

AV3-primjeri

Potreba za novim standardom

- U prošlom poglavlju smo naučili da je HTML razvijen kao jezik za strukturirani prikaz podataka. U HTML-u postoje oznake pomoću kojih se može uljepšati izgled web stranica, ali s rastom web stranica uvidjelo se nedostatak oznaka koje bi mogle urediti izgled stranice.
- Ukazala se potreba za uvođenjem dodatnih stilova pomoću kojih se može oblikovati izgled web stranice.
- Krajem 1994. Godine Håkon Wium Lee i Bert Bos objavili su prvu specifikaciju CSS-a.
- CSS (eng. *Cascading Style Sheets*), u prijevodu kaskadni stilovi, jednostavan je mehanizam za oblikovanje web stranica pomoću kojeg možemo dodavati fontove, boje, margine itd.

Sintaksa CSS-a

- Prije nego krenemo sa sintaksom CSS-a usporedit ćemo web stranicu oblikovanu pomoću čistog HTML koda i web stranicu oblikovanu pomoću CSS-a.
- Za primjer treba napraviti stranicu čiji izgled u web pregledniku izgleda ovako:

<p>Najbolje stvari na svijetu su besplatne, i vrijede svakog novčića kojeg ste za njih dali!</p>
--

HTML kôd

```
<p>  
<font face="Courier New">  
<font color="#0000FF" size="+2">  
<b>Najbolje</b></font>  
  stvari na svijetu su besplatne,</font>  
</p>  
<p>  
<font face="Courier New">  
<font color="#0000FF" size="+2">  
<b>i vrijede </b></font>  
  svakog novčića kojeg ste za njih dali! </font>  
</p>
```

Isti ispis pomoću CSS-a

- HTML kôd:

```
<p>  
<span class="mojstil">Najbolje</span>  
  stvari na svijetu su besplatne,  
</p>  
<p>  
<span class="mojstil">i vrijede</span>  
  svakog novčića kojeg ste za njih dali!  
</p>
```

- CSS dio koda:

```
p  
{  
  font-family:"Courier New";  
}  
.mojstil  
{  
  color:#0000FF;  
  font-size:24px;  
  font-weight:bold;  
}
```

Na osnovi gornjeg primjera objasnimo sintaksu CSS-a

- CSS kôd se sastoji od tri glavna elementa:
 1. **selektori** – u gornjem primjeru to su **p** i **.mojstil** – oni identificiraju pojedine elemente na HTML stranici i utječu na njihov izgled.
 2. **svojstva** – u gornjem primjeru to su **font-size**, **font-weight** i **color** – pomoću njih opisujemo pojedina svojstva kao što su boja teksta, veličina teksta itd.
 3. **vrijednosti** – u gornjem primjeru to su **#0000FF**, **24px** i **bold** – i predstavljaju vrijednost koju može poprimiti pojedino svojstvo.
- Osnovna sintaksa CSS-a izgleda ovako:

```
selektor { svojstvo: vrijednost; }
```
- Nikako ne smijemo zaboraviti **komentare** - tekst napisan između znakova **/*** i ***/**.

Povezivanje CSS-a i HTML-a

- Postoje četiri moguća načina na koji možemo povezati CSS sa HTML-om:
 1. Umetanje u zaglavlje dokumenta (unutar oznake HEAD).
 2. Dodavanje unutar linije HTML koda pomoću atributa STYLE.
 3. Povezivanje sa vanjskim dokumentom.
 4. Umetanje vanjskog dokumenta.

- Zamislimo da imamo HTML kôd spremljen u datoteku pod imenom **mojstranica.html**, a CSS u datoteku pod imenom **mojstil.css**. Vidimo da datoteka sa CSS kodom mora biti spremljena sa ekstenzijom **css**.

Mudra izreka

Najbolje stvari na svijetu su besplatne,

i vrijede svakog novčića kojeg ste za njih dali!

mojstil.css

```
/* Jednostavan primjer CSS-a */
p {
  text-decoration: none;
  color: red;
  font-family: Arial, Courier;
  font-size: 14pt;
}
.naslov {
  text-decoration: bold;
  color: green;
  font-family: Verdana, Courier;
  font-size: 20pt;
  font-weight: heavy;
}
.podcrtani {
  text-decoration: underline;
  color: blue;
  font-family: Arial, Courier;
  font-size: 14pt;
}
```

Umetanje u zaglavlje dokumenta

- Kada koristimo ovaj način povezivanja CSS se ubacuje unutar zaglavlja HTML dokumenta pomoću oznake `<style>` koja ima svoj atribut `type="text/css"`, i ne postoji vanjska datoteka **mojstil.css**. Taj atribut nam govori da se radi o tekstu i da pripada tipu `css`. U ovome slučaju **mojstranica.html** izgleda ovako

```
<html>
<head>
<style type="text/css">
/* Jednostavan primjer CSS-a */
p {
  text-decoration: none;
  color: red;
  font-family: Arial, Courier;
  font-size: 14pt;
}
.naslov {
  text-decoration: bold;
  color: green;
  font-family: Verdana, Courier;
  font-size: 20pt;
  font-weight: heavy;
}
.podcrtani {
  text-decoration: underline;
  color: blue;
  font-family: Arial, Courier;
  font-size: 14pt;
}
```

```
</style>
</head>
<body>
<h1 class="naslov"> Mudra izreka
</h1>
<p>Najbolje stvari na svijetu su
besplatne,</p>
<p class="podcrtani">i vrijede svakog
novčića kojeg ste za njih dali! </p>
<body>
</html>
```

Dodavanje unutar linije HTML koda

- Ovaj način se koristi kada želite napraviti oblikovanje unutar samo retka HTML koda, tj. Kada želite napraviti promjenu samo unutar određene oznake. Dodavanje stila za unutar linije HTML koda se obavlja pomoću atributa `style`, te se unutar dvostrukih navodnika dodaju CSS svojstva i vrijednosti. Atribut `style` se može dodati skoro svim HTML oznakama i taj stil vrijedi samo unutar HTML oznake u kojoj je definiran. Tako bi odlomak iz gornjeg primjera izgledao:

```
<p style="text-decoration:none; color:red; font-family:Arial, Courier; font-size:14pt;">
```

Tekst odlomka koji je oblikovan ovim stilom.

```
</p>
```

Povezivanje sa vanjskim dokumentom

- Povezivanje sa vanjskim dokumentom podrazumijeva postojanje vanjske CSS datoteke i u ovom primjeru mi koristimo CSS datoteku **mojstil.css**. Povezivanje HTML-a sa vanjskom CSS datotekom izvodi se pomoću oznake `<link>` koju dodajemo u zaglavlje dokumenta tj. unutar oznake `<head>`. Oznaka `<link>` je dio HTML standarda i jedna je od rijetkih oznaka koja nema svoj završni dio `</link>`. Oznaka `<link>` ima svoje attribute koje moramo navesti pri povezivanju. Njeni atributi su:

- rel** atribut koji definira odnos između oznake i dokumenta s kojim se povezujemo. Može poprimiti nekoliko vrijednosti ovisno o vrsti dokumenta s kojim se HTML povezuje, a kad služi za povezivanje CSS-a, poprima vrijednost "stylesheet"
- type** definira tip dokumenta s kojim se povezujemo. U slučaju CSS-a tip dokumenta je "text/css"
- href** URI dokumenta na koji se povezujemo

- Dakle, datoteka **mojstranica.html** ima sljedeći sadržaj i povezuje se na vanjsku CSS datoteku sa imenom **mojstil.css**:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="mojstil.css">
</head>
<body>
<h1 class="naslov"> Mudra izreka </h1>
<p>Najbolje stvari na svijetu su besplatne,
</p>
<p class="podcrtani">i vrijede svakog novčića kojeg ste za
njih dali!
</p>
</body>
</html>
```

Umetanje vanjskog dokumenta

- Umetanje vanjskog dokumenta također podrazumijeva postojanje vanjske CSS datoteke i u ovom primjeru također koristimo CSS datoteku **mojstil.css**. Umetanje vanjskog dokumenta u HTML izvodi se pomoću ključne riječi `@import` koja se ubacuje u zaglavlje i to unutar oznake `<style>`. Oznaka `<style>` obavezno mora imati atribut `type="text/css"` koji označava tip dokumenta s kojim se povezujemo. U ovom primjeru uvozimo vanjsku CSS datoteku sa imenom **mojstil.css** i datoteka **mojstranica.html** ima sljedeći sadržaj:

```
<html>
<head>
<style type="text/css">
@import url("mojstil.css");
@import "mojstil.css";
@import "http://www.etfos.hr/css/mojstil.css";
</style>
</head>
<body>
<h1 class="naslov"> Mudra izreka </h1>
<p>Najbolje stvari na svijetu su besplatne,
</p>
<p class="podcrtani">i vrijede svakog novčića kojeg ste za njih dali!
</p>
</body>
</html>
```

- Iz gornjeg primjera vidimo da dokument možemo uvesti na nekoliko načina. Prva dva načina uvoze dokument koji se nalazi u istom direktoriju kao i datoteka **mojstranica.html**, jedina razlika je u načinu korištenja ključne riječi @import. Treći način se razlikuje po tome što se datoteka **mojstil.css** ne mora nalaziti u našem lokalnom direktoriju nego se može laziti na bilo kojem serveru u svijetu.

Kombiniranje načina povezivanja

```
<html>
<head>
<link rel="stylesheet" type="text/css" ref="mojstil.css">
</head>
<body>
<h1 class="naslov"> Mudra izreka </h1>
<p>Najbolje stvari na svijetu su besplatne,
</p>
<p style="color: red; font-family: Arial;">i vrijede svakog
novčića kojeg ste za njih dali!
</p>
</body>
</html>
```

Tipovi selektora

selektor { **svojstvo**:vrijednost }

- Svaki selektor odnosi se na jedan ili više elemenata HTML stranice na koju se primjenjuje dotični stil. Postoji čitav niz selektora pa potpuno upoznavanje primjene svakog od njih nije nimalo banalan zadatak. Ovdje ćemo spomenuti i detaljnije obraditi nekoliko osnovnih tipova selektora:
 1. Jednostavni selektori
 2. Klasni selektori
 3. Id selektori
 4. Kontekstni selektori
 5. Pseudoklase.

Jednostavni selektori

- Jednostavni selektori su najjednostavniji tip selektora i imaju imena jednaka html oznakama. Prednost ovih selektora je što ne zahtijevaju izmjene HTML koda. Ako želimo promijeniti boju teksta u odlomku potrebno je samo napraviti selektor sa imenom **p** u CSS-u, ili ako želimo odrediti izgled unutar oznake **<body>** potrebno je napraviti selektor **body**.
- Zbog toga što je selektor **body** u samom vrhu hijerarhije vidljivog dijela HTML dokumenta - svi elementi koji se hijerarhijski nalaze niže od njega, poprimit će svojstva navedena u selektoru **body**. Nedostatak jednostavnih selektora je što se oni odnose na sve elemente unutar HTML stranice koji imaju isto ime, tako će se selektor **p** odnositi na sve odlomke unutar stranice, stoga ako želimo da različiti odlomci izgledaju različito morat ćemo koristiti druge selektore.

```
p {  
  text-decoration: none;  
  color: red;  
  font-family: Arial, Courier;  
  font-size: 14pt;  
}  
h1 {  
  text-decoration: bold;  
  color: green;  
  font-family: Verdana, Courier;  
  font-size: 20pt;  
  font-weight: heavy;  
}
```

- U gornjem primjeru imamo dva jednostavna selektora **h1** i **p**. Prvi selektor kaže da će svi glavni naslovi na stranici biti zelene boje, pisani fontom Verdana i veličine 20 točkica, dok drugi selektor kaže da će svi odlomci na stranici biti crvene boje, veličine 14 točkica i pisani fontom Arial.

2.3.2. Klasni selektori

- Vidjeli smo da je nedostatak jednostavnih selektora u tome što se oni odnose na sve oznake istog imena unutar stranice. Često je slučaj da mi ne želimo da nam sve oznake budu jednake i tada nam jednostavni selektori ne mogu pomoći. Tu na scenu stupaju klasni selektori. Klasni selektor se definira tako da se ispred imena selektora stavi točka kao što je pokazano u sljedećem primjeru:

```
<html>
<head>
<style type="text/css">
body {color:red}
h1 {color:#00ff00}
.plava {color:rgb(0,0,255)}
</style>
</head>
<body>
<h1>Ovo je naslov 1</h1>
<p>Ovo je jedan vrlo običan odlomak. Primijetite da je tekst napisan crvenom bojom,
<br>jer odlomak nasljeđuje oznaku body u kojoj je definirana crvena boja teksta.
</p>
<p class="plava">Boja ovog odlomka je definirana u klasnom selektoru .plava i stoga je
tekst plave boje, <br> jer klasni selektor ima veći prioritet od jednostavnog selektora
što ćemo učiti u kasnijim poglavljima.
</p>
</body>
</html>
```

- U primjeru imamo dva jednostavna selektora **h1** i **body**, te klasni selektor **.plava** ispred čijeg imena se nalazi točka. Klasni selektor se poziva tako da unutar oznake u kojoj ga želimo upotrijebiti navedemo atribut **class** i kao vrijednost predamo ime selektora. Primijetimo da se kod poziva ne stavlja točka ispred imena selektora tako da imamo poziv **class="plava"**

- Ovo je prikaz koda u pregledniku

Ovo je naslov 1

Ovo je jedan vrlo običan odlomak. Primjetite da je tekst napisan crvenom bojom, jer odlomak nasljeđuje oznaku body u kojoj je definirana crvena boja teksta.

Boja ovog odlomka je definirana u klasnom selektoru .plava i stoga je tekst plave boje, jer klasni selektor ima veći prioritet od jednostavnog selektora što ćemo učiti u kasnijim poglavljima.

- Postoji još jedan način definiranja klasnih selektora, a to je **oznaka.naziv**. Razlika je u tome što se ovako definirani selektori mogu primijeniti samo na oznake istog tipa

```
<html>
<head>
<style type="text/css">
body { color:red }
h1 { color:#00ff00 }
p.plava { color:rgb(0,0,255) }
</style>
</head>
<body>
<h1>Ovo je naslov 1</h1>
<p>Ovo je jedan vrlo običan odlomak. Primijetite da je tekst napisan crvenom bojom,
<br>jer odlomak nasljeđuje oznaku body u kojoj je definirana crvena boja teksta.
</p>
<p class="plava">Boja ovog odlomka je definirana u klasnom selektoru .plava i stoga je
tekst plave boje, <br> jer klasni selektor ima veći prioritet od jednostavnog selektora
što ćemo učiti u kasnijim poglavljima.
</p>
</body>
</html>
```

Primjer je isti kao i prošli ali sada je ime selektora **p.plava**. Ovako definirani selektor primjenjuje se jednako kao i prethodni, ali za razliku od prethodnog selektora koji se mogao primijeniti na različitim oznakama, ovaj selektor se može primijeniti samo na odlomke tj. na oznaku **<p>**.

Id selektori

- Id selektor se definira tako da se ispred imena selektora oznaka # kao što je pokazano u sljedećem primjeru:

```
<html>
<head>
<style type="text/css">
#zelena {color:red}
.plava {color:rgb(0,0,255)}
</style>
</head>
<body>
<p id="zelena">Ovo je jedan odlomak čiji je stil definiran id selektorom
#zelena. Tekst je napisan crvenom bojom, <br>kao što je definirano id
selektorom #zelena.
</p>
<p class="plava">Boja ovog odlomka je definirana u klasnom selektoru
.plava i stoga je tekst plave boje. <br> Sada se vjerojatno pitate koja je
razlika klasnih i id selektora?
</p>
</body>
</html>
```

- Ovo je prikaz koda u pregledniku

Ovo je jedan odlomak čiji je stil definiran id selektorom #zelena. Tekst je napisan crvenom bojom, kao što je definirano id selektorom #zelena.

Boja ovog odlomka je definirana u klasnom selektoru .plava i stoga je tekst plave boje. Sada se vjerojato pitate koja je razlika klasnih i id selektora?

Pseudoklase

- Sintaksa za pseudoklase je ponešto drugačija nego kod ostalih selektora:

selektor: pseudoklasa { svojstvo: vrijednost; }

- Postoji više pseudoklasa, a mi ćemo nabrojati najčešće korištene:

link - izgled neposjećenog linka

visited - izgled posjećenog linka

hover - izgled linka kada se pokazivač miša nalazi iznad linka

active –izgled linka u trenutku kad kliknemo na njega tj. izgled aktivnog linka

focus – izgled elementa koji trenutno ima fokus na sebi

first-child – izgled elementa koji je prvi potomak nekom drugom elementu.

- Da bismo dobili željen rezultat, zbog određenih svojstava CSS-a, pseudoklase se pri definiranju uvijek moraju navesti sljedećim redoslijedom:

```
a:link{}  
a:visited{}  
a:hover{}  
a:active{}
```

- Ako ih navedemo pogrešnim redoslijedom CSS stilovi nam neće raditi. Sada ćemo navesti nekoliko primjera koji će nam pokazati primjenu pseudoklasa.

```
<html>  
<head>  
<style type="text/css">  
a:link { color: red; text-decoration: none; }  
a:visited { color: green; text-decoration: none; }  
a:hover { color: blue; }  
a:active { color: yellow; }  
</style>  
</head>  
<body>  
<p><b><a href="http://www.etfos.hr" target="_blank">WWW.ETFOS.HR  
</a></b></p>  
</body>  
</html>
```

- Za neposjećeni link dobijemo:

WWW.ETFOS.HR

- Za posjećeni link dobijemo:

WWW.ETFOS.HR

- Kad se pokazivač miša nalazi iznad linka:

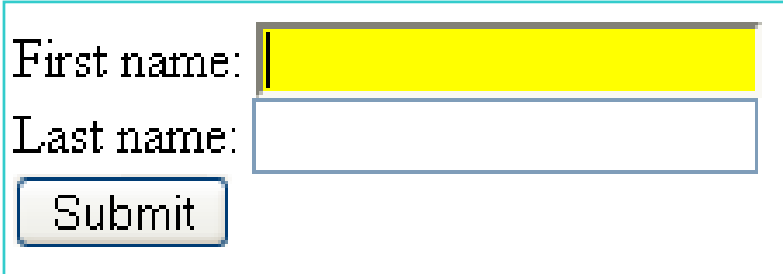
WWW.ETFOS.HR

- Za aktivni link dobijemo:

WWW.ETFOS.HR

Primjer pseudoklase **focus**

```
<html>
<head>
<style type="text/css">
input:focus { background-color:yellow; }
</style>
</head>
<body>
<form action="nekaskripta.php" method="get">
First name: <input type="text" name="ime"><br>
Last name: <input type="text" name="prezime"><br>
<input type="submit" value="Submit">
</form></body>
</html>
```



First name:

Last name:

The screenshot shows a web form with a light blue border. It contains two text input fields and a submit button. The 'First name:' label is followed by a text input field that is highlighted in yellow, indicating it is the current focus. The 'Last name:' label is followed by an empty text input field. Below these fields is a 'Submit' button.

Primjer pseudoklase **first-child**

```
<html>
<head>
<style type="text/css">
p:first-child { color:blue; }
</style>
</head>
<body>
<p>Ovo je tekst odlomka koji je prvo dijete unutar oznake body.</p>
<p>Ovo je tekst odlomka koji nije prvo dijete unutar oznake body i <br>
pseudoklasa first-child ne utječe na njega.</p>
</body>
</html>
```

Ovo je tekst odlomka koji je prvo dijete unutar oznake body.

Ovo je tekst odlomka koji nije prvo dijete unutar oznake body i
pseudoklasa first-child ne utječe na njega.

2.3.5. Grupiranje selektora

```
<html>
<head>
<style type="text/css">
h1 {color:black; background-color:yellow; font-family:Arial; font-size:16pt;}
p {color:black; background-color:blue; font-family:Arial; font-size:14pt;}
.odlomak {color:black; background-color:green; font-family:Arial; font-
size:12pt;}
</style>
</head>
<body>
<h1> Moj naslov </h1>
<p>Ovo je jedan odlomak čiji je stil definiran jednostavnim selektorom.
</p>
<p class="odlomak">Ovo je odlomak koji je definiran klasnim selektorom
.odlomak. <br> Koristimo ga kao primjer grupiranja zajedničkih svojstava.
</p>
</body>
</html>
```


- Iz gornjeg primjera vidimo da su za sve selektore zajednički boja i tip fonta, stoga ta svojstva možemo grupirati na jednome mjestu. U gornjem primjeru ako želimo promijeniti boju i tip fonta to moramo učiniti na 3 mjesta, no kad ih grupiramo to činimo na samo jednome mjestu kao što pokazuje idući primjer:

```
<html>
<head>
<style type="text/css">
.odlomak , h1, p { color:black; font-family:Arial; }
h1 { background-color:yellow; font-size:16pt; }
p { background-color:blue; font-size:14pt; }
.odlomak { background-color:green; font-size:12pt; }
</style>
</head>
<body>
<h1> Moj naslov </h1>
<p>Ovo je jedan odlomak čiji je stil definiran jednostavnim selektorom.</p>
<p class="odlomak">Ovo je odlomak koji je definiran klasnim selektorm
.odlomak. <br> Koristimo ga kao primjer grupiranja zajedničkih svojstava.
</p>
</body>
</html>
```

Nasljeđivanje i kaskadna svojstva selektora

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

```
td {color:black; font-size: 14pt;}
```

- Tada će sav tekst unutar stupca tablice biti crvene boje, veličine 14pt i sa žutom pozadinom jer oznaka `<td>` nasljeđuje oznaku `<body>` i time nasljeđuju žutu boju pozadine. No ako na nekom od elemenata unutar oznake `<td>` primijenimo neki drugi stil, taj stil će imati prioritet pri prikazivanju.

- Razmotrimo još jedan primjer korištenja nasljeđivanja:

```
body { background-color:yellow; font-size: 10pt; }  
h1 { color:black; font-size: 120%; }
```

- Oznaka `<h1>` također nasljeđuje oznaku `<body>`, u ovome slučaju osim boje pozadine nasljeđuje i veličinu fonta pa će veličina fonta biti 120% veličine unutar oznake `<body>` tj. bit će 12pt.

- Na sljedećem primjeru ćemo prikazati prioritet selektora:

```
<html>
<head>
<style type="text/css">
#naslov { color:red; }
h1 { font-size: 16pt; color:blue; }
p { font-size: 14pt; color:green; }
.plavi { font-size: 12pt; color:blue; }
</style>
</head>
<body>
<h1 id="naslov"> Moj naslov </h1>
<p class="plavi">Ovo je jedan odlomak čiji je stil definiran klasnim
selektorom .plavi. <br> I on ima veći prioritet od jednostavnog
selektora.
</p>
<p class="plavi" style="color:black;">Ovo je odlomak koji je definiran
klasnim selektorom .plavi i trebao bi imati tekst plave boje. <br> Ali je
atributom style koji je većeg prioriteta od klasnog selektora definirano
da je tekst crne boje.
</p>
</body>
</html>
```

Rezultat u pregledniku:

Moj naslov

Ovo je jedan odlomak čiji je stil definiran klasnim selektorom `.plavi`.

I on ima veći prioritet od jednostavnog selektora.

Ovo je odlomak koji je definiran klasnim selektorm `.plavi` i trebao bi imati tekst plave boje.

Ali je atributom `style` koji je većeg prioriteta od klasnog selektora definirano da je tekst crne boje.

- Vidimo da je naslov crvene boje jer je id selektor `#naslov` većeg prioriteta od jednostavnog selektora `h1`. Prvi odlomak je plave boje jer je klasni selektor `.plavi` većeg prioriteta od jednostavnog selektora `p`. Kako je u drugom odlomku tekst definiran atributom `style`, koji je većeg prioriteta od klasnog selektora `.plavo`, tekst je crne boje.

CSS font

- Pošto se tekst nalazi na većini internet stranica, a njega treba oblikovati svojstva fonta spadaju u najčešće korištena CSS svojstva. Prikazat ćemo najčešće korištena svojstva fonta i vrijednosti koje mogu poprimiti:
 - font-variant
 - font-weight
 - font-size
 - font-style
 - font-family

- Tablica Generičke porodice fontova:

Generička porodica	Ime fonta	Opis
Serif	Times New Roman Georgia	Fontovi porodice serif imaju crtive na kraju znakova.
Sans-serif	Arial Verdana	Sans oznazava fontove porodce serif bez crtica na kraju.
Monospace	Courier New Luci da Consol e	Svi monospace fontovi imaju istu širinu.

- Svojstva fonta možemo koristiti da navedemo generičku porodicu:

```
body {
    font-family: Serif;
}
```

- Ili možemo koristiti stvarna imena fontova:

```
body {
    font-family: Georgia, "Times New Roman";
}
```

```
<html>
<head>
<style type="text/css">
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}
</style>
</head>
<body>
<p class="normal">Normalan tekst.</p>
<p class="italic">Italic tekst.</p>
<p class="oblique">Oblique tekst.</p>
</body>
</html>
```


- Slijedi primjer koji koristi nekoliko načina određivanja veličine teksta:

```
<html>
<head>
<style type="text/css">
body { font-size:100%;}
h1 { font-size:2.5em;}
h2 { font-size:1.875em;}
p {font-size:14px;}
.odlomak {font-size:14pt;}
</style>
</head>
<body>
<h1> Naslov veličine 2.5em.</h1>
<h2> Naslov veličine 1.875em.</h2>
<p> Tekst veličine 14px.</p>
<p class="odlomak"> Tekst veličine 14pt.</p>
</body>
</html>
```

- U web pregledniku dobijemo slijedeći izgled:

Naslov veličine 2.5em.

Naslov veličine 1.875em.

Tekst veličine 14px.

Tekst veličine 14pt.

font-weight

- Ovo svojstvo definira debljinu znakova a može imati sljedeće vrijednosti normal, bold, bolder, lighter, a može poprimiti i brojčane vrijednosti 100, 200, 300, 400, 500, 600, 700, 800 i 900.

font-variant

- ima dvije vrijednosti normal i small-caps. Ako koristimo small-caps dobijemo efekt teksta pisanog velikim slovima različitih veličina. Najjasnije ćemo to pokazati primjerom:

```
<html>
<head>
<style type="text/css">
body { font-size:100%;}
p.normal { font-variant:normal;}
p.small { font-variant:small-caps;}
</style></head>
<body>
<p class="normal"> Tekst koji je napisan normalno.</p>
<p class="small"> Tekst napisan sa efektom velikih slova.</p>
</body>
</html>
```

- Kao rezultat u web pregledniku dobijemo:

Tekst koji je napisan normalno.

TEKST NAPISAN SA EFEKTOM VELIKIH SLOVA.

Skraćeno pisanje svih selektora

- Sva gore navedena svojstva možemo napisati preko skraćenog selektora **font**, na sljedeći način:

```
p {  
    font: bold italic small-caps 12pt Arial;  
}
```

CSS tekst

- Pomoću svojstava teksta određujemo razmještaj teksta na našoj stranici, razmake među redovima, razmak među slovima itd.
- Prikazat ćemo najčešće korištena svojstva teksta:
 - text-align
 - text-decoration
 - text-transform
 - letter-spacing
 - line-height
 - text-indent

text-align

- Definira vodoravno poravnanje teksta koje se primjenjuje na blok elemente i može imati sljedeće vrijednosti: left, right, center i justify. Najjasnije će nam biti ako pogledamo primjer:

```
<html>
<head>
<style type="text/css">
h1 { text-align:center;}
p.lijevo { text-align:left;}
p.desno { text-align:right;}
p.obostrano { text-align:justify;}
</style></head>
<body>
<h1> Naslov koji je poravnat centralno.</h1>
<p class="lijevo"> Tekst poravnat na lijevu stranu.</p>
<p class="desno"> Tekst poravnat na desnu stranu.</p>
<p class="obostrano"> Tekst koji nije poravnat niti lijevo niti desno, nego
obostrano. No da bismo pokazali da se naš tekst poravnava obostrano
treba nam malo veća količina teksta.</p>
</body>
</html>
```

- Kao rezultat u web pregledniku dobijemo:

Naslov koji je poravnat centralno.

Tekst poravnat na lijevu stranu.

Tekst poravnat na desnu stranu.

Tekst koji nije poravnat niti lijevo niti desno, nego obostrano. No da bismo pokazali da se naš tekst poravnava obostrano treba nam malo veća količina teksta.

text-decoration

- Pomoću ovog svojstva tekstu se dodaju ili uklanjaju različite dekoracije. Ovo svojstvo može imati sljedeće vrijednosti: none (nema dekoracije), underline (linija ispod teksta), line-through (linija precrtava tekst), overline (linija iznad teksta), i blink (tekst koji treperi). Slijedi primjer:

```
<html>
<head>
<style type="text/css">
h1 { text-decoration:blink;}
h2 { text-decoration:underline;}
h3 { text-decoration:line-through;}
h4 { text-decoration:overline;}
.bezdekoracije { text-decoration:none;}
</style></head>
<body>
<h1> Naslov koji treperi.</h1>
<h2> Naslov koji je podcrtan.</h2>
<h3> Naslov koji je precrtan.</h3>
<h4> Naslov koji ima liniju iznad.</h4>
<a href="http://www.etfos.hr">Ovo je obični link</a><br><br>
<a class="bezdekoracije" href="http://www.etfos.hr">Ovo je link kojem smo
uklonili podcrtanu liniju</a>
</body>
</html>
```

- Kao rezultat u web pregledniku dobijemo:

Naslov koji treperi.

Naslov koji je podcrtan.

~~Naslov koji je precrtan.~~

Naslov koji ima liniju iznad.

Ovo je obični link

Ovo je link kojem smo uklonili podcrtanu liniju

text-transform

- svojstvo koje pretvara sva slova u velika ili mala. Može poprimiti slijedeće vrijednosti: capitalize (postavlja početno slovo u svakoj riječi u veliko slovo), uppercase (postavlja sva slova u velika) i lowercase (postavlja sva slova u mala). Slijedi primjer:

```
<html>
<head>
<style type="text/css">
p.uppercase { text-transform:uppercase;}
p.lowercase { text-transform:lowercase;}
p.capitalize { text-transform:capitalize;}
</style></head>
<body>
<p class="uppercase">Ovo je tekst kome su sva slova velika.</p>
<p class="lowercase">Ovo je tekst kome su sva slova mala.</p>
<p class="capitalize">Ovo je tekst kome su sva prva slova u riječi velika.</p>
</body>
</html>
```

- Kao rezultat u web pregledniku dobijemo:

OVO JE TEKST KOME SU SVA SLOVA VELIKA.

ovo je tekst kome su sva slova mala.

Ovo Je Tekst Kome Su Sva Prva Slova U Riječi Velika

letter-spacing

- Svojsvo koje određuje razmak među znakovima u tekstu zovemo letter-spacing.

```
<html>
<head>
<style type="text/css">
p.razmak1 { letter-spacing:3px;}
p.razmak2 { letter-spacing:0.3em;}
p.razmak3 { letter-spacing:0.6em;}
</style>
</head>
<body>
<p class="razmak1">Ovo je tekst sa razmakom 3px.</p>
<p class=" razmak2">Ovo je tekst sa razmakom 0.3em.</p>
<p class=" razmak3">Ovo je tekst sa razmakom 0.6em.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je tekst sa razmakom 3px.

Ovo je tekst sa razmakom 0.3em.

Ovo je tekst sa razmakom 0.6em.

line-height

- Definira vertikalni razmak između dvije linije teksta.

```
<html>
<head>
<style type="text/css">
p.manja { line-height:10pt;}
p.veca { line-height:20pt;}
</style>
</head>
<body>
<p class="manja">Ovo je tekst sa manjim vertikalnim razmakom od
10pt.<br>
Vidimo da je vrlo mal vertikalni razmak među redovima.</p>
<p class="veca">Ovo je tekst sa većim razmakom od 20pt.<br>
Vidimo da je povećan vertikalni razmak među redovima.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je tekst sa manjim vertikalnim razmakom od 10pt.
Vidimo da je vrlo mal vertikalni razmak među redovima.

Ovo je tekst sa većim razmakom od 20pt.

Vidimo da je povećan vertikalni razmak među redovima.

text-indent

- Je svojstvo koje određuje koliko će biti uvučen prvi redak u odlomku.

```
<html>
<head>
<style type="text/css">
p.odlomak1 { text-indent:20px;}
p.odlomak2 { text-indent:40px;}
</style>
</head>
<body>
<p class="odlomak1">Ovo je odlomak kojem je prvi redak uvučen 20 piksela.
Najbolje stvari na svijetu su besplatne, <br> i vrijede svakog novčića kojeg ste za
njih dali!
</p>
<p class="odlomak2"> Ovo je odlomak kojem je prvi redak uvučen 40 piksela.
Najbolje stvari na svijetu su besplatne, <br> i vrijede svakog novčića kojeg ste za
njih dali!
</p>
</body>
</html>
```

- Slijedi rezultat u web pregledniku:

Ovo je odlomak kojem je prvi redak uvučen 20 piksela. Najbolje stvari na svijetu su besplatne, i vrijede svakog novčića kojeg ste za njih dali!

Ovo je odlomak kojem je prvi redak uvučen 40 piksela. Najbolje stvari na svijetu su besplatne, i vrijede svakog novčića kojeg ste za njih dali!