# რა არის **HTML?**

HTML5 არის ტექნოლოგია, რომელიც გამოიყენება იმისათვის, რომ აიგოს და წარმოდგენილი იქნას საიტები. ახალი ვერსიის განახლება ოქტომბრის 2014 წელს მოხდა. ეს არის ჯერ ჯეროვნად საბოლოო და სრული მეხუთე ვერსია HTML-სა, რომელიც მსოფლიო ქსელის კონსორციუმის მიერ არის შემუშავებული (*World Wide Web Consortium -W3C*).

| | |
|---|---|
| *HTML* | **1991** |
| *HTML 2.0* | 1995 |
| *HTML3.2* | 1997 |
| *HTML4.01* | 1999 |
| *XHTML* | 2000 |
| *HTML5* | 2014 |

დავიწყოთ მარტივად, წიგნთან მუშაობისას და ენი შესწავლისას, მკითხველს ვურჩევ გამოყენოს პროგრამები, მაგალითად გადმოიწერეთ *google* საშუალებით რომელიმე პროგრამა:

- *Adobe Dreamweaver cc 2017*
- *Microsoft Expression Web*
- *Visual Studio Code*
- *CoffeeCup HTML Editor*
- *notepad++*
- *HTMLPad 2015*
- *Sublime Text*

სწავლისათვის გირჩევთ უბრალოდ *notepad*-ში აკრიფოთ სკრიპტები. (კოდებს რომელსაც დავწერეთ WEB-ტექნოლოგიებში მას სკრიპტები ჰქვია)

*ScriptS* -(სკრიპტები) ეს არის კოდების სცენარი, რის მეშვეობითაც შესაძლებელია დაიწეროს ან შედგეს, სხვადასხვა ვებ-პროგრამირების ენებით ვებსაიტები.

*Notepad*-ნებისმიერ Windows-ს მოყვება, *Windows Xp, Windows 7,Windows 8.1 windows 10*-ს. წინამდებარე ვერსიებს არ ვეხები რადგანაც დღეს დღეისობით XP-საც და მის წინამდებარე ვერსიას აღარ გამოიყენებაა

ვაწვებით *Start Screen*, კლავიატურაზე ანუ ვინდოუსის ლოგოს და ძებნის ველში ვწერთ-*notepad*-ს რომელიც გვიგდებს შესაბამის პროგრამას, ეს იმათთვის, ვინც არ იცოდა. როდესაც სკრიპტს ავკრეფავთ შესაბამის რედაქტორში, არ აქვს მნიშვნელობა *notepad*-ი იქნება თუ *Dreamweaver*-ი მას შენახვის დროს მას გაფართოება უნდა მიუთითოთ *.html.*

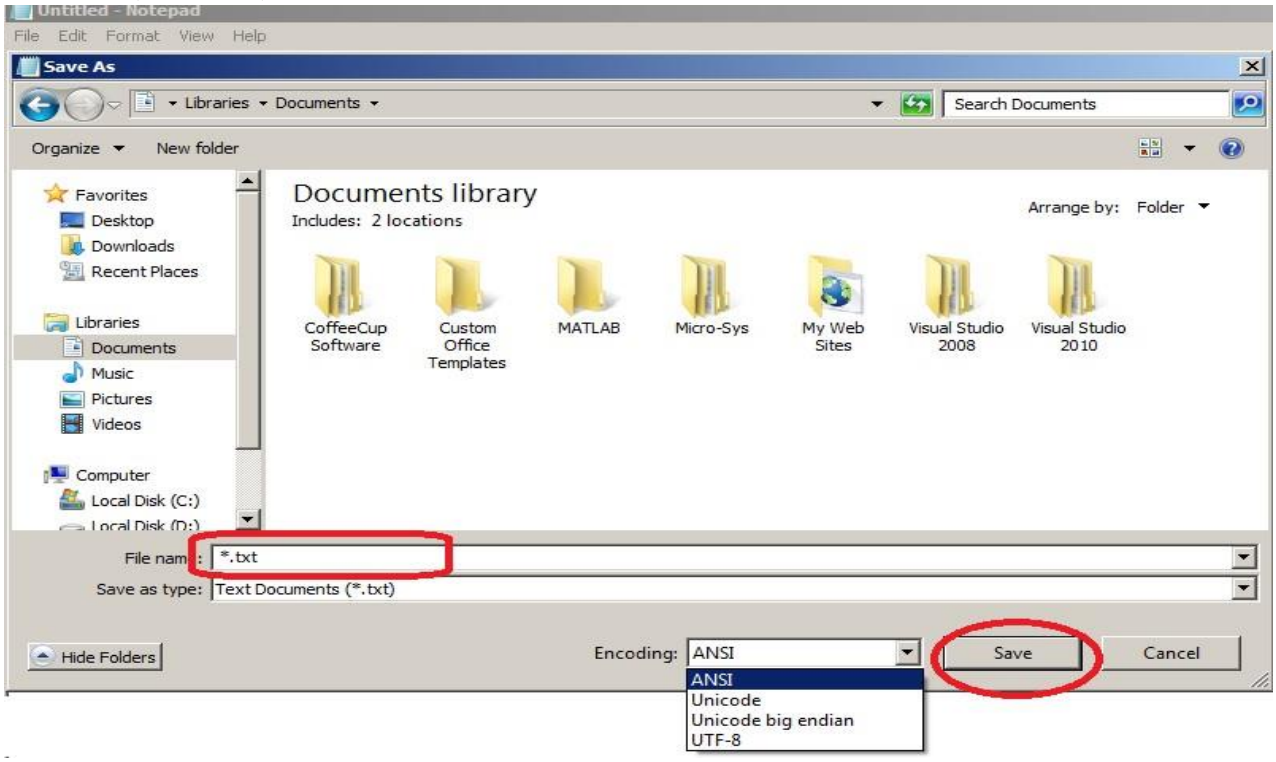
HTML რომელიც ითარგმნება, როგორც ჰიპერტექსტის მარკირების ენა (*Hyper Text Markup Language*). ჰიპერტექსტი განიმარტება როგორც ჩვეულებრივ

ტექსტზე უფრო მეტი ინფორმაციული და ფუნქციური მონაცემების შემცველი დოკუმენტი.

მაგალითი:

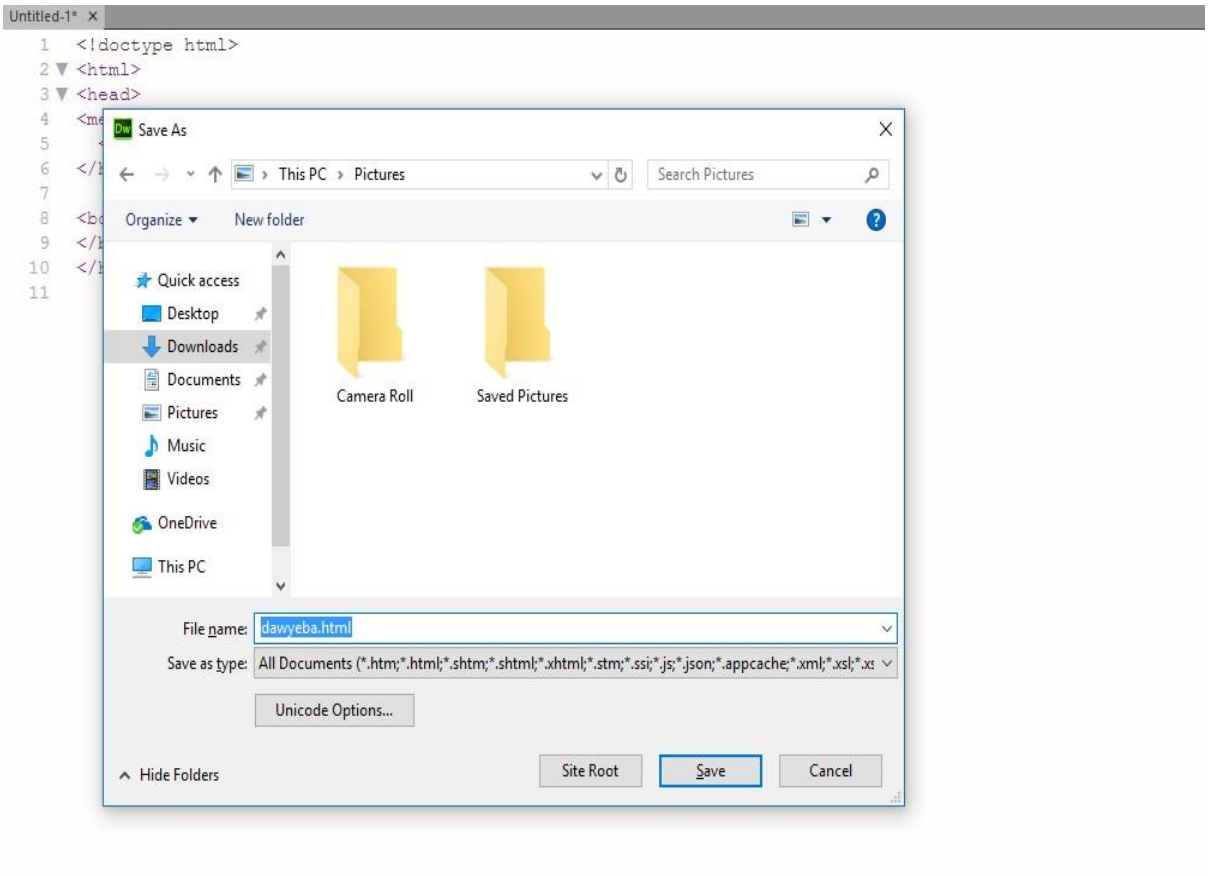
```
<!DOCTYPE html>
<html>
<body>
</body>
</html>
```

შევინახოთ რაიმე სახელით (მაგალითი თუ გსურთ თქვენი სახელი დაარქვით,) მე პირობითად ავიღებ *dawyeba.html*-ს, აუცილებლად სახელს, რომ მივუწერთ .html არ დაგავიწყდეთ.




ქართულად ასოებით აკრეფის დროს დეკოდირება უნდა მივ უთითოთ UTF-8 ხოლო *Dreamweaver*-ში კი, *Modify > Page properties > Title/encoding > Encoding -> UTF-8* და თუ არ ქნა *Edit>Preferences>Fonts > Code View> to Sylfaen > Apply*- და ქართულად დაწერს ნებისმიერი ტექსტი, რომლიც <აქ> მოთავსებულია ეწოდება ტეგი.

რედაქტორები არსებობს, რომელებსაც არ ჭირდება ენკოდირების მითითება და ავტომატურად ინახავს ქართული უნიკოდით, მაგალითად *Visual Studio Code*, რომლის გადმოწერა *microsoft* საიტიდან თავისუფლადაა შესაძლებლი, ნებისმიერი ოპერაციული სისტემისათვის.




```
<!DOCTYPE html>
<html>
<head>
  <title>პირველი გვერდი</title>
</head>
<body>
```

# <h1>ეს არის სათაური</h1>

## <h2>ეს არის სათაური</h2>

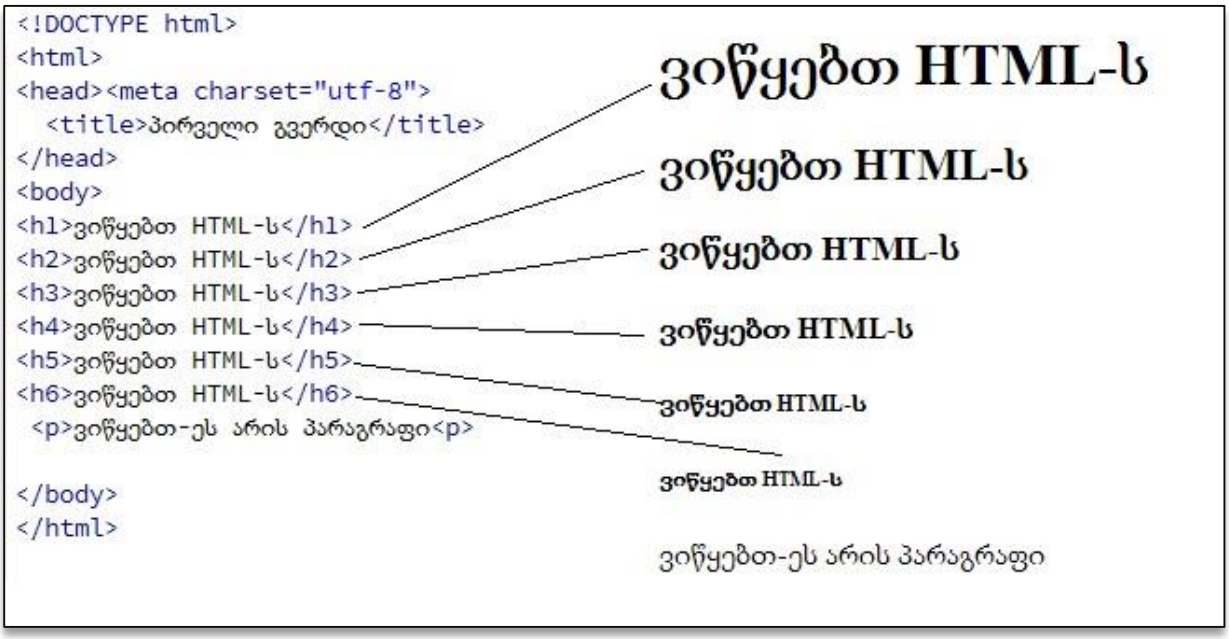
### <h3>ეს არის სათაური</h3>

#### <h4>ეს არის სათაური</h4>

##### <h5>ეს არის სათაური</h5>

###### <h6>ეს არის სათაური</h6>

```
 <p>ვიწყებთ-ეს არის პარაგრაფი<p>
</body>
</html>
```




სურ. 1.2

DOCTYPE განსაზღვრავს HTML დუკემენტის ტიპს, მისი წერის სტრუქტურა ნაჩვენებია სურ1.2 ტეგის ტიპი <html> და </html> აღიწერება HTML დუკუმენტს.

ტეგები *<head>* და *</head>* დუკუმენტის შესახებ გვაწვდის ინფორმაციას.

*<title>* და *</title>* გვაწვდის ინფორმაციას სათაურის შესახებ.

*<body>* და *</body>* შინაარსობრივად აღწერს გვერდს .

*<h1>* დან *<h6>* ამდე , განისაზღვრება სათაურები <h1>-ი არის ყველაზე დიდი ზომის, ხოლო <h6> ანალოგიურად ყველაზე პატარა ზომის სათაური.

HTML 5 ში არის სემტიკური ახალი ტეგები, როგორიცაა მაგალითად *<header>, <footer>, <article>* და *<section>* რომელბმაც გაამარტივა სკრიპტის წერა და საბოლოდ უფრო მოქნილი გახადა ვებ გვერდები ვიდრე 4.01-ის შემთხვევაში, ასევე უნდა აღინიშნოს რომ დაემატა გრაფიკული ელემენტები *<svg>* და *<canvas>*, ასევე მულტიმედიის ელემენტები, როგორიცაა *<audio>* და *<video>* ტეგები.

HTML 5-ში დიდ ინტერესს იწვევს API-ები (ანუ *Application Programming Interfaces*აპლიკაციის პროგრამული ინტერეფეისი), საკითხები, რომლებსაც შევეხებით:

*HTML Geolocation*
*HTML Drag and Drop*
*HTML Local Storage*
*HTML Application Cache*
*HTML Web Workers*
*HTML SSE*

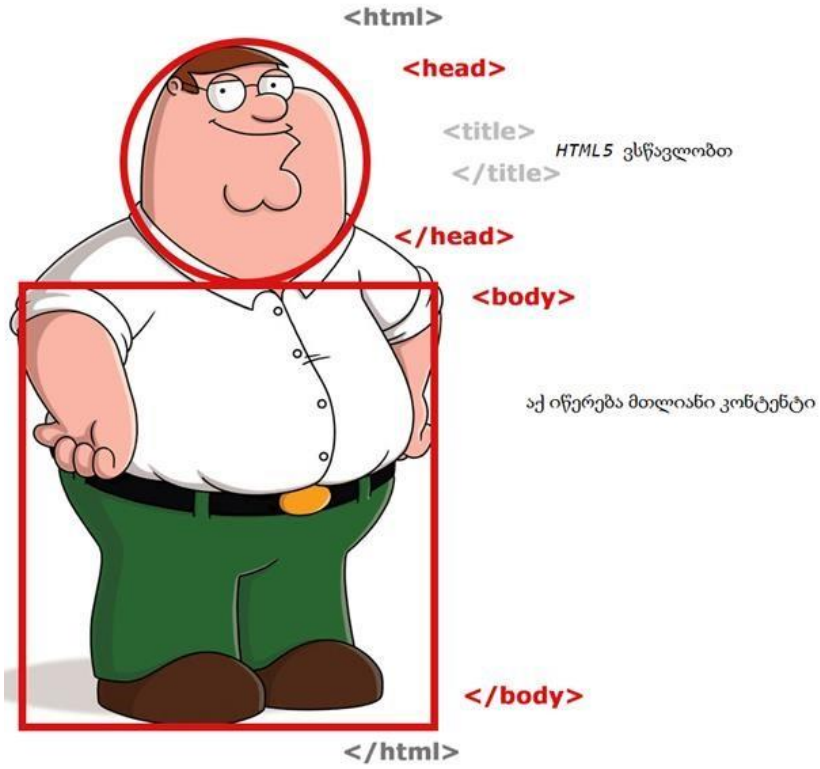
# თავი **I**
## ტეგები და ატრიბუტები

HTML-ში დოკუმენტის შექმნა იწყება ტეგით *<!DOCTYPE html>* ამით ვეუბნებით ბრაუზერს რომ დეკლარირება ხდება HTML-ი დოკუმენტის:

```
<!DOCTYPE html>
<html>
<body>
მოთავსებულ კონტენტს შეუძლია ინტეგრაცია სხვადასხვა ენასთან როგორიცა javascripts, css და სხვა.
</body>
</html>
```

მაგრამ ამის შესახებ ქვემოთ ვისუბრებთ

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




*<!DOCTYPE html>*

ეხმარება ბრაუზერს, რომ სწორად წარმოადგინოს ეკრანზე ინფორმაცია. არსებობს სხვადასხვა ტიპის აღწერა, რომელიც ქვემოთაა მოცემული:

*<!DOCTYPE html>*
*<!DOCTYPE HTML>*
*<!doctype html>*
*<!Doctype Html>*

ყველა ზემოთ მოყვანილი მათგანი დასაშვებია და არ წარმოდგენას ბრაუზერისთვის ან უშუალოდ სკრიპტისათვის შეცდომას.

გადავიდეთ უშუალოდ ტეგებზე, როგორც ზემოთ აღვნიშნე, ამათ შორის მოთავსებულ < სიტყვას> ტეგი ეწეოდება. მაინც რა არის ელემენტი? ელემენტი არის ის, რაც იწერება გახსნილ და დახურულ ტეგში, მაგალითი:

***<a> </a>***

ამ ტეგის მეშვეობით შესაძლებელია გადასვლა ერთი ბმულიდან (ლინკიდან) მეორე ბმულზე (ლინკზე) მაგალითი:

*<a href="http://scripts.ge">* დაკლიკება *</a>* გამოჩნდება „დაკლიკება“ დაჭერის შემდეგ გადავა მითითებულ საიტზე.

სანამ ტეგი დაიხურება, რაიმე ტექსტი უნდა ჩავწეროთ, რომ გავიგოთ ლინკი სად იმყოფება მაგალითი 1.1

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> სალამი ყველას </title>
</head>
<body>
<a href="http://scripts.ge">საიტი</a>
</body>
</html>
```

***href***

ეს გახლავთ დამხმარე ატრიბუტი *<a>* ტეგის, რომელშიც აღიწერება ლინკები, ხოლო მისი სინტაქსი ასე გამოიყურება *<a href="*ლინკი*">*

***<***

***a***

***b***

***b***

***r***

***>***

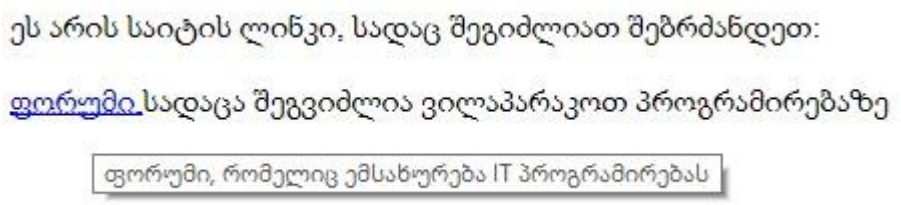
მოცემული ტეგი განსაზღვრავს აბრივიატურას ინფორმაციას რომელსაც საიტზე განვათავსებთ, ან სტატიაში, მაგალითი ავიღოთ:

მაგალითი 1.2

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>სალამი ყველას</title>
</head> <body> ეს არის საიტის ლინკი, სადაც
შეგიძლიათ შებრძანდეთ:
<p>
<a href="http://scripts.ge">
<abbr title="ფორუმი, რომელიც ემსახურება IT პროგრამირებას " >
ფორუმი
```

*</abbr> </a>* სადაცა შეგვიძლია
ვილაპარაკოთ პროგრამირებაზე *</p>*
*</body>*
*</html>*
როდესაც ლინკთან მიიტანთ კურსორს, გამოჩნდება წარწერა „ფორუმი, რომელიც ემსახურება IT-პროგრამირებას“

შედეგი




***<img>***

სურათის ტეგი, სადაც შესაძლებელია სურათების ჩასმა. მასში აუცილებლად მოთავსებული არის ატრიბუტი *src*, რომელიც აღწერს სად იმყოფება სურათი, ის ერთგვარად წარმოადგენს სურათის ლინკს და ზუსტად ეს გახლავთ ატრიბუტი. ქვემოთ მოცემულ *img* ტეგში ასევე გვხდება სურათის სიგანის-*width* და სიმაღლის-*height* ატრიბუტები. მაგალითი:

*<img src="surati.jpg“ width="140" height="342">*

ალბათ, გასაგებია, რომ 140 და 342 მისი ზომებია, საერთოდ კი ატრიბუტის სინტაქსი ასე გამოიყურება:

<ტეგი სახელი=“ცვლადი“>

მოცემულ ტეგს აქვს რამდენიმე ატრიბუტი, რომელსაც HTML5 ის მხარდაჭერა აქვს, HTML4.01-ისგან განსხვავებით, HTML4.01-ში იყო და: *align, border, hspace, vspace* ხოლო დარჩენილი ატრიბუტები გახლავთ:

- ***alt***- ამ ატრიბუტში იწერება ტექსტი, რომელიც შემდეგ არ ჩნდება საიტზე

*<img alt="ტექსტი">*

მის აზრი ის არის რომ ეს ატრიბუტი აღწერს ინფრომაციას სურათის შესახებ, ჩვენ შეგვიძლია საერთოდ არ მიუთითოთ ეს ატრიბუტი და ისე დავწეროთ *<img>* ტეგი

- ***height*** და ***width***- აღწერს სურათის სიმაღლეს და სიგანეს, *height="342"* ხოლო 342 არის პიქსელი, ე.ი სიმაღლისა და სიგანის ზომის ერთეულად აღებულია პიქსელები.
- ***ismap-*** ამ ატრიბუტის გამოყენება ხდება იშვიათად, ძირითადად სურათზე დაკლიკების შედეგად წარმოადგენს ამ სურათის კოორდინატებს.
- ***Usemap-*** ატრიბუტი მონათასავე ატრიბუტია *<map>* ელემენტის, რომელიც უკავშირდება # სიმბოლოთი, ის მისი სინტაქსი ასე გამოიყურება:

*<img usemap="#saxeli">*

***<head>***

*head* ელემენტი მოიცავს ინფორმაციას დოკუმენტის შესახებ და მასში შესაძლებელია აღიწეროს *meta* ინფორმაცია, *<style>,<base>,<link>,<script>,<noscript>* და *<title>.*

***<meta>***

რაც შეეხება *meta* ტეგს, იგი შეიცავს ინფორმაციას HTML დოკუმენტის შესახებ და არსად საიტზე არ ჩნდება, მისი მიზანი აღწეროს საიტი, მიუთითოს დეკოდირება, თუ საკვანძო სიტყვები, მაგალითად, რა განსხვავება დეკოდირებას შორის *html 4.01* თან შედარებით: ***HTML 4.01:***

*<meta http-equiv="content-type" content="text/html; charset=UTF-8">*

***HTML 5:***

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

ასე რომ, *HTML 5* უფრო მეტად თანამედროვე და დახვეწილია და მარტივად აღწერს *html*ის დოკუმენტს.

ასევე *<meta>* ტეგში შესაძლებელია აღიწეროს საიტი , საკვანძო სიტყვები, ავტორის მითითება. მისი არსი ის არის, რომ საძიებო სისტემება მარტივად მოახდინოს საიტის სკანირება, როგორც შინაარსობრივად, ისე საკვანძო სიტყვებითაც, რომელსაც ვებ სამყაროში ***SEO*** ჰქვია.

- SEO არის „*Search Engine Optimization*" , რომელის მეშვეობითაც შესაძლებელია საძიებო სისტემაში გამოჩნდეს. ამისთვის კი საჭიროა მოვახდინოთ <title>, ტეგში აღიწეროს საიტის დასახელება , *<meta* > ტეგებში კი მოხდეს საკვანძო სიტყვებისა და საიტის აღწერა.
- რაც შეეხება SEO-ს, მოვახდინოთ მოკლე გადახვევა, როდესაც ჩავწერთ რაიმე ტექსტს ძებნის ველში და დავაჭერთ „ენთერ"ს, მივიღებ WEB შედეგების სიას, რითაც შეიცავენ დასმულ კითხვაზე პასუხს ან მიახლოებულია მასთან თემატურად. მომხმარებლები ჩვეულებრივ სტუმრობენ იმ საიტებს, რომლებიც ამ სიის თავში არის, რადგან მათ უფრო შესაბამება და დაისმის კითხვა: რატომაა ზოგი საიტის რეიტინგი უკეთესი ვიდრე სხვების? ეს ხდება ძლიერი მარკეტინგული ტექნოლოგიის მეშვეობით ანუ „საძიებო ოპტიმიზების" მეშვეობით-*(SEO).* სეო არის ტექნიკა, რომელიც ეხმარება საძიებო სისტემას, მოძებნოს და წამოწიოს თქვენთვის საინტერესო საიტი საიტების სიაში სხვა საიტებზე წინ.

მიუხედავად იმისა, რომ ტექნოლოგიური პროგრესი სწრაფად ცვალებადია, საძიებო სისტემები შორსაა ინტელიგენტური არსებებისაგან, რომლებსაც შეუძლიათ შეიგრძნონ კარგი დიზაინის სილამაზე, ან ისიამოვნონ ფილმებში ხმებით და მოქმედებებით. ამის მაგივრად, საძიებო სისტემები საიტების ძებნისას ნახულობენ საიტის კონკრეტულ პუნქტებს (ძირითადად ტექსტს), რომ გამოიტანოს აზრი, რის შესახებაა ესა თუ ის საიტი. ეს მცირე განმარტებები არ არის გასაგები, როგორც ჩვენ ვნახავთ შემდეგ, საძიებო სისტემები ახდენენ რამდენიმე ქმედებას, იმისათვის, რომ მიიღონ ძიების შედეგი-–სკანირება, ინდექსირება, დამუშავება და შედეგის პოვნა.

პირველ რიგში, საძიებო სისტემა ასკანირებს საიტს, რათა რომ ნახოს, რა არის იქ. ეს ამოცანა სრულდება კომპიუტერული პროგრამის მიერ, რომელსაც ქვია *„crawler„* ან **„spider“** (ან *„Googlebot“,* როგორცაა გუგლის შემთხვევაში). *Spider* მიჰყვება ლინკებს ერთი გვერდიდან მეორეზე და აფიქსირებს ყველაფერს, რასაც პოულობს ძებნის პროცესში და პოულობს მრავალი გვერდს (მიახლოებით 20 მილიარდამდე).

მისთვის შეუძლებელია ესტუმროს საიტს ყოველდღიურად და ნახოს, რა ცვლილებები მოხდა ან უკვე არსებულმა გვერდმა ხომ არ შეიცვალა სახე. ზოგჯერ *crawler*-ს არ შეუძლია ესტუმროს თქვენს საიტს მთელი თვის განმავლობაში, როგორც უკვე ვახსენეთ *crawler* არ არის ადამიანი და საძიებო სისტემები რომელზიც ვერ ხედავენ გამოსახულებებს, ფილმებს, ჩარჩოებს, პაროლით დაცულ გვერდებს და კატალოგებს. მას მერე, რაც გვერდი დასკანერდება, შემდეგი ნაბიჯია მისი შემადგენლობის ინდექსირება. ინდექსირებული გვერდი ინახება მონაცემების ბაზაში, საიდანაც მოგვიანებით შესაძლებელია მისი პოვნა. ინდექსაციის პროცესი არის სიტყვებისა და გამონათქვამების განსაზღვრა, რომლებიც ყველაზე კარგად აღწერს გვერდების მნიშვნელობას. ადამიანისათვის შეუძლებელია დიდი ინფორმაციის გადამუშავება, მაგრამ საძიებო სისტემა კარგად ართმევს ამ ამოცანას თავს. ზოგჯერ ისინი შეიძლება არ იგებენ გვერდის სწორ მნიშვნელობას, მაგრამ თუ დაეხმარები (*meta* ტეგის გამოყენებით) მათ მისი ოპტიმიზებით, უფრო მარტივი იქნება, თქვენი გვერდის კლასიფიცირება მოხდეს სწორედ, რათა მიიღოს უფრო მაღალი რეიტინგი. როდესაც საძიებო მოთხოვნა მოდის, საძიებო სისტემა ამუშავებს მას, ის ადარებს საძიებო მოთხოვნაში ინდექსირებული გვერდებით მონაცემთა ბაზაში, თუ ერთზე მეტი გვერდი შეიცავს საძიებო სიას, საძიებო სისტემა იწყებს თითოეული გვერდის შესაბამისობის თვლას მის ინდექსში. არსებობს მრავალი შესაბამისობის გამომთვლელი ალგორითმი. თითოეულ ამ ალგორითმს აქვს განსხვავებული შეფარდებითი წონა ყველა საერთო ფაქტორისათვის, როგორც საკვანძო სიტყვისათვის, ბმულისათვის. ამიტომ სხვადასხვა საძიებო სისტემები იძლევა სხვადასხვა შედეგს, ერთი და იმავე საძიებო სიტყვაზე.

ცნობილი ფაქტია, რომ ისეთი საძიებო სისტემები, როგორიცაა *Yahoo!,* Google, Bing, ა.შ. პერიოდულად ცვლის თავის ალგორითმს და თუ გინდა დარჩე სიის თავში, უკანასკნელ ცვლილებებთან საჭიროა მოახდინო შენი გვერდის ადაპტაცია. ეს არის ერთი მიზეზი (სხვა შენი კონკურენტებია) მუდმივად დახაარჯო ძალისხმევა SEO-ზე, თუ გინდა, იყო სიის სათავეში.

საიტი უფრო სწრაფად ახდენს ინდესაქციას, თუ გავაკეთებთ *sitemap*-ს, გავუწერთ მოკლე აღწერას თუ საკვანძო სიტყვებს *meta* ტეგებში და ასევე საიტს დავარეგისტირებთს *google webmaster tool*-ში ან *bing webmaster tool*-ში ან სხვა საძიებო სისტემებში მარტივია მოხდეს ინფორმაციას საიტის შესახებ. რაც შეეხება საძიებო სისტემის მუშაობის პრინციპს, უნდა ითქვას, რომ ბოლო ნაბიჯი საძიებო სისტემის აქტივიზაციაა არის შედეგის პოვნა. ეს პრინციპულად სხვა არაფერია, თუ არაა ბრაუზერზე მისი ჩვენება.

დაუსრულებელი რაოდენობის ძიების რეზულტატები ხარისხდება უფრო შესაბამისი საიტებიდან ნაკლებად შესაბამის საიტებამდე, თუმცა ყველა საძიებო სისტემის ძირითადი პრინციპების მუშაობისა ერთი და იგივეა, სხვადასხვა საძიებო

სისტემებისათვის სხვადასხვა ფაქტორებია მნიშვნელოვანი, იყო დრო, როდესაც *SEO* -ს სპეციალისტები ხუმრობდნენ, რომ *Bing*-ის ალგორითმი შექმნილია *Google*ის საპირისპიროდ, თუმცა ამაში შეიძლება იყოს ჭეშმარიტები მარცვალიც, თუ შენ (თქვენ) გეგმავ დაიპყრო ერთერთი მათგანი, აუცილებელია ფრთხილად მოახდინო ოპტიმიზება.

არსებობს მრავალი მაგალითი საძიებო სისტემის განსხვავებისა. მაგალითი *Yahoo!* და Bing, საძიებო სიტყვას აქვს დიდი მნიშვნელობა, ხოლო Google-ისთვის ბმულებია ძალიან მნიშვნელოვანი. ასევე Google-ისთვის საიტები არის ღვინოსავით-რაც უფრო ძველია მით უკეთესი,იმ დროს როცა Yahoo-მ საერთოდ არ მიანიჭა უპირატესობა საიტებს. ამასთან უფრო მეტი დროა საჭირო, რომ საიტი შევიდეს Google-ის ტოპ საიტებში, ვიდრე *Yahoo*-ში.

**<!--** კომენატრი**-->** ეს არის კომენტარი, რომელსაც სკრიპტისათის დამალულია.

### *<style>*

აღწერს დოკუმენტის სტილს. ძირითადად სტილი სტატიკური თუ დინამიკური საიტებისა იწერება კასკადურ ენაში რომელსაც CSS ჰქვია *(Cascading Style Sheets)*, და მისი გაფართოება არის *.css*, სკრიპტშივე *head* ტეგში შიგნით მოვათავსოთ *<style>* ლინკი, სადაც აღიწერება საიტის დიზაინი, შიგნით შესაძლებელია ასევე ცალკე ფაილის შექმნა და *<link>*-ით დაკავაშირება შესაბამის მისამართზე, მაგალითი *<link>* თვითონ გვეუბნება, რომ გადამისამართება ხდება.

*CSS* ჩვენი კიდევ დავუბრუნდებით, ჯობს გადავიდეთ შემდეგ ტეგებზე, რაც შეეხება *rel* ატრიბუტს, ის დამხმარე ატრიბუტია, რომელიც ურთიერთქმედებს არსებულ დოკუმენტსა და მიბმულ დოკუმენტს შორის.

### *<base>*

*HTML* არის ერთ-ერთი ტეგი, რომელსაც არ აქვს დამხურავი ტეგი, მისი ატრიბუტებში შედის *href* და *_target*, თვითონ ტეგის მიზანიაა, რომ განთავსდეს ლინკი რომელიც შეიცვას სხვა ლინკებს, ის შეიძლება მხოლოდ ერთხელ განთავსდეს დოკუმენტში.

*_target* ატრიბუტის მიზანია სხვადასხვა ფორმაში გახსნას გვერდები.

| ელემენტები | სინტაქსი |
|---|---|
| *_blank* იხსნება ახალ ფანჯარაში | *<base target="_blank”>* |
| *_parent* იხსნება იმავე გვერდზე | *<base target="_parent”>* |
| *_self* იხსნება მშობლიურ გვერდზე | *<base target="_self”>* |
| *_top* იხსნება სრულიად ახალ გვერზე | *<base target="_top”>* |

**<script>** და ***<noscript>*** ეს ორი ტეგი ორიენტირებულია *javascript* ფაილების წასაკითხავად, ხოლო *<noscript>* თუ ბრაუზერს არ აქვს მისი

მხარდაჭერა შეტყობინების გამოსატანად, მაგრამ თანამედროვე ყველა ბრაუზერს აქვს მისი მხარდაჭერა,ამიტომაც ხშირად აქტიურად არ გამოიყენება.

მაგალითი

1.3 *<!*

*DOCTYPE*

*HTML>*

*<html>*

*<head> <script>*

*document.write("*სალამი

ყველას*")*

*</script>*

*<- script>*თქვენს ბრაუზერს არ აქვს *JavaScript*-ის მხარდაჭერა*</-script>*

* * ბრაუზერს, რომელსაც არ აქვს მხარდაჭერა - *script* ელემენტის გამოჩნდება ის ტექსტი, რაც წერია ტეგებს შორის.* *

*</head>*

*<body>*

*</body>*

*</html>*

ახლა უბრალო ვახდენთ ტეგის დემონსტრირებას, რაც შეეხება ტეგში ჩაწერილი სტრიქონს *document.write()*, მას მოგვიანებით განვიხილავთ.

*** ***

*div* ტეგი html 4.01 -ის გან განსხვავებით *html5* -ში გახდა ერთ- ერთი ძირითადი მთავარი ტეგი, მასში აღიწერება ლინკები, მენიუს ტეგები, ინფორმაცია სხვადასხვა ობიექტზე, ასევე *div* ტეგი გამოიყენება სტილის აღსაწერად, რაც ბლოკების ანუ განლაგების ე.წ „ლეიაუთების“ *(Layouts)* და ასევე კლასების აღსაწერად, მოკლედ ძალიან დიდი და ფართო დანიშნულება აქვს *div* ტეგს.

დავიწყოთ ძალიან მარტივი მაგალითით, შევქმნათ რაიმე ტექსტი, რომელზეც გამოყოფილია და გაფერადებულია სხვადასხვა სიტყვები, ასევე დამატებით გავეცნოთ ახალ ტეგებს, რომლებიც *div* ტეგის გარეშეც მშვენივრად მუშაობს, მაგალითის საფუძველზე ვნახოთ, * * და * * -ს შორის რა განსხვავება.

პარაგრაფი ანუ  მასში მოთავსებულ ტექსტებზეა პასუხისგებელი და ამავე ტექსტის რეგულირება შესაძლებელია მასშივე მითითებული პოზიციით, რომელიც ასე გამოიყურება *HTMLpad 2015*-ში. **left** ტექსტს მარცხენა კიდეზე ასწორებს. ***right*** ტექსტს მარჯვენა კიდეზე ასწორებს.

***center*** ტექსტს გვერდის ვერტიკალური შუახაზის სიმეტრიულად ასწორებს (გამოიყენება სათაურების კეთებისას). ***justify*** იგივეა, რაც *left+right,* ანუ ტექსტს ორივე კიდეზე ასწორებს.

*<body text="#1E90FF" bgcolor="#CFCFCF">* -მოდის გავშიფროთ აქ რაც წერია, ამით ვამბობთ, რომ *body text*-ი უნდა იყოს *#1E90FF*-ფერის ანუ მოცისფრო, ამ კოდს რომელიც დავწერე ***hex color***-ი ჰქვია და ამის თაობაზე ორიოდე სიტყვით ცოტა ქვემოთ განვმარტავთ, რაც შეეხება *bgcolor=““* ეს იგივეა, რაც *background*

*color*-ანუ უკანა ფონის ფერი-ნაცრისფერი, რა თქმა უნდა თქვენ სხვა ფერების მითითებაც შეგიძლიათ და ჩაწეროთ სკრიპტში სიტყვიერად მსგავსი ფერი, (არა მარტო ფერები) საკითხი CSS-სის მეშვეობით წყდება.

*<body text="blue" bgcolor="grey">* მაგრამ ასე უფრო მუქ ფერებს მივიღებთ, მხოლოდ სკრიპტის მიხედვით „მოგესალმებით“ არის ლურჯი, ამიტომაც სკრიპტი მარტო მას გაალურჯებს, რაც შეეხება *<font color*>, აქ გასაგებია რომელ ტექსტზეა ლაპარაკი „ჩემი პირველი ნაბიჯები HTML 5-ში“ იქნება წითლად.

- `<font color=””>` განსაზღვრავს ფონტის ფერს `</font>`

*html4.0.1* ტეგებია *html5*-ს არ აქვს ამ

- *<fontface=”acadnusx”> text</font>* ლათინურად

ტეგების მხარდაჭერა, დაწერილ სიტყვებს გადაიყვანს აკადნუსხურ მაგრამ ბრაუზერი მაინც ფონტში. ქართული ალფავიტით, რა თქმა უნდა წაიკითხავს ნებისმიერი ფონტის მითითებაა შესაძლებელი.

- `<font size=”12”>` ფონტის ზომა `</font>`

```
<p><font size="3" color="red">ტექსტი წითლად 3 ზომის </font></p>
<p><font size="2" color="blue">ტექსტი ლურჯად 2 ზომის</font></p>
<p><font face=" acadnusx " color="green">text mwvaned </font></p>
```

რაც შეეხება ჩვენს *div* ტეგს, მისი მეშვეობითაც შესაძლებელია ფონტის რეგულირებაც, მაგალითი ასე:

```
<div align="center">...</div>
<div align="left">...</div>
<div align="right">...</div>
<div align="justify">...</div>
```

შეიძლება შევამჩნიოთ კიდეც, რომ ტექსტის განლაგება *default*-ად *left*-ზეა დაყენებული. ასევე საგულისხმოა, რომ დამხურავი ტეგი </p>-ს შემდეგ ავტომატურად გადადის ხაზზე, ანუ <br>-ს აკეთებს, მაშინ დაისმის კითხვა, ეს ფუნქცია თუ ხელს გვიშლის?

*br* -იგივეა *break* ანუ გამოტოვება, საჭიროდ არ ჩავთვალეთ ამისათვის დიდი ყურადღება დამეთმო, ერთი სტრიქონით გამოტოვება არის და მეტი არაფერი

გამოვიყენოთ * * და პრობლემაც მოგვარებულია, მაგრამ, როგორც ზემოთ აღვნიშნე, * *-ს გარდა, მასა დაემატა სხვა ფუქნციებიც, მოდი, შევქმნათ სტატიკური საიტი რომელშიც იქნება ქალაქები აღწერილი; თბილისი, თელავი და ბათუმი, ამისათვის დაგვჭირდება * * ები, შემდეგ *<style>* ტეგში მისი დიზაინის აღვწეროთ. სანამ დავუბრუნდებით CSS საფუძვლებს, სკრიპტში არსებულ ტეგებს ავხსნით:

სტატიკური არის საიტი, რომელიც მონაცემთა ბაზას არ იყენებს და ხელით ხდება მაში

სანამ *div* ტეგებს დავუბრუნდებოდით კიდევ რაც უნდა ვიცოდეთ:

-  გასქელებული ტექსტი 
-  დახრილი ტექსტი 
- <tt>ნაბეჭდი ტექსტი</tt>
- <u> გახაზული ტექსტი </u>
- <strike>გადახაზული ტექსტი</strike>
- <s>გადახაზული ტექსტი</s
- <li> განსაზღვრავს სიას</li>
- <ol> გადანომრის დროს<ol>

მაგალითი 1.4

```
<html>
<body>
<b> გასქელებული ტექსტი </b>
<i> დახრილი ტექსტი </i>
<tt>ნაბეჭდი ტექსტი</tt>
<u> გახაზული ტექსტი </u>
<strike>გადახაზული ტექსტი</strike>
<s>გადახაზული ტექსტი</s
<li> განსაზღვრავს სიას</li>
<ol> გადანომრის დროს<ol>
<!-- <ol> გამოიყენება ნუმერაციის დროს, ხოლო <ul> ნომერეციის გარეშე
-->
<ol>

<li> </li>
<li> ჯავახიშვილი</li>
<li> ალასანია </li>
<ol>
<p> დაულაგებელია სია:</p>
<ul>საქართველოს ტექნიკური უნივერსიტეტი </ul>
<ul> ჯავახაშივლი </ul>
<ul> ალასანია </ul>
<li>
</ul>
</ol>
</body>
</html>
```

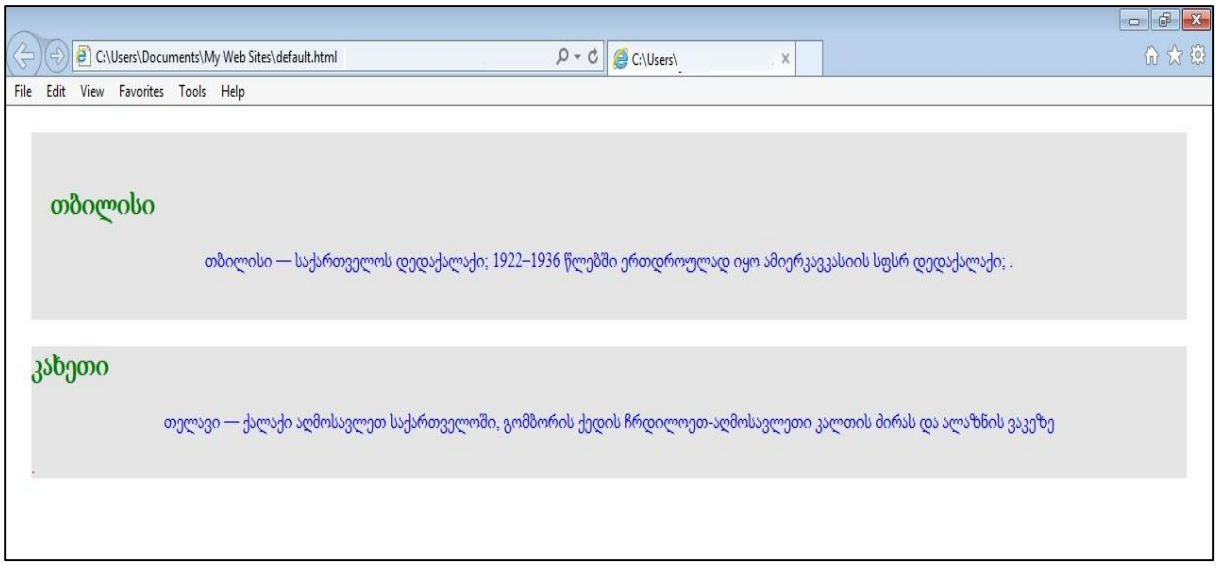
მაგალითი 1.5

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<!--მისი სინატქსია <ol reversed>, რომლის მხარდაჭერა მხოლოდ Internet Explorer-ის ბრაუზერს 11-ის ჩათვლით არა აქვს-->
<ol reversed>
<li> Internet Explorer </li>
<li> Google Chrome   </li>
<li> Mozilla Firefox </li>
</ol>
 <!--სინტაქსი <ol start="number"> და იწყება მაგალითისათვის 2-დან და ატრიბუტია start="" -->
<ol start="2">
  <li>Internet Explorer</li>
<li> Google Chrome</li>
<li>Mozilla Firefox</li>
</ol>
 <!-- რომაული ციფრებით გადანოვმრა იწყება ნუმერაცია 20-დან გადანომრავს 20 დან და მისი ატრიბუტია type=““ -->
<ol type="I" start="20">
  <li>Internet Explorer </li>
<li>Google Chrome </li>
<li>Mozilla Firefox </li>
</ol>
</body>
</html>
```

*div* ტეგის ატრიბუტია ერთ *class=””*, სადაც აღიწერება ჩვენთვის სასურველი სახელი, ასევე * * აქვს კიდევ ერთი *id* ატრიბუტი, რომელიც *class*-სგან განსხვავებით უნიკალურია, შეიძლება გვქონდეს ერთი კლასი მრავალი ელემენტით და ასევე ბევრი ელემენტები ერთი კლასით, ხოლო თითოეულ ელემენტს შეიძლება ჰქონდეს ერთი id და თითოეულ გვერდს შეიძლება ჰქონდეს ერთი ელემენტი *id*-ით ხოლო რაც შეეხება *CSS* მაში ჩაწერის ფორმა სხვადასხვანაირია. მაგალითი

```
<div id="header_id" class="header_class">ტექსტი</div>
#header_id {color:#fff}
.header_class {color:#000}
```

როგორც ვხედავთ, პრეფიქსი # გამოყენება *id*-ის, ხოლო წერტილი-კლასის შემთხვევაში, დაუბრუნდეთ ჩვენს საიტის შექმნას.




მისი სკრიპტი ასე გამოყურება:

მაგალითი 1.6

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style> div.Tbilisi  {
background-color: #e5e5e5;
color:#000000;
margin:20px;
padding:20px; text-align:
justify; } div.Telavi  {
background-color: #e5e5e5;
color:#CC0000;
margin:20px;     padding:
       text-align: justify;
} p{
color:blue;
text-align:
center; } h2 {
color:green; }
</style>
</head>
<body>
<div class="Tbilisi">
<h2>თბილისი
</h2>
<p> თბილისი – საქართველოს დედაქალაქი; 1922–1936 წლებში
ერთდროულად იყო ამიერკავკასიის სფსრ დედაქალაქი;
</p>
</div>
<div class="Telavi">
<h2>კახეთი
```

```
</h2>
<p> თელავი–ქალაქი აღმოსავლეთ საქართველოში, გომბორის ქედის ჩრდილოაღმოსავლეთი კალთის ძირას, ალაზნის ვაკეზე.
</p>
</div>
</body>
</html>
```

- სტილში და სემანტიკაში შედის შემდეგი სახის ტეგები *<style>, , , <header>,<footer>,<main>, <section>, <article>, <aside>, <details>, <dialog>, <summary>* ამ ტეგებს ეტაპობრივად განვიხილავთ, რაც შეეხება სემანტიკას. სემანტიკა სწავლობს სიტყვებს და ფრაზებს ენაში, ზოგი ლიტერატორი და პროგრამისტის მტკიცებით * * და **, ტეგები არ აღწერენ არაინაირ კონტენტს, ამიტომაც არასემატიკური არიან, ზოგნი კი პირიქით ამტკიცებს, ჩვენ არ დავიწყებთ ამ ტეგების ერთი ან მეორე მხარის მტკიცებას, თუმცა ჩვენი აზრით ორივე ტეგი აღწერს დოკუმენტის კონტექსტს და ამიტომაც შევიყვანეთ სემატიკურობაში.