

## VJEŽBA 2 – HTML, CSS

U prošloj vježbi, upoznali smo se s HTML-om (HyperText Markup Language), jezikom za definiranje strukture i sadržaja web aplikacije.

HTML se piše u obliku **HTML elementa** koji se sastoje od oznaka (eng. tag) ugniježđenih između znakova „<“ i „>“ (npr. <html>). HTML oznake uglavnom dolaze u parovima, npr. <h1> i </h1> gdje je <h1> otvarajuća, a </h1> zatvarajuća oznaka. Između otvarajuće i zatvarajuće oznake se nalazi sadržaj tog elementa. Sadržaj može biti bilo kakva kombinacija običnog teksta i drugih html elemenata (npr. <h1>Naslov</h1> kao sadržaj ima čist tekst, dok npr. <div> Tekst <span> tekst </span><span> tekst </span></div> sadrži tekst i dva span elementa). Postoje i HTML elementi koji nemaju sadržaj, npr. br element koji služi za prelazak u novi red. Takvi elementi se mogu definirati pomoću samozatvarajuće oznake (npr. <br/>).

Svaki element može sadržavati i **atribute** pomoću kojih označavamo neka dodatna svojstva ili definiramo funkcionalnosti elemenata. Atributi elementa se definiraju unutar otvarajuće oznake u obliku:

imeAtributa = "vrijednost atributa"

Svaki element može sadržavati proizvoljan broj atributa:

<element attr1="vrijednost atributa 1" attr2="vrijednost atributa 2"></element>

Korištenjem HTML elemenata gradimo HTML stranice. Primjer jednostavne web stranice nalazi se u Tablici 1.

<!DOCTYPE HTML>	← doctype deklaracija koja definira verziju HTML-a
<html>	← glavni html element unutar kojeg se nalazi stranica
<head>	← zaglavlje, uključuje vanjske resurse, meta-informacije
<meta charset="utf-8" />	← meta-informacija, potrebno za korištenje utf-8 znakova
<title>Title</title>	← naslov stranice
</head>	
<body>	← tijelo stranice, prikazuje se korisniku
<h1>Hello World!</h1>	← najveći naslov
</body>	
</html>	

Tablica 1 – Primjer jednostavne web stranice

HTML isključivo služi za definiranje strukture i sadržaja neke web stranice, a za definiranje prezentacijskih svojstava strukture i sadržaja definiranog u HTML-u koristimo **CSS** (Cascading Style Sheets).

# 1. CSS (CASCADING STYLE SHEETS)

CSS (Cascading Style Sheets) je jezik za definiranje prezentacijskih svojstava (izgleda, formatiranja) markup dokumenata.

CSS dokument se sastoji od niza CSS pravila. Na primjer, CSS pravilo koje specificira da će svi paragrafi biti crvene boje se definira na sljedeći način:

```
p { color: red; }
```

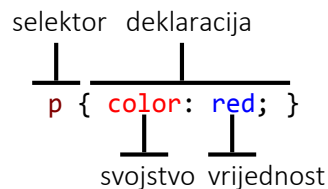
HTML:

```
<p>
Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt
mollit anim id est laborum.
</p>
```

Rezultat:

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua.  
Ut enim ad minim veniam, quis nostrud  
exercitation ullamco laboris nisi ut aliquip  
ex ea commodo consequat. Duis aute irure  
dolor in reprehenderit in voluptate velit  
esse cillum dolore eu fugiat nulla pariatur.  
Excepteur sint occaecat cupidatat non  
proident, sunt in culpa qui officia deserunt  
mollit anim id est laborum.

CSS pravilo se sastoji od dva dijela (Slika 1):



Slika 1. Primjer CSS pravila

**Selektor** definira na koje html elemente se primjenjuju CSS deklaracije. Unutar jednog CSS pravila moguće je specificirati više CSS deklaracija međusobno odvojenih s „;“. Svaka deklaracija se sastoji od CSS svojstva i CSS vrijednosti međusobno odvojenih s „:“.

## 1.1. UKLJUČIVANJE CSS STILOVA

Razlikujemo tri načina uključivanja CSS stilova: *i)* korištenjem *style* html elementa, *ii)* uključivanjem preko *style* atributa html elementa na kojeg želimo primijeniti određeni stil, te *iii)* korištenjem link elementa pomoću kojeg uključujemo vanjsku datoteku s CSS pravilima.

### 1.1.1. Korištenjem *style* html elementa

Uključivanje CSS stilova korištenjem *style* elementa se koristi ako definiramo stilove koji se koriste samo unutar jednog HTML dokumenta. U tom slučaju *style* element se definira unutar zaglavlja dokumenta (*head* element) na sljedeći način:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Title</title>
    <style>
      p { color: red; }
    </style>
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
      tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
      quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
      consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
      cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
      non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>
  </body>
</html>
```

### 1.1.2. Uključivanje preko *style* atributa

CSS stilovi se mogu definirati za pojedinačne HTML elemente korištenjem *style* atributa. *Style* atribut se može definirati na bilo kojem html elementu, a kao vrijednost se navodi niz CSS deklaracija (**svojstvo: vrijednost**, međusobno odvojeni s „;“). Nema potrebe za definiranjem selektora jer se deklaracija primjenjuju na HTML element u kojem se nalazi *style* atribut.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Title</title>
  </head>
  <body>
    <p style="color: red; background-color: gray">Lorem ipsum dolor sit amet,
      consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
      dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
      ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
      dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
      pariatur. Excepteur sint occaecat cupidatat non proident.
    </p>
  </body>
</html>
```

### 1.1.3. Uključivanje vanjske CSS datoteke

Najčešći način uključivanja CSS koda je kroz vanjske CSS datoteke. CSS pravila se definiraju unutar posebne datoteke s ekstenzijom „.css“. Ta datoteka se uključi u HTML dokument korištenjem *link* elementa unutar zaglavlja (*head*). *Link* element mora imati dva atributa: *rel*="stylesheet" s kojim se označi da se radi o css datoteci te *href*="put do datoteke" s kojim se definira točna lokacija css datoteke.

Datoteka: `index.html`

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Title</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur
      adipiscing elit, sed do eiusmod tempor
      incididunt ut labore et dolore magna
      aliqua. Ut enim ad minim veniam, quis
      nostrud exercitation ullamco laboris
      nisi ut aliquip ex ea commodo
      consequat. Duis aute irure dolor in
      reprehenderit in voluptate velit esse
      cillum dolore eu fugiat nulla pariatur.
      Excepteur sint occaecat cupidatat non
      proident, sunt in culpa qui officia
      deserunt mollit anim id est laborum.
    </p>
  </body>
</html>
```

Datoteka: `style.css`

```
p
{
  color: red;
  font-style: italic;
}

body
{
  background-color: gray;
}
```

Osim CSS pravila, unutar css koda koji je smješten ili unutar style elementa ili u posebnoj datoteci, moguće je definirati i CSS naredbe. Npr. naredba za uključivanje vanjske CSS datoteke se može smjestiti na početku CSS koda (ili unutar style elementa ili unutar posebne .css datoteke):

```
@import url("MyStyle.css");
```

## 1.2. GRUPIRANJE CSS PRAVILA

Ponekad se događa da imamo slična pravila, npr s pravilima:

```
h1 { color: red; background: black; }
h2 { color: red; background: black; }
```

definiramo da svi elementi tipa h1 imaju crveni boju teksta i crnu pozadinu, i da svi elementi tipa h2 također imaju crveni boju teksta i crnu pozadinu. U tom slučaju, ta dva pravila je moguće spojiti u jedno pravilo na sljedeći način:

```
h1, h2 { color: red; background: black; }
```

ovo pravilo čitamo na sljedeći način: svi elementi tipa h1 i svi elementi tipa h2 imaju crvenu boju teksta i crnu pozadinu

### 1.3. SELEKTIRANJE PO TIPU ELEMENTA

Najjednostavniji način odabiranja elemenata je po njihovim tipu. CSS pravilo:

```
<h1>Naslov1</h1>
<h1>Naslov2</h1>
<h2>Naslov3</h2>
```

**h1** { color: red; background: black; }

↓  
tip elementa

se primjenjuje na **sve elemente tipa h1**. Ovakav selektor se samo sastoji od imena tipa elementa.

### 1.4. SELEKTIRANJE ELEMENATA PO KLASI I ID-U

Svakom HTML elementu možemo pridružiti dva posebna atributa: **class** – kojim definiramo da taj HTML element pripada određenoj grupi HTML elemenata, te atribut **id** – s kojim definiramo jedinstveni identifikator nekog elementa. Atributi **class** i **id** se često koriste prilikom specificiranja sučelja i jako je bitno da ih ispravno koristite. Na primjer:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Title</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h3>User</h3>
    

    <h3>Friends </h3>
    
    
    
  </body>
</html>
```

U HTML dokumentu postoji točno jedna slika glavnog korisnika i zbog toga joj pridružujemo jedinstven identifikator (**id="userPicture"**), međutim postoji više slika prijatelja i zbog tog svakoj od tih slika pridružujemo atribut **class="friendPicture"**. Na taj način smo označili da unutar stranice postoji točno jedan element s id-em „userPicture“, a više elemenata klase „friendPicture“.

HTML element može imati više klasa, npr.

```

```

image element je i klase „friendPicture“ i klase „colleaguePicture“.

#### 1.4.1. Selektiranje po ID-u

Selektiranje po ID-u se koristi za odabiranje jednog elementa. Ukoliko želimo odabrati element po ID-u pišemo selektor u sljedećem formatu:

```

```

Id elementa  
#userPicture { color: red; }  
# oznaka da se radi o ID selektoru

Jednostavno prije identifikatora elementa stavimo znak ljestve („#“) s čime označimo da se radi o identifikatoru.

#### 1.4.2. Selektiranje po klasi

Selektiranje po klasi se koristi kad želimo odabrati više elemenata koji su članovi neke grupe (što je naznačeno tako da svi imaju isti class atribut). Odabiranje elementa iste klase obavljamo prema sljedećem formatu:

```
  
  

```

klasa  
.friendPicture { color: red; }  
. oznaka da se radi o class

Jednostavno prije klase elementa stavimo točku („.“) s čime označimo da se radi o klasi.

### 1.5. SELEKTORI ATRIBUTA

Uz svaki selektor možemo (bilo po id, klasi, ili tipu elementa) možemo dodati selektor atributa, npr:

```
<input />  
<input type="text" />  
<input type="password" />
```

input[type] { background-color: red; }  
[imeAtributa] – mora postojati atribut s tim imenom

Selektor „input[type]“ označava da želimo selektirati sve elemente tipa input koji imaju definiran atribut type, pri čemu nije bitna točna vrijednost atributa, nego samo činjenica da je atribut definiran.

Osim postojanja atributa možemo provjeravati:

- točnu vrijednost atributa, tako da koristimo [atribut=vrijednost]. Npr u HTML kodu:

```
<div class="container" type="imageContainer"></div>  
<div class="container" type="tableContainer"></div>  
<div class="container" type="textContainer"></div>
```

za odabiranje prvog elementa možemo koristiti css selektor: .container[type=imageContainer]  
{ }

- da li atribut sadrži neku riječ, tako da koristimo [atribut~=vrijednost]

```
<input type="text" value="word">
<input type="text" value="word word2">
```

Za odabiranje drugog elementa možemo koristiti: `input[value~= "word"] {}`

- da li atribut sadrži bilo koji pod-niz znakova, tako da koristimo [atribut\*=vrijednost]

```
<p title="my paragraph"></p>
<p title="my text"></p>
```

Za odabiranje drugog elementa možemo koristiti: `p[title*="para"] {}`

- da li atribut počinje s nizom znakova, tako da koristimo [atribut^=vrijednost]
- da li atribut završava s nizom znakova, tako da koristimo [atribut\$=vrijednost]

## 1.6. PSEUDO SELEKTORI

Selektorima možemo dodati pseudo-klase i pseudo-elemente, da bi ih učinili specifičnijima. Od glavnog selektora se odvajaju korištenjem znaka „:“.

### 1.6.1. Pseudo-elementi

Pseudo-elementi omogućavaju definiranje stilova za dijelove elementa:

- First-letter – primjenjuje stil samo na prvo slovo nekog elementa. (selector: `first-letter`{})

```
<p>Moj tekst</p> p:first-letter { font-size: 120%; } Moj tekst
```

- First-line – primjenjuje stil na prvu liniju elementa (selector: `first-line` {})

```
<p>Moj tekst u više linija</p> p:first-line {color: red; } Moj tekst u više linija
```

- Before – primjenjuje stil na lokaciju prije elementa (selector: `before`{}). Najčešće služi za ubacivanje sadržaja prije nekog elementa.

```
<p> za</p> p:before { content: "Programiranje "; } Programiranje za
```

- After – primjenjuje stil na lokaciju nakon elementa (selector: `after` {}).

```
<p>Za </p> p:after { content: "Internet "; } Za Internet
```

### 1.6.2. Pseudo-klase

Pseudo-klase omogućavaju odabir elemenata na temelju odnosa između elemenata, te na temelju određenog stanja aplikacije. Razlikujemo sljedeće pseudo-klase:

- Link (selektor:**link**) – koristi se za poveznice (elementi tipa **a**). Odabire poveznice koje nisu do sad posjećene.
- Visited (selektor: **visited**) – koristi se za poveznice (elementi tipa **a**). Odabire poveznice koje je korisnik do sad već posjetio.
- Active (selektor: **active**) – odnosi se na elemente koji su aktivirani (npr. Ako je miš kliknut nad elementom, najčešće se koristi za definiranje izgleda elementa u trenutku kad je korisnik kliknuo na element).
- Hover (selektor:**hover**) – odnosi se na elemente iznad kojih se nalazi „uređaj za pokazivanje“ (npr. ukoliko mišem dođemo iznad elementa).
- Focus (selektor:**focus**) – odnosi se na elemente koji su fokusirani (npr. ukoliko korisnik klikne na polje za unos teksta, taj element je fokusiran sve dok se u njega može unositi tekst).
- First-child (selektor:**first-child**) – odabire prvo dijete elementa opisanih selektorom.

```
<p>                                p:first-child { color: red; }    First child Text
  <span>First child</span>
  <span>Text</span>
</p>
```

- Last-child (selektor:**last-child**) – odabire zadnje dijete elemenata opisanih selektorom.

```
<p>                                p:last-child { color: red; }    First child Text
  <span>First child</span>
  <span>Text</span>
</p>
```

- Only-child (selektor:**only-child**) – odabire element dijete elemenata opisanih selektorom, ali samo ako je jedino.

```
<p>                                p:only-child { color: red; }    Only child
  <span>Only child </span>
</p>                                First child Text

<p>
  <span>First child </span>
  <span>Text</span>
</p>
```

- Only-of-type (selektor:**only-of-type**) – odabire element samo ako nema je jedini element tog tipa unutar roditelja.

```
<div>                                p:only-of-type { color: red; }    Text
  <h1>Text</h1>
  <p>Paragraph</p>                                Paragraph
</div>
<div>
  <h1>Text</h1>
```



```
<p> Paragraph </p>
<p> Paragraph </p>
</div>
```

Text

Paragraph

Paragraph

- First-of-type (selektor:**first-of-type**) – odabire prvi element tog tipa unutar roditelja.

```
<div>
  <h1>Text</h1>
  <p>Paragraph</p>
</div>
<div>
  <h1>Text</h1>
  <p> Paragraph </p>
  <p> Paragraph </p>
</div>
```

```
p:first-of-type { color: red; }
```

Text

Paragraph

Text

Paragraph

Paragraph

- Last-of-type (selektor:**last-of-type**) – odabire zadnji element tog tipa unutar roditelja.

```
<div>
  <h1>Text</h1>
  <p>Paragraph</p>
</div>
<div>
  <h1>Text</h1>
  <p> Paragraph </p>
  <p> Paragraph </p>
</div>
```

```
p:last-of-type { color: red; }
```

Text

Paragraph

Text

Paragraph

Paragraph

- N-th child (selektor:**nth-child(an + b)**) – odabire svako **b**-to dijete elemenata opisanih selektorom nakon što se dijeca podijele u grupe s **a** elemenata.

```
<p>
  <span>Prvi, neparan </span>
  <span>Paran </span>
  <span>Neparan </span>
</p>
```

```
p:nth-child(1){ font-weight: bold;}
p:nth-child(2n){ color: green; }
p:nth-child(2n+1){ color: red; }
```

Prvi, neparan

Paran

Neparan

1 – odaberi prvi element; 2n (ili jer je česta operacija može se koristiti konstanta **even**) – odaberi parne elemente; 2n + 1 (ili konstanta **odd**) – odaberi neparne elemente.

## 1.7. SELEKTORI ODNOSA (RELATIONSHIP SELECTORS)

Selektori odnosa odabiru elemente prema njihovom odnosu na druge elemente:

- Selektori prvog susjedstva (selektor+selektor) – drugi element se odabire samo ako se nalazi odmah nakon prvog elementa

<code>&lt;div&gt;Element&lt;/div&gt;</code>	<code>div+p {color: red}</code>	Element
<code>&lt;p&gt;Prvi p&lt;/p&gt;</code>		Prvi p
<code>&lt;p&gt;Drugi p&lt;/p&gt;</code>		Drugi p

- Selektori potomka (selektor selektor) (stavlja se **razmak** između dva selektora) – odabire element opisan s drugim selektorom ako se je potomak (ako se nalazi negdje unutar elementa) opisanog s prvim selektorom.

<code>&lt;div&gt;</code>	<code>div p {color: red}</code>	Prvi p
<code>&lt;p&gt; Prvi p&lt;/p&gt;</code>		Drugi p
<code>&lt;div&gt;</code>		Treći p
<code>&lt;p&gt;Drugi p&lt;/p&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;p&gt; Treći p&lt;/p&gt;</code>		

- Selektor djece (selektor > selektor) – odabire element opisan s drugim selektorom samo ako je direktno dijete elementa opisanog s prvim selektorom.

<code>&lt;div&gt;</code>	<code>div&gt;p {color: red}</code>	Prvi p
<code>&lt;p&gt; Prvi p&lt;/p&gt;</code>		Drugi p
<code>&lt;div&gt;</code>		Treći p
<code>&lt;p&gt;Drugi p&lt;/p&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;p&gt; Treći p&lt;/p&gt;</code>		

- Selektor zajedničkog roditelja (selektor~selektor) – odabire elemente opisane s drugim selektorom ako imaju istog roditelja kao i elementi opisani s prvim selektorom (ako su „braća“).

<code>&lt;div&gt;</code>	<code>.con~p {color: red}</code>	Prvi p
<code>&lt;div class='con'&gt;</code>		Drugi p
<code>&lt;p&gt; Prvi p&lt;/p&gt;</code>		Treći p
<code>&lt;div&gt;</code>		
<code>&lt;p&gt;Drugi p&lt;/p&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;/div&gt;</code>		
<code>&lt;p&gt; Treći p&lt;/p&gt;</code>		
<code>&lt;/div&gt;</code>		

## 2. ZADATAK

Dodati stilove na Vježbu 1 i realizirati sljedeće sučelje:

# Programiranje za Internet - uvod u CSS

Ja sam **Ime i Prezime** i ovo je moja vježba iz *Programiranja za Internet*. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut **labore** et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Anim id est *laborum*. Više o meni možete saznati u mom [životopisu](#).



## Životopis

- Rođen sam u [Gradu](#), XX.XX.XXXX
- Završio sam [Opću Gimnaziju](#)
- Trenutno studiram računarstvo na FESB-u

## Položeni predmeti

Predmet	Profesor	Semestar	Opterećenje
Matematika 1	Ivan Slapničar	1	3/2
Fizika 1	Ivica Puljak	1	3/2
Osnove Elektrotehnike	Slavko Vujević	1	3/2
@FESB		2014	

Username:

Password:

Login