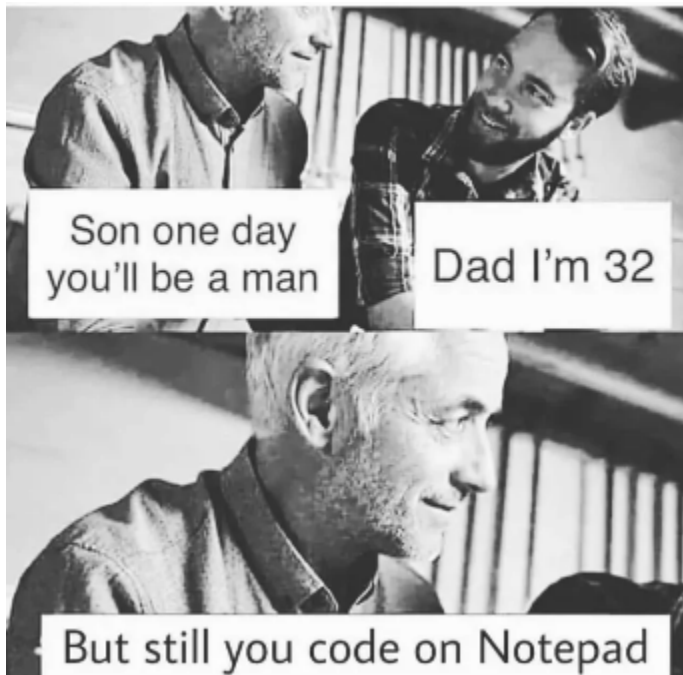


HTML დოკუმენტის ანატომია

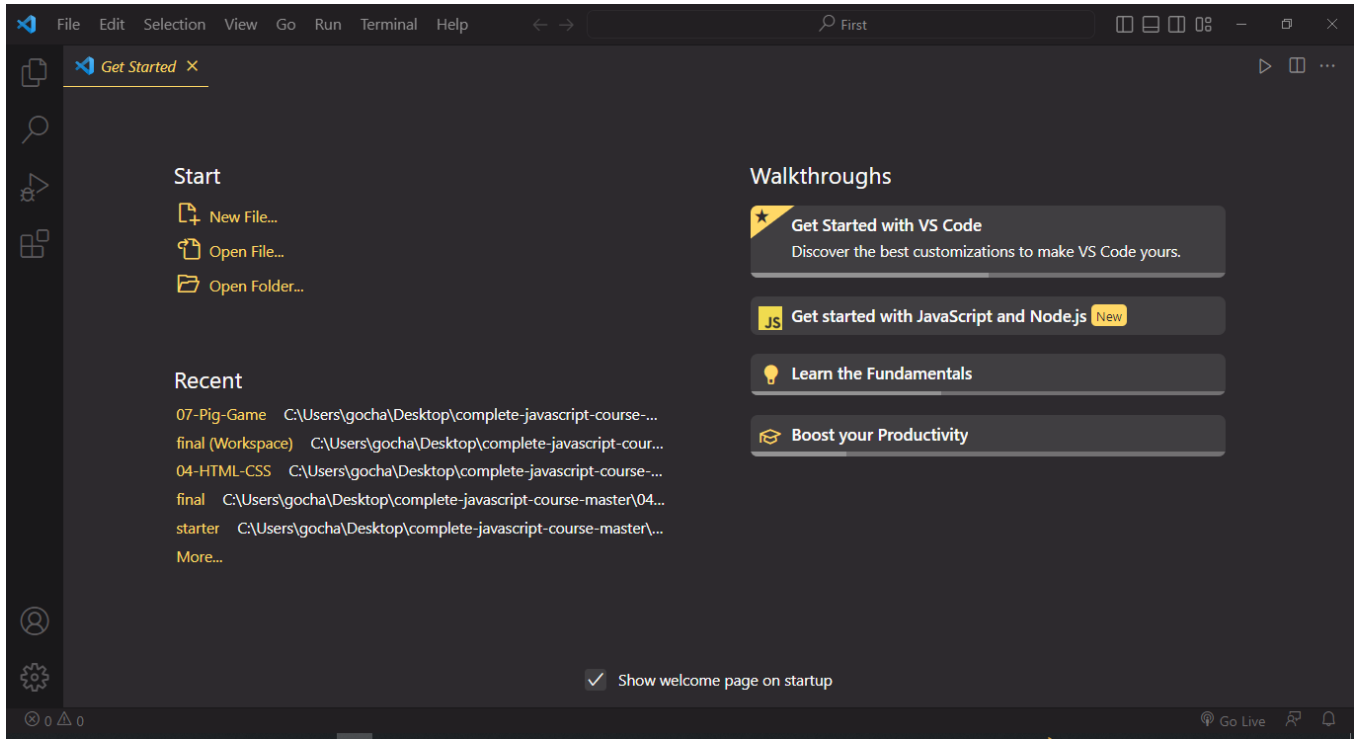
გამარჯობა, შენ უკვე შექმენი ჩემთან ერთად პირველი და ყველაზე მარტივი ვებ-გვერდი, თუმცა ახლა იწყება ნამდვილი თავგადასავალი და მოდი გავაგრძელოთ წინსვლა. სანამ უშუალოდ HTML-ს დავუბრუნდები, უფრო ფართოდ განვმარტავ რა არის და რაში გამოიყენება და ასევე როგორია მისი დოკუმენტის ანატომია, იქამდე დაგამშვიდებ და გეტყვი რომ კოდის წერა არ მოგიწევს Notepad-ში. თანამედროვე სამყაროში არსებობს უშუალოდ კოდის საწერად განკუთვნილი Code editor-ები ანუ იგივე IDE (integrated development environment- ინტეგრირებული პროექტირების გარემო) - პროგრამული აპლიკაცია, რომელიც პროგრამისტებს მრავალმხრივი დამხმარე საშუალებებით უზრუნველყოფს და აგრეთვე პროგრამის გაშვების პროცესსაც უადვილებს.



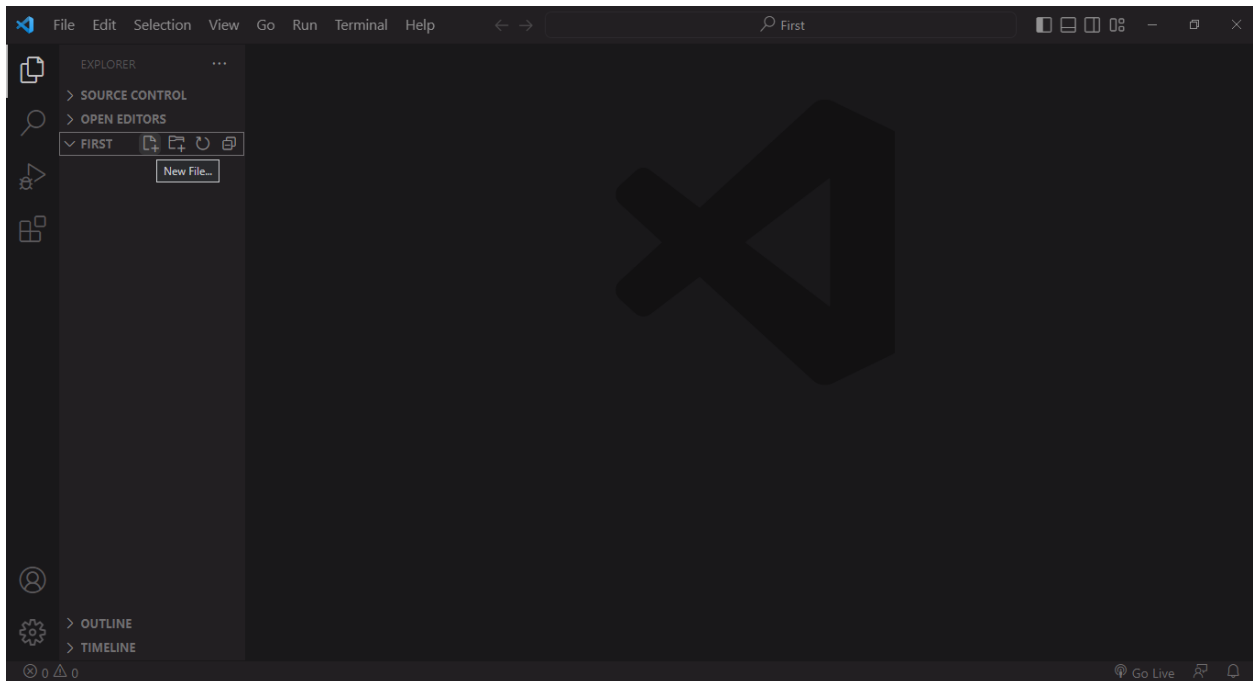
Code editor-ებიდან რამდენიმე ძალიან ცნობილია, მაგალითად Visual Studio (ეს სხვა ედიტორია და სახელის მსგავსებამ არ დაგაბნოს), Sublime Text, ATOM, Eclipse, CLion, IntelliJ, WebStorm და ა.შ, ზოგი მათგანი ფასიანია, მაგრამ ერთ-ერთი ყველაზე ცნობილი, ხშირად გამოყენებადი და თანაც უფასო code editor არის Visual Studio Code, რომელსაც შემოკლებით VSC-ს ვუწოდებთ, რომლის გადმოწერასა და დაინსტალირებასაც მარტივად შეძლებ თუ youtube-ზე უბრალოდ საძიებო ველში ჩაწერ : How to install Visual Studio Code-ს. VSC-ს აქვს უამრავი სხვადასხვა ხელსაწყო (tools), რაც ვებ-დეველოპერებს და არა მარტო, ძალიან გვიმარტივებს მუშაობას, თუმცა სანამ "დავიმსახურებთ" თავის ასე განებივრებას, უმჯობესია notepad-ში წერის სტილს მივყვეთ, ხელით ვწეროთ თითოეული html ტეგი და ბევრი ვივარჯიშოთ, რათა ინფორმაცია ჩვენმა თითებმაც დაიმსხოვროს და შემდგომ კოდის წერისას მის შინაარსობრივ გამართვაზე ვკონცენტრირდეთ ხოლმე და არა მხოლოდ ტეგების სისწორეზე.

იმედია VSC წარმატებით დააინსტალირე, code editor-ის გახსნამდე დესკტოპზე შექმენი ჩვეულებრივი folder, რომელსაც დაარქვი ნებისმიერი სახელი, ამ შემთხვევაში მე დავარქმევ First-ს.

გახსენი VSC და აირჩიე Open Folder, მოძებნე შენ მიერ შექმნილი ფოლდერი და გახსენი ედიტორში.



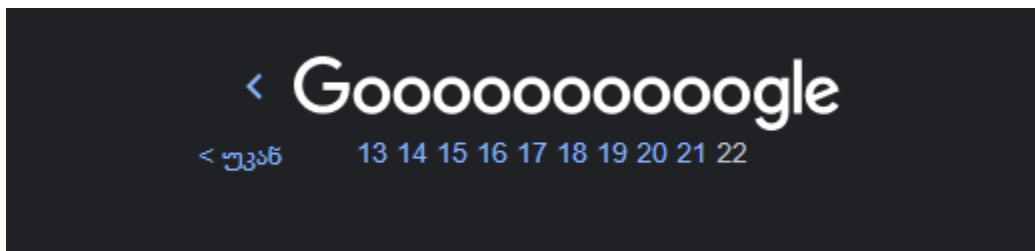
ამის შემდეგ გამოჩნდება ასეთი ფანჯარა, სადაც შენვე შეიძლებ არსებულ ფოლდერში ფაილებისა თუ ახალი ფოლდერების დამატებას.



ახლა რამდენიმე რჩევას მოგცემ, რაც აუცილებლად გამოგადგება : ფოლდერებსა და ფაილებს ნუ დაარქმევ სახელს, რომელიც შეიცავს გამოტოვებას იგივე **space**-ს. თუ მაინცა და მაინც ორი სიტყვით გინდა აღწერო, გამოიყენე - ან _ მათ გამოსაყოფად და არა **space**. ზოგადად მთავარ **html** დოკუმენტს **index.html**-ს ან **homepage.html**-ს არქმევენ ხოლმე, თუმცა შეზღუდული არ ხარ ამ ეტაპზე და რაც მოგესურვება დაარქვი. როდესაც შენ მიერ არჩეულ ფოლდერს გახსნი **VSC**-ში და შემდეგ შექმნი მასში **index.html**-ს თუ სხვა გაფართოებების ფაილებს ან თუნდაც ახალ ფოლდერებს და მასში სხვადასხვა ფაილებს, ეს ყველაფერი შენს დესკტოპზე არსებულ ფოლდერში რა თქმა უნდა აისახება ავტომატურად.

ახლა კი, სანამ უშუალოდ პრაქტიკულ ნაწილს გავაგრძელებთ, მოდი გაგახსენებ და უფრო ფართოდ განვმარტავ თუ რა არის **HTML** და თეორიულად მოგიყვები **html** დოკუმენტის ანატომიაზე, რასაც შემდგომ რა თქმა უნდა გაჩვენებ კიდევ.

HTML - HyperText Markup Language ანუ ჰიპერტექსტული მარკირების ენა არის ის, რის გარეშეც წარმოუდგენელია ვებ-საიტის არსებობა, იქნება ეს ვებ-საიტი ნასას საკუთრება თუ ჩვენ მიერ შექმნილი ერთგვერდიანი ვებ-გვერდი. **2014** წელს დასრულდა მუშაობა და გამოვიდა **HTML5** ანუ **html**-ის ახალი სტანდარტი და ჩვენ სწორედ ამას ვისწავლით. როგორც წინა ბლოგში ვახსენე, **html**-ის საშუალებით ჩვენ ვქმნით საიტის ჩონჩხს. იმისათვის რომ ჩვენ მიერ საიტზე განთავსებული ინფორმაცია, იქნება ეს ტექსტები, სურათები თუ ვიდეოები არ იყოს ქაოტური, იყოს ორგანიზებული და მარტივად გარჩევადი (რომელია მთავარი სათაური, ქვე-სათაური, რომელი პარაგრაფი და ა.შ.), ჩვენ უნდა გამოვიყენოთ კონკრეტული დანიშნულებით შექმნილი ტეგები საკუთარი დანიშნულებისამებრ. მაგალითად თუკი ვებ-გვერდზე გვინდა დიდი ტექსტის განთავსება, ჩავსვათ ეს ტექსტი `<p></p>` ტეგებს შორის, რომელიც სწორედ ამისთვის არსებობს 😊. შეიძლება გაგიჩნდეს კითხვა, იმუშავებს თუ არა `<p></p>` ტეგებში ჩანერილი სათაური. რა თქმა უნდა იმუშავებს! თუმცა არ იქნება სწორი და მარტივად აღქმადი და ეს ჩვენი საიტის ხარისხზე ცუდად აისახება (**google**-ის და სხვა დიდი კომპანიების ე.წ რობოტები ჩათვლიან რომ არ ვიცით ჩვენი საქმე და არც ჩვენ მიერ შექმნილი ვებ-საიტი გამართული და **search**-ში ჩვენი ვებ-საიტი არასახარბიელო პოზიციაზე აღმოჩნდება.

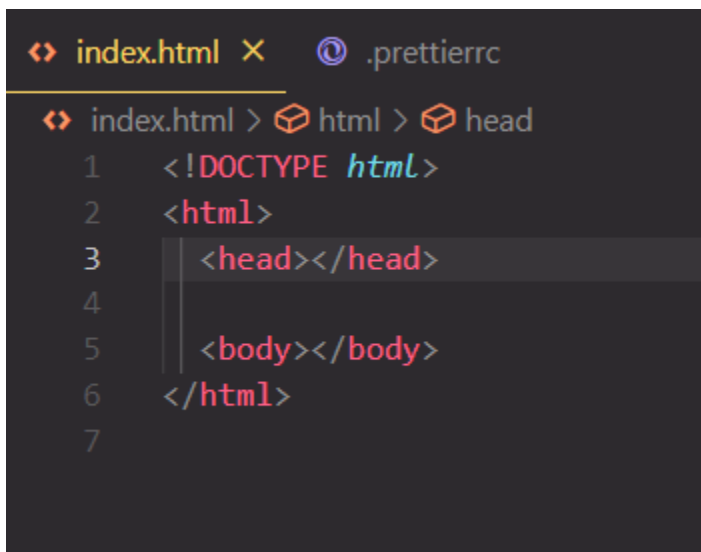


საიტზე არსებული ინფორმაციის სწორ ტეგებში განთავსება გარდა იმისა რომ საიტის **search**-ის წინა პოზიებზე გადმოსვლას უზრუნველყოფს (ზედმეტი თანხის გადახდის გარეშე), ასევე მნიშვნელოვანია მაგალითად უსინათლო ადამიანებისთვის, რომლებიც საიტზე ვიზიტისას იყენებენ ისეთ ტექნოლოგიებს, რაც საიტის შინაარსს სწორედ ჩვენ მიერ სწორ ტეგებში განთავსებული ინფორმაციის მიხედვით გადასცემენ მათ. შესაბამისად როდესაც შეცდომით ვწერთ სათაურს პარაგრაფის ტეგში, ეს ინფორმაციაც შეცდომით

იქნება გადაცემული.იმედია ახლა უკვე კარგად გაიაზრე რატომ არსებობს **html** და რატომ არის ტეგების დანიშნულებისამებრ გამოყენება ასეთი მნიშვნელოვანი.

HTML დოკუმენტის შესაქმნელად ჩვენ დაგვჭირდება ორი მთავარი ელემენტი,ესენია **DOCTYPE** ანუ **Document Type Declaration** და **html** ტეგი,რაშიც განთავსდება ყველა დანარჩენი. **Doctype** იწერება შემდეგნაირად **<!DOCTYPE html>** და ბრაუზერს აწვდის ინფორმაციას დოკუმენტის ტიპზე,ამ შემთხვევაში **html**-ზე,რაც ასევე იმაზე მიუთითებს რომ ჩვენ ვიყენებთ **HTML5**-ს და არა მის რომელიმე წინა ვერსიას.მომდევნო მთავარი,ასე ვთქვათ მშობელი ტეგია, **<html> </html>**, ალბათ გასაგებია და ლოგიკურიც რატომ ქვია მთავარ ტეგს **html** და არა სხვა რამე.

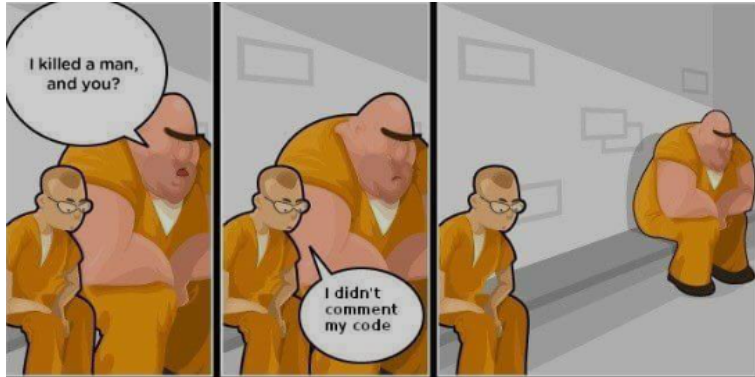
მასში ვათავსებთ ორ მთავარ ელემენტს : **head**-სა და **body**-ს,რომელიც დოკუმენტში გამოისახება ტეგების საშუალებით **<head></head>** და **<body></body>**. კარგად დაიმახსოვრე ეს სტრუქტურა,რადგან ის სტანდარტულია და არასდროს შეიცვლება. **<head></head>**-ი შეიცავს ვებგვერდის მეტამონაცემებს,ასევე მიმართებს სხვა რესურსებზე,იქნება ეს **css**, **javascript** თუ სხვა და ა.შ. ამ ყველაფერს უფრო ღრმად ამავე ბლოგში მოგიყვები და გაჩვენებ კიდევ ვიზუალურად.უბრალოდ დაიმახსოვრე რომ ინფორმაციას,რომელსაც ჩვენს **html** დოკუმენტს **head**-ტეგებში ვანვდით,ვიზუალურად ვერ ვხედავთ,თუმცა თუ რაოდენ დიდი მნიშვნელობა აქვს ამ ინფორმაციას,ცოტა ხანში გაიგებ და ნახავ. იქამდე კი ვისწავლოთ რისთვისაა **body** ელემენტი.მასში თავსდება ვებგვერდის მთელი შინაარსი,ის რასაც შემდეგ ჩვენ ვხედავთ ვიზუალურად.დაიმახსოვრე რომ ერთ **html** დოკუმენტში მხოლოდ თითო-თითო **head** და **body** ელემენტებია და ასევე მათ შორის არაფერი ჩანერო,შეიძლება იმუშავოს და ბრაუზერმა გაპატიოს,თუმცა ეს არასწორია,ჩვენ კი აქედანვე სწორად წერას უნდა შევეჩვიოთ.



```
index.html x .prettierrc
index.html > html > head
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4
5   <body></body>
6 </html>
7
```

შეიძლება შეგემჩნია კიდევ რომ კოდი იერარქიულადაა დალაგებული, მთავარი ტეგები დასაწყისშია,ვგულისხმობ **<!DOCTYPE html>**-სა და **<html></html>**, შემდეგ ცოტატი მარჯვნივაა განეული მასში მოთავსებული ორი თანაბარი მნიშვნელობის ტეგი : **<head></head>** და **<body></body>** და ასე იქნება სულ,თუ თავად არ დაარღვევ ამ პრინციპს ხელოვნურად.ეს გვეხმარება ჩვენც და მასაც,ვისაც მომავალში ჩვენს კოდზე მოუწევს მუშაობა,მარტივად გაარჩიოს და წაიკითხოს ის.

კიდევ ერთი ძალიან კარგი საშუალება,რასაც **html** (და არა მარტო **html**) გვთავაზობს მომავალში გაუგებრობების თავიდან ასაცილებლად,არის **html** კომენტარი. ის პირველ რიგში გვეხმარება ჩვენვე.მაგალითად, მრავალგვერდიანი საიტის შექმნის დროს რთული დანვრილებით გახსოვდეს ყოველ გვერდზე რა დანერე და რატომ დანერე,ასევე შენს კოდზე მომუშავე სხვა დეველოპერს შეგიძლია კომენტარებით გაუადვილო საქმე ან რაიმე სახის მითითება დაუტოვო (მაგ: არ ნამიშალო ეს ნაწილი!!!!!! 😊).



კომენტარი იწერება ნებისმიერ ადგილზე დოკუმენტში,თუმცა რა თქმა უნდა ყველაზე ხშირად მას **body** ელემენტში გამოვიყენებთ, ის არ აისახება ბრაუზერში ასე პირდაპირ,თუმცა თუ მარჯვენა კლიკზე დაჭერით ავირჩევთ **View page source**-ს ან **Inspect**-ს, მის ნახვას ნებისმიერი შევძლებთ,ასე რომ გაითვალისწინე - კომენტარი არ არის საიდუმლო ინფორმაცია.

კომენტარი იწერება **<!-- კომენტარი -->** ასე და ის გამოირჩევა ვიზუალურად დანარჩენი კოდისგან (სულ ერთია რომელი თემა გეყენება **VSC**-ზე.მიხვდები რომ კომენტარი კომენტარია).

```
index.html X .prettierrc
index.html > html
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4
5   <body>
6     <!-- გამარჯობა, მე შენი პირველი კომენტარი ვარ :) ხშირად გამოვიყენე!..-->
7   </body>
8 </html>
```

```
index.html X .prettierrc
index.html > html > head > link
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta
6   name="description"
7   content="პროგრამირების კურსები,UI/UX დიზაინის კურსი, Front-end React კურსი"
8 />
9 <meta name="author" content="მიშო ფაილომე" />
10 <meta property="og:image" content="https://novatori.ge/images/main.png" />
11 <meta
12   property="og:description"
13   content="პროგრამირების კურსები,UI/UX დიზაინის კურსი, Front-end React კურსი"
14 />
15 <meta property="og:title" content="ციფრული პროფესიების სკოლა - ნოვატორი" />
16 <link rel="icon" href="https://novatori.ge/images/favicon.svg" />
17 <link rel="stylesheet" href="styles.css" />
18 <title>ციფრული პროფესიების სკოლა - ნოვატორი</title>
19 <script src="my-js-file.js"></script>
20 </head>
21 <body></body>
22 </html>
23
```

ახლა კი ვისწავლოთ რა ხდება და რატომ ხდება **head**-ში და რა ინფორმაციის მატარებლები არიან მასში არსებული სხვადასხვა ტეგები.

Head-ში გვხვდება ბევრნაირი ატრიბუტის მატარებელი ე.წ **meta** ელემენტი.

<meta/> ტეგს ვაყოლებთ ატრიბუტების დახმარებით რაიმე სახის ინფორმაციას, მაგალითად ერთ-ერთი ასეთია **charset="UTF-8"**, ეს არის სიმბოლოების კოდირების სისტემა, რაც საშუალებას გვაძლევს რომ სხვადასხვა სიმბოლოები ტექსტის სახით აისახოს ვებგვერდზე, მაგალითად ლათინური, ჩინური, არაბული სიმბოლოები ან თუნდაც ქართული.

ყველა **meta** ტეგი ე.წ **void** ელემენტია, რადგან მას არ აქვს შიგთავსი კონტენტის სახით, არამედ მას ინფორმაციას ვაყოლებთ ატრიბუტების საშუალებით, საიდანაც ერთ-ერთი ატრიბუტია თავად **content**. მოდი განვიხილოთ რამდენიმე მათგანი. მაგალითად ზოგიერთ **meta** ტეგს ატრიბუტად ვაყოლებთ **name**-ს, სადაც ვუნეროთ თუ რა ტიპის ინფორმაცია უნდა გადაეცეს და **content** ატრიბუტში ვანვდით უკვე უშუალოდ ინფორმაციას.

ყველაზე ხშირად აღწერის (**description**) სახით გადავცემთ ხოლმე **name** ატრიბუტს ინფორმაციას და შემდეგ **google search**-ში ეს ინფორმაცია გვხვდება საიტის აღწერის სახით.

მესამე **meta** ტეგში გადავცემა ინფორმაცია საიტის ავტორის შესახებ.

მომდევნო სამი **meta tag** კი განსაზღვრავს ამ კონკრეტულ შემთხვევაში როგორ გამოჩნდეს **Facebook**-ში გაზიარებისას ჩვენი სასწავლო საიტი ანუ პირველ შემთხვევაში რა ფოტო გამოჩნდეს, მეორე შემთხვევაში რა განმარტება გამოჩნდეს და მესამე შემთხვევაში რა დასახელება. **og:** არის იგივე **open graph** ანუ ეს კონკრეტული არის **facebook**-ის პროტოკოლი, შესაბამისად ყველა **meta** ტეგი, რომელიც **og:** -ით იწყება განკუთვნილია სწორედ **facebook**-სთვის.

ასევე `<head></head>`-ში შენ ხედავ `<link/>` ტეგს.მას გადაეცემა ორი ატრიბუტი : `rel` (იგივე `relationship`) ანუ რა სახის ინფორმაციას ვაწვდით,იქნება ეს “`icon`” თუ “`stylesheet`” და `href` ატრიბუტი,სადაც გადავცემთ ხოლმე რისი ჩვენებაც გვინდა იმ კონკრეტული საიტის, `icon`-ის ან ფაილის მისამართს,როგორც ამ მაგალითებშია.

რაც შეეხება `<title></title>` ტეგს,მასში იწერება ვებსაიტის დასახელება,რომელსაც შემდეგ ბრაუზერები იყენებენ `Tab`-თვის სახელის დასარქმევად.

დღეისათვის ბოლო ტეგია `<script></script>`,რომელიც გამოიყენება `html`-თვის `javascript` ანუ `.js` გაფართოების ფაილების მისაბმელად,რომელსაც გადავცემთ `src (source)` ატრიბუტის დახმარებით.

გილოცავ! შენ უკვე იცი `HTML` დოკუმენტის ანატომია და ასევე `<head></head>`-ში არსებული თითქმის ყველა მნიშვნელოვანი ტეგი და მისი დანიშნულება! თუ უფრო მეტი `meta tag` და მისი დანიშნულებები გაინტერესებს,ყოველთვის შეგიძლია `google search`-ის გამოყენება,თუმცა მგონი დასაწყისისთვის შენ უკვე `HTML` დოკუმენტის ანატომიის მასტერი ხარ! წინ ბევრი საინტერესო თავგადასავალი გველის...