**Stilizovanje teksta**

**00:22:22**

Jedinica: 13 od 30

**+Rezime**

U prethodnim lekcijama ovog modula objašnjeno je kako se postiže stilizovanje HTML elemenata i na koji način se može obaviti selektovanje željenog elementa ili elemenata koje je potrebno stilizovati. Nakon što je element selektovan, unutar CSS deklaracije, potrebno je definisati odgovarajuća svojstva i njihove vrednosti, kako bi se stilizovanje nad elementom primenilo. Različiti tipovi HTML elemenata dozvoljavaju definisanje različitih CSS svojstava. Ova lekcija biće posvećena tehnikama koje se mogu koristiti za stilizovanje teksta.

**Tipografija**

Pre nego što pređemo na različite tehnike za stilizovanje fonta unutar jednog HTML dokumenta, potrebno je pozabaviti se osnovnim načelima tipografije. Tipografija je zapravo tehnika za prikaz teksta na razumljiv i čitljiv način.

Svi fontovi dele se na sledeće osnovne kategorije:

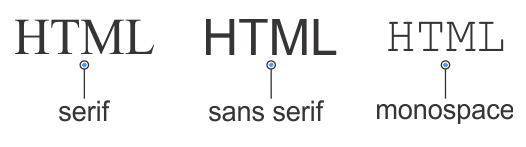
* Serif
* Sans Serif
* Monospace

**Serif** fontovi karakterišu se detaljima na krajevima linija. Ti detalji se nazivaju *serif*, pa odatle i naziv za ovaj tip fontova. Nekada se ova kategorija fontova naziva i*roman*, a najpoznatiji font ove vrste je svakako *Times New Roman*.

**Sans Serif** fontovi su oni fontovi koje odlikuje odsustvo detalja koje poseduju serif fontovi. Ovo je potpuno logično i može se zaključiti iz samog naziva ove vrste fontova, s obzirom da *sans* na francuskom jeziku znači *bez*. Zbog odsustva takozvanih ukrasa, fontovi ove kategorije izgledaju čistije i modernije. Najpopularniji fontovi ove vrste su Arial, Tahoma, Verdana, Helvetica itd.

**Monospace** su oni fontovi kod kojih svi karakteri zauzimaju isti horizontalni prostor. Za početak njihove primene se uglavnom smatra pojava pisaćih mašina, kada su zbog tehnoloških ograničenja bili potrebni takvi fontovi. Kasnije su našli primenu na prvim računarima i terminalima, a i dalje se koriste u programiranju. Najčešći *monospace* font na vebu je Courier New (ili Courier).

Razlike između tri navedene kategorije fontova prikazane su slikom 13.1.



*Slika 13.1 – Sintaksa selektora na osnovu postojanja atributa*

|  |
| --- |
| **Napomena**    *U dosadašnjem toku kursa reč font se upotrebljavala da označi familiju fonta. Ovo je česta praksa koja se može sresti u govoru i literaturi. Tako se može reći da je, na primer, Verdana familija fonta, ali će se zbog jednostavnosti, familija fonta u nastavku kursa označavati terminom font.* |

**Tipografija unutar HTML dokumenta**

Unutar HTML dokumenta, fontovi koji se koriste za stilizovanje teksta mogu se čitati sa klijentskog računara ili neke druge lokacije. Kada se dokument oslanja na fontove na klijentskom računaru, neophodno je da klijent poseduje font koji je iskorišćen za stilizovanje instaliran na svom računaru. Na taj način, slično kao i kod korišćenja fontova u Word ili nekim drugim dokumentima, fontovi se ne ugrađuju u sam dokument, već se čitaju sa računara klijenta. Ovo je posebno važna činjenica kada se razvija web aplikacija, koja je obično hostovana na nekom udaljenom serveru. Iako je kompletan sadržaj na serveru, fontovi koji se koriste u dokumentu podrazumevano će biti korišćeni sa računara klijenta.

Pored opisanog scenarija korišćenja fontova sa klijentskog računara, postoji mogućnost da se font ugradi u samu HTML aplikaciju. Oba opisana scenarija predstavljena su u nastavku.

**Fontovi koje karakteriše prisustvo određenih ukrasa na krajevima karaktera pripadaju grupi koja se naziva:**

 Serif

 Sans Serif

 Monospace

**Izbor fonta**

Izbor fonta za prikaz teksta nekog elementa vrši se postavljanjem font-family svojstva. Sledeći primer ilustruje upotrebu ovog svojstva:

|  |  |
| --- | --- |
| 1  2  3 | body {      font-family: Georgia;  } |

Prikazanim CSS opisom definisano je da će cela strana biti prikazana korišćenjem Georgia fonta. Ukoliko definisani font ne postoji na korisničkom računaru, moguće je za vrednost svojstva postaviti više vrednosti, pri čemu će se koristiti vrednost prvog fonta koji postoji na korisničkom računaru. Primer definisanja svojstva font-family sa više vrednosti je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | h1 {      font-family: Arial, Helvetica, sans-serif;  } |

Prikazanim CSS opisom, parser će pokušati da koristi font *Ariel*, a ukoliko ga ne pronađe, pokušaće da koristi font *Helvetica*. Ako ni *Ariel* ni *Helvetica* nisu dostupni, biće korišćen bilo koji sans-serifni font koji je dostupan na sistemu korisnika, jer je kao poslednja vrednost svojstva font-familypostavljeno *sans-serif.*

Ukoliko naziv fonta sadrži više od jedne reči, prilikom definisanja vrednosti, naziv se postavlja između znaka navodnika, kao u sledećem primeru:

|  |  |
| --- | --- |
| 1  2  3 | body {      font-family: "Times New Roman", Times, Georgia, serif;  } |

Svi dosadašnji primeri ilustrovali su upotrebu fontova koji se nalaze na klijentskom računaru. Ipak, moguće je fontove koji se koriste unutar HTML dokumenta ugraditi u samu aplikaciju, i na taj način koristiti fontove koji ne postoje na klijentskom računaru. Da bi se koristili takvi fontovi, njih je neophodno definisati korišćenjem specijalne rezervisane reči *@font-face*. Sledeći blok koda ilustruje definisanje fonta koji se može koristiti unutar HTML aplikacije, a koji je ugrađen u samu aplikaciju:

|  |  |
| --- | --- |
| 1  2  3  4  5 | @font-face {  font-family: "My Nicer Font";  src: url('fonts/my\_woff\_font.eot');  src: url('fonts/my\_woff\_font.woff');  } |

Specijalna konstrukcija @font-face je zapravo blok koda, unutar koga se postavljaju vrednosti različitih svojstava. Svojstvom font-family definiše se naziv fonta, odnosno familije fonta. Svojstvom src definiše se lokacija na kojoj se nalazi font. Font se može nalaziti na istoj lokaciji kao i čitava aplikacija, ali se može koristiti i iz različitih online repozitorijuma. Takođe, fontovi mogu biti definisani fajlovima različitih ekstenzija. U primeru su prikazani fajlovi sa ekstenzijama eot i woff, a pored njih postoje i fajlovi koji mogu imati ekstenzije ttf ili svg. Neophodno je znati da različiti browseri podržavaju fontove različitih formata.

Font definisan korišćenjem konstrukcije @font-face može se iskoristiti na nekom od HTML elemenata na sledeći način:

|  |  |
| --- | --- |
| 1  2  3 | p {  font-family: 'My Nicer Font';  } |

Na ovaj način, nije neophodno da klijent poseduje font kojim je stilizovan određeni tekst aplikacije, s obzirom da će font biti učitan sa putanje navedene unutar @font-face konstrukcije.

**Veličina teksta**

Izbor veličine teksta može se postići korišćenjem svojstva font-size. Sledeći primer ilustruje definisanje veličine fonta:

|  |  |
| --- | --- |
| 1  2  3 | body {     font-size: 12px;  } |

U prikazanom CSS opisu postavljena je veličina teksta za kompletan tekstualni sadržaj HTML dokumenta. Veličina teksta izražena je korišćenjem vrednosti u pikselima. Pored vrednosti izraženih pikselima, CSS omogućava da se veličina teksta definiše korišćenjem vrednosti iskazanih raznim drugim jedinicama. Tako se sve jedinice za izražavanje veličine mogu podeliti na apsolutne i relativne.

Apsolutne jedinice su:

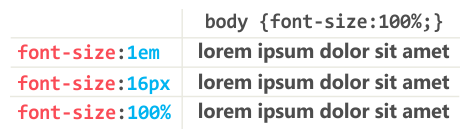
* in – inč (2.54cm)
* pt – point, tipografska tačka (1 point je 1/72 inča)
* pc – pika (1 pika je 12 pointa)
* cm – centimetar
* mm – milimetar
* px – piksel (jednobojna, nedeljiva tačka na ekranu monitora/uređaja)

Relativne jedinice su:

* em – ems
* % – procenat
* ex – x-height

Apsolutne jedinice koriste se da apsolutno odrede veličinu teksta. Drugim rečima, apsolutna veličina slova će uvek biti ista, bez obzira na veličinu prikaza ili bilo koji drugi parametar. Ovakve jedinice preuzete su iz štampanih medija, a prilikom kreiranja HTML aplikacija, dobro ih je izbegavati, osim ako nije potrebno eksplicitno postaviti veličinu nekog teksta koja se ne sme menjati.

Relativne jedinice su one koje svoju finalnu vrednost formiraju na osnovu bazne veličine teksta. Na primer, podrazumevana veličina teksta u većini browsera je 16 px. Tako se može reći da je 1 em=16 px. Ukoliko se podrazumevana veličina teksta poveća na 20 px, 1 em će imati vrednost od 20 px. Slično je i sa procentima, s tim što bi 100% predstavljalo 16 px. Kako bi se razlika između apsolutnih i relativnih jedinica na pravi način razumela, na slikama 13.2 i 13.3 prikazane su različite situacije i efekti koje one proizvode.



*Slika 13.2 – Upotreba različitih jedinica za postavljanje veličine fonta*

Na slici 13.2 može se videti da je na body elementu postavljena veličina teksta korišćenjem relativnih jedinica, i to na 100%. U takvoj situaciji, veličina teksta direktnih potomaka body elementa postavljena na različite načine biće kao u tabeli na slici 13.2. Može se zaključiti da je veličina teksta istovetna za sve tri vrednosti (1 em, 16 px, 100%).

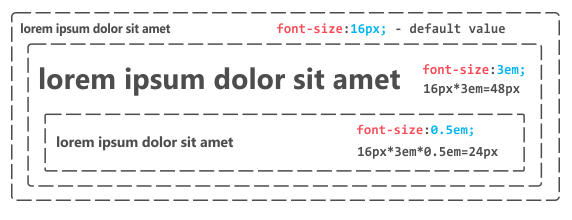
Prava razlika između relativnih i apsolutnih jedinica vidi se kada se na body elementu poveća veličina teksta. Upravo takvu situaciju ilustruje slika 13.3.



*Slika 13.3 – Upotreba različitih jedinica za postavljanje veličine fonta(2)*

Ukoliko se veličina teksta na body elementu poveća na 130%, može se zapaziti da tekst sa apsolutno izraženim jedinicama ostaje nepromenjene veličine, dok tekst sa relativnom veličinom adekvatno prati promenu veličine na roditeljskom body elementu.

Relativne vrednosti koje se izražavaju u procentima i em jedinicama relativne su u odnosu na roditeljski element elementa na kojem se primenjuju. To ilustruje sledeći primer, prikazan na slici 13.4.



*Slika 13.4 – Način na koji relativne jedinice za postavljanje veličine teksta funkcionišu*

Struktura elementa prikazanih na slici je sledeća:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <body style="font-size:1em;">   body text   <div style="font-size:3em;">    first div text    <div style="font-size:0.5em;">      second div text    </div>   </div>  </body> |

Unutar body dela se nalazi jedan div i unutar njega još jedan div. Body element, kao i svaki od divova poseduje pripadajući tekst. Na body elementu je postavljena veličina teksta od *1 em*. Na prvom direktnom potomku elementa body, postavljena je veličina teksta od *3 em*. Na ugnežđenom divu postavljena je veličina teksta na *0.5 em*. Sa slike 13.4 se može zaključiti da je veličina teksta poslednjeg diva veća od teksta koji se nalazi direktno unutar body elementa, i to zato što je njegova veličina relativna roditeljskom elementu. Veličina teksta poslednjeg ugnežđenog div elementa, izražena u pikselima, može se proračunati na sledeći način (ukoliko je veličina teksta podrazumevanih 16 px):

|  |  |
| --- | --- |
| 1 | 16px\*3\*0.5=24px |

Iz računa se može zaključiti da su u kalkulaciji veličine fonta ugnežđenog diva učestvovale veličine slova svih roditeljskih elemenata.

|  |
| --- |
| **Napomena**    *Sve navedene jedinice koje se mogu koristiti za definisanje veličine fonta mogu se koristiti i prilikom definisanja vrednosti nekih drugih svojstava kao što su margine, visina, širina i slično. Zbog toga je veoma bitno razumeti razliku između relativnih i apsolutnih jedinica, jer će se one intenzivno koristiti u nastavku kursa.* |

**Boja teksta**

Boja teksta u HTML dokumentu može se promeniti korišćenjem color svojstva CSS opisa. Vrednost ovoga svojstva može se definisati na jedan od tri sledeća načina:

* heksadecimalnom vrednošću
* nazivom boje
* RGB funkcijom

Vrednost color svojstva se može definisati u [**heksadecimalnom**](https://www.link-elearning.com/linkdl/opisPojma.php?id=144303) obliku. Heksadecimalnom vrednošću se zapravo definiše RGB kod. Primer definisanja boje u heksadecimalnom obliku je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | h1{  color:#00FF00;  } |

Prve dve heksadecimalne cifre definišu vrednost za crvenu boju, druge dve za zelenu, a poslednje dve za plavu.

CSS omogućava definisanje boje i korišćenjem **naziva**. Tako je jednostavno moguće napisati nešto ovako:

|  |  |
| --- | --- |
| 1  2  3 | h1{  color: green;  } |

Na kraju, boju teksta je moguće definisati i korišćenjem **RGB funkcije**. Ova funkcija prihvata tri parametra, od kojih svaki predstavlja spektar osnovne boje iskazan vrednošću od 0 do 255. Sledeći primer ilustruje boju teksta definisanu korišćenjem RGB funkcije:

|  |  |
| --- | --- |
| 1  2  3 | h1{  color: rgb(0,255,0);  } |

U svim prikazanim primerima, ilustrovano je postavljanje boje teksta h1 naslova na zelenu.  
  
Definisanje boje teksta korišćenjem RGB funkcije ili heksadecimalne vrednosti omogućavaju znatno bolju kontrolu i granularnost prilikom odabira nijanse boje. Drugim rečima, postoji znatno veći broj boja od onih koje su pokrivene CSS nazivima, tako da je za preuzimanje pune kontrole nad bojom teksta, uvek dobro koristiti heksadecimalnu vrednost ili RGB funkciju.

**Težina fonta – Bold**

Pored korišćenja <b> taga za kreiranje teksta koji je boldovan, moguće je korišćenje i CSS svojstva font-weight. Ovim svojstvom definiše se koliko će tanki ili debeli biti karakteri fonta koji se koristi. Primer korišćenja ovog svojstva je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | p {  font-weight: bold;  } |

Prikazanim CSS opisom je naznačeno da će tekst unutar svih paragraf elemenata na stranici biti boldovan.

Svojstvo font-weight može da prihvati četiri različite tekstualne vrednosti i devet numeričkih vrednosti. Tako je za definisanje debljine karaktera fonta moguće koristiti neku od sledećih tekstualnih vrednosti:

* Normal
* Bold
* Bolder
* lighter

Pored tekstualnih vrednosti, svojstvo font-weight može da prihvati i numeričke vrednosti od 100 do 900, sa korakom od 100. Vrednost 400 ima identičan efekat kao i vrednost normal, dok je vrednost 700 identična vrednosti bold. Primer definisanja debljine karaktera korišćenjem brojčane vrednosti je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | p {  font-weight: 900;  } |

**Italic**

Da bi se određeni tekst formatirao kao italic, moguće je koristiti svojstvo font-style. Sledeći primer ilustruje upotrebu ovog svojstva:

|  |  |
| --- | --- |
| 1  2  3 | p{  font-style:italic;  } |

Svojstvo font-style, pored vrednosti italic, može imati i vrednosti normal i oblique. Podrazumevana vrednost ovoga svojstva je normal, što znači da se inicijalno italic formatiranje ne primenjuje. Vrednost oblique definiše pojačan italic efekat, kreiranjem dodatnog zakošenja.

**Poravnanje teksta**

Za horizontalno poravnanje teksta koristi se svojstvo text-align. Dostupne vrednosti za horizontalno poravnanje su: left, right, center i justify. Left i Right vrednosti izvršiće poravnanje teksta po levoj, odnosno desnoj ivici okvira. Vrednost center će centrirati tekst, dok će vrednost justify poravnati tekst i po levoj i po desnoj ivici i pritom uticati na razmake između reči.

Sledeći primer ilustruje upotrebu svojstva za poravnanje teksta:

|  |  |
| --- | --- |
| 1  2  3 | #footer {     text-align: center;  } |

Pored horizontalnog poravnanja, CSS omogućava da se tekst poravna i po **vertikalnoj** osi. Poravnanje po vertikalnoj osi postiže se korišćenjem svojstva vertical-align. Ovo svojstvo zapravo omogućava da se izvrši vertikalno poravnanje različitog teksta u jednoj liniji. Primer upotrebe ovog svojstva je sledeći:

|  |  |
| --- | --- |
| 1  2  3 | .textBaseline {     vertical-align: baseline;  } |

Primer ilustruje poravnanje teksta po osnovnoj liniji kontejnerskog elementa. Uključujući navedenu vrednost baseline, za vrednost svojstva vertical-align mogu se navesti sledeće vrednosti:

* baseline – poravnava tekst po osnovnoj liniji roditelja
* top – poravnava vrh teksta sa vrhom najvišeg elementa u liniji
* text-top – poravnava vrh teksta sa vrhom roditeljskog elementa
* middle – postavlja tekst na sredinu roditeljskog elementa po visini
* bottom – poravnava dno teksta sa dnom najnižeg elementa u liniji
* text-bottom – poravnava dno teksta sa dnom roditeljskog elementa

**Uvlačenje teksta**

Uvlačenje teksta je moguće postići korišćenjem svojstva text-indent. Ovo svojstvo može da prihvati neku od apsolutnih vrednosti opisanih nešto ranije, pri čemu vrednost oslikava koliko će test biti uvučen. Primer upotrebe svojstva za uvlačenje teksta je sledeći:

|  |  |
| --- | --- |
| 1 | p {    text-indent: 50px; } |

Vrednost svojstva text-indent se može iskazati i u procentima, i tada se ona odnosi na širinu roditeljskog kontejnera u kome se nalazi tekst. To ilustruje sledeći primer:

|  |  |
| --- | --- |
| 1 | p {    text-indent: 10%;} |

**Razmak između reči i karaktera**

Korišćenjem CSS-a vrlo lako se može kontrolisati razmak između reči i karaktera teksta. Za kontrolisanje razmaka između karaktera koristi se svojstvo letter-spacing, dok se za definisanje razmaka između reči koristi svojstvo word-spacing. Sledeći primer ilustruje upotrebu ova dva svojstva:

|  |  |
| --- | --- |
| 1  2  3  4 | p {  letter-spacing: 8px;  word-spacing: 16px;  } |

**Razmak između redova teksta**

Pored razmaka između karaktera i pojedinačnih reči teksta, korišćenjem CSS-a moguće je uticati i na razmak između linija teksta. Preciznije rečeno, CSS omogućava definisanje visine jedne linije teksta, što direktno utiče na razmak između linija. Za definisanje visine jedne linije teksta koristi se svojstvo line-height, a korišćenje ovog svojstva prikazano je u sledećem primeru:

|  |  |
| --- | --- |
| 1  2  3 | p {     line-height:1.4em;  } |

**Transformacija teksta**

Kada se kaže transformacija teksta, u CSS-u se prevashodno misli na upotrebu svojstva text-transform. Ovo svojstvo može imati jednu od sledećih vrednosti:

* uppercase
* lowercase
* capitalize

Prikazanim svojstvima vrši se transformacija teksta koji je definisan unutar elementa i to tako što se sva slova pretvaraju u mala ili u velika, ili se postavlja prvo veliko slovo svake reči. Prilično je jasno da se sva slova pretvaraju u velika korišćenjem vrednosti uppercase i da se sva slova pretvaraju u mala korišćenjem svojstva lowercase. Postavljanje velikog početnog slova svake reči u tekstu postiže se korišćenjem vrednosti capitalize.

Sledeći primer ilustruje upotrebu svojstva text-transform:

|  |  |
| --- | --- |
| 1  2  3  4 | h1 {     text-transform: uppercase;  }  <h1>this is Heading</h1> |

U prikazanom primeru, definisano je svojstvo text-transform sa vrednošću uppercase, koje će se primeniti na svim h1 naslovima. U primeru je takođe naveden jedan takav naslov. Prilikom renderovanja na stranici, naslov će biti prikazan na sledeći način:

THIS IS HEADING

Može se primetiti da su sva slova naslova pretvorena u velika. Bitno je razumeti da u izvornom HTML dokumentu tekst ostaje onakav kakav jeste, a da se transformiše samo prikaz za korisnika. Otuda se ovo svojstvo i naziva text-transform.

**Dekorisanje teksta**

Pod dekorisanjem teksta smatra se postavljanje linije koja se može naći ispod, iznad ili na sredini teksta. Za postizanje nečeg takvog, koristi se svojstvo text-decoration, koje može imati sledeće vrednosti:

* none
* underline
* overline
* line-through

Nazivi vrednosti su prilično slikoviti, a primer koji ilustruje njihovu upotrebu je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4 | a  {     text-decoration: none;  } |

U navedenom primeru, stilizovan je prikaz teksta jednog linka. U primeru nije slučajno uzeto upravo stilizovanje jednog linka, s obzirom da se podrazumevano linkovi na stranici prikazuju podvučeni. Ukoliko je tako nešto potrebno ukinuti, može se iskoristiti upravo prikazani CSS opis.

**Senka teksta**

Korišćenjem CSS-a, moguće je postaviti i senku koju tekst postavlja na element ispod. To se postiže korišćenjem svojstva text-shadow. Primer upotrebe ovog svojstva je sledeći:

|  |  |
| --- | --- |
| 1 | h1 {    text-shadow: 2px 2px #ff0000; } |

U primeru se može primetiti da se vrednost svojstva text-shadow sastoji iz nekoliko delova. Sintaksa ovog svojstva je sledeća:

|  |  |
| --- | --- |
| 1 | text-shadow: h-shadow v-shadow blur-radius color; |

Analizom sintakse može se zaključiti da se prvi deo vrednosti odnosi na horizontalnu poziciju senke, drugi deo na njenu vertikalnu poziciju, treći deo na prečnik blur efekta, dok se poslednji deo vrednosti odnosi na boju senke.

[**Povratak na vrh strane**](https://www.link-elearning.com/linkdl/preview/index.php?pk=51bf85667ac477fb83423c0ba4cc29a5_ita_17&IDJedinica=1068962&c=jedinice)