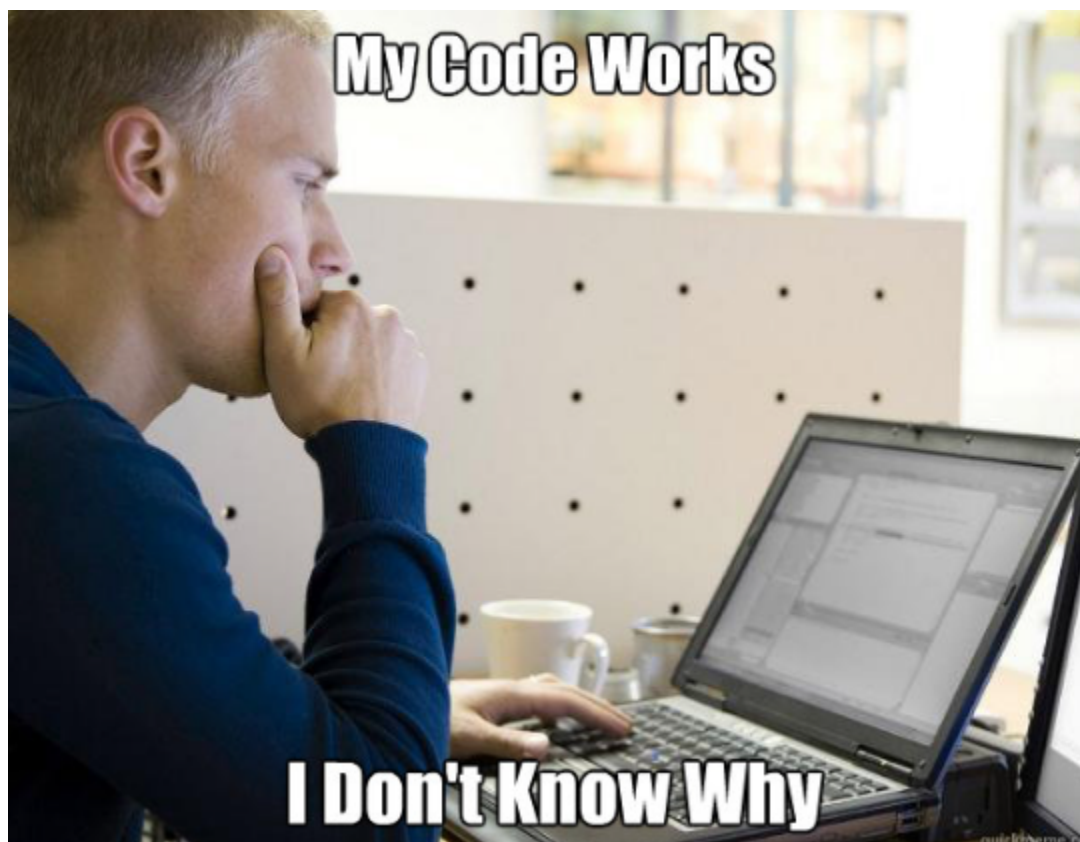


ტექსტური ინფორმაცია HTML- ში

გამარჯობა, მეორე ბლოგიდან უკვე ნაწილობრივ იცი რომ ყველაფერი,მათ შორის ტექსტური ინფორმაციაც,რასაც ჩვენ ანუ მომხმარებლები (users) ბრაუზერში ვხედავთ იწერება `<body></body>` ტეგებს შორის,ხოლო ტექსტური ინფორმაცია HTML-ში შინაარსობრივად გადანაწილდება ტექსტის რაობიდან გამომდინარე, კონკრეტული ინფორმაციის მატარებელი ტექსტი → კონკრეტულ,მისთვის შექმნილ ტეგში (აქ ტექსტის უშუალო შინაარსი არ იგულისხმება,იგულისხმება მაგალითად რა არის კონკრეტული ტექსტი: სათაურია,ქვესათაური,რაიმეს აღწერაა,რაიმეს ჩამონათვალია თუ სხვა) .ისიც იცი თუ რატომაა მნიშვნელოვანი ტექსტური ინფორმაციის ტეგებში სწორად განთავსება და რა მოუვა ჩვენს საიტს თუ ტეგებს არასწორად გამოვიყენებთ (აღმოჩნდება არასასურველ პოზიციაზე google search-ში).

წარმოიდგინეთ როგორი იქნებოდა საიტი,რომ არა სტრუქტურული იერარქიები,რომ არა ტექსტის "სწორ" ტეგებში ასახვის პრაქტიკა,ეს იქნებოდა უბრალოდ ერთი მთლიანი ტექსტი და მისგან რაიმე შინაარსს ძნელად თუ გამოვიტანდით.

დიდი ტექსტების საწერად `html` გვთავაზობს `<p></p>` ტეგს, `p` ანუ იგივე პარაგრაფი,დიდი ალბათობით პირდაპირ `<body></body>`-ში დანერილი ტექსტი ბრაუზერისთვის მაინც ხილვადი იქნება,თუმცა არ დაუშვა ასეთი შეცდომები.ნუ დაემსგავსები ათასობით მოქმედ დეველოპერს,რომელთა კოდი მუშაობს,თუმცა თავადაც არ იციან რატომ! 😊



იმისათვის რომ უკეთ გავიაზროთ ვიზუალურად თუ რაზე ვსაუბრობ, მოდი შევხედოთ ამ ძველი გაზეთის მთავარ გვერდს.



ალბათ დამეთანხმები რომ ენის ცოდნაც კი არ არის საჭირო იმის მისახვედრად რომელია გაზეთის სათაური, რომელი სტატიის სათაურები, სად არის განთავსებული სტატიის ტექსტები, რეკლამა და ა.შ. სწორედ ასეთი, ერთი შეხედვით მარტივად აღქმადი უნდა იყოს ჩვენი ვებ-გვერდიც და გვახვედრებდეს თუნდაც მხოლოდ ვიზუალურად, სად რა არის. ვიზუალური მართალია უშუალოდ ვებ-დეველოპერებს არ გვეხება და საიტის “სწორი” , “user-friendly” დიზაინის შექმნა მთლიანად UI/UX დიზაინერის საქმეა, თუმცა ჩვენ მათ მიერ შექმნილი ვიზუალური გადაგვაქვს კოდის წერის საშუალებით ეკრანზე ანუ ისინი გვაძლევენ საიტის დიზაინს, ჩვენ კი სწორი ტექსტებისა თუ სტილების გამოყენებით ავსახავთ ამ ყველაფერს ბრაუზერში.

მოდი გავეცნოთ ტექსტური ინფორმაციის ასახვისთვის არსებულ ტექსტს html-ში და მათ დანიშნულებებს. ჩვენ უკვე ვიცით ტეგი `<p></p>`, რომელიც გამოიყენება html დოკუმენტში დიდი ტექსტების განსათავსებლად. მოდი ახლა ერთ ჯადოქრობას გასწავლი. იმისათვის რომ დეველოპერები ვარჯიშისას და ჯერაც დაუსრულებელი პროექტების წერისას, ტექსტის მოფიქრებაზე ბევრს არ ვფიქრობდეთ, გვაქვს საშუალება `<p></p>` ტეგებს შორის ჩაწეროთ სიტყვა `lorem`, მასვე მიუზღწეროთ გამოტოვების გარეშე სასურველი სიტყვების რაოდენობა რიცხვით და დავაჭიროთ `Enter`-ს და ჰოი საოცრებავ, ჩვენს `<p></p>` ტეგში გაჩნდება 20 სიტყვისგან შემდგარი ლათინური ტექსტი. `lorem20→Enter`;

```
<p>
Lorem ipsum dolor sit amet consectetur adipiscing elit. Velit quod fugit
consequatur rerum ullam fuga labore saepe veniam placeat quam!
</p>
```

ერთ-ერთი მნიშვნელოვანი ტეგი არის Heading tag ანუ სათაურის ამსახველი ტეგი,რომელიც სულ 6 არსებობს: <h1></h1>, <h2></h2>, <h3></h3>, <h4></h4>, <h5></h5> ,<h6></h6>.

Heading Tags for SEO:

- <h1> Most important
- <h2> Second Most Important
- <h3> Third Most Important
- <h4> Fourth Most Important
- <h5> Fifth Most Important
- <h6> Sixth Most Important

სათაურის ტეგების გამოყენებისას გაითვალისწინე რომ,როგორც წესი,ერთ ვებ-გვერდზე ვიყენებთ მხოლოდ ერთ <h1></h1> ტეგს,რადგან ლოგიკურია რომ ერთ ვებ-გვერდზე მთავარი მნიშვნელობის სათაური მხოლოდ ერთი იყოს (თუ არსებობს რაიმე სახის გამონაკლისი,შესაძლოა ორის გამოყენებაც),უფრო ნაკლები მნიშვნელობის,მაგალითად ქვე-სათაურებს,ვათავსებთ <h2></h2>- ში და ა.შ. თუმცა იშვიათია რომ ერთ ვებ-გვერდზე რაიმეს “ჩაშლა” <h6></h6>-მდე დაგვჭირდეს,მაგრამ თუ დაგჭირდება არ არის პრობლემა,გამოიყენე.

ალბათ შეამჩნევდი რომ სათაურები თავისით იღებენ ზომასა და ოდნავ მუქ ფერს, ჩვენ ამ ყველაფრის ცვლილება მომავალში შეგვიძლია CSS-ის საშუალებით,თუმცა ახლა უბრალოდ დაინახსოვრე რომ html-ში ზოგიერთ ტეგებს მოყვება თავისი, default (ნაგულისხმევი) ინფორმაციები ზომის, ფერის სიმუქის, დანომვრის, უჯრების გაჩენის და ა.შ შესახებ, ისეც და ისეც, შინაარსიდან გამომდინარე.

ახლა დროა გავიცნოთ ტეგები,რომელთა საშუალებითაც შეგვიძლია შევექმნათ სიები. HTML-ში არსებობს ორი სახის სია: **Unordered Lists** და **Ordered Lists**. შინაარსზე უშუალოდ ტეგის სახელებიც მიგვითითებს,პირველი არის unordered ანუ დაუნომრავი სია და მეორე ordered ანუ დანომრილი სია.

Unordered List-ის ამსახველი ტეგია ტეგი ,რომელშიც თავსდება ტეგები. (ალბათ დააკვირდებოდი რომ ტეგის სახელები ლოგიკურია, ნუ შეგეშინდება ახალი ელემენტების გაცნობა,შეიძლება მოგეჩვენოს რომ ეს ზედმეტად ბევრია დასაზებირებლად,თუმცა არ მოგინწევს ტეგების დაზებირება, შენს თითებს ვარჯიშისას

ისინი თავისით დაამახსოვრდება, სხვა შემთხვევაში კი უბრალოდ საჭიროა სწორად და-google-ო.). **Ordered List**-ის ამსახველი ტეგი კი `` ტეგია, რომელშიც ასევე `` ტეგები თავსდება. მოდი ვნახოთ ამ ორივე სიის მაგალითი ცალ-ცალკე და ასევე მათი შერწყმით (ერთმანეთში განთავსებით, რომელზეც ჩვენ უკვე გვსმენია, გაიხსენე **nesting** ანუ ერთი ელემენტის მეორეში განთავსების პრინციპი) მიღებული შედეგიც. შეიძლება ეს ყველაფერი ახლა ვიზუალურად არ მოგეწონოს, რადგან მხოლოდ HTML არც ისე კარგად გამოიყურება და ჩვენ მიჩვეული ვართ ბევრად ლამაზ ვებ-გვერდებს, თუმცა სანამ CSS-ზე გადავალთ საჭიროა ბევრი ვარჯიში და მონდომება.

ეს არის შენი სავარჯიშოს კოდი:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  </head>
  <body>
    <h1>სიები</h1>

    <h2>დაუნომრავი სია</h2>
    <ul>
      <li>გაკვეთილები</li>
      <li>პრაქტიკა</li>
      <li>შეხვედრები</li>
      <li>პროგრესი</li>
    </ul>

    <h2>დანომრილი სია</h2>
    <ol>
      <li>გაკვეთილები</li>
      <li>პრაქტიკა</li>
      <li>შეხვედრები</li>
      <li>პროგრესი</li>
    </ol>
```

```
<h2>სიების ნესტინგის მაგალითი</h2>
<ol>
  <li>
    გაკვეთილები
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>Javascript</li>
    </ul>
  </li>
  <li>
    პრაქტიკა
    <ul>
      <li>კოდის წერა</li>
      <li>დავალებები</li>
      <li>თეორიის ათვისება</li>
    </ul>
  </li>
  <li>
    შეხვედრები
    <ul>
      <li>კვირაში 3</li>
      <li>პროგრამირება</li>
      <li>ინგლისური</li>
    </ul>
  </li>
  <li>
    პროგრესი
    <ul>
      <li>ვარჯიში</li>
      <li>ისევ ვარჯიში</li>
      <li>ვარჯიში=პროგრესი</li>
    </ul>
  </li>
</ol>
</body>
</html>
```

ეს ყველაფერი კი ჩვენს ვებ-გვერდზე გამოიყურება ასე :

სიები

დაუნომრავი სია

- გაკვეთილები
- პრაქტიკა
- შეხვედრები
- პროგრესი

დანომრილი სია

1. გაკვეთილები
2. პრაქტიკა
3. შეხვედრები
4. პროგრესი

სიების ნესტინგის მაგალითი

1. გაკვეთილები
 - HTML
 - CSS
 - Javascript
2. პრაქტიკა
 - კოდის წერა
 - დავალებები
 - თეორიის ათვისება
3. შეხვედრები
 - კვირაში 3
 - პროგრამირება
 - ინგლისური
4. პროგრესი
 - ვარჯიში
 - ისევ ვარჯიში
 - ვარჯიში=პროგრესი

მოდით ახლა რამდენიმე საინტერესო მაგალითსაც გაჩვენებ პირველ ბლოგში ნასწავლი ელემენტების გამოყენებით, რომ მათი ვიზუალური მნიშვნელობაც დაინახო, თუმცა გახსოვდეს რომ ამ ეტაპზე ეს ტეგები აღარ გამოიყენება და მსგავსი ეფექტის მისაღებად ვიყენებთ CSS-ს, მაგრამ ეს არ გვაძლევს იმის ფუფუნებას რომ არ ვიცოდეთ მათი არსებობის შესახებ, რადგან ჩვენ ვსწავლობთ საფუძვლებს და ასევე ყველაფერს საფუძვლიანად!

**** - იგივე **emphasized text** მოქმედებს იგივენაირად როგორც *<i></i>* ტეგი და ტექსტს ხრის.

ასეთივე შემთხვევა გვაქვს -სა და ჩვენ მიერ უკვე ნასწავლ ტეგებში, ორივე ტექსტს ამუქებს.

```
<body>
<!-- როგორც ხედავ შედეგი ორივე შემთხვევაში იქნება ერთნაირი, გამოვიყენებთ <em></em>-სა თუ <i></i>-ს-->
<p>
  მოხარული ვიქნები თუ ჩემი ბლოგი <em>და ა გ ე ხ მ ა რ ე ბ ა</em> და მოგცემს
  სწორ მიმართულებას.
</p>
<p>
  მოხარული ვიქნები თუ ჩემი ბლოგი <i>და ა გ ე ხ მ ა რ ე ბ ა</i> და მოგცემს
  სწორ მიმართულებას.
</p>
<!-- როგორც ხედავ შედეგი ორივე შემთხვევაში იქნება ერთნაირი, გამოვიყენებთ <strong></strong>-სა თუ <b></b>-ს-->
<p>
  მოხარული ვიქნები თუ ჩემი ბლოგი <strong>და ა გ ე ხ მ ა რ ე ბ ა</strong> და
  მოგცემს სწორ მიმართულებას.
</p>
<p>
  მოხარული ვიქნები თუ ჩემი ბლოგი <b>და ა გ ე ხ მ ა რ ე ბ ა</b> და მოგცემს
  სწორ მიმართულებას.
</p>
<!-- მოდი ახლა ერთ პარაგრაფში გამოვიყენოთ რამდენიმე სხვადასხვა ელემენტი-->
<p>
  მოხარული ვიქნები თუ ჩემი ბლოგი <i>და ა გ ე ხ მ ა რ ე ბ ა</i> და
  <b> მოგცემს </b>
  <a>სწორ მიმართულებას.</a>
</p>
</body>
```

ეს ყველაფერი ვებ-გვერდზე გამოჩნდება ასე :

მოხარული ვიქნები თუ ჩემი ბლოგი და ა გ ე ხ მ ა რ ე ბ ა და მოგცემს სწორ მიმართულებას.

მოხარული ვიქნები თუ ჩემი ბლოგი და ა გ ე ხ მ ა რ ე ბ ა და მოგცემს სწორ მიმართულებას.

მოხარული ვიქნები თუ ჩემი ბლოგი და ა გ ე ხ მ ა რ ე ბ ა და მოგცემს სწორ მიმართულებას.

მოხარული ვიქნები თუ ჩემი ბლოგი და ა გ ე ხ მ ა რ ე ბ ა და მოგცემს სწორ მიმართულებას.

მოხარული ვიქნები თუ ჩემი ბლოგი და ა გ ე ხ მ ა რ ე ბ ა და მოგცემს სწორ მიმართულებას.

დღეისათვის ბოლო რჩევაა რომ VSC-ში დააყენო **Live Server**, რასაც მარტივად შეძლებ თუ საძიებო ველში ჩანერ : how to set up live server in visual studio code-ს და დიახ, ხშირად მოგიწევს რალაცების სასწავლად, დასატესტად, პრობლემის გადასაჭრელად თუ საქმის გასამარტივებლად დასერჩვა, ამიტომაც აქედანვე მიეჩვიე.

Live server-ს ჩართავ VSC-ში ფაილის შიგთავსზე მარჯვენა click-ზე გამოტანილი ბრძანება : Open with live server-ის საშუალებით და ის დაგეხმარება VSC-ში დაწერილი და შენახული კოდი ავტომატურად, refresh ღილაკზე დაჭერის გარეშე გამოისახოს ბრაუზერში, რაც უდიდეს დროსა და ნერვებს დაგიზოგავს.

არაფრის 😊
დროებით...