-ის დახმარებით: <ol>...</ol>

```
მაგალითი: <html>
    <head>
    <title> list of people </title>
    </head>
    <body>
    <ol>
    <li> Vano
    <li> Dato
    <li> Elene
    </ol>
    </body>
</html>
```

ბრაუზერის ფანჯარაში მიღებულ HTML დოკუმენტს ექნება შემდეგი სახე:

1. Vano
2. Dato
3. Elene

</ol>- ის ატრიბუტებია start და type:

type = A            ნუმერაციის ტიპი (A,B,C და ა.შ).

type = a            ნუმერაციის ტიპი (a, b, c...)

type = I            ნუმერაციის ტიპი (I, II, III...)

type = i            ნუმერაციის ტიპი (i, ii, iii...)

type = 1            ნუმერაციის ტიპი (1, 2, 3...)

Start = n            ნუმერაცია იწყება n-იდან

მარკირებული სია იქმნება ტეგ <ul>-ის საშუალებით: <ul>...</ul>

```
მაგალითი: <html>
    <head>
    <title> List </title></head>
    <body>
    <ul>
    <li> Programming
    <li> Algorithm
    <li> Design
```

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

შედეგად მიიღება მარკირებული სია:

- Programming
- Algorithm
- Design

<ul> მარკერს გააჩნია ატრიბუტი type, რომელიც საშუალებას იძლევა წრის ნაცვლად სიაში ჩასვას კვადრატი:

```
<ul type = square>
<li> first element </li>
<li> second element </li>
</ul>
```

განსაზღვრებათა სია იწყება <DL> ტეგით და მთავრდება დახურვის </DL> ტეგით. თითოეული ტერმინი იწყება <DT>-თი, აღწერა კი <DD>თი.

```
<DL>
<DT> I ტერმინი
<DD> I განსაზღვრება
<DT> II ტერმინი
<DD> II განსაზღვრება
...
</DL>.....
```

ხშირად განსაზღვრებათა სიას იყენებენ ოდნავ მარჯვნივ ტექსტის გასაწვეად. მაგალითი:

```
<html>
<head>
<title> List and definition
</title>
</head>
<body>
<DL>
<DT> A
<DD> first symbol of alphabet
<DT> B
```

```
<DD> Second symbol of alphabet
<DT> C
<DD> third symbol of alphabet
</DL>
</body>
</html>
```

ზემოთ მოცემული კოდის საშუალებით ბრაუზერის ფანჯარაში მიიღება შემდეგი სურათი:

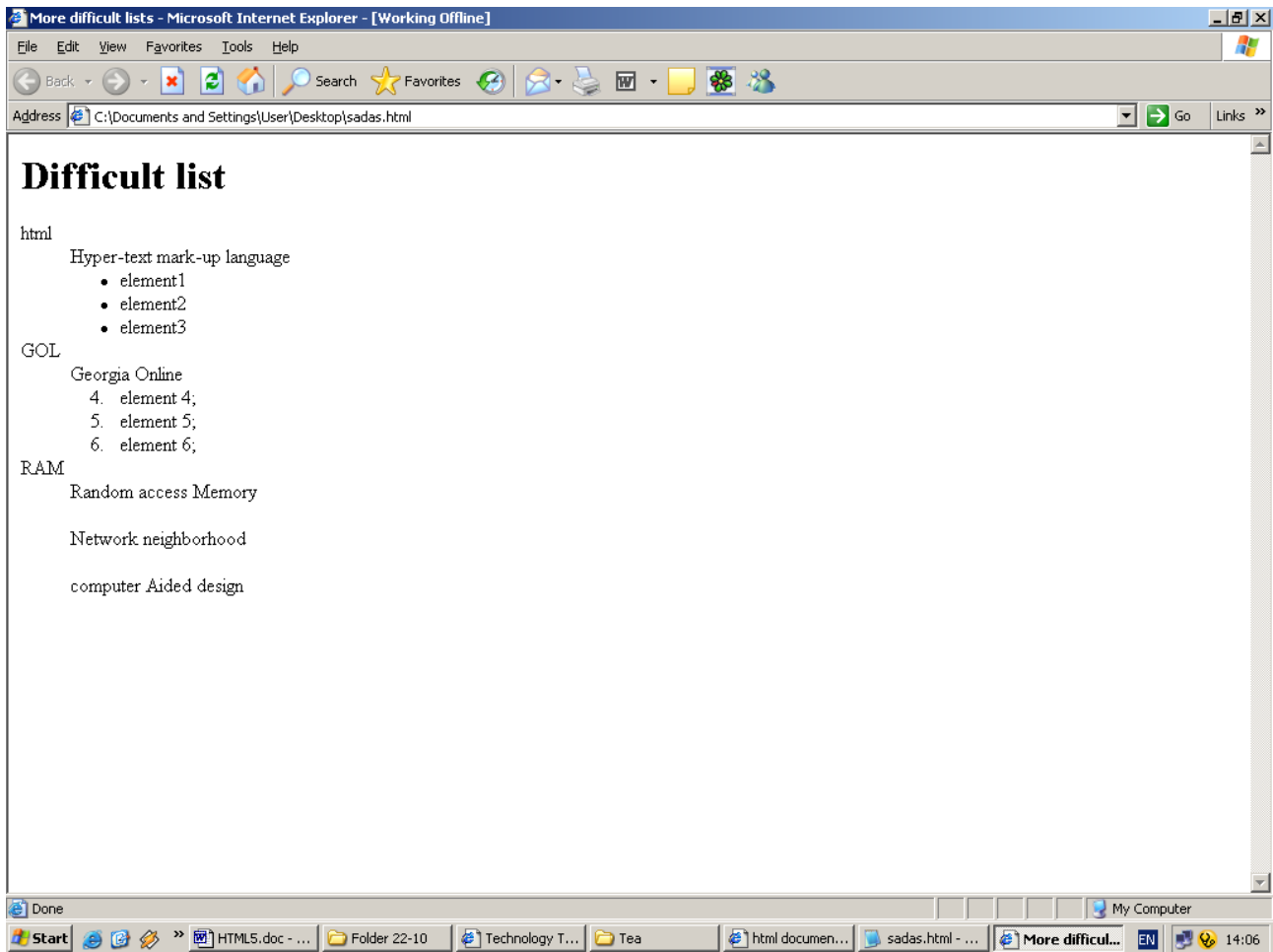
```
A      first symbol of alphabet
B      Second symbol of alphabet
C      third symbol of alphabet
```

ზემოთ განხილული სამივე ტიპის სიის ერთმანეთში ჩადგმა შესაძლებელია.

```
მაგალითი: <html>
<head>
<title> More difficult lists </title>
</head>
<body>
<H1> Difficult list </H1>
<dl>
<dt> html
<dd> Hyper-text markup language
<ul>
<li> element1
<li> element2
<li> element3
</li>
</ul>
<dt>GOL
<dd>Georgia Online
<ol start=4>
<li>element 4;
<li>element 5;
```

```
<li>element 6;
</ol>
<dt> RAM
<dd> Random access Memory
<p> Network neighborhood
<p> computer Aided design </p>
</dl>
<body>
</html>
```

ბრაუზერის ფანჯარაში მიღებულ შედეგს აქვს შემდეგი სახე:



## 12. გამოსახულება HTML დოკუმენტში

გამოსახულების ჩასასმელად HTML დოკუმენტში გამოიყენება ტეგი <img>:

```
<html>
<head>
<title> Image </title>
</head>
<body>
<H1> Image </H1>
<p><img src = "picture. gif" ALT = "This is picture" Height = 20 Width = 30 ></p>
</body>
</html>
```

img src = "Picture. gif"- დოკუმენტში სვამს გამოსახულებას, რომელიც მოცემულ HTML დოკუმენტთან ერთად ერთ საქალაქშია მოთავსებული.

Height = 20 უთითებს გამოსახულების სიმაღლეს პიქსელებში.

Width = 30 უთითებს გამოსახულების სიგანეს პიქსელებში.

გამოსახულების ზომების შესახებ ინფორმაციის მითითებით, გვერდის ავტორი აჩვენებს მისი ჩამოტვირთვის პროცესს, რაც მომხმარებელს საშუალებას აძლევს, დოკუმენტი უფრო ადრე დაინახოს. ბრაუზერი იყენებს height-ისა და width-ის მნიშვნელობებს, რათა გვერდზე დატვირთვას ადგილი გამოსახულებისათვის. როდესაც გამოსახულება ჩამოტვირთება, ის გამოჩნდება ამ რეზერვირებულ ადგილას. height-ისა და width-ის მნიშვნელობების მითითების გარეშე ბრაუზერს მოუწევდა მთლიანად გამოსახულების ჩატვირთვა, მერე დაიწყებდა მისი ზომების გამოთვლას, განათავსებდა ეკრანზე და მხოლოდ ამის შემდეგ დაიწყებდა ეკრანზე დანარჩენი ელემენტების მოთავსებას.

ტეგ <IMG>-ის ატრიბუტები:

ALT - გამოსახულების ტექსტური ალტერნატივა;

ALIGN – გამოსახულების გარშემო მოთავსებული ტექსტის გასწორება;

Height - გამოსახულების სიმაღლე პიქსელებში;

Width - გამოსახულების სიგანე პიქსელებში

Hspace - განსაზღვრავს გამოსახულების მარცხნიდან და მარჯვნიდან თავისუფალი ადგილის ზომას

Vspace - განსაზღვრავს გამოსახულების ზემოთა და ქვემოთა მხრიდან თავისუფალი ადგილის ზომას

SRC - უთითებს გამოსახულების URL-ს

Border - უთითებს გამოსახულების ჩარჩოს ზომას პიქსელებში.

ALIGN –ს შეუძლია მიიღოს ნებისმიერი ხუთი მნიშვნელობიდან: Left, Right, Top, Middle, Bottom.

LEFT- სურათი იქნება მარცხენა მხარეს, ტექსტი კი დაიკავებს დოკუმენტის დანარჩენ ადგილს სურათის მარჯვენა მხარეს.

RIGHT – სურათი იქნება მარჯვნივ, ტექსტი – მარცხნივ;

TOP – ტექსტი სურათის ზედა კიდის გასწვრივ;

MIDDLE – ტექსტი სურათის ცენტრის გასწვრივ;

BOTTOM – ტექსტი სურათის ქვედა კიდის გასწვრივ.

<img src = “giko.jpg” align=“left”> - ნიშნავს, რომ სურათი მოთავსებული იქნება, მარცხენა მხარეს. ტექსტი მას გარს შემოუვლის მარჯვნიდან. იმისათვის, რომ სურათი მოთავსდეს მარჯვნივ, ტექსტი კი – მარცხნივ, უნდა დაიწეროს:

<img src = “giko.fpg” align = “right”>

როგორც უკვე იცით, ერთი ტეგის ატრიბუტები, შეიძლება ერთდროულად იყოს გამოყენებული:

<img src = “tt.gif” align = “left” hspace = “30”, vspace = “5” , alt = “My picture”>

ეს ჩანაწერი აღნიშნავს, რომ სურათი მოთავსებული იქნება ეკრანის მარცხენა მხარეს. ტექსტი მას შემოუვლის მარჯვნიდან, დაშორება ტექსტამდე ჰორიზონტალურად - 30 პიქსელი, ხოლო ვერტიკალური მიმართულებით 5 პიქსელი იქნება. თუ თქვენ სურათთან მიიტანთ კურსორს, იქ გაჩნდება წარწერა: “My picture”.

თქვენი ცოდნის განსაზოგადებლად შექმენით html დოკუმენტი:

<html>

<head>

<title> html document

</title>

</head>

<body text = “#336699” bgcolor = “#000000”>

<H3 align = “center”> Hello, this is my html document</H3>

<br>

<font color = “#CC0000”> Welcome </font>

<p align=“justify”>

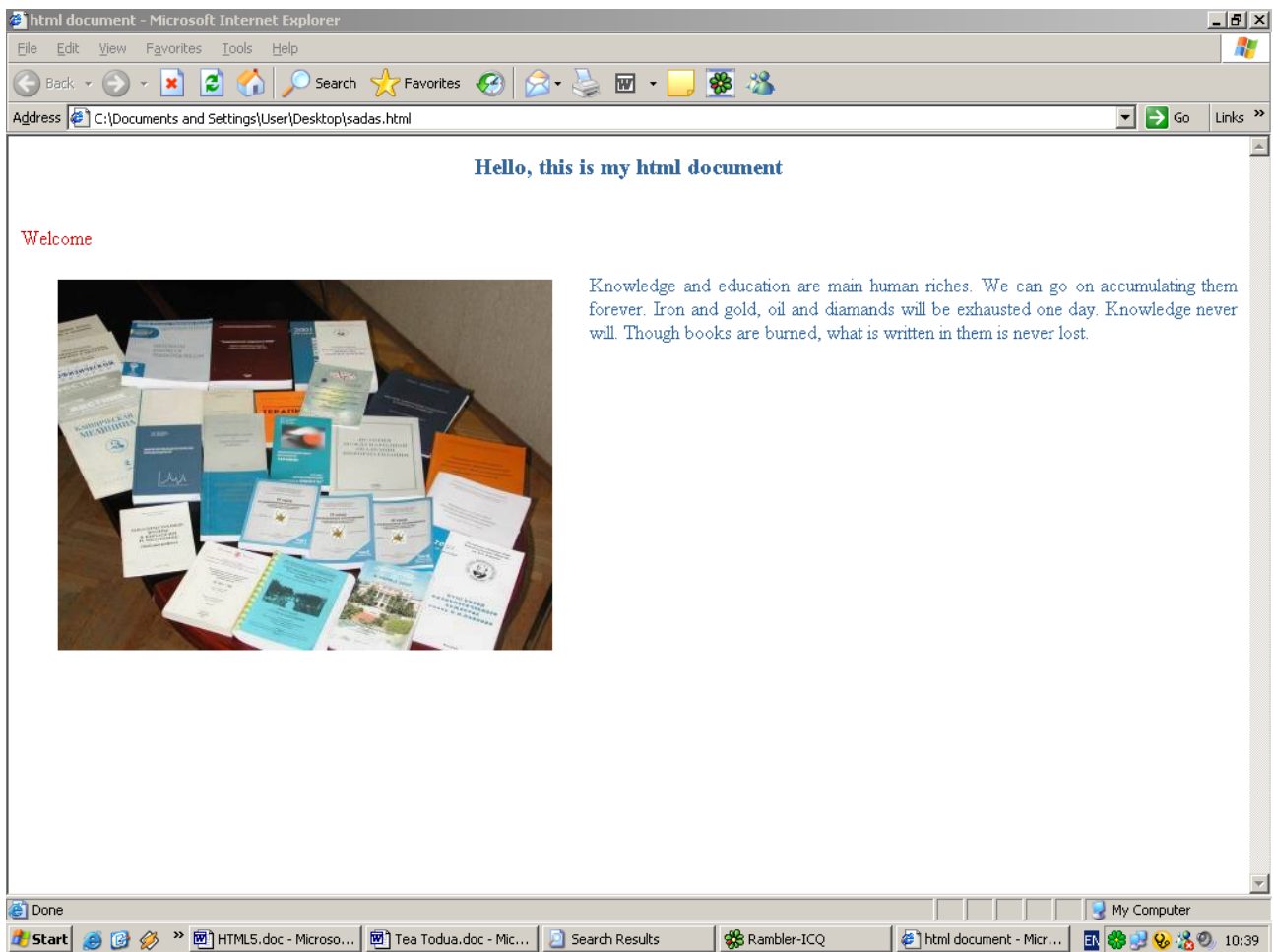
<img src = “tt.gif” align = “left” height=200 width=300 Hspace = 30 Vspace = 5 alt = “My picture”>

Knowledge and education are main human riches. We can go on accumulating them forever. Iron and gold, oil and diamonds will be exhausted one day. Knowledge never will. Though books are burned, what is written in them is never lost.

</p>

</body>

</html>



### 13. ბმული

ბმული საშუალებას გვაძლევს განვახორციელოთ გადასვლა ტექსტის ერთი ფრაგმენტიდან მეორეზე.

მაგალითი:

```
<html>
<head>
<title> Hypertext </title>
</head>
<body>
<a href = "h2. html"> About us </a>
<a href = "Tea / Lion.html"> Lion </a>
</body>
</html>
```

html-ში ტექსტის ერთი ფრაგმენტიდან მეორეში გადასვლა შეიძლება შემდეგი სახით:

```
<a href = “[ გადასვლის მისამართი]”> ტექსტის გამოყოფილი ფრაგმენტი </a>
```

პარამეტრის სახით [გადასვლის მისამართი] შეიძლება გამოყენებულ იქნეს არგუმენტების რამდენიმე ტიპი. ყველაზე მარტივია მიუთითოთ სხვა HTML დოკუმენტის სახელი, რომელზეც უნდა გადავიდეთ: <A href = “h2.html> About us </A> შედეგად ჩნდება ტექსტი About us, რომელზე დაჭერითაც ბრაუზერის ფანჯარაში ჩაიტვირთება h2.html დოკუმენტი. მიაქციეთ ყურადღება: თუ გადასვლის მისამართში არ არის მითითებული შესაბამისი საქალაქის სახელი, გადასვლა განხორციელდება მიმდინარე საქალაქის შიგნით. შეიძლება ბმულის ფორმატირება, მაგალითად:

```
<a href = “h2.html”> <h2> About us </h2> </A>
```

თქვენ შეგიძლიათ გადასვლა დოკუმენტზე, რომელიც მდებარეობს ნებისმიერ საქალაქში, დოკუმენტამდე სრული გზის მითითებით. ასე მაგალითად, გადასვლა ფაილზე Lion.html, რომელიც მოთავსებულია ფოლდერში Tea, ხორციელდება შემდეგნაირად:

```
<a href = “Tea /Lion.html”>Lion </a>
```

ზემოთ ჩვენ განვიხილეთ ბმულები მხოლოდ ჩვენს მიერ შექმნილ HTML დოკუმენტზე. ქვემოთ ნაჩვენებია, როგორ შეიძლება გადასვლა ჩვენს მიერ შექმნილ HTML დოკუმენტიდან უკვე არსებულ web-გვერდზე.

მაგალითი:

```
<html>
<head>
<title>
New links
</title>
</head>
<body>
<img src = "medi.gif">
<br><a href = "http://www.yahoo.com">Yahoo</a>
<br><a href = "http://www.rambler.ru"> Rambler </a>
</body>
</html>
```

გადასვლა საფოსტო ყუთში ხდება ოდნავ განსხვავებულად:

```
<a href = "mailto: tea_todua@yahoo.com"> My E-mail </a>
```

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

მაგალითი:

```
<html>
<head><title> Links</title></head>
<body>
<h1>Image</h2>
<p><a href = "h2.html"><img src = "picture.gif"></a></p>
</body>
</html>
```

ბმულზე მათხის მიტანისას შესაძლებელია გამოჩნდეს ინფორმაცია ბმულის შესახებ. ამ ფუნქციას ასრულებს ატრიბუტი title.

მაგალითი:

```
<html>
<head><title> xxxx</title></head>
<body>
<h2> Prompt </h2>
<a href = "index.html" title = Prompt>Hypertext </a>
</body>
</html>
```

ზოგჯერ საჭირო ხდება ბმულის გაკეთება არა სხვა დოკუმენტზე, არამედ იმავე დოკუმენტის შიგნით. ერთი დოკუმენტის შიგნით ამგვარი ნავიგაცია სშირად ძალიან მოხერხებულია. ამის გაკეთება შეიძლება ორი ხერხით.

I ხერხი - A ტეგის ატრიბუტ Name-ის დახმარებით:

მაგალითი:

```
<html>
<head>
<title> Link 1
</title>
</head>
<body>
<H2> politics </H2> <Br>
<a href = "#citation 1" > reference on the citation 1</a><Br>
<a href = "#citation 2"> reference on the citation 2</a><Br>
<a href = "#citation 3"> reference on the citation 3</a><Br>
<pre>
```

Humankind learns a lot from its history, but does it very slowly. So, unfortunately, do most people discussing politics.

Maybe we should discuss it in a calmer and more important way, influencing the progress mainly not by speeches,

demonstrations, wars, etc., but just by living and working honestly. That is the way common people make history.

And Their way is the most important one.

```
</pre>

<h3> <a name = "#citation 1"> Citation 1 </a> </h3>
```

```
<pre>
```

“Political ability is the ability to foretall what is going to happen tomorrow, next week, next month and next year. And to have the ability afterward to explain why it didn’t happen”.

Winston Churchil

```
</pre>

<h3><a name="#citation 2">Citation 2</a></h3>
<pre>
```

“Since the politician never believes what he says, he is surprised when others believe him.”

Charles de Gaulle

</pre>



<h3><a name="#citation 3">Citation 3</a></h3>

<pre>

“Political language – and with variations this is true of all political parties, from conservatives to anarchists – is designed to make lies sound truthfull. . . “

George Orwell

</pre>

</body>

</html>

თქვენ შეგიძლიათ ბმული გააკეთოთ სხვა დოკუმენტებიდანაც:

<a href="ancorpri.html#citation1">reference on the citation 1 from the another document</a>

მეორე ხერხი – ატრიბუტ id-ის დახმარებით. ყველაფერი იგივე რჩება, მხოლოდ ზემოთ მოყვანილ მაგალითში <h3> ტეგებთან ვწერთ:

<h3 id="#citation 1">Citation 1</h3>

<h3 id="#citation 2">Citation 2</h3>

<h3 id="#citation 3">Citation 3</h3>

თქვენ შეგიძლიათ გამოიყენოთ ნებისმიერი ამ ორი ხერხიდან.

არასოდეს არ მიუთითოთ ამგვარად:

<a href="#citation1">Reference on the citation 1</a>

<a name="CITATION 1">Citation1</a>

ბრაუზერმა თქვენს მიერ სხვადასხვა რეგისტრში დაწერილი Citation 1 და CITATION 1 შეიძლება აღიქვას როგორც ორი სხვადასხვა სახელი, ამიტომ შეცდომებისაგან თავის დაზღვევის მიზნით სახელები დაწერეთ ერთ რეგისტრში.

## 14. ცხრილები

ცხრილების შექმნის პრინციპი მდგომარეობს შემდეგში: web-გვერდზე იქმნება ცხრილი, უჯრების უზინარი საზღვრებით და ელემენტები, რომლებიც მოითხოვენ ზუსტ პოზიციონირებას, განლაგდებიან ცხრილის უჯრებში. თითოეული უჯრისთვის თქვენ შეგიძლიათ მიუთითოთ ფორმატირების საკუთარი პარამეტრები. შესაბამისად, ფორმატირების ბრძანებები მოქმედებენ მხოლოდ უჯრის საზღვრებში. პრაქტიკაში ხშირად გვჭირდება ტექსტი რამდენიმე სვეტად დავეყოთ. ცხრილები სწორედ ამის გაკეთების საშუალებას იძლევა. გარდა ამისა, ცხრილს, რომელიც ერთი უჯრისგან შედგება, შეუძლია ძალზე ეფექტურად გამოყოს ტექსტის ფრაგმენტი, რომელზეც მკითხველის ყურადღების მიპყრობაა საჭირო.

ცხრილი იწყება `<table>` ტეგით და მთავრდება `</table>`-ით. `table`-ს აქვს რამდენიმე ატრიბუტი:

`Align` - განსაზღვრავს ცხრილის ადგილმდებარეობას დოკუმენტის ველებთან მიმართებაში: `Align = left` (გასწორება მარცხენა მხრიდან); `align=center` (ცენტრირება). `align = right` (გასწორება მარჯვენა მხრიდან).

`Width` - განსაზღვრავს ცხრილის სიგანეს პიქსელებში ან პროცენტებში ბრაუზერის ფანჯრის სიგანისგან დამოკიდებულებით. (მაგ. `width = 400` ან `width = 80%`)

`Height` - განსაზღვრავს ცხრილის სიმაღლეს პიქსელებში ან პროცენტებში.

`Border`- აყენებს ცხრილის ჩარჩოს სისქეს პიქსელებში. მაგ. `Border =2`. თუ ეს ატრიბუტი არ არის მითითებული, მაშინ ცხრილი ჩნდება ჩარჩოს გარეშე.

`Cellspacing` - განსაზღვრავს უჯრებს შორის მანძილს პიქსელებში.

`Cellpadding` – განსაზღვრავს მანძილს პიქსელებში უჯრის ჩარჩოსა და უჯრაში მოთავსებულ ელემენტს (ტექსტი, სურათი) შორის. (მაგ. `Cellpadding = 10`)

`bgcolor` - ცხრილის ფონის ფერი

`background (Background=eli.jpg)` - ეს ატრიბუტი გამოიყენება ცხრილის ფონად რაიმე სურათის ჩასასმელად.

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

```
<table>
<tr> <!... I უჯრის სტრიქონის აღწერა></tr>
<tr> <!... II უჯრის სტრიქონის აღწერა></tr>
</table>
```

ყველაფერი, რაც ჩვენს ცხრილში უნდა გამოჩნდეს, იწერება <td>-სა და </td>-ს შორის; მაგალითად, ქვემოთ მოყვანილი კოდი საშუალებას გვაძლევს მივიღოთ ერთუჯრიანი ცხრილი:

```
<Html>
<head><title>Table</title></head>
<table border=1>
<tr>
<td>ერთუჯრიანი ცხრილი</td>
</tr>
</table>
</body>
</html>
```

სამუჯრიანი ერთსტრიქონიანი ცხრილის მისაღებად უნდა დავწეროთ:

```
<html>
<head><title>Table</title></head>
<table border=1>
<tr>
<td>პირველი უჯრა</td>
<td>მეორე უჯრა</td>
<td>მესამე უჯრა</td>
</tr>
</table>
</body>
</html>
```

შევექმნათ სამი სტრიქონისა და ოთხი სვეტისაგან შემდგარი ცხრილი:

```
<html>
<head><title>Table</title></head>
<table border=1>
<tr>
<td>September</td>
<td>October</td>
<td>November</td>
<td>December</td>
</tr>
```

```

<tr>
<td>January</td>
<td>February</td>
<td>March</td>
<td>April</td>
</tr>
<tr>
<td>May</td>
<td>June</td>
<td>July</td>
<td>August</td>
</tr>
</table>
</body>
</html>

```

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

tr-ს შეიძლება ჰქონდეს რამდენიმე ატრიბუტი:

Align = left უჯრაში ასწორებს ტექსტს მარცხენა მხრიდან; align = center (ცენტრირება), align = right (ტექსტის გასწორება მარჯვენა მხრიდან).

Valign - მიუთითებს ტექსტის ვერტიკალურ მდებარეობაზე.

valign = top (ტექსტს უახლოებს უჯრის ზედა ნაწილს)

valign = middle (ტექსტი უჯრის შუაში, ცენტრში)

valign = Bottom (ტექსტი უჯრის ქვედა კიდეში)

bgcolor - აყენებს ცხრილის ფერს, მოცემული სტრიქონისათვის.

<td>-ს აქვს შემდეგი ატრიბუტები:

colspan - შეიძლება უჯრის სიდიდის გაზრდა ჰორიზონტალურად.

მაგ: colspan = 3 ნიშნავს, რომ უჯრის სვეტის სიგანე 3-ჯერ გაიზარდა.

Rowspan - Rowspan =2 ნიშნავს, რომ უჯრა იკავებს ორ სტრიქონს.

Align - უჯრაში ტექსტის განლაგებას ცვლის. დასაშვები მნიშვნელობებია:

Align = left; Align=right; Align=center.

valign - მიუთითებს ტექსტის ვერტიკალურ მდებარეობაზე

valign = top (ტექსტი მიახლოებულია უჯრის ზედა კიდესთან);

valign = middle (ტექსტის ცენტრში დაყენება)

valign = bottom (ტექსტი უჯრის ქვედა კიდეშია)

Width - უჯრის სიგანე პიქსელებში (მაგ: width = 200)

Height - უჯრის სიმაღლე პიქსელებში

bgcolor - აყენებს ცხრილის ფერს მითითებული უჯრისათვის.

უფრო დაწვრილებით განვიხილოთ Colspan და Rowspan ატრიბუტები.

მაგალითი:

```
<html>
<head><title>Table</title></head>
<body>
<table border="1" align="center">
<tr align="center">
<td width="300" colspan="3">1.1</td>
</tr>
<tr align="center">
<td>2.1</td>
<td>2.2</td>
<td>2.3</td>
</tr>
<tr align="center">
<td>3.1</td>
<td>3.2</td>
<td>3.3</td>
</tr>
</table>
</body>
</html>
```

მიიღება ცხრილი, რომლისთვისაც მითითებულია ატრიბუტი colspan=3, რის შედეგადაც ცხრილის პირველი სტრიქონი მოიცავს სამ სვეტს:

1.1		
2.1	2.2	2.3
3.1	3.2	3.3

ქვემოთ მოცემული ცხრილის მისაღებად კოდი უნდა დაიწეროს ასე:

```

<html>
<head><title>Table</title></head>
<body>
<table border="1" align="center">
<tr align="center">
<td width="100" Rowspan="2">1.1</td>
<td width="100">1.2</td>
<td width="100">1.3</td>
<td width="100">1.4</td>
</tr>
<tr align="center">
<td>2.1</td>
<td>2.2</td>
<td>2.3</td>
</tr>
</table>
</body>
</html>

```

1.1	1.2	1.3	1.4
	2.1	2.2	2.3

ცხრილს შეიძლება ჰქონდეს სათაური, ეს არ არის აუცილებელი ელემენტი. ცხრილის სათაური თავსდება <caption> და </caption> ტეგებს შორის. Caption-ს შეიძლება ჰქონდეს ატრიბუტი align: <caption align=top> (სათაური ცხრილის ზემოთ) და <Caption align=bottom> (სათაური თავსდება ცხრილის ქვემოთ). ცხრილის შიგნით სათაურის მითითება ხდება <th> . .</th> ტეგების საშუალებით. იგი td ტეგისგან განსხვავდება მხოლოდ იმით, რომ ტექსტი უფრო მსხვილი შრიფტით გამოჰყავს.

შევქმნათ უფრო რთული ცხრილი:

<b>ცხრილის სათაური</b>			
<b>ცხრილის ქვესათაური</b>			
რთული ცხრილი	1.1	1.1	რთული ცხრილი
	1.1	1.1	
	1.1	1.1	
	1.1	1.1	
ეს ცხრილის ბოლო სტრიქონია			

ასეთი ცხრილის მისაღებად უნდა დაიწეროს:

```
<html>
<head><title>Table</title></head>
<body>
<table border="1" align="center" width="400">
<caption align="top"><b>ცხრილის სათაური</b></caption>
<tr align="center">
<th colspan="4">ცხრილის ქვესათაური</th>
</tr>
<tr align="center">
<td rowspan="4">რთული ცხრილი</td>
<td>1.1</td>
<td>1.1</td>
<td rowspan="4">რთული ცხრილი</td>
</tr>
<tr align="center">
<td>1.1</td>
<td>1.1</td>
</tr>
<tr align="center">
<td>1.1</td>
<td>1.1</td>
</tr>
<tr align="center">
<td>1.1</td>
<td>1.1</td>
</tr>
<tr align="center">
<td colspan="4">ეს ცხრილის ბოლო სტრიქონია</td>
</tr>
</table>
</body>
</html>
```

## 15. ცხრილების მაგალითები

მაგალითი I. 3 სტრიქონისა და 4 სვეტისგან შემდგარი ცხრილი:

```
<html>
<head><title>Tables</title></head>
<body>
<table>
<tr> <th> one </th> <th> two </th> <th> three </th><th> four </th> </tr>
<tr> <td> 1.1 </td> <td>1.2 </td> <td> 1.3</td> <td> 1.4 </td> </tr>
<tr> <td> 2.1 </td> <td> 2.2 </td> <td> 2.3</td> <td> 2.4 </td> </tr>
</table>
</body>
</html>
```

შედეგად ბრაუზერის ფანჯარაში მიიღება:

<b>One</b>	<b>two</b>	<b>three</b>	<b>four</b>
1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4

მაგალითი II.

```
<html>
<head>
<title> table2 </title>
</head>
<body>
<H1> უმარტივესი ცხრილი </H1>
<table border = 1> <!-- ცხრილის დასაწყისი -->
<Caption> <!-- ცხრილის სათაური -- >
ცხრილი
</Caption>
<tr> <!-- I სტრიქონის დასაწყისი -- >
<td> <!-- I უჯრის დასაწყისი -- >
პირველი სტრიქონი, პირველი სვეტი
</td> <!-- I უჯრის დასასრული -- >
<td> <!-- II უჯრის დასაწყისი -- >
```

პირველი სტრიქონი, მეორე სვეტი

```

</td> <!-- II უჯრის დასასრული -- >
</tr> <!-- I სტრიქონის დასასრული-- >
<tr> <!-- II სტრიქონის დასაწყისი -- >
<td>
მეორე სტრიქონი, პირველი სვეტი
</td>
<td>
მეორე სტრიქონი, მეორე სვეტი
</td>
</tr>
</table>
</body>
</html>

```

შედეგად ბრაუზერის ფანჯარაში მიიღება:

## უმარტივესი ცხრილი

ცხრილი	
პირველი სტრიქონი, პირველი სვეტი	პირველი სტრიქონი, მეორე სვეტი
მეორე სტრიქონი, მეორე სვეტი	მეორე სტრიქონი, პირველი სვეტი

მაგალითი III. : <Html>

```

<head>
<title> table3</title></head>
<body> <font face = "Sylfaen">
<table border =10 width = 400 align = center bordercolor=red>
<tr>
<td> ამ ცხრილისთვის მითითებულია პარამეტრი Border=10 </td>
</tr>
</table>
</font>
</body>
</html>

```

#### მაგალითი IV:

```
<html>
<head>
<title> table4</title>
</head>
<body>
<font face = "Sylfaen">
<table border = 5 cellspacing = 0 width = 400 align = center Bgcolor = "cyan">
<tr align = center>
<td width = 10%> უჯრის სიგანე მთლიანად ცხრილის 10%-ს შეადგენს</td>
<td width = 60% > უჯრის სიგანე მთლიანად ცხრილის 60%-ია</td>
<td width = 30%> უჯრის სიგანე მთლიანად ცხრილის 30%-ია</td>
</tr>
<tr>
<td align = center colspan = 3> ამ უჯრისთვის მიეთითა პარამეტრი colspan = 3
</td>
</tr>
</table>
</font>
</body>
</html>
```

#### მაგალითი V:

```
<html>
<head>
<title> table5 </title></head>
<body>
<font face = "Sylfaen">
<table border = 5 cellspacing = 0 width = 400 Align = center Bgcolor = "cyan">
<tr align = center>
<td width = 30% Rowspan = 2 Bgcolor = "00FF00">
ამ უჯრისთვის მითითებულია Rowspan = 2 და width = 30% </td>
<td> ამ უჯრის სიგანე არ არის მითითებული, ამიტომ ის იკავებს
დარჩენილ ნაწილს, ე.ი. 70%-ს</td>
</tr>
```

```

<tr align = center>
<td> ეს უჯრა კი წარმოქმნილია იმ უჯრის სიგანისგან დამოკიდებულებით,
რომელიც მის ზემოთ მდებარეობს </td>
</tr>
</table>
</font>
</body>
</html>

```

მაგალითი VI:

```

<html>
<head>
<title>table6 </title>
</head>
<body>
<table>
<tr>
<td height = "35" bgcolor = "#FFCC33" colspan = "2">
<center> 1×1 </center> </td>
<td width = "50" bgcolor = "#336699" rowspan = "2">
<center> 1×2 </center> </td>
</tr>
<tr>
<td height = "35" width = "50" bgcolor = "#336699">
<center> 2×1 </center> </td>
<td width = "50" bgcolor = "#FFCC33"> <center> 2×2</center > </td>
</tr>
</table>
</body>
</html>

```



მაგალითი VII:

```
<html>
<head>
<title> table7 </title>
</head>
<body>
<table border = "0" cellspacing = "0" height = "100% " width = "100%">
<td bgcolor = "#FFFF80" align = "center"> <small>
<strong> <p align = "center"> <font face = "Arial">
<a href = "index 1. html"> Main </a> </font> </strong></small> </td></tr>
<tr>
<td bgcolor = "#66FFCC" align = "center"><small> <strong>
<p align = "center"> <font face = "Arial">
<a href = "rg. html"> Registration </a> </font> </strong>
</small> </td></tr>
<tr>
<td align = "center" bgcolor = "#FFFF80"> <small><strong>
<p align = "center"> <a href = "tsr. html"> <font face = "Arial">
<a href = "rv. html"> Links </a> </font> </strong> </small>
</td></tr>
<tr>
<td bgcolor = "#66FFCC" align = "center"> <small> <strong>
<p align = "center"> <font face = "Arial">
<a href = "r. html"> Forum </a> </font> </strong> </small></td></tr>
</table>
</body>
</html>
```

მაგალითი VIII:

ქვემოთ განხილულია ცხრილის მონაცემების ვერტიკალური მდებარეობის საკითხი. ცხრილის მონაცემების ვერტიკალურ მდებარეობას განსაზღვრავს ატრიბუტი `valign = "middle"` ("top", "bottom"). კონკრეტული უჯრის მონაცემი მოთავსებული იქნება უჯრის შუაში, ზემოთ ან ქვემოთ. კოდი და მისი შესაბამისი შედეგი, ნაჩვენებია ქვემოთ.

```

<html>
<head> <title> table8 </title> </head>
<body>
<table>
<tr>
<td height = "35" width = "50" bgcolor = "#FFCC33" valign = "top"> <center> 1×1
</center> </td>
<td width = "50" bgcolor = "#336699"> <center> 1×2 </center> </td>
<td width = "50" bgcolor = "FFCC33" valign = "bottom"> <center> 1×3</center>
</td></tr>
<tr>
<td height = "35" width = "50" bgcolor = "#336699" valign = "bottom">
<center> 2×1 </center> </td>
<td width = "50" bgcolor = "FFCC33"> <center> 2×2 </center> </td>
<td width = "50" bgcolor = "#336699" valign = "top"> <center> 2×3 </center> </td>
</tr>
</table>
</body>
</html>

```

1×1	1×2	1×3
2×1	2×2	2×3

მაგალითი IX:

```

<html>
<head> <title> table9 </title> </head>
<body>
<table border = "3" bordercolor = blue bgcolor = red>
<tr> <td> 1-one </td> <td> 2-two </td> <td> 3-three </td> </tr>
<tr> <td> 4-four </td> <td> 5-five </td> <td> 6-six </td> </tr>
<tr> <td> 7-seven</td> <td> 8-eight </td> <td> 9-nine </td> </tr> </table>

```

Changing the entire tables color

```

<hr>
<table border><tr bordercolor = yellow bgcolor =green>

```

```
<td> 1-one </td><td> 2-two </td> <td> 3-three </td> </tr>
<tr> <td> 4-four </td> <td> 5-five </td> <td> 6-six </td> </tr>
<tr> <td> 7-seven</td> <td> 8-eight </td> <td> 9-nine </td> </tr> </table>
```

Changing a single rows color

```
<hr>
<table border> <tr><td bordercolor = red bgcolor = yellow> 1-one </td>
<td> 2-two </td> <td> 3-three </td> </tr>
<tr> <td> 4-four </td> <td> 5-five </td> <td> 6-six </td> </tr>
<tr> <td> 7-seven</td> <td> 8-eight </td> <td> 9-nine </td> </tr> </table>
```

Changing a single cells color

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

მაგალითი X: ქვემოთ განხილულ მაგალითში, ცხრილი გამოყენებულია ტექსტის განსათავსებლად გამოსახულების გასწვრივ.

```
<html>
<head><title> table borders </title></head>
<body>
<table>
<tr><td <img src = "georgia. gif"> </td>
<td>
<p align="justify">
```

Two Georgian Kingdoms of late antiquity, Iberia, in the east of the country and Colchis in the west, were among the first nations in the region to adopt Christianity (317 AD and 523 AD, respectively). Egrisi often saw battles between rivals Persia and the Byzantine Empire, both of which managed to conquer Western Georgia from time to time. As a result, those Kingdoms were disintegrated into various feudal regions in the early Middle Ages. This made it easy for Arabs to conquer Georgia in the 7<sup>th</sup> century. The rebellious regions were liberated and united into the Georgian Kingdom at the beginning of the 11<sup>th</sup> Ccentury. Starting in the 12<sup>th</sup> century the rule of Georgia extended over the significant part of Southern Caucasus, including northeastern parts and almost entire northern coast of what is now Turkey.</p>


```
</td></tr>
</table>
</body>
</html>
```

table borders - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail Internet Options

Address C:\Documents and Settings\User\Desktop\sadas.html Go Links



Two Georgian Kingdoms of late antiquity, Iberia, in the east of the country and Colchis in the west, were among the first nations in the region to adopt Christianity (317 AD and 523 AD, respectively). Egrisi often saw battles between rivals Persia and the Byzantine Empire, both of which managed to conquer Western Georgia from time to time. As a result, those Kingdoms were disintegrated into various feudal regions in the early Middle Ages. This made it easy for Arabs to conquer Georgia in the 7th century. The rebellious regions were liberated and united into the Georgian Kingdom at the beginning of the 11th Century. Starting in the 12th century the rule of Georgia extended over the significant part of Southern Caucasus, including northeastern parts and almost entire northern coast of what is now Turkey.

Done

Start HTML.doc - Mi... Dictionary - Mic... My Computer Image:Gelati.jp... table borders ... sadas.html - No... 14:44

მაგალითი XI: ამ მაგალითის საშუალებით შეძლებთ ცხრილებთან დაკავშირებული ცოდნის განზოგადებას:

```

<html>
<head>
<title> Page with tables </title>
</head>
<body bgcolor = "#FFFFFF" text = "#000000">
<table cellspacing = "5">
<tr>
<td height = "35" bgcolor = "#FFCC33" colspan = "2"> <center> 1×1 </center> </td>
<td width = "50" bgcolor = "#336699" rowspan = "2"> <center> 1×2 </center> </td>
</tr>
<tr>
<td height = "35" width = "50" bgcolor = "#336699"> <center> 2×1 </center> </td>
<td width = "50" bgcolor = "#FFCC33"> <center> 2×2 </center> </td>
</tr>

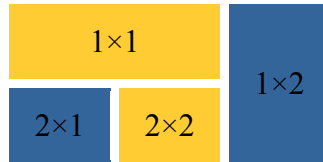
```

```

</table>
</body>
</html>

```

შედგებად მიიღება ასეთი სურათი:



ცხრილის უჯრაში (უჯრა 1x1) შევიტანოთ ტექსტი:

```

<html>
<head><title> Page with tables </title>
</head>
<body bgcolor = "#FFFFFF" text = "#000000">
<center>
<table cellspacing = "5" width = "700">
<tr>
<td width = "400" bgcolor = "#FFCC33" colspan = "2">
<div align = "left">
Celebration of the New Year is one of the oldest traditions. For Europe the 1st January as the
New Year Day was introduced by Julius Caesar.
Janus was the Roman god of beginnings and endings, openings and closings. 600 years later
church proclaimed Annunciation (the 25th of March) as the New Year Day but at the end of
the 16th century the 1st of January again was declared by church as the New Year Day.
</div>
</td>
<td width = "300" bgcolor = "#336699" rowspan = "2" ><center> 1x2 </center> </td>
</tr>
<tr>
<td bgcolor = "#336699"> <center> 2x1 </center> </td>
<td bgcolor = "#FFCC33"><center> 2x2 </center> </td>
</tr>
</table>
</center>
</body>
</html>

```

ტექსტი გავასწორეთ მარცხენა მხრიდან DIV ტეგის დახმარებით, ცხრილი მოთავსდა ეკრანის შუაში ტეგ <center>-ის საშუალებით. თუ თქვენ ზემოთ მოყვანილი კოდის რეალიზებით მიღებულ შედეგს შეხედავთ ბრაუზერის ფანჯარაში, შენიშნავთ რომ ტექსტი საკმაოდ ახლოსაა უჯრის კიდეებთან და გამოიყურება უმნოდ. ამის გამოსწორება შესაძლებელია, cellpadding ატრიბუტის დახმარებით.

```

<html>
<head>
<title> Page with tables </title>
</head>
<body bgcolor = "#FFFFFF" text = "#000000">
<center>
<table cellspacing = "5"
width = "700" cellpadding="20">
<tr>
<td width = "400" bgcolor = "#FFCC33"
colspan = "2" valign="top">
<div align = "left">
<h3>Welcome on my page!</h3>
Celebration of the New Year is one of the oldest traditions. For Europe the 1st January as the
New Year Day was introduced by Julius Caesar.
Janus was the Roman god of beginnings and endings, openings and closings. 600 years later
church proclaimed Annunciation (the 25th of March) as the New Year Day but at the end of the
16th century the 1st of January again was declared by church as the New Year Day.
</div>
</td>
<td width = "300" bgcolor = "#336699"
rowspan = "2" valign="top">
<div align="center">
<br><br>
</div> </td>
</tr>
<tr>
<td bgcolor = "#336699">
<center> 2×1 </center>

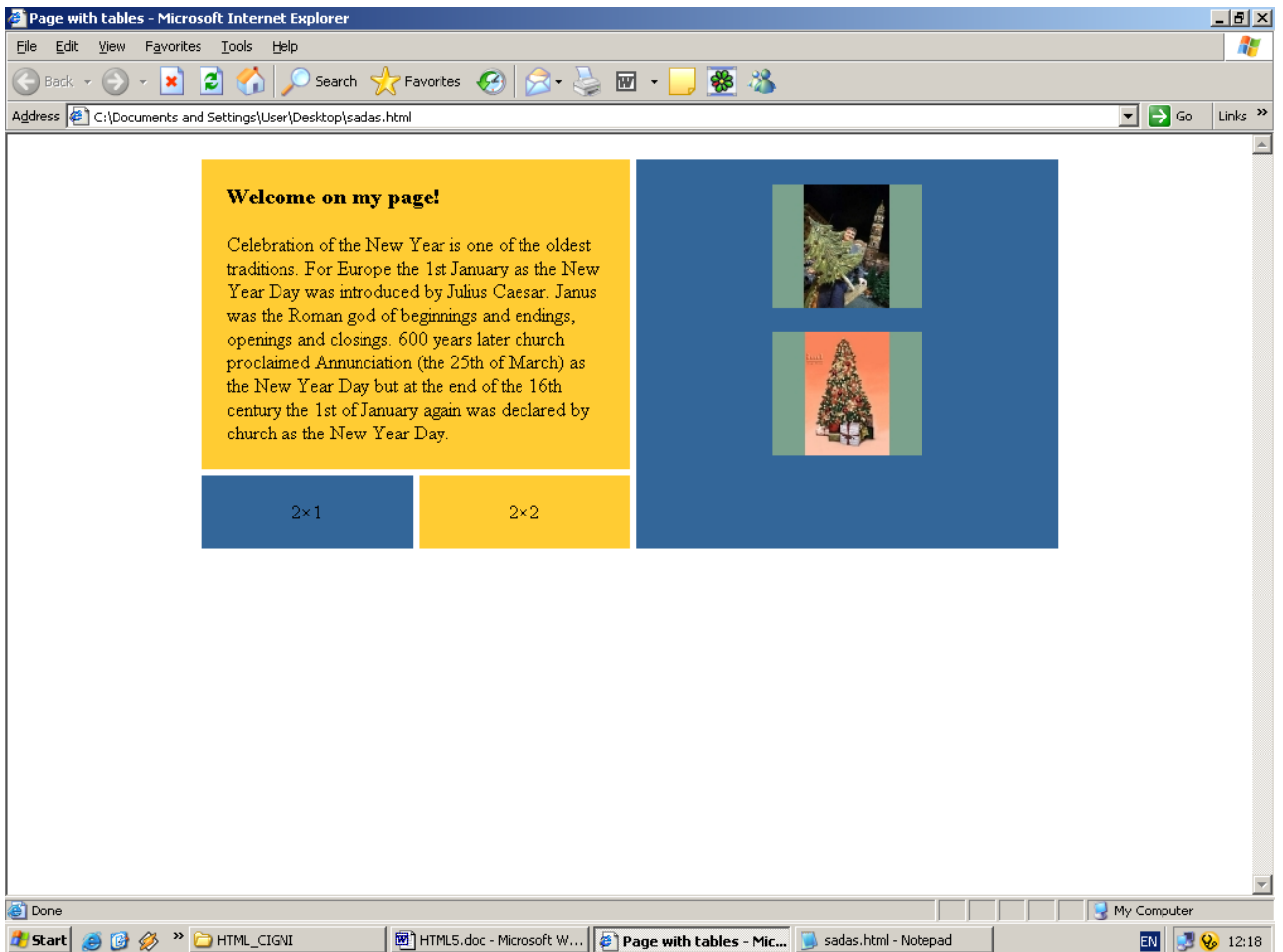
```

```

</td>
<td bgcolor = "#FFCC33">
<center> 2×2 </center>
</td>
</tr>
</table>
</center>
</body>
</html>

```

გარდა cellpadding ატრიბუტისა, აქ დაემატა <h3> ტეგი და მონაცემები მეორე სვეტისათვის. ატრიბუტ valign-ის დახმარებით ტექსტი და სურათი მოთავსდა უჯრის ზედა ნაწილში.



შევიტანოთ მონაცემები დარჩენილი ორი უჯრისათვის:

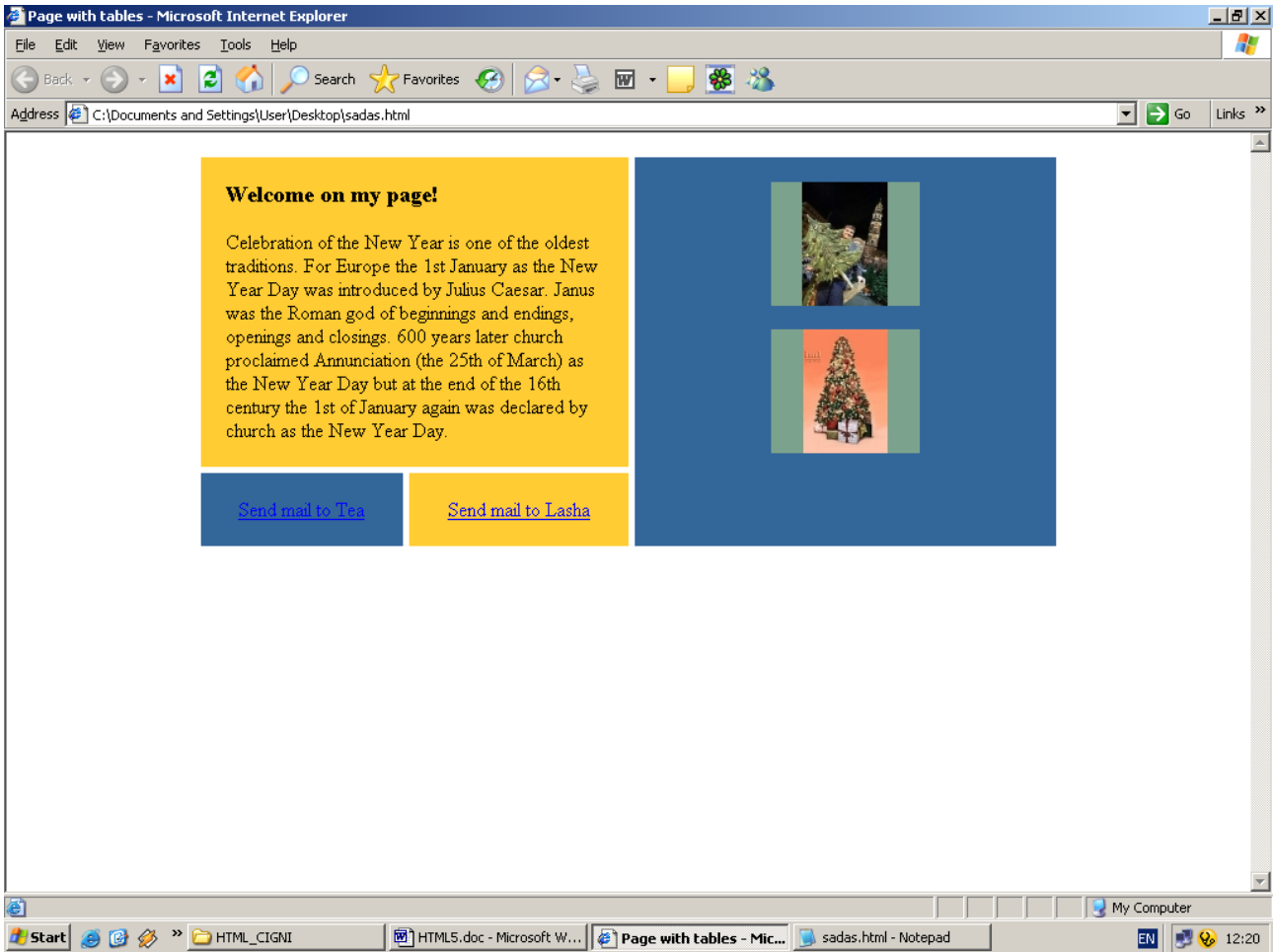
```

<td bgcolor = "#336699"> <div align="center"><a href="mailto:tea_todua@yahoo.com">Send
mail to Tea</a></div> </td>

```

<td bgcolor = "#FFCC33"><div align="center"><a href="mailto:"lasha@gtu.ge">Send mail to Lasha</a></div></td>

მიღებული ცხრილი ეკრანზე გამოიყურება შემდეგნაირად:



## 16. სპეციალური სიმბოლოები

სპეციალური სიმბოლოები html დოკუმენტში შეიძლება შეტანილ იქნეს &-ს (ამპერსანდი) მეშვეობით

სიმბოლო	კოდი	მაგალითი
საავტორო უფლების ნიშანი	&copy;	copyright@2002gn.ge
დარეგისტრირებული მარკა	&reg;	Mgt Co®
სავაჭრო მარკა	&#8482;	Wepfarer™
ნაკლებია	&lt;	<
მეტია	&gt;	>
ამპერსანდი	&amp;	&
სიტყვის შუაზე არგამსლენი პრობელი	&nbsp;	
ტირე	&#821;	-
ბრჭყალები	&quot;	“
გრადუსი	&deg;	°
პლიუს-მინუსი	&plusmn;	±
ტაბულაციის ნიშანი	&para;	¶
ხარისხი	&sup2;	2

ყველაფერს, რაც მოთავსებულია < > -ს შორის, ბრაუზერი აღიქვამს როგორც ტეგს. ამიტომ >-ს შესატანად ტექსტში მოიფიქრეს ნიშანი &gt;. დაუშვათ, თქვენ გჭირდებათ მიყოლებით ხუთი პრობელის ჩასმა ტექსტში. სწორედ ამ შემთხვევაში მოსახერხებელია პრობელის სპეციალური სიმბოლოს გამოყენება. ის შეიძლება დაიწეროს თუნდაც ათასჯერ. ჩვეულებრივ პრობელის იგნორირება ხდება ბრაუზერის მიერ, თუ ის ერთზე მეტია სიტყვებს ან ტეგებს შორის.

## 17. საზღვრები ბრაუზერის ფანჯარაში

html-ის შესწავლისას აუცილებლად უნდა აღინიშნოს, რომ არსებობს ტეგი და ატრიბუტი, რომელსაც ყველა ბრაუზერი ვერ აღიქვამს. ქვემოთ ვისაუბრებთ ტეგ body-ს ატრიბუტებზე (topmargin, leftmargin):

topmargin - განსაზღვრავს დოკუმენტის ზედა მინდვრის სიგრძეს  
(დაშორებას ზედა კიდესთან)

leftmargin- დოკუმენტის მარცხენა მინდვრის სიგანეს (დაშორებას  
მარცხენა კიდესთან)

ამ მინდვრების გასაუქმებლად პარამეტრებს leftmargin და topmargin-ს მივანიჭოთ ნულის ტოლი მნიშვნელობა:

```
<body text = "#000000" bgcolor = "ffffff" topmargin = "0" leftmargin = "0">
```

leftmargin და topmargin ატრიბუტებს აღიქვამს მხოლოდ Internet Explorer (IE). Netspace Navigator (NN)-სათვის არსებობს სხვა ატრიბუტები - marginheight და marginwidth . marginheight - topmargin-ის ანალოგიურია. marginwidth კი leftmargin-ის ანალოგიური.

იმისათვის რომ დაშორება ზედა და მარცხენა კიდესთან იყოს ნული, კოდი უნდა დაიწეროს შემდეგნაირად:

```
<body text = "#000000" bgcolor = "ffffff" topmargin = "0" leftmargin = "0" marginwidth = "0" marginheight = "0">
```

ასეთ შემთხვევაში ყველაფერი კარგად იქნება IE-ს, NN-ისა და სხვა ბევრი ბრაუზერისთვისაც.

leftmargin და topmargin დოკუმენტის მხოლოდ ზედა და მარცხენა მხრიდან დაშორებას განსაზღვრავენ. იმისათვის, რომ მოვაცილოთ მარჯვენა და ქვედა მინდვრები, არსებობს ატრიბუტები rightmargin და bottommargin:

```
<body text = "000000" bgcolor = "ffffff" rightmargin = "0" bottommargin = "0" topmargin = "0" leftmargin = "0" marginwidth = "0" marginheight = "0">
```

## 18. მორბენალი სტრიქონი

ტეგ Marquee-ს რომლის საშუალებითაც იქმნება მორბენალი სტრიქონი, აღიქვამს მხოლოდ IE:

```
<marquee height = "10" width = "270" bgcolor = "#99CCFF">  
მორბენალი სტრიქონი  
</marquee>
```

როგორც ხედავთ, ყველაფერი მარტივია. ტექსტი, რომელიც მოთავსდება ტეგებს `<marquee>` `</marquee>`-ს შორის, ხდება მორბენალი სტრიქონი.

ტეგ `marquee`-ს ყველა ატრიბუტი თქვენთვის უკვე ცნობილია. `bgcolor` - მორბენალი სტრიქონის ფონის ფერი, `height` - სტრიქონის სიმაღლე, `width` - სტრიქონის სიგანე.

მორბენალი სტრიქონისთვის შეიძლება მიუთითოთ თქვენთვის ცნობილი სხვა ატრიბუტებიც: `hspace`, `vspace` და `align`.

```
<marquee height = "10" width = "270" loop = "2"> მორბენალი სტრიქონი  
</marquee>
```

ატრიბუტი `loop` განსაზღვრავს მორბენალი სტრიქონის ეკრანზე გავლის რაოდენობას. ზემოთ აღწერილ შემთხვევაში, მორბენალი სტრიქონი ეკრანზე გაირბენს მხოლოდ ორჯერ.

ატრიბუტი `direction` განსაზღვრავს მორბენალი სტრიქონის მოძრაობის მიმართულებას - `direction = "left"` (`right`, `up`, `down`) - მოძრაობა მარცხნივ (მარჯვნივ, ზემოთ, ქვემოთ).

`behavior` - სტრიქონის მოძრაობის სახეს განსაზღვრავს. `behavior = "scroll"` (`slide`, `alternate`). `Scroll` - დაყენებულია ავტომატურად, შეგიძლიათ არც მიუთითოთ. `slide` - სტრიქონი მირბის კიდემდე და ჩერდება. `Alternate` - სტრიქონი იმოძრაებს კიდიდან კიდემდე. `Scrollamount` - სტრიქონის მოძრაობის სიჩქარე, შეიძლება მიიღოს მნიშვნელობა 1-იდან 10-მდე: 1 - ყველაზე ნელი მოძრაობაა, 10 - ყველაზე სწრაფი.

## 19. ფრეიმები

ფრეიმები ბრაუზერის ფანჯარას ცალკეულ დამოუკიდებელ პანელებად ყოფენ, რომელთაგან თითოეულს თავისი ფაილი გამოჰყავს ეკრანზე. სხვა სიტყვებით, თითოეული ასეთი პანელი “მინი ბრაუზერს” წარმოადგენს.

მაგალითი:

```
<html>
<head>
<title> Creating Frames </title>
</head>
<frameset rows = "100, *, 150">
<frame src = "logo.html">
<frame src = "content.html">
<frame src = "menu.html">
</frameset>
</html>
```

ბრაუზერის ფანჯარა იყოფა სამ კორიზონტალურ ფრეიმად:

logo.html
content.html
menu.html

წინასწარ არ დაგაფიწყდეთ შექმნათ ჩვეულებრივი html დოკუმენტები ლოგოტიპით, მენიუთი და ძირითადი ტექსტით (logo.html, content.html, menu.html). ატრიბუტ rows მეშვეობით, მიუვითეთ თითოეული ფრეიმის სიმაღლე (rows = “100, \*, 150”). პირველი ფრეიმის სიმაღლე - 100 პიქსელია, მესამის - 150, მეორე იკავებს დარჩენილ ნაწილს. (ეს მიეთითა \* ნიშნის მეშვეობით).

ტეგი Frame მიუთითებს, რომ პირველ ფრეიმში ჩაიტვირთება logo.html დოკუმენტი, მეორეში იქნება მოთავსებული Content.html, მესამეში - menu.html. თუ თქვენ გნებავთ, რომ menu.html იყოს მეორე რიგში, მაშინ menu.html-მა უნდა გაუცვალოს ადგილი Content.html-ს. ცვლილება გაკეთდეს ისე, რომ menu.html-მა კვლავ 150 პიქსელი დაიკავოს სიმაღლეში.

მაგალითი:

```
<html>
<head>
<title> Creating Frames </title>
</head>
<frameset rows = "100, 150, *">
<frame src = "logo.html">
<frame src = "menu.html">
<frame src = "content.html">
</frameset>
</html>
```

## 20. კონტეინერი Frameset

ფრეიმი მოთავსებული უნდა იყოს კონტეინერ frameset-ში, რომელიც ჩვენთვის უკვე ნაცნობი კონტეინერ body-ს ადგილს იკავებს. Html დოკუმენტი რომელსაც გააჩნია სტრუქტურა Frameset, არ შეიცავს body-ს და პირიქით. თუკი body-ს ჩავრთავთ დოკუმენტში, სადაც გამოიყენება კონტეინერი Frameset და Frame, ფრეიმები იგნორირებული იქნებიან ბრაუზერის მიერ და მათი ინფორმაცია არ აისახება ეკრანზე. გამომჩნდება მხოლოდ ის ინფორმაცია, რომელსაც შეიცავს body.

იყავით ყურადღებით და არ გამოიყენოთ სტრუქტურები body და Frameset ერთ დოკუმენტში. ტეგ Frameset-ის შიგნით შეიძლება მოთავსებული იყოს მხოლოდ ტეგი <Frame> ან კიდევ თვით ტეგი <Frameset>.

## 21. ფრეიმების პარამეტრების განსაზღვრა

ტეგ <Frameset>-ს აქვს ორი ძირითადი ატრიბუტი: Rows და cols. მათი საშუალებით ხდება სტრიქონებისა (რიგების) და სვეტების რაოდენობის მითითება. თუკი თქვენ ერთ სტრიქონზე ან სვეტზე მეტი არ განსაზღვრეთ, მაშინ ბრაუზერი იგნორირებას უკეთებს კონტეინერ <Frameset>-ს და ეკრანი ცარიელი რჩება. სხვა სიტყვებით, ფრეიმი არ შეიძლება იყოს ერთადერთი.

```
<Frameset rows = "100, 240, 140">
```

მიუთითებს, რომ გვაქვს სამი ჰორიზონტალური ფრეიმი, რომელთა სიმაღლე 100, 240 და 140 პიქსელია შესაბამისად. თუმცა სტრიქონების სიღღის მითითება

პიქსელებში არ არის რეკომენდებული, რადგან ამ დროს მხედველობაში არ მიიღება ის ფაქტი, რომ ბრაუზერის ფანჯრები შეიძლება სულ სხვადასხვა ზომისანი იყვნენ. უმჯობესია ვისარგებლოთ ფარდობითი სიდიდეებით. მაგალითად: Frameset Rows = “25%,50%,25%” რაც ქმნის სამ ფრეიმს 25%,50%,25% სიმაღლით, ბრაუზერის ფანჯრისაგან დამოკიდებულებით.

ფრეიმების პარამეტრები ფარდობით ერთეულებში შეიძლება მივუთითოთ შემდეგი სახითაც: <Frameset cols = “\*, 2\*, 3\*”>, სადაც სიმბოლო \* აღნიშნავს ბრაუზერის ფანჯრის პროპორციულ დაყოფას. ამგვარად, ფანჯარა იყოფა სამ ვერტიკალურ ფრეიმად (სამ სვეტად), პირველის სიგანეა ბრაუზერის ფანჯრის 1/6, მეორის 2/6 (ანუ 1/3) და მესამის 3/6 (ანუ 1/2). ერთიანის მითითება ფარდობითი მნიშვნელობების დროს აუცილებელი არ არის.

Rows და Cols ატრიბუტები შეიძლება შერეული სახითაც მიეთითოს, მაგალითად: <Frameset cols = “100,25%,\*,2\*”>. აქ პირველ ფრეიმს ენიჭება აბსოლუტური მნიშვნელობა – 100 პიქსელი, მეორეს 25% ფანჯრის ზომისაგან დამოკიდებულებით. დანარჩენი სივრცე იყოფა მესამე და მეოთხე ფრეიმს შორის პროპორციით 1/3 და 2/3.

პირველ რიგში გამოიყოფა ადგილი იმ ფრეიმისთვის, რომლისთვისაც აბსოლუტური მნიშვნელობაა მითითებული. შემდეგ ფრეიმისთვის, რომლის მნიშვნელობაც პროცენტებშია მითითებული, ბოლოს კი ფრეიმისთვის – ფარდობითი მნიშვნელობით.

თუ თქვენ იყენებთ აბსოლუტურ სიდიდეებს Rows და Cols-ის მითითებისას, ნუ გააკეთებთ ასეთ ფრეიმს დიდს. ისინი ხომ ნებისმიერი ზომის ბრაუზერის ფანჯარაში უნდა ჩაეტიონ. ამიტომ გარკვეული ბალანსირების გაკეთების მიზნით რეკომენდებულია თუნდაც ერთი ისეთი ფრეიმის მითითება, რომელიც განსაზღვრულია პროცენტებში ან ფარდობით ერთეულებში.

ორივე ატრიბუტის განსაზღვრისას, მაგალითად: <Frameset Rows = “\*,2\*,\*” Cols = “2\*,\*”> მიიღება სამი სტრიქონი (რიგი) და ორი სვეტი. პირველი და ბოლო სტრიქონები იკავებენ ფანჯრის სიმაღლის 1/4-ს, შუა 1/2-ს. პირველი სვეტი ბრაუზერის ფანჯრის სიგანის 2/3-ს, ხოლო მეორე 1/3-ს იკავებს.

## 22. ტეგი <Frame>

ტეგი <Frame> განსაზღვრავს ცალკეულ ფრეიმს. ის უნდა განთავსდეს კონტეინერ <Frameset>-ის შიგნით:

```
<Frameset Rows = “*,2*”>
<Frame>
<Frame>
</Frameset>
```

ავლნიშნავთ, რომ ეს ტეგი არ წარმოადგენს კონტეინერს და </Frameset>-ისგან განსხვავებით არ გააჩნია დახურვის ტეგი. <Frame> ტეგების რიცხვი უნდა იყოს ტოლი ფრეიმების იმ რაოდენობისა, რომელიც განსაზღვრულია ტეგ <Frameset>-ით.

ჩვენს მაგალითში ორი ფრეიმია განსაზღვრული, ამიტომაც კონტეინერი შეიცავს <Frame> ტეგების შესაბამის რაოდენობას. ჯერჯერობით ჩვენს ფრეიმებში არანაირი ინფორმაცია არ წერია.

Html-ში ტეგ Frame-ს გააჩნია ექვსი ატრიბუტი: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING და NORESIZE. ამ ატრიბუტების გამოყენების სინტაქსი ასეთია: <Frameset src = “url” Name = “window\_name”, scrolling = Yes/No/Auto, Marginwidth = “value”, Marginheight = “value” NORESIZE >

ფრეიმში მოთავსებული ინფორმაციის მითითება ხორციელდება ატრიბუტ SRC-ს გამოყენებით: <Frame src = “sample.html”>. sample.html - სრულფასოვანი Html დოკუმენტია, შეიცავს ყველა აუცილებელ შემადგენელ ნაწილს (კონტეინერები: Html, Head, Body და ა.შ.) და მდებარეობს იმავე საქალაქოში, რომელშიც დოკუმენტი, რომელიც შეიცავს კონტეინერ Frameset-ს. Sample.html-ს შეიძლება ჰქონდეს შემდეგი სახე:

```
<Html>
<head>
<title>
</head>
<body>
This is some sample text
</body>
</html>
```

თუ თქვენ გნებავთ თქვენს ფრეიმში მოათავსოთ გამოსახულება GIF ფორმატში, დოკუმენტის კოდში შეგიძლიათ ჩართოთ ასეთი სტრიქონი:

```
<Frame src = “world.gif”>
```

იმ შემთხვევაში, როდესაც ბრაუზერი ვერ პოულობს მითითებულ ფაილს, ფრეიმი არ აიგება და ბრაუზერი გამოიტანს შეტყობინებას შეცდომის შესახებ. თუ ტეგში Frame არ არის მითითებული ატრიბუტი Src, ფრეიმი ნებისმიერ შემთხვევაში შეიქმნება, მხოლოდ ის იქნება ცარიელი.

ტეგსტი, სათაური, გრაფიკა და სხვა ელემენტები არ შეიძლება პირდაპირ იქნენ ჩართული დოკუმენტში, რომელიც შეიცავს ფრეიმებს. ისინი მხოლოდ ელემენტის URL მისამართის დახმარებით უნდა იყვნენ ჩართული დოკუმენტში. თუკი კონტეინერი Frameset შეიცავს “უცხო სხეულს”, ის უგულებელყოფილი იქნება, ფრეიმები იგნორირებულნი იქნებიან.

## 23. ფრეიმების გამოყენების რამდენიმე

### მარტივი მაგალითი

ფრეიმების მინიმალური შესაძლო რაოდენობა შეიძლება იყოს 2-ის ტოლი.

მაგალითი:

```
<Html>
<head></head>
<Frameset cols = *, 2*>
<Frame src = “label.html”>
<Frame src = “info.html”>
</Frameset>
</html>
```

განხილულ მაგალითში ფრეიმები ბრაუზერის ფანჯარას ორ ვერტიკალურ ნაწილად ყოფენ. მარცხენა ნაწილი იკავებს ფანჯრის 1/3-ს, მარჯვენა ნაწილი კი - 2/3-ს. პირველი ფრეიმი შეიცავს დოკუმენტს label.html-ს, მეორე კი - info.html-ს. ასეთი სახით ადვილად შეიძლება შეიქმნას ათი ან მეტი ვერტიკალური (ან ჰორიზონტალური, ატრიბუტ Row-ს გამოყენებით) ფრეიმი. თუმცა, როგორც წესი, გვერდზე არ უნდა იყოს ერთდროულად სამ-ოთხ ფრეიმზე მეტი.

თუ გნებავთ გვერდზე გქონდეთ ერთდროულად ოთხი ან ოთხზე მეტი ჰორიზონტალური ან ვერტიკალური ფრეიმი, უმჯობესია ცხრილების გამოყენება.

განვიხილოთ html დოკუმენტი, რომლის შედეგადაც ბრაუზერის ფანჯარაში დოკუმენტი განლაგდება შემდეგნაირად:

logo.html	
menu.html	Content.html

ფანჯარა იყოფა ორ რიგად (სტრიქონად). პირველ რიგში მოთავსდება logo.html დოკუმენტი, მეორე რიგი იყოფა ორ სვეტად, რომლებშიც მოთავსდებიან დოკუმენტები menu.html და content.html.

მაგალითი:

```
<html>
<head><title> Creating Frames </title></head>
<frameset rows = "100,*">
<frame src = "logo.html">
<frameset cols = "150,*">
<frame src = "menu.html">
<frame src = "content.html">
</frameset>
</frameset>
</html>
```

ახლა განვიხილოთ მეორე სურათზე გამოსახული ვარიანტი:

logo.html	content.html
menu.html	

ამ შემთხვევაში ფანჯარა იყოფა სვეტებად. მეორე სვეტი შეიცავს დოკუმენტს content.html, პირველი სვეტი კი დაყოფილია ორ რიგად და მასში მოთავსებულია დოკუმენტები logo.html და menu.html.

შესაბამისი კოდი მოცემულია ქვემოთ:

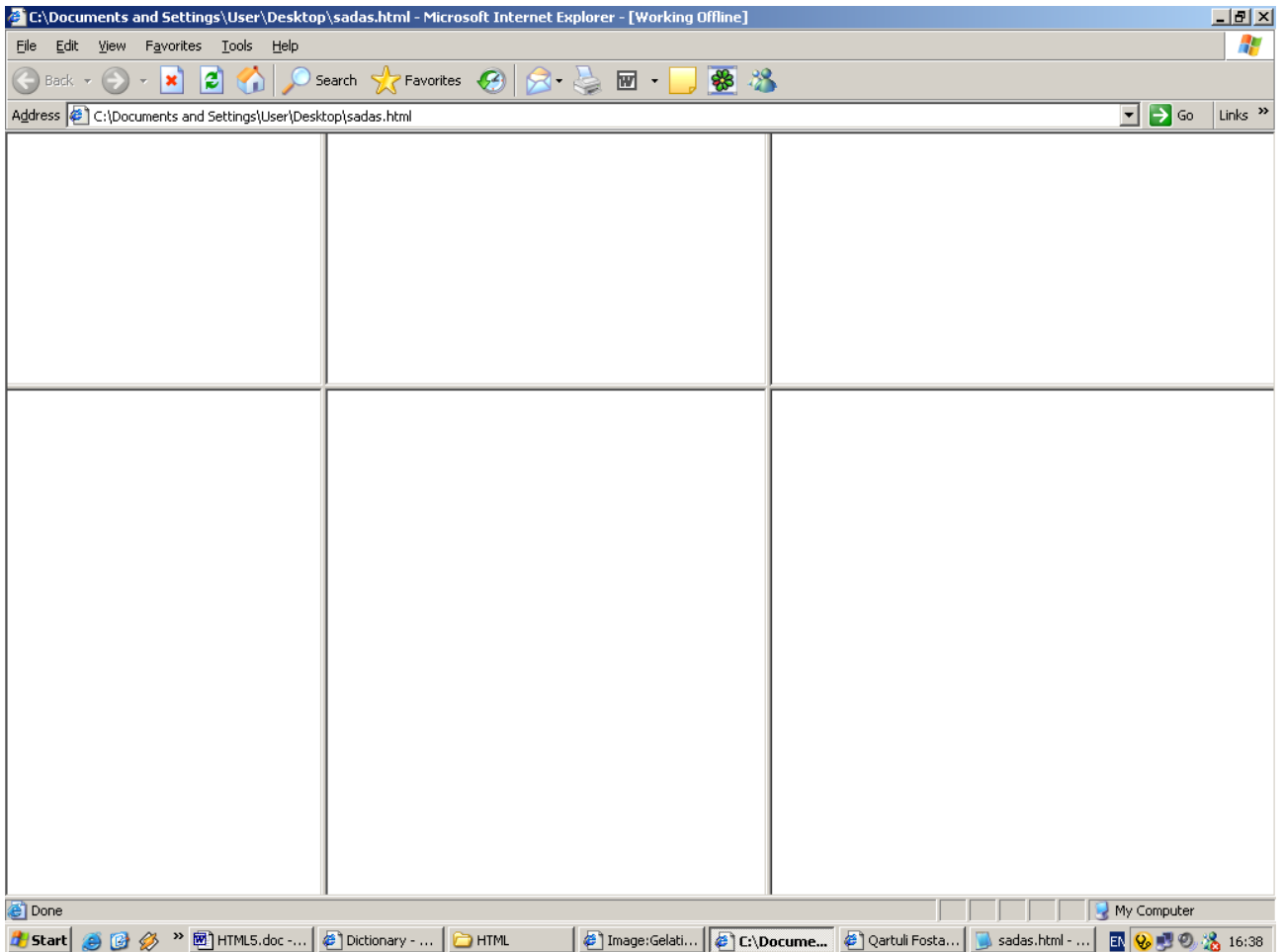
```
<html>
<head>
<title> Creating Frames </title>
</head>
<frameset cols = "100, *">
<frameset rows = "100, *">
<frame src = "logo.html">
<frame src = "menu.html">
</frameset>
```

```
<frame src = "content.html">
</frameset>
</html>
```

მაგალითი:

```
<html>
<head><title></title></head>
<Frameset Rows = "*,2*" Cols = "25%, 35%, 40%">
<Frame src = "labela.html">
<Frame src = "labelb.html">
<Frame src = "labelc.html">
  <Frame src = "infoa.html">
  <Frame src = "infob.html">
  <Frame src = "infoc.html">
</frameset>
</html>
```

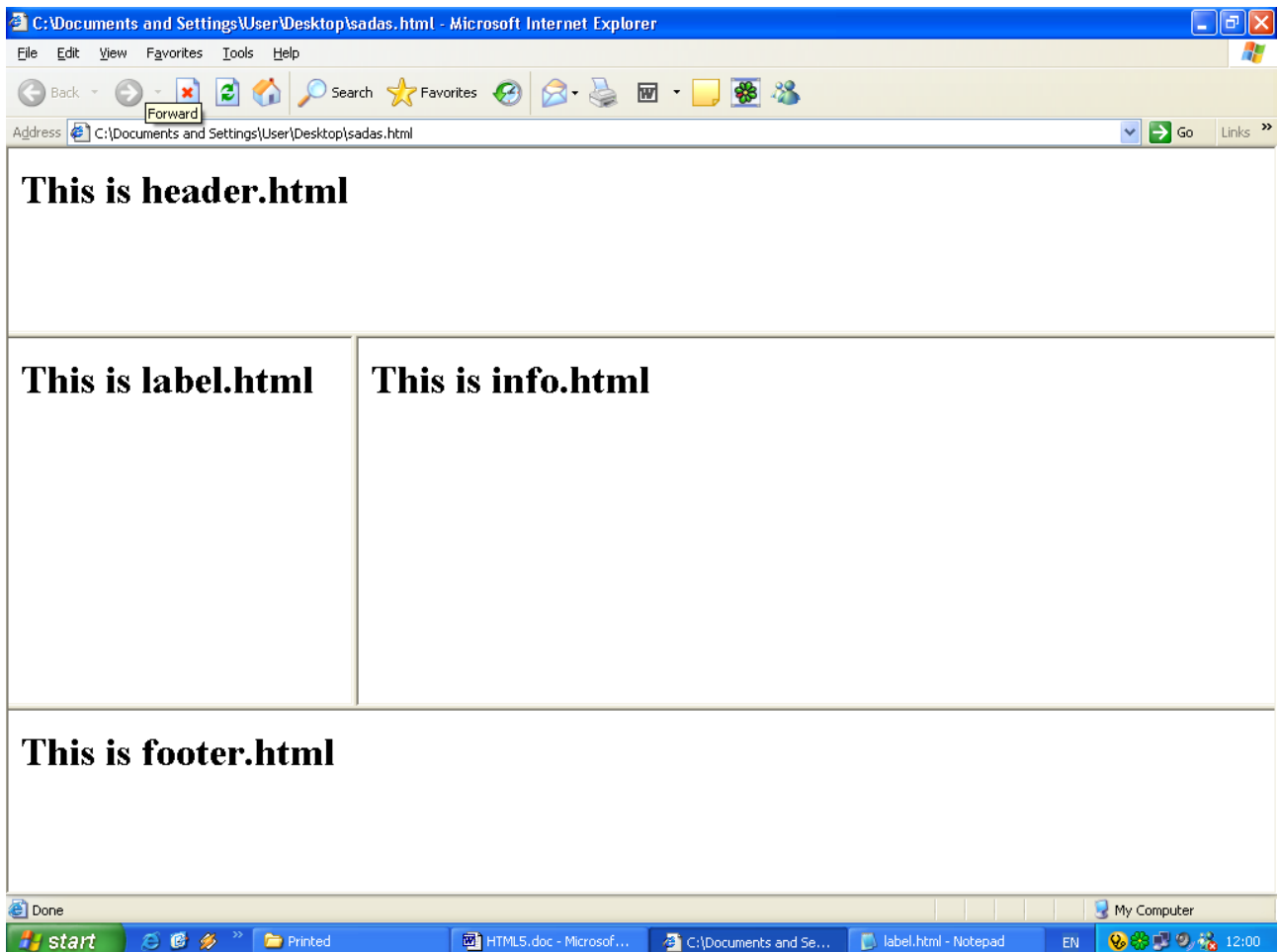
მიიღება შემდეგი სურათი:

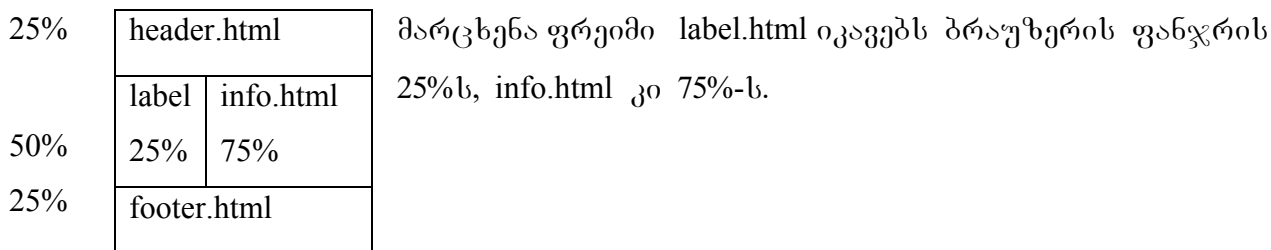


ფრეიმების უფრო რთული კომბინაციის მაგალითი:

```
<Html>
<head></head>
<Frameset Rows = "25%, 50%, 25%">
<Frame src = "header.html">
<Frameset cols = "25%, 75%">
<Frame src = "label.html">
<Frame src = "info.html">
</Frameset>
<Frame src = "footer.html">
</Frameset>
</html>
```

ამ დოკუმენტში ერთი ფრეიმი ჩადგმულია მეორეში, გარეთა კონტეინერი Frameset განსაზღვრავს სამ ჰორიზონტალურად განლაგებულ ფრეიმს, რომლებიც იკავებენ ფანჯარას მთელ მის სიგანეზე, მათი ვერტიკალური განზომილება კი შესაბამისად 25%, 50% და 25%-ია: <Frameset Rows = "25%, 50%, 25%">





## 24. ფრეიმების გარეგნული სახის მოდიფიკაცია

ატრიბუტები `Marginwidth` და `Marginheight` ფრეიმის შიდა საზღვრების კონტროლირების საშუალებას იძლევიან.

ატრიბუტების მნიშვნელობები მოიცემა ყოველთვის პიქსელებში, მაგალითად:

```
<Frame marginwidth = "5" marginheight = "7">
```

იქმნება ფრეიმის შიგნითა ჩარჩო ზედა და ქვედა საზღვარზე სიგანით 5 პიქსელი, მარცხენა და მარჯვენა საზღვარზე სიგანით 7 პიქსელი. ამ ჩარჩოს შიგნით მონაცემები არ იქნებიან მოთავსებული. ტეგ `<Frame>`-ს გააჩნია ატრიბუტი `SCROLLING`.

```
<Frame scrolling = "YES/NO/AUTO">
```

ამ ატრიბუტმა შეიძლება მიიღოს ერთ-ერთი სამი მნიშვნელობიდან: YES, NO ან AUTO. ეს უკანასკნელი ავტომატურად ყენდება საჭიროების შემთხვევაში. თუ მითითებულია yes, მაშინ მცოცავი ზოლი გამოჩნდება ყველა შემთხვევაში, no კრძალავს მის გამოჩენას.

ფრეიმის ზომები ადვილად შეიძლება შეიცვალოს მომხმარებლის მიერ. არსებობს სპეციალური პარამეტრი, რომელიც თქვენს მიერ შექმნილი გვერდის დამთავალიერებელს საშუალებას არ მისცემს შეცვალოს ფრეიმების ზომები. ვებმასტერები იყენებენ ატრიბუტ `NORESIZE`-ს.

მაგალითი:

```
<html>
<head>
<title> Creating frames </title></head>
<frameset rows = "100, *">
<frame src = "logo.html" scrolling = "no" noresize>
<frame src = "menu.html" noresize>
</frameset>
</html>
```

ფრეიმების საზღვრების წარმოსადგენად html-ში არსებობს სამი ატრიბუტი: BORDER, FRAMEBORDER და BORDERCOLOR. პირველი მათგანი ამ ატრიბუტებიდან გამოიყენება მხოლოდ ტეგთან <Frameset> და აყენებს ყველა საზღვრის სისქის ზომას პიქსელებში ყველა ფრეიმისთვის.

```
<Frameset Border = "5">
```

თუ ეს ატრიბუტი 0-ის ტოლია, მაშინ ყველა ფრეიმი იქნება ხილული ჩარჩოს (საზღვრის) გარეშე. ქვემოთ განხილულ მაგალითს დავარქვათ სახელი, ვთქვათ, Ali.html

```
<html>
<head>
<title> Creating frames </title>
</head>
<frameset cols = "100, *", border = "0">
<frameset rows = "100, *">
<frame src = "logo.html" scrolling = "no" >
<frame src = "menu.html">
</frameset>
<frame src = "content.html">
</frameset>
</html>
```

ატრიბუტ Border-ის მნიშვნელობა ავტომატურად 5-ის ტოლია. ატრიბუტი Frameborder გამოიყენება ტეგებთან <Frameset> და <Frame> და იღებს ორ მნიშვნელობას: yes ან no. როცა Frameborder = 0, ფრეიმებს შორის საზღვარი არ ჩანს.

ატრიბუტი Bordercolor შეიძლება გამოიყენოთ ტეგებთან <Frameset> და <Frame>.

მაგალითი:

```
<html>
<head>
<title> Creating frames </title></head>
<Frameset bordercolor = "red" Rows = " *, *">
<Frame src = "info.html" >
<Frame src = "info1.html">
</Frameset>
</html>
```

## 25. ფრეიმის სახელის განსაზღვრა

ფრეიმის სახელის განსაზღვრა ხდება Name-ის საშუალებით. მაგალითად, სტრიქონი <Frame src = “info.html” name = “Tamri”> შექმნის კადრს სახელად “Tamri”, რომელზეც შეიძლება შეიქმნას ჰიპერტექსტური გადასვლა შემდეგი სახით:

```
A href = “moreinfo.html” Target = “Tamri”>
```

```
Click here to jump to Tamri </A>
```

ატრიბუტი Target შეიცავს ფრეიმის სახელს. ამ ჰიპერტექსტის აქტივიზაციის დროს ფრეიმ “Tamri”-ს შემცველობა ანუ იგივე info.html, შეიცვლება ფაილით moreinfo.html. ატრიბუტი Target რომ არ ყოფილიყო, moreinfo.html იმ ფანჯარაში ან ფრეიმში იქნებოდა, სადაც არის ჰიპერტექსტი. ატრიბუტი TARGET სწორედ იმ ფრეიმის მისათითებლადაა განკუთვნილი, რომელშიც განთავსდება Href ატრიბუტით განსაზღვრული ფაილი.

დავუბრუნდეთ მაგალითს, რომელსაც სახელად Ali.html დავარქვით. მასში განსაზღვრულია სამი ფრეიმი – logo.html, menu.html და content.html. იმისათვის, რომ კარგად გამოჩნდეს თითოეული ფრეიმის საზღვარი, კოდში წაშაღეთ border=”0”. დავუშვათ, menu.html დოკუმენტს აქვს შემდეგი სახე:

მაგალითი:

```
<html>
<head>
<title> Document with menu </title>
</head>
<body>
<center>
<a href = “content.html”>Main </a>
<a href = “book 1.html”> Book 1 </a>
<a href = “book 2.html”> Book 2 </a>
<a href = “book 3.html”> Book 3 </a>
<a href = “book 4.html”> Book 4 </a>
<a href = “book 5.html”> Book 5 </a>
</center>
</body>
</html>
```

დოკუმენტები, რომლებზეც ხდება გადასვლა a href-ის საშუალებით, იხსნება იმავე ფრეიმში, რომელშიც მოთავსებულია menu.html დოკუმენტი. თქვენ გჭირდებათ, რომ

ის გაიხსნას ფრეიმში, რომელშიც მოთავსებულია ფაილი “content.html”, მენიუ კი ეკრანზე დარჩეს უცვლელი სახით. როგორ უნდა გაკეთდეს ეს? სწორედ აქ გვჭირდება ზემოთხსენებული ატრიბუტი name. Ali.html შეიცვლება შემდეგნაირად:

```
<html>
<head><title> Creating frames </title></head>
<frameset cols = "100, *">
<frameset rows = "100, *">
<frame src = "logo.html" scrolling = "no">
<frame src = "menu.html">
</frameset>
<frame src = "content.html" name = "tea">
</frameset>
</html>
```

პარამეტრის სახელი ფრეიმს ანიჭებს უნიკალურ სახელს. ჩვენს შემთხვევაში ფრეიმს, რომელიც შეიცავს ფაილს content.html, მიენიჭა სახელი tea. ფრეიმის სახელი შეიძლება შემდეგში გამოყენებული იყოს სხვა დოკუმენტებიდან (ფრეიმებიდან) მასზე მიმართვისათვის ატრიბუტ target-ის საშუალებით. შემოვიტანოთ ატრიბუტი target ფაილში menu.html. თითოეული ბმულისათვის მივუთითოთ პარამეტრი target="tea", სადაც tea არის ფრეიმის სახელი, რომელშიც მოთავსებულია ფაილი content.html.

```
<html>
<head>
<title> Document with menu </title>
</head>
<body>
<center>
<a href = "content.html" target="tea">Main </a>
<a href = "book 1.html" target="tea"> Book 1 </a>
<a href = "book 2.html" target="tea"> Book 2 </a>
<a href = "book 3.html" target="tea"> Book 3 </a>
<a href = "book 4.html" target="tea"> Book 4 </a>
<a href = "book 5.html" target="tea"> Book 5 </a>
</center>
</body>
</html>
```

ახლა ყველა ბმული გაიხსნება ჩვენთვის საჭირო ფრეიმში, მენიუ კი უცვლელად დარჩება. ამისათვის საჭირო გახდა მხოლოდ ორი დოკუმენტის შეცვლა. პირველი ეს არის ფრეიმების შემცველი დოკუმენტი, მეორე - menu.html.

ზოგჯერ საჭიროა დოკუმენტი გაიხსნას მთლიან ფანჯარაში, ანუ დაირღვეს ფრეიმების მთელი სტრუქტურა. ამისათვის საჭიროა პარამეტრ target -ს მიეცეს მნიშვნელობა -top: `<a href = "content.html" target="_top">Main </a>`

შესაძლებელია ერთდროულად რამდენიმე ფრეიმის შემცველობის განახლება ერთი ბმულის აქტივიზაციით. დაეუშვათ, გვაქვს ფაილი Frames.html

```
<html>
<head>
</head>
<Frameset Rows = "25%, 50%, 25%">
<Frame src = "header.htm">
<Frameset cols = "25%, 75%">
<Frame src = "label.html">
<Frame src = "info.html">
</Frameset>
<Frame src = "footer.html">
</Frameset>
</html>
```

ეს ყველაფერი შესაძლებელია წარმოვადგინოთ სხვაგვარადაც:

```
<Html>
<head>
</head>
<Frameset Rows = "25%, 50%, 25%">
<Frame src = "header.html">
<Frame src = "nested.html" Name = "Inner">
<Frame src = "testlink.html">
</Frameset>
</html>
```

ავლნიშნოთ რომ სულ ბოლო, ქვედა ფრეიმის შესაბამის დოკუმენტს ჰქვია testlink.html. სწორედ testlink.html შეიცავს ბმულს, რომლის შემოწმებასაც ჩვენ ახლა მოვახდენთ. ქვემოთ მოყვანილია nested.html ფაილის კოდი. ეს ფაქტიურად არის ორი ფრეიმი (label.html და info.html), რომელიც აღინიშნება ერთი სახელით Inner.

```

<Html>
<head>
</head>
<Frameset cols = "25%, 75%, ">
<Frame src = "label.html">
<Frame src = "info.html">
</Frameset>
</html>

```

ფაილს testlink.html აქვს შემდეგი სახე:

```

<html>
<head>
<title>
</title>
</head>
<body>
<a href = "Newstuff.html" Target= "Inner">
Click me </a>to put new stuff into the center frameset
</body>
</html>

```

Click me ბმულის აქტივიზაციის დროს მოხდება ორი ცენტრალური ფრეიმის განახლება, რომლებიც ადრე nested.html ფაილით იყო განსაზღვრული. მათში იქნება მოთავსებული დოკუმენტი Newstuff.html.

```

<html>
<head>
<title>
</title>
</head>
<body>
Here is some New stuff!
</body>
</html>

```

ამგვარად, ორი ცენტრალური ფრეიმი შევცვალეთ ახალი ინფორმაციის შემცველი ერთი კადრით.

## 26. ობიექტები Html დოკუმენტში

იმისათვის, რომ Html დოკუმენტი გახდეს დინამიური, თქვენ შეგიძლიათ სპეციალური ტეგების საშუალებით მასში სხვადასხვა ობიექტები მოათავსოთ. ერთი ტეგი, რომლის საშუალებითაც Html დოკუმენტში შეიძლება ობიექტის მოთავსება, თქვენთვის უკვე ცნობილია: <img> ტეგის საშუალებით Html დოკუმენტში შეგიძლიათ ჩასვათ სურათი. გარდა სურათისა, Html დოკუმენტში შესაძლებელია მუსიკის, სხვა დოკუმენტის, flash-ანიმაციის, კლიპების და სხვა მრავალი ობიექტის ჩასმა.

ტეგი iframe ერთი html დოკუმენტის მეორე html დოკუმენტში მოთავსების საშუალებას იძლევა. iframe-სთვის არ არის საჭირო frame დოკუმენტის შექმნა.

ტეგი <iframe> მიეთითება შემდეგნაირად: <iframe src = "nino.html" width = "300" height = "250" scrolling = "auto" frameborder = "1"> </iframe>

ზოგადად html დოკუმენტში ობიექტების ჩასასმელად არსებობს ტეგი <object> ეს ტეგი საკმაოდ მრავალფუნქციურია. მისი გაცნობა ამ სახელმძღვანელოში იქნება მოკლე და მიმოხილვითი. უფრო დაწვრილებით შეგიძლიათ გაეცნოთ Html-ის ოფიციალურ დოკუმენტაციას, სადაც დეტალურადაა აღწერილი ტეგ <object>-ის გამოყენების ყველა შესაძლებლობა. სანამ გაეცნობოდეთ ამ ტეგსა და მის ფუნქციებს, გავისხენოთ ცოტაოდენი, ადრე განვლილი მასალიდან.

თქვენთვის უკვე ნაცნობია ტეგი <img>. თქვენს დოკუმენტში სურათის ჩასასმელად დაწერილი კოდი გამოიყურება შემდეგნაირად:

```
<img src = "bona.jpg" alt = "this is my dog">
```

თქვენთვის აქ ყველაფერი გასაგებია. src მიუთითებს გზას დოკუმენტისაკენ bona.jpg, alt - ალტერნატიული აღწერაა, ყოველი შემთხვევისათვის, თუ სურათი არ ჩაიტვირთება.

იგივეს გაკეთება შეიძლება ტეგ <object>- ის საშუალებით:

```
<object data = "bona.jpg" type = "image/jpeg">
```

```
this is my dog
```

```
</object>
```

ახლა განვიხილოთ ზემოთ მოყვანილი კოდი. ატრიბუტი data უთითებს გზას დოკუმენტისაკენ. ატრიბუტი type გვიჩვენებს მონაცემების ტიპს, ანუ მიუთითებს ბრაუზერს რა ტიპის ობიექტთან აქვს მას საქმე. ჩვენს შემთხვევაში ეს არის image - გამოსახულება, სურათი, jpeg-სურათის (გამოსახულების) ტიპი. ტექსტს, რომელიც მოთავსებულია <object> </object> ტეგის შიგნით, მომხმარებელი

(დამთვალეირებელი) დაინახავს იმ შემთხვევაში, თუ ბრაუზერი ვერ შეძლებს ეკრანზე ასახოს თქვენი ობიექტი - სურათი.

ვიდეო-კლიპის Html დოკუმენტში ჩატვირთვისათვის კოდი ანალოგიურია:

```
<object data = "film.mpeg" type = "application/ mpeg">  
there was a film  
</object>
```

მაგალითი: დავამატოთ ჩვენს ინტერნეტ გვერდს ხმა ან ვიდეოკლიპები.

```
<html>  
<head>  
<title>Video files and sound</title>  
</head>  
<body>  
<I>Sound</I>  
<a href="shl.wav">Sound1</a><br>  
<i>Music background</i><br>  
<Bgsound src="1.mid" Loop=Infinite>  
<i>Advertising Video with standard control board:</i>  
<embed src="name.avi" width=300 heigth=200>  
</body>  
</html>
```

ბრაუზერების უმრავლესობა საშუალებას იძლევა მიმართოს ბგერით ფაილებს .AU, MID/MIDI, WAV და ვიდეოფაილებს .MPG/.MPEG, .QT და .AVI ფორმატებში:

```
<a href="აუდიოფაილის URL">აღწერა</a>;  
<a href="ვიდეოფაილის URL">აღწერა</a>
```

გვერდის ჩატვირთვისას მუსიკალური ფონის შესაქმნელად ვწერთ:

<Bgsound src="აუდიოფაილის URL" Loop=n|Infinite>, სადაც Loop=n; n არის განმეორებების რიცხვი (n=1, 2, 3...); თუ მითითებულია Loop=infinite, მაშინ ვიდეოფაილის ჩვენება უსასრულოდ გაგრძელდება;

შეიძლება პირდაპირ web-გვერდზე იყოს ვიდეოკლიპი:

```
<embed name = სახელი src="ვიდეოფაილის URL" Height=Y Width=X Autostart=true>
```

ტეგ Embed - ის ატრიბუტები

Src= name.avi (ვიდეო კლიპის სახელი);

Width = X; Height = y; გამოსახულების სიგანე (სიმაღლე) პიქსელებში

Autostart = true; თუ არ მიუთითებთ Autostart=true-ს, მაშინ ვიდეოკლიპი დოკუმენტის ჩატვირთვისას, ავტომატურად არ ჩაირთვება. მისი ჩართვა მოხდება მართვის ღილაკებით (true/false); Repeat = true საშუალებას იძლევა ან კრძალავს ვიდეოფაილის გამეორებას (true / false).

Play\_loop = z; თუ თქვენ მიუთითებთ Repeat = true, z-ის ნაცვლად აქ მიუთითებთ განმეორებების რიცხვი.

Controls = Smallconsole გამოიყენეთ ეს ატრიბუტი, თუ გნებავთ რომ მართვის პულტი პატარა იყოს.

ვიდეოფაილი შეიძლება ჩავდგათ HTML დოკუმენტში <Img> ტეგის დახმარებით. ამისათვის არის Img-ის ატრიბუტი Dynsrc = "--" და არასავალდებულო ატრიბუტები LOOP = n/Infinite (განმეორებების რიცხვის მისათითებლად) და Controls (მართვის ღილაკების დამატება)

IMG DYN SRC = " ვიდეო ფაილის URL" CONTROLS LOOP = n/INTINITE>

ზემოთ ჩვენ ვისაუბრეთ Html დოკუმენტში ობიექტების ჩასმის შესაძლებლობის შესახებ. თუმცა მუსიკის, ვიდეოკლიპების, flash-ისა და სხვა ობიექტების გარდა, არსებობენ საშუალებები, რითაც შეგიძლიათ "გააცოცხლოთ" და საინტერესო გახადოთ თქვენი web-გვერდი. სწორედ ასეთ საშუალებას წარმოადგენენ სკრიპტები და CSS. ორივე Html-საგან განსხვავებული სტანდარტია. ისინი აფართოებენ html-ის შესაძლებლობებს. აქვე ავღნიშნავთ, რომ მხოლოდ Html-ის შესაძლებლობები საკმაოდ შეზღუდულია. სწორედ ამის გამო მას უკვე დიდი ხანია სწრაფ სიკვდილს უწინასწარმეტყველებენ, როგორც web-ის სტანდარტს. თუმცა, როგორც ხედავთ, ის ჯერ კიდევ ცოცხალია და კიდევ დიდხანს იცოცხლებს. მიზეზი მარტივია. Html-ის შესწავლა ძალიან ადვილია, ნებისმიერ ადამიანს შეუძლია მისი ათვისება. რაც შეეხება მის ნაკლოვანებებს, ამისათვის სტანდარტის შემმუშავებლებმა Html-ში შემოიტანეს ტეგები, რომლის დახმარებითაც Html დოკუმენტში შესაძლებელია სხვადასხვა ობიექტების მოთავსება, მისთვის სხვა ტექნოლოგიების (CSS, Javascript და ა.შ.) მიბმა.

ამ ტექნოლოგიების შესწავლა აუცილებელია, თუ გნებავთ პროფესიონალურად შექმნათ თქვენი web-გვერდები. მათ შესახებ სახელმძღვანელოს II ნაწილში ვისაუბრებთ. ყველასათვის გასაგები ჭეშმარიტებაა, ჯერ უნდა ვისწავლოთ უფრო მარტივი ტექნოლოგიები, იმისათვის, რომ შემდგომში ადვილად გავიგოთ უფრო რთული.

## 27. ფორმები

ფორმები ყველაზე პოპულარულ საშუალებას წარმოადგენენ მსოფლიო აბლაბუდაში ინტერაქტიული ურთიერთქმედებისათვის. Html-ის ფორმები მკითხველისაგან ინფორმაციის მიღების საშუალებას იძლევიან. Html-ის დახმარებით თქვენ შეგიძლიათ შექმნათ მარტივი ფორმები, რომლებიც ითვალისწინებენ “დიახ” და “არა” ტიპის პასუხებს. შეგიძლიათ შექმნათ რთული ფორმები შეკვეთებისთვის ან იმისათვის, რომ თქვენი მკითხველებისაგან მიიღოთ რაიმე კომენტარი ან კეთილი სურვილები.

ფორმები რამდენიმე ველისაგან შედგება, სადაც მომხმარებელს შეუძლია შეიტანოს გარკვეული ინფორმაცია ან აირჩიოს რომელიმე ოფცია. მას შემდეგ, რაც მომხმარებელი გააგზავნის ინფორმაციას, ის დამუშავდება პროგრამის (სკრიპტის) მიერ, რომელიც მოთავსებულია სერვერზე. სკრიპტი - ეს მოკლე პროგრამაა, სპეციალურად შექმნილი თითოეული ფორმის დასამუშავებლად.

ბევრი ამჯობინებს არ ისწავლოს თეორია და ისარგებლოს ვიზუალური html-რედაქტორებით, რომლებიც ფორმას თვითონ ქმნიან და მისგან არ მოითხოვენ html-ის ცოდნას. თქვენ რა თქმა უნდა, შეგიძლიათ ისარგებლოთ ასეთი რედაქტორებით, მაგრამ ამ შემთხვევაში თქვენ იქნებით საკმაოდ შეზღუდული. ასეთი რედაქტორები ყოველთვის არ იძლევიან იმ შედეგების მიღწევის საშუალებას, რომელიც თქვენ მიზნად დაისახეთ ისევ და ისევ თავიანთი მოუქნელობის და შესაძლებლობებში შეზღუდულობის გამო. საუკეთესო რედაქტორები, ეს არიან ტექსტური რედაქტორები, სადაც თქვენ შეგიძლიათ კოდი თქვენ თვითონ საკუთარი ხელებით დაწეროთ.

ფორმები იქმნებიან html-ის საშუალებით. თუმცა ამას ხშირად შეჰყავს ხალხი შეცდომაში. ბევრი თვლის, რომ საკმარისია ისწავლონ ფორმების შექმნა Html-ის საშუალებით, რომ მათ საიტზე გამოჩნდეს, მაგალითად ე.წ. სტუმართა წიგნი. ეს არასწორია.

ფორმა - ეს მხოლოდ კარკასია. Html-ის დახმარებით ჩვენ მხოლოდ განვსაზღვრავთ, რომ ვთქვათ, აქ გვექნება ფორმის ველი, აქ - ღილაკი, ღილაკზე კი იქნება წარწერა და ა.შ. იმისათვის რომ ფორმის ღილაკზე დაჭერისას თქვენი შეტყობინება დაემატოს სტუმართა წიგნს, საჭიროა სხვა ტექნოლოგია, საჭიროა რაიმე პროგრამა, სკრიპტი, რომელიც ჩვენ უნდა მივაბათ ჩვენს ფორმას. ჩვენ ამ სახელმძღვანელოს ფარგლებში შევისწავლით მხოლოდ იმას, როგორ შექმნათ ფორმები Html-ის საშუალებით.

Html დოკუმენტში ფორმები შემოდის ტეგ `<form>` `</form>` -ის საშუალებით:

```
<form>
</form>
```

მონაცემები ფორმიდან შეიძლება გაეგზავნოს დასამუშავებლად რომელიმე სკრიპტს (გზა სკრიპტამდე, რა თქმა უნდა, უნდა მიეთითოს):

```
<form action = "http://www.mysite.ge/cgi_bin/guestbook/">
</form>
```

ან რომელიმე საფოსტო სერვერს:

```
<form action: "mailto:tea_todua@yahoo.com">
</form>
```

## 28. ფორმის ტეგებთან მუშაობა

ფორმის ელემენტები არც ისე ბევრია. ესენია ღილაკები, ალმები, გადამრთველები, ველები და სიები. გარდა ამისა, ფორმებში შეიძლება სხვა ობიექტები ჩაისვას.

ფორმების შესწავლას შევუდგეთ კონკრეტული მაგალითების განხილვით. დავიწყოთ ასეთი ფორმით:

Your Name:

Your Comments:

OK
Cancel

მოცემულ ფორმაში გვაქვს შემდეგი ელემენტები: ველი რაიმე ინფორმაციის შესატანად (name), ველი ტექსტის შესატანად (Comments), გაგზავნისა და ინფორმაციის გაგზავნის შეწყვეტის ღილაკები.

წარმოიდგინეთ, რომ ჩვენი კოსტერი (კომპანია რომელიც გვაძლევს ადგილს web-გვერდის განსათავსებლად), არ გვაძლევს საშუალებას გამოვიყენოთ სკრიპტები, ამიტომ არ შეგვიძლია მოცემული ფორმა მივაბათ ე.წ. სტუმართა სკრიპტის წიგნს, რათა გაგზავნის ღილაკზე დაჭერისას მონაცემები ფორმიდან მაშინვე დაემატოს საიტს. თუმცა გვინდა, რომ გვერდის დამთვალიერებელს

ჩვენთვის შეეძლოს თავისი გამოხმაურების გამოგზავნა, ამიტომ ამ ფორმაში მიუვითითებთ, რომ ფორმიდან მონაცემები გამოიგზავნოს ჩვენს E-mail-ზე:

```
<form action="mailto:lasha_verulava@yahoo.com"> </form>
```

შემოვიტანოთ ამ კონსტრუქციაში ფორმის ელემენტები. ფორმის მრავალი ელემენტის შექმნა შეიძლება, ტეგ <input>-ის მეშვეობით. ელემენტის ტიპი განისაზღვრება ამ ტეგის ატრიბუტებით.

შევქმნათ ველი რაიმე ინფორმაციის შესატანად, მაგალითად შევქმნათ ველი იმ ადამიანის სახელის შესაყვანად, რომელიც ავსებს ფორმას:

```
<html>
<head><title>input1.html</title></head>
<body>
<form action="mailto:lasha_verulava@yahoo.com">
input type="text" name="firstname"
</form>
</body>
</html>
```

მაშ ასე, ტეგი <input>, როგორც უკვე ვიცით, ქმნის ფორმის ელემენტს. კონკრეტულად რომელ ელემენტს – ეს ჩვენ უკვე განვსაზღვრეთ ატრიბუტ type-ის მეშვეობით, როდესაც მას მივანიჭეთ მნიშვნელობა text. type=text ქმნის ელემენტს ერთსტრიქონიანი ტექსტის შესაყვანად.



Name ელემენტის სახელია, აუცილებელი ატრიბუტია. გარდა აუცილებელი ატრიბუტებისა type და name, ფორმის ელემენტს აქვს სხვა ატრიბუტებიც, value ტექსტის შეტანის ველისათვის (და არა მხოლოდ მისთვის), size - ტექსტის შესატანი ველის სიგრძე (იზომება ნაბეჭდ სიმბოლოებში, მისი მნიშვნელობა ავტომატურად 20-ის ტოლია), maxlength - განსაზღვრავს იმ სიმბოლოების მაქსიმალურ რიცხვს, რომელიც მომხმარებლის მიერ იქნება შეტანილი ფორმის ველში.

```
<html>
<head><title>input1.html</title></head>
<body>
<form action="mailto:lasha_verulava@yahoo.com">
<input type="text" name="firstname"
```

```

value = "Enter your name" size = "40" maxlength = "20">
</form>
</body>
</html>

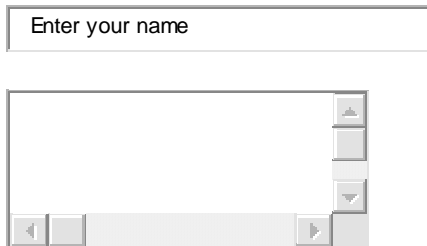
```

ყველაფერი ეს გამოიყურება შემდეგნაირად:



მაშ ასე, ველი გახდა უფრო გრძელი, რადგან ჩვენ შევიტანეთ მისთვის მნიშვნელობა `size = "40"`. მომხმარებელს არ შეუძლია ამ ველში დაბეჭდოს 20-ზე მეტი სიმბოლო (`maxlength = "20"`), `value` კი უბრალოდ მიუთითებს, კონკრეტულად რა უნდა იქნეს შეტანილი ამ ველში.

ჩვენს ფორმაში შემოვიტანოთ მომდევნო ელემენტი: ველი კომენტარის შესატანად.



ასეთი ველი ჩნდება ტეგ `<textarea>` `</textarea>`-ს მეშვეობით. `<textarea>` ქმნის ელემენტს მრავალსტრიქონიანი ტექსტის შესატანად.

```

<html>
<head>
<title>input1.html</title>
</head>
<body>
<form action="mailto: lasha_verulava @yahoo.com">
<input type="text" name="firstname"
value = "Enter your name" size = "40" maxlength = "20"><br><br>
<textarea name="comments"></textarea>
</form>
</body>
</html>

```

განვიხილოთ ზემოთ აღწერილი კოდი: `name` - როგორც უკვე იცით, აუცილებელი ატრიბუტია, ამიტომაც შემოვიტანეთ ის. იმისათვის, რომ ფორმის

ელემენტები განგვეთავსებინა ერთმანეთის ქვეშ, ჩვენ შემოვიტანეთ თქვენთვის უკვე ნაცნობი ტეგი <br>. იმის განსაზღვრა, როგორ განთავსდებიან ფორმის ელემენტები ერთმანეთის მიმართ, თქვენ შეგიძლიათ თქვენთვის უკვე ნაცნობი ტეგებისა და ატრიბუტების მეშვეობით. ახლა ჩვენს ფორმას დავამატოთ ტექსტები (Your name და Your comments):

Your Name:

Your Comments:

დაწვრილებით განვიხილოთ <textarea> და მისი ატრიბუტები. მათი საშუალებით შეიძლება მიუთითოთ ველის სიმაღლე (rows - სტრიქონების რაოდენობა), ველის სიგანე (cols - ნაბეჭდი სიმბოლოების რაოდენობა სტრიქონში), შევიტანოთ ტექსტი, რომელიც აისახება ფორმაში.

```
<html>
<head>
<title>
input1.html
</title>
</head>
<body>
<form action="mailto: lasha_verulava @yahoo.com">
Your Name:<br>
<input type="text" name="firstname"
value = "Enter your name" size = "40" maxlength = "20"><br><br>
Your Comments:<br>
<textarea rows =7 cols=40 name="comments"></textarea>
</form>
</body>
</html>
```

მიიღება შემდეგი სახის ფორმა:

Your Name:

Your Comments:

ფორმის შესაქმნელად სამუშაოს გარკვეული ნაწილი უკვე შესრულდა. დარჩა ორი ტიპის ღილაკის შექმნა. ინფორმაციის გაგზავნის ღილაკი (თქვენს ელფოსტაში ან სკრიპტისათვის დასამუშავებლად) იქმნება თქვენთვის უკვე ნაცნობი ტეგ <input>-ის და მისი ატრიბუტ type-ის მეშვეობით.

```
<html>
```

```
<head>
```

```
<title>
```

```
input1.html
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<form action="mailto: lasha_verulava @yahoo.com">
```

```
Your Name:<br>
```

```
<input type="text" name="firstname"
```

```
value = "Enter your name" size = "40" maxlength = "20"><br><br>
```

```
Your Comments:<br>
```

```
<textarea rows =7 cols=40 name="comments"></textarea><br><br>
```

```
<input type=submit>
```

```
</form>
```

```
</body>
```

```
</html>
```

შედგეი:

Your Name:

Your Comments:

როგორც უკვე ნახეთ, ღილაკისთვის ატრიბუტი name არ არის აუცილებელი. Reset ღილაკი იქმნება ანალოგიურად. მხოლოდ type-სთვის ჩვენ შეგვყავს მნიშვნელობა არა submit, არამედ reset:

```
<input type=reset>
```

შედგეი:

Your Name:

Your Comments:

ღილაკებზე სტანდარტული წარწერები შეგიძლიათ შეცვალოთ საკუთარით, თქვენთვის უკვე ნაცნობი ტეგის value-ს საშუალებით:

```
<html>
<head>
<title>input1.html</title>
</head>
<body>
<form action="mailto: lasha_verulava @yahoo.com">
Your Name:<br>
```

```

<input type="text" name="firstname"
value = "Enter your name" size = "40" maxlength = "20"><br><br>
Your Comments:<br>
<textarea rows =7 cols=40 name="comments"></textarea><br><br>
<input type=submit value="OK">
<input type=reset value="Cancel">
</form>
</body>
</html>

```

შედეგი:

Your Name:

Your Comments:



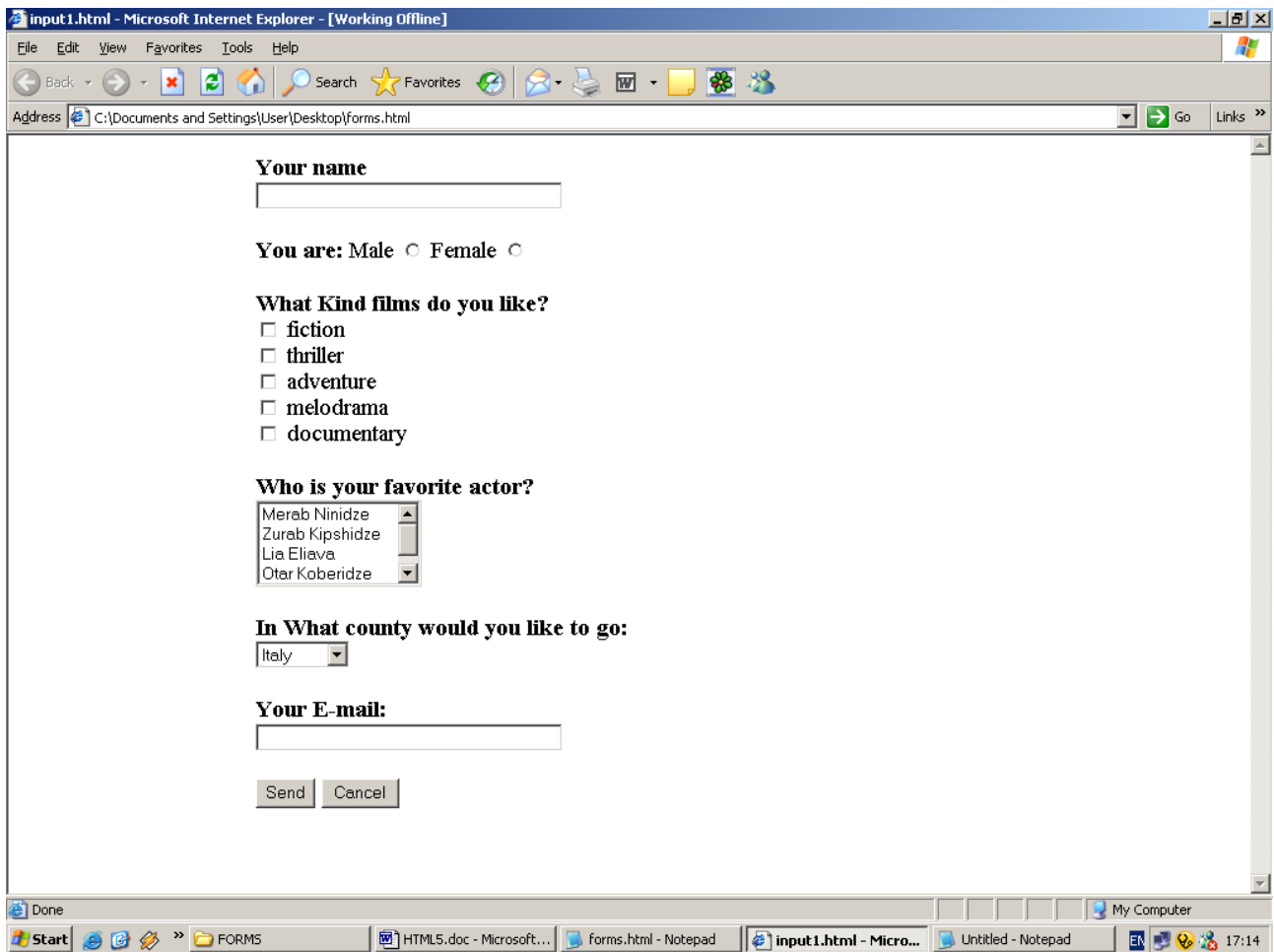

ფორმა შეიქმნა, მაგრამ ამით ყველაფერი არ დამთავრებულა. ჩვენ დაგვაუწყდა ჩაგვეხედა ერთი ძალიან მნიშვნელოვანი ატრიბუტი, რომლის გარეშეც ფორმა არ იმუშავებს:

```
<form action="mailto: lasha_verulava @yahoo.com" method=post>
```

Method განსაზღვრავს ფორმიდან მონაცემების გაგზავნის მეთოდს (სკრიპტისათვის ან საფოსტო სერვერზე). შესაძლო მნიშვნელობებია: get ან post.

ახლა შევქმნათ უფრო რთული ფორმა. დავუშვათ თქვენ შექმენით საიტი კინოს შესახებ. საიტზე გინდათ განათავსოთ გამოკითხვის ფორმა, რათა გაარკვიოთ რომელი ფილმები და მსახიობები მოსწონს იმ ხალხს, რომელთაც უყვართ კინოს ყურება. პრიზის სახით ერთ-ერთი მათგანი მიიღებს საგზურს მის მიერ არჩეულ ქვეყანაში სამოგზაუროდ.

ანკეტას, რომელსაც თქვენ შესთავაზებთ თქვენი საიტის დამთავალიერებლებს, ექნება შემდეგი სახე:



ამ ანკეტაში ბევრია ფორმის თქვენთვის უცნობი ელემენტები. თუმცა არის ის ელემენტებიც, რომლებიც უკვე განვიხილეთ. თავიდან შევქმნათ ის ელემენტები, რომლებიც ჩვენთვის ნაცნობია.

ჩვენთვის ცნობილია როგორ შემოდის ფორმა დოკუმენტში:

```
<form name = "anketa" method = "post" action = "http://mysite.ge/script">
</form>
```

Name - აუცილებელი ელემენტი, ფორმის სახელი.

Method - ფორმის დამუშავების მეთოდი.

Action - გზა სკრიპტისაკენ, რომელიც დაამუშავებს ფორმას, ან ელ.ფოსტა სადაც გაიგზავნება მონაცემები ფორმიდან.

თქვენ უკვე იცით ტექსტის შეყვანის ველის შექმნა (Your name და Your E-mail)

```
<html>
<head><title>input1.html</title></head>
<body>
<form name = "anketa" method = "post"
action = "http://mysite.ge/script">
```

```
<b> Your name </b> <br>
<input type = text name = "fns" size = 37 maxlength = 30> <br> <br>
<b> Your E-mail: </b> <br>
<input type = text name = "email" size = 37 maxlength = 30">
</form>
</body>
</html>
```

type = text-ის საშუალებით ჩვენ მივუთითეთ, რომ უნდა შეიქმნას ველი ტექსტის შესატანად. Name - ელემენტების სახელებია. Size - ტექსტში შესატანი ველების ზომა. Maxlength - სიმბოლოების ის მაქსიმალური რაოდენობაა, რომლის შეტანაც შეუძლია აღნიშნულ ველებში ანკეტის შემვსებს.

ჩვენს მიერ შექმნილი ფორმა, ჯერჯერობით ასე გამოიყურება:

**Your name**

**Your E-mail:**

როგორც ხედავთ, საბოლოო შედეგამდე ჯერ კიდევ შორია. თქვენთვის ასევე ცნობილია, როგორ უნდა შეიქმნას Send და Cancel ღილაკები:

```
<input type = Submit value = "Send">
<input type = reset value = "Cancel">
```

მიიღება შემდეგი სახის ფორმა:

**Your name**

**Your E-mail:**

ფორმაში, რომლის შექმნასაც ვაპირებთ, სხვა ჩვენთვის აქამდე ცნობილი ელემენტი არ არის. ასე რომ, გავლილი მასალის გამეორების სასიამოვნო პროცესი დასრულებულია. იწყება ახალი მასალის შესწავლა. დავიწყოთ ე.წ. “გადამრთველებით”:

You are: Male o Female o

ფორმის ამ ელემენტის თავისებურება მდგომარეობს იმაში, რომ ისინი მომხმარებელს აძლევენ არჩევის საშუალებას: ან-ან. განვიხილოთ კოდი, რომელიც უნდა დაემატოს ჩვენს მიერ შექმნილ ფორმას:

```
<b> You are: </b>
```

```
Male <input type = radio name = "sex" value = "Male">
```

```
Female <input type = radio name = "sex" value = "Female">
```

```
<br> <br>
```

ამ კოდის დამატებით მიიღება შემდეგი სურათი:

**Your name**

**You are:** Male  Female

**Your E-mail:**

ამრიგად, “გადამრთველები” ფორმაში შემოდიან თქვენთვის უკვე ნაცნობი ტეგ `<input>`-ის მეშვეობით. მიაქციეთ ყურადღება, რომ “გადამრთველებისათვის” სახელი ერთიდაიგივე იყოს. რადგან ისინი ფაქტიურად ერთ ელემენტს წარმოადგენენ. Value, ცხადია, რომ უნდა იყოს სხვადასხვა.

გადავიდეთ ფორმის აქამდე უცნობ ელემენტებზე, ე.წ. “ალმებზე”. ეს ელემენტიც იძლევა არჩევანის გაკეთების საშუალებას. ე.წ. “ალმები” “გადამრთველებისაგან” განსხვავდება იმით, რომ აქ მომხმარებელს შეუძლია მონიშნოს როგორც ერთი, ასევე რამდენიმე ვარიანტი ერთდროულად. “გადამრთველები” კი, როგორც უკვე იცით, შემოთავაზებული ვარიანტებიდან მხოლოდ ერთის მონიშნვის საშუალებას იძლევა.

ქვემოთ მოცემულია შესაბამისი კოდი და მიღებული შედეგი:

```
<html>
```

```
<head><title>input1.html</title></head>
```

```
<body>
```

```
<form name = "anketa" method = "post" action = "http://mysite.ge/script">
```

```
<b> Your name </b> <br>
```

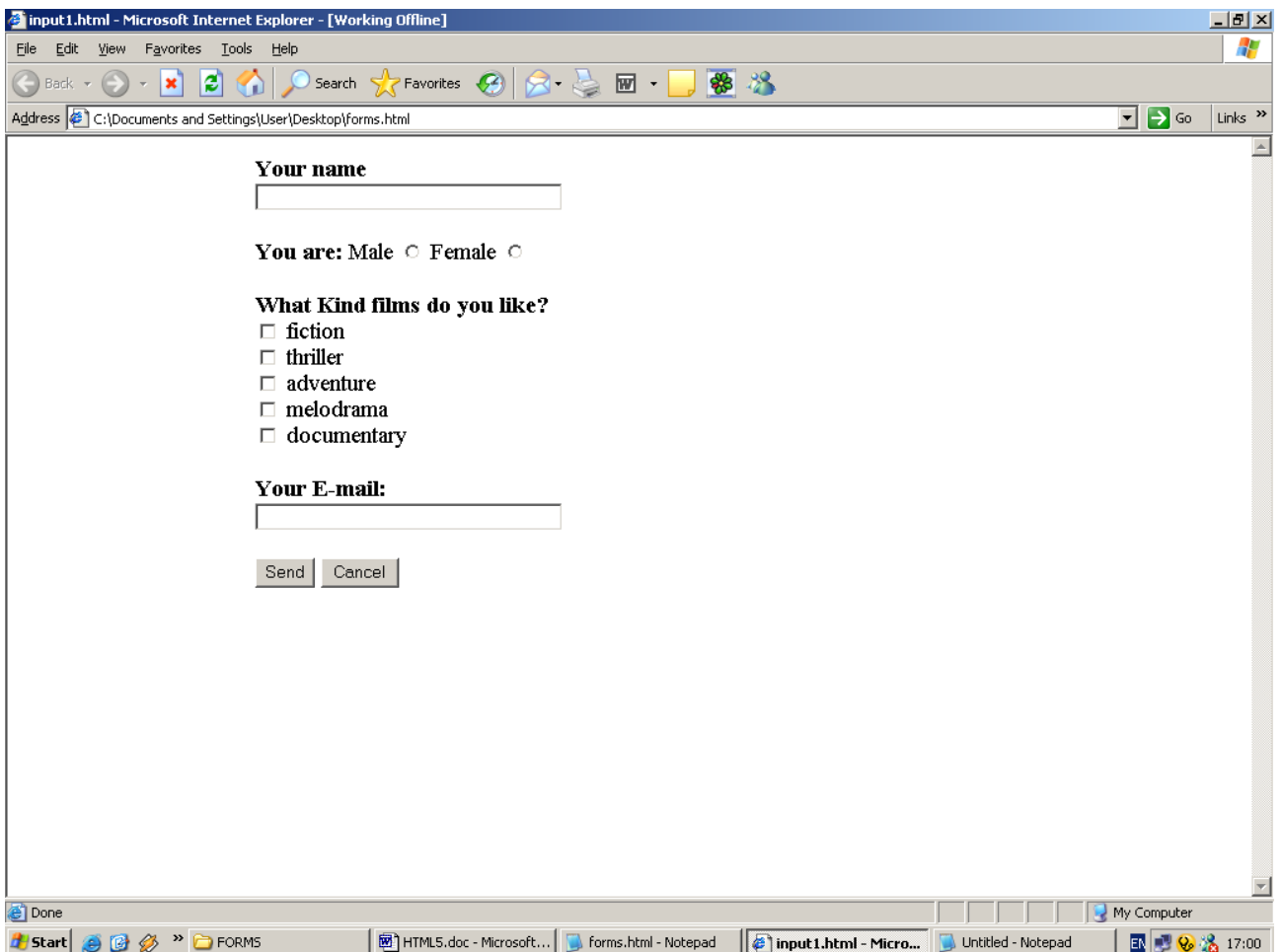
```
<input type = text name = "fns" size = 37 maxlength = 30> <br> <br>
```

```
<b> You are: </b>
```

```

Male <input type = radio name = "sex" value = "Male">
Female <input type = radio name = "sex" value = "Female">
<br> <br>
<b> What Kind films do you like?
</b> <br>
<input type = checkbox name = "fiction" value = "yes"> fiction <br>
<input type = checkbox name = "thriller" value = "yes"> thriller <br>
<input type = checkbox name = "adventure" value = "yes"> adventure <br>
<input type = checkbox name = "melodrama" value = "yes"> melodrama <br>
<input type = checkbox name = "documentary" value = "yes"> documentary <br> <br>
<b> Your E-mail: </b> <br>
<input type = text name = "email" size = 37 maxlength = 30"><br><br>
<input type = Submit value = "Send"><input type = reset value = "Cancel">
</form>
</body>
</html>

```



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

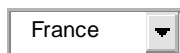
თუ გსურთ, რომელიმე ალაში ან გადამრთველი ავტომატურად იყოს მონიშნული, კოდში ამ “აღმისთვის” ან გადამრთველისთვის შეიტანეთ ატრიბუტი `checked`:

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

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

გადავიდეთ თქვენთვის უცნობ შემდეგ ელემენტზე:

**In What county would you like to go:**



```
<b> In What county would you like to go:
```

```
</b>
```

```
<br>
```

```
<select name = "country">
```

```
<option value = "France"> France
```

```
<option value = "Japan"> Japan
```

```
<option value = "England"> England
```

```
<option value = "Italy"> Italy
```

```
<option value = "China"> China
```

```
</select>
```

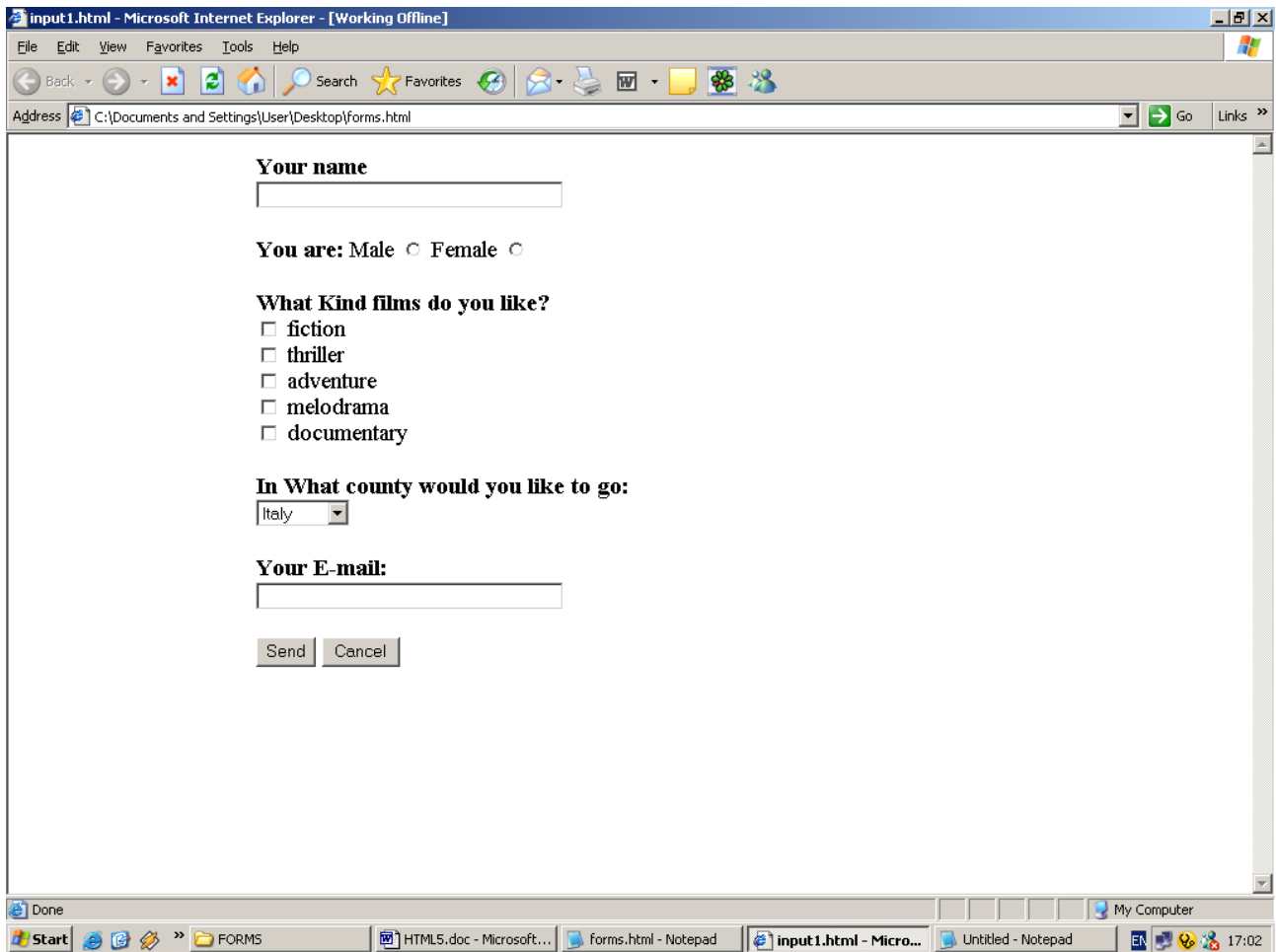
```
<br>
```

```
<br>
```

სიაში პირველი პუნქტი ჩანს ეკრანზე როგორც არჩეული. თუმცა ატრიბუტი `selected`-ის საშუალებით, შესაძლებელია ეკრანზე, როგორც უკვე არჩეული, გამოჩნდეს ის ქვეყანა, რომლისთვისაც ამ ატრიბუტს მივუთითებთ.

```
<option value = “Italy” selected> Italy
```

მიიღება შემდეგი სურათი:



დარჩა ფორმის მხოლოდ ერთი ელემენტი. კოდი იწერება შემდეგნაირად:

```
<b> Who is your favorite actor? </b>  
<br>  
<select name: "actor" size = "4">  
<option value ="Ninidze"> Merab Ninidze  
<option value ="Kipshidze"> Zurab Kipshidze  
<option value ="Eliava"> Lia Eliava  
<option value ="Koberidze"> Otar Koberidze  
<option value ="Archvadze"> Tengiz Archvadze  
</select>
```

ატრიბუტი `size` მიუთითებს სიაში იმ სტრიქონების რაოდენობაზე, რომელიც ეკრანზე ჩანს. ჩვენს შემთხვევაში, `size = 4`. ეკრანზე ჩანს ოთხი სტრიქონი.

საბოლოოდ, მთლიანი კოდი მიიღებს სახეს.

```
<html>
<head>
<title>input1.html</title>
</head>
<body leftmargin=200>
<form name = "anketa" method = "post"
action = "http://mysite.ge/script">
<b> Your name </b> <br>
<input type = text name = "fns" size = 37 maxlength = 30> <br> <br>
<b> You are: </b>
Male <input type = radio name = "sex" value = "Male">
Female <input type = radio name = "sex" value = "Female">
<br> <br>
<b> What Kind films do you like?
</b> <br>
<input type = checkbox name = "fiction" value = "yes"> fiction <br>
<input type = checkbox name = "thriller" value = "yes"> thriller <br>
<input type = checkbox name = "adventure" value = "yes"> adventure <br>
<input type = checkbox name = "melodrama" value = "yes"> melodrama <br>
<input type = checkbox name = "documentary" value = "yes"> documentary <br> <br>
<b> Who is your favorite actor? </b> <br>
<select name: "actor" size = "4">
<option value ="Ninidze"> Merab Ninidze
<option value ="Kipshidze"> Zurab Kipshidze
<option value ="Eliava"> Lia Eliava
<option value ="Koberidze"> Otar Koberidze
<option value ="Archvadze"> Tengiz Archvadze
</select>
<br>
<br>
<b> In What county would you like to go:
</b> <br>
<select name = "country">
<option value = "France"> France
```

```

<option value = "Japan"> Japan
<option value = "England"> England
<option value = "Italy" selected> Italy
<option value = "China"> China
</select>
<br> <br>
<b> Your E-mail: </b> <br>
<input type = text name = "email" size = 37 maxlength = 30"><br><br>
<input type = Submit value = "Send">
<input type = reset value = "Cancel">
</form>
</body>
</html>

```

ქვემოთ განხილული კოდების კომპიუტერული რეალიზაცია დაგეხმარებათ უკეთ აითვისოთ ზემოთ გადმოცემული მასალა ფორმებზე. ქვემოთ განხილული კოდი ქმნის სიას სახელწოდებით occupation, რომელსაც აქვს ხუთი ოფცია: programmer, doctor, actor, chemist, teacher.

მაგალითი I:

```

<html>
<head>
<title> Select1 </title>
</head>
<Body>
Your Speciality:
<form>
<select name = "occupation">
<Option selected value = "programmer"> Programmer
<option value = "doctor"> Doctor
<option value = "chemist"> Chemist
<option value = "actor"> Actor
<option value = "teacher"> Teacher
</select>
</form>
<body>
</html>

```

ამ კოდში select name = "occupation"-ის შეცვლა ბრძანებით select multiple name = "occupation" მოგვცემს ფანჯარას, რომელშიც გამოჩნდება ერთდროულად სიის რამდენიმე წევრი.

გარდა არჩევანის გაკეთებისა, მკითხველებს უნდა ჰქონდეთ საშუალება თვითონაც დაუმატონ ახალი ელემენტი.

```
<html>
<head>
<title> Select1 </title>
</head>
<Body>
Your Speciality:
<form>
<select name = "occupation">
<Option selected value = "programmer"> Programmer
<option value = "doctor"> Doctor
<option value = "chemist"> Chemist
<option value = "actor"> Actor
<option value = "teacher"> Teacher
<option value="other">Other. . .
</select>
</form>
<body>
</html>
```

ქვემოთ მოცემული მაგალითი საშუალებას იძლევა პაროლის შესაბამისი სიმბოლოები ვარსკვლავებით შევცვალოთ.

მაგალითი II:

```
<html>
<head>
<title>Password</title>
</head>
<body>
<form>
Enter the secter word
<input type="password" name="secret_word" size="30"
```

```
Maxlength="30">
</form>
</body>
</html>
```

მაგალითი III:

```
<html>
<head>
<title> Radio1.html </title>
</head>
<body>
Form # 1:
<Form>
<input type = "radio" Name = "choice" value = "choice 1"> Yes
<input type = "radio" Name = "choice" value = "choice 2"> No
</form>
<HR>
Form # 2:
<Form>
<input type = "radio" Name = "choice" value = "choice 1" checked> yes
<input type = "radio" Name = "choice" value = "choice 2"> No
</form>
</body>
</html>
```

ამ კოდის მოდიფიკაციით მივიღებთ:

```
<html>
<head>
<title> Radio 2.html </title>
</head>
<body>
<Form>
One choice: <Br>
<Input type = "radio" name = "choice " value = "choice 1" checked> (1)
<Input type = "radio" name = "choice 1" value = "choice 2"> (2)
```

```

<Input type = "radio" name = "choice 1" value = "choice 3"> (3)
<Br>
One choice: <Br>
<Input type = "radio" name = "choice 2" value = "choice 1" checked> (1)
<Input type = "radio" name = "choice 2" value = "choice 2"> (2)
<Input type = "radio" name = "choice 2" value = "choice 3"> (3)
<Input type = "radio" name = "choice 2" value = "choice 4"> (4)
<Input type = "radio" name = "choice 2" value = "choice 5"> (5)
</form>
</body>
</html>

```

მაგალითი IV: <html>

```

<head>
<title> Reset.html </title>
</head>
<body>
<Form>
<Input type = "reset"><br>
<Br>
<Input type = "reset" value = "clear that form!"><br><br>
<Input type = "submit" value = "send data!">
</form>
</body>
</html>

```

მაგალითი V:

```

<html>
<head>
<title> Form Design </title>
</head>
<body>
<H1> Forms</H1>
<form>
please enter the new title for the message: <Br>
<Input name = "name" Size = "40">

```

```

<HR>
Your comments: <Br>
<Textarea Rows = "6" Cols = "70"> </textarea>
</form>
</body>
</html>

```

მაგალითი VI. წინასწარ ფორმატირებული ტექსტის გამოყენება:

```

<html>
<head><title> Form Design </title></head>
<body>
<H1> Using PRE Tags </H1>
<form>
<pre>
Name:           <Input type = "text" Name = "name" size = "50">
E-mail:         <Input type = "text" Name = "email" size = "50">
Street Address: <Input type = "text" Name = "street 1" size = "30">
                <Input type = "text" Name = "street 2" size = "30">

City:           <Input type = "text" Name = "city" size = "50">
State:          <Input type = "text" Name = "state" size = "2">
Zip:            <Input type = "text" Name = "zip" size = "10">
</PRE>
</form>
</body>
</html>

```

სხვა მეთოდს ფორმების ველების გასწორებისათვის წარმოადგენს ცხრილების გამოყენება. ეფექტი ისეთივე იქნება, როგორც წინასწარ ფორმატირებული ტექსტის გამოყენებისას:

მაგალითი VII:

```

<html>
<head><title> Form Design </title></head>
<body>
<H1> Using Html tables </H1>
<form>

```

```

<table>
  <TR> <TD> Name: </TD> <TD> <Input type = "text" Name = "name" Size = "50">
  </TD> </TR>
  <TR> <TD> E-mail: </TD> <TD> <Input type = "text" Name = "email" size= "50">
  </TD> </TR>
  <TR> <TD> Street Address: </TD> <TD> <Input type = "text" Name ="street 1"
  size = "30"> </TD> </TR>
  <TR> <TD> </TD> <TD> <Input type = "text" Name = "street 2" size = "30"> </TD> </TR>
  <TR> <TD> City: </TD> <TD> <Input type = "text"
  name = "city" size = "50"> </TD> </TR>
  <TR> <TD> State: </TD> <TD> <Input type = "text"
  Name = "state" size = "2"> </TD> </TR>
  <TR> <TD> Zip: </TD> <TD> <Input type = "text"
  Name = "zip" size = "10"> </TD> </TR>
</table>
</form>
</body>
</html>

```

თუ თქვენ ქმნით დიდსა და რთულ ფორმას, მაშინ აზრი აქვს მის ლოგიკურ ნაწილებად დაყოფას. ამ მიზნით შეიძლება აბზაცის <p> ტეგის გამოყენება.

ხშირად ფორმების სასურველი კომპონირებისათვის საკმაოდ მოხერხებელია სიების ტეგების გამოყენება.

ფორმის კომპონირება სიების ტეგების გარეშე:

მაგალითი VIII:

```

<html>
  <head><title> Form Layout and Design </title></head>
  <body>
    <H1> checkboxes and Radio buttons </H1>
    <Form>
      What programming languages do you use? <Br>
      <Input name = "size" Type = "radio" Value = "sm"> C++
      <Input name = "size" Type = "radio" Value = "sd"> Pascal
      <Input name = "size" Type = "radio" Value = "lg">Basic
      <Input name = "size" Type = "radio" Value = "x" >Cobol
      <Input name = "size" Type = "radio" Value = "xx"> Algol

```

```
</form>
```

```
</body>
```

```
</html>
```

ფორმა სივრცის ტეგების გამოყენებით:

მაგალითი IX:

```
<html>
<head><title> Form Design </title></head>
<body>
<H1> Ordered list Tegn </H1>
<Form>
What are your three favorite books?
<OL>
<LI><Input name = "1st" size = "20">
<LI><Input name = "2nd" size = "20">
<LI> <Input name = "3nd" size = "20">
</OL>
</Form>
</body>
</html>
```

მაგალითი X:

```
<html>
<head><title> Form Layout and Design </title></head>
<body>
<H1> checkboxes </H1>
<Form>
<DL>
<DT> What machines do you work on?
<DD> <Input name = "mac" Type = "checkbox"> Macintosh
<DD> <Input name = "pc" Type = "checkbox"> IBM PC
<DL>
<DT> Unix Workstation
<DD> <Input name = "sun" Type = "checkbox"> Sun
<DD> <Input name = "sgi" Type = "checkbox"> SGI
<DD> <Input name = "next" Type = "checkbox"> NeXT
<DD> <Input name = "aix" Type = "checkbox"> AIX
```

```

<DD> <Input name = "lin" Type = "checkbox"> Linux
<DD> <Input name = "other" Type = "checkbox"> Other ...
</DL>
</DL>
</form>
</body>
</html>

```

მაგალითი XI:

```

<html>
<head><title> Form Layout and Design </title></head>
<body>
<Form>
What combinations?
<table>
<TR> <TD> </TD> <TD> Red </TD> <TD> Blue </TD> </TR>
<TR> <TD> Small </TD> <TD> <Input name = "sr"
Type = "checkbox"> </TD>
<TD> <Input name = "sb" Type = "checkbox"> </TD> </TR>
<TR> <TD> Medium </TD> <TD> <Input name = "mr"
Type = "checkbox"> </TD>
<TD> <Input name = "mb" Type = "checkbox"> </TD> </TR>
<TR> <TD> Large </TD> <TD> <Input name = "lr"
Type = "checkbox"> </TD>
<TD> <Input name = "lb" Type = "checkbox"> </TD> </TR>
</Table>
</form>
</body>
</html>

```

მაგალითი XII:

```

<html>
<head><title> Forms </title></head>
<body>
<H1> Multiple Forms in a document </H1>
<Form>

```

```

What ball would you like? <Br>
<Input name = "size" Type = "radio" Value = "rd"> Red
<Input name = "size" Type = "radio" Value = "bl"> Blue
<Input name = "size" Type = "radio" Value = "gr"> Green
<p>
<Input type = "submit"
</form>
<HR>
<Form>
<table>
<TR> <TD> Name: </TD> <TD> <Input type = "text" name = "name"
size = "50"> </TD> </TR>
<TR> <TD> E-mail: </TD> <TD> <Input type = "text" name = "E-mail"
size = "50"> </TD> </TR>
<TR> <TD> street address: </TD> <TD> <Input type = "text" name ="street 1"
size = "30"> </TD> </TR>
</table>
<p>
<Input type = "submit">
</form>
<HR>
<Form>
<DL>
<DT> How would you like to pay for this?
<DD> <Input name = "pay" Type = "radio" Value = "cash" checked>Cash
<DD> <Input name = "pay" Type = "radio" Value = "check"> check
</DL>
<p>
<Input type = "submit">
</form>
</body>
</html>

```

### ლიტერატურა:

1. Учебник по Html. [www.dvgu.ru/lemoi/lect/](http://www.dvgu.ru/lemoi/lect/)
2. Html & Web - Шаг за Шагом. <http://www.firststeps.ru/>
3. Учебник по Html для чайников. <http://postroika.ru>
4. Учебник по формам. <http://postroika.ru>
5. Html Tutorial. <http://www.w3schools.com/html/>
6. Writing Html. <http://www.mcli.dist.maricopa.edu/tut/lessons.html>
7. Html Tutorial. <http://www.december.com/html/>