

Stilizovanje teksta

Tekst je svakako jedan od najznačajnijih tipova sadržaja koje HTML dokumenti mogu posedovati. U prvom delu ovoga kursa prikazan je veliki broj različitih HTML elemenata čija je primarna uloga bila markiranje teksta (naslovi, paragrafi, liste, tabele itd). Ipak, sve do sada tekst naših HTML dokumenata prikazivao se onako kako je to browser želeo. Stoga će lekcija pred vama biti posvećena različitim tehnikama koje će nam omogućiti da samostalno utičemo na izgled i formatiranje teksta u HTML dokumentima.

Tipografija

Pre nego što pređemo na različite tehnike za stilizovanje teksta unutar jednog HTML dokumenata, potrebno je pozabaviti se osnovnim načelima tipografije. Tipografija je zapravo veština oblikovanja teksta kako bi se on prezentovao na razumljiv i čitljiv način.

Nekada davno, u vremenima nastanka prvih štamparija, tekst se na papir utiskivao korišćenjem karaktera koji su bili izliveni od metala. Skup takvih metalnih karaktera danas su u digitalnom obliku drugačije naziva **font**.

Svi fontovi dele se na sledeće osnovne tipove:

- Serif;
- Sans Serif;
- Monospace.

Serif fontovi karakterišu detalji 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 najpoznatija familija fontova ove vrste svakako je 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 na to da sans na francuskom jeziku znači „bez”. Zbog odsustva takozvanih ukrasa, fontovi ove kategorije izgledaju čistije i modernije. Najpopularnije familije fontova ove vrste su Arial, Tahoma, Verdana, Helvetica itd.

Monospace su oni fontovi kod kojih svi karakteri zauzimaju isti horizontalni prostor. Za početak njihove primene uglavnom se 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 webu je Courier New (ili Courier).

Razlike između tri navedene kategorije fontova prikazane su slikom 17.1.



Slika 17.1. Razlike između tri osnovne kategorije fontova

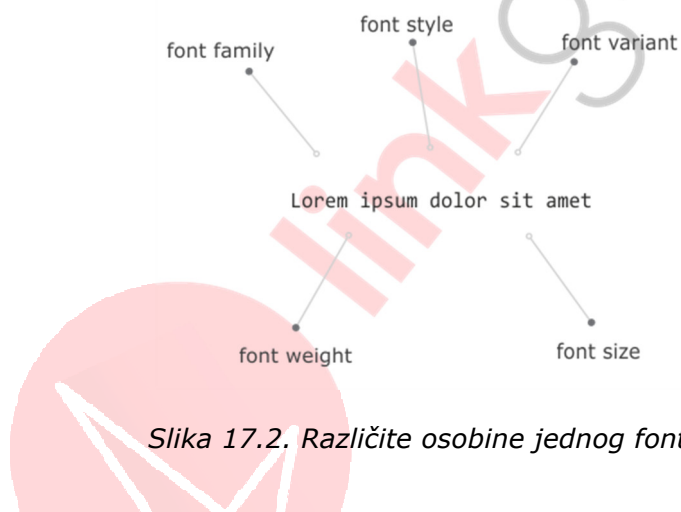
Familija fonta

Više srodnih fontova grupiše se u jednu familiju. Tako se može reći da je, na primer, Verdana familija fonta, a Verdana Bold font. Familija fonta određuje vizuelne karakteristike fonta. Tako svi fontovi jedne familije poseduju zajedničke vizuelne karakteristike. Ono što ih razlikuje može se nazvati varijacijom na jednu istu temu. Neke od najznačajnijih takvih varijacija su: light, italic, regular, bold itd.

Veoma često se pojam familije fonta izjednačava sa pojmom fonta, pa se tako može čuti: Verdana font. To je potpuno legitimno i u većini slučajeva misli se na regular, odnosno normal varijaciju iz neke familije fontova.

Osobine fontova na webu

U dosadašnjem toku ovog programa jedina osobina fonta na koju smo direktno uticali bila je veličina karaktera. Sve ostale osobine korišćenog fonta u potpunosti su zavisile od browsera unutar koga su prikazivani naši HTML dokumenti. To praktično znači da je za prikaz teksta bio korišćen podrazumevani font browsera (u Google Chromeu je to font Times New Roman). Kako bismo bili u mogućnosti da promenimo osobine fonta koji se koristi u dokumentu, prvo je potrebno znati koje su to sve osobine na koje je moguće uticati (slika 17.2).



Slika 17.2. Različite osobine jednog fonta

Pitanje

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

- a) **Serif;**
- b) Sans Serif;
- c) Monospace;

Objašnjenje

Serif fontovi karakterišu detalji na krajevima linija. Ti detalji se nazivaju serif, pa odatle i naziv za ovaj tip fontova.

Na slici su prikazane najznačajnije osobine fontova na koje je moguće uticati korišćenjem CSS-a. Reč je o osobinama koje direktno utiču na izgled karaktera. Pored nabrojanih osobina, CSS poseduje i razna druga svojstva kojima je moguće uticati na pojedinačne reči, redove ili kompletne pasuse. Sva takva svojstva biće obrađena u nastavku ove lekcije.

Izbor familije fonta

Prvi korak u promeni osobina fonta HTML dokumenta podrazumeva definisanje njegove familije. Kao što je rečeno, familija kojoj neki font pripada osnovna je osobina koja određuje njegove vizuelne karakteristike. Zbog toga se pojam familije fonta veoma često poistovećuje se pojmom fonta, pa ćete i u ovoj lekciji na dosta mesta pročitati pojam font umesto familija fonta.

Familija fonta može se postaviti korišćenjem `font-family` svojstva. Sledeći primer ilustruje upotrebu ovog svojstva:

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

Prikazanim CSS opisom familija fonta kompletnog HTML dokumenta postavljena je na vrednost `Georgia`.

U ovom trenutku veoma je bitno razumeti da je neophodno da definisani font postoji na korisničkom kompjuteru. U protivnom, neće biti moguće njegovo korišćenje i browser će upotrebiti svoj podrazumevani font. Identična je situacija i kod, na primer, Word dokumenata. Ukoliko na kompjuteru na kome se pregleda neki Word dokument ne postoji font koji je u njemu korišćen, tekst će biti prikazan korišćenjem nekog drugog fonta koji je dostupan na sistemu. Drugim rečima, podrazumevano **fontovi nisu sastavni deo web sajta ili aplikacije, već se čitaju sa korisničkog kompjutera**. Uzimajući u obzir rečeno, kao logično nameće se pitanje *kako da znam koje fontove mogu koristiti prilikom kreiranja web sajtova i aplikacija*. I, zaista, nikada ne možete biti potpuno sigurni da će neki font koji definišete `font-family` svojstvom postojati na klijentskim uređajima. Ipak, postoji određeni broj fontova koji su sigurno dostupni na svakom uređaju, te se stoga nazivaju *Web safe fontovi*.

Web safe fontovi

Određeni broj fontova smatra se sigurnim web fontovima, zato što postoje na gotovo svim operativnim sistemima koji su danas u upotrebi (Windows, macOS, Linux, Android, iOS). Ti fontovi su sledeći:

- Arial;
- Courier New;
- Georgia;
- Times New Roman;
- Trebuchet MS;
- Verdana.

CSS svojstvo `font-family` može da prihvati i veći broj vrednosti. U takvoj situaciji koristiće se vrednost prvog fonta koji postoji na korisničkom uređaju. Primer definisanja svojstva `font-family` sa više vrednosti je sledeći:

```
h1 {  
  font-family: "Times New Roman", Helvetica, sans-serif;  
}
```

Prikazanim CSS opisom browser će pokušati da koristi font Times New Roman, a ukoliko ga ne pronađe, potražiće font Helvetica. Ako ni Times New Roman ni Helvetica nisu dostupni, biće korišćen bilo koji Sans Serif font koji je dostupan na sistemu korisnika, jer je kao poslednja vrednost svojstva `font-family` postavljeno `sans-serif`.

Takođe, veoma je bitno primetiti još jedno značajno pravilo u upravo prikazanom primeru. Ukoliko se naziv familije fonta sastoji iz više reči, takav naziv postavlja se između navodnika.

Generički fontovi

Upravo prikazani primer prvi put je ilustrovao upotrebu jedne od generičkih vrednosti koje `font-family` može da prihvati. Naime, `sans-serif` nije naziv bilo kojeg fonta, odnosno familije fontova, već je reč o jednoj od kategorija ilustrovanih na početku ove lekcije.

CSS omogućava da se za vrednost `font-family` svojstva postavi nekoliko generičkih naziva fontova, od kojih su najznačajnije:

- Serif;
- Sans Serif;
- Monospace.

Korišćenjem generičkih naziva fontova odabir konkretnog fonta zavisi samo od browsera. Jedino što je potrebno da browser ispuni prilikom odabira jeste to da odabrani font pripada kategoriji definisanoj generičkom vrednošću. Tako će u slučaju navođenja `sans-serif` generičke vrednosti browser odabrati svoj podrazumevani font iz grupe Sans Serif fontova.

Generičke vrednosti najčešće se navode kao poslednje, baš kao što je to prikazano u prethodnom primeru. Na taj način one će biti aktivirane samo ukoliko browser ne pronađe nijedan od prethodno definisanih fontova.

Težina fonta – Bold

Sledeća osobina fonta koju ćemo razmotriti odnosi se na debljinu karaktera. Ona se definiše korišćenjem `font-weight` CSS svojstva. Ovim svojstvom definiše se koliko će tanki ili debeli biti karakteri fonta koji se koristi. Primer korišćenja ovog svojstva je sledeći:

```
p {  
  font-weight: bold;  
}
```

Prikazanim CSS opisom naznačeno je 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:

```
p {
  font-weight: 900;
}
```

U jednoj od prethodnih lekcija u kojoj su obrađeni različiti HTML elementi za markiranje teksta, prikazani su i `b` i `strong` elementi. Reč je o elementima koji podrazumevano od strane browsera dobijaju sledeću stilizaciju:

```
strong, b {
  font-weight: bold;
}
```

Svakako se preporučuje korišćenje CSS opisa za postavljanje *bold* teksta. Ipak, ukoliko je određeni deo teksta potrebno i semantički naglasiti, dobro je takav tekst smestiti unutar `strong` elementa.

Italic

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

```
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 zakojenja.

Veličina teksta

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

```
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

Relativne jedinice su:

- % – procenat;
- em;
- rem.

Apsolutne i relativne jedinice

Apsolutne jedinice koriste se da apsolutno odrede veličinu teksta. Drugim rečima, apsolutna veličina slova uvek će biti ista, bez obzira na veličinu prikaza ili bilo koji drugi parametar. Ovakve jedinice uglavnom su preuzete su iz štampanih medija, sa izuzetkom piksela, koji su karakteristični za digitalni prikaz slike.

Relativne jedinice su one koje svoju finalnu vrednost formiraju na osnovu bazne veličine teksta. Na primer, podrazumevana veličina teksta u većini browsera je 16px. Tako se može reći da je 1 em = 16 px. Ukoliko se podrazumevana veličina teksta poveća na 20 px, 1 em imaće 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 17.3. i 17.4 prikazane su različite situacije i efekti koje one proizvode.

	body {font-size:100%;}
font-size:1em	lorem ipsum dolor sit amet
font-size:16px	lorem ipsum dolor sit amet
font-size:100%	lorem ipsum dolor sit amet

Slika 17.3. Upotreba različitih jedinica za postavljanje veličine fonta

Na slici 17.3. 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 17.3. Može se zaključiti da je veličina teksta istovetna za sve tri vrednosti (1em, 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 17.4.

	<code>body {font-size:130%;}</code>
<code>font-size:1em</code>	lorem ipsum dolor sit amet
<code>font-size:16px</code>	lorem ipsum dolor sit amet
<code>font-size:100%</code>	lorem ipsum dolor sit amet

Slika 17.4. 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.

em i procenti

Kada je u pitanju definisanje veličine teksta, može se reći da se procenti i `em` jedinice ponašaju istovetno, stoga će u nastavku biti ilustrovano korišćenje `em` jedinica, a sve što bude prikazano važi i za procenete. Procenti i `em` jedinice relativni su direktnom roditeljskom elementu. To ilustruje primer prikazan slikom 17.5.



Slika 17.5. Kako funkcionišu `em` jedinice?

Struktura elemenata prikazanih slikom je sledeća:

```
<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 nalazi se jedan `div` i unutar njega još jedan `div`. Svi ovi elementi poseduju tekstualne sadržaje. Na `body` elementu postavljena je veličina teksta od *1 em*. Na prvom direktnom potomku elementa `body` postavljena je veličina teksta od *3 em*. Na ugnežđenom `div`-u postavljena je veličina teksta na *0,5 em*. Sa slike 17.5. može se zaključiti da je veličina teksta poslednjeg `div`-a veća od teksta koji se nalazi direktno unutar `body` elementa, i to zato što je njegova veličina relativna roditeljskom elementu. Veličina teksta poslednjeg `div` elementa u prikazanoj strukturi može se proračunati na sledeći način (ukoliko je veličina teksta podrazumevanih 16 px):

$$16\text{px} * 3 * 0.5 = 24\text{px}$$

Iz računa se može zaključiti da su u kalkulaciji veličine fonta ugnežđenog `div` elementa učestvovala veličine slova svih roditeljskih elemenata.

rem

Još jedna od relativnih jedinica koja se može koristiti za definisanje veličine teksta unutar HTML dokumenata jeste `rem`. Reč je zapravo o jedinici koja se interpretira kao *Root EM*, pa otuda i njen naziv – `rem`.

Jedinica `rem` veoma je slična već opisanoj `em` jedinici. Razlika se ogleda u baznom elementu koji se uzima kao referenca na osnovu koje će biti proračunata konačna vrednost. Kod `em` jedinice to je direktan roditeljski element. Sa druge strane, `rem` jedinica uvek je relativna korenom elementu HTML dokumenta – `html`. Tako, prilikom korišćenja `rem` jedinica, u finalnu kalkulaciju ne ulaze vrednosti koje se nalaze na svim roditeljskim elementima koji se nalaze u hijerarhiji, kao što je to bio slučaj na slici 17.5. Jedini element koji se uzima kao relevantan jeste `html` element. Naravno, ukoliko na `html` elementu nije definisana veličina teksta, za baznu se uzima osnovna veličina teksta browsera, što je uglavnom 16 px. Osobine `rem` jedinica ilustrovane se slikom 17.6.



Slika 17.6. Kako funkcionišu `rem` jedinice?

Za demonstraciju osobina `rem` jedinica poslužio je primer istovetan prethodnom. HTML dokument poseduje baznu veličinu teksta od 16 px. To je veličina teksta i unutar `body` elementa. Veličina teksta prvog `div` elementa ne menja se bilo da se koriste `em` ili `rem` jedinice. Ipak, razlike postoje na poslednjem `div` elementu u strukturi. Kada se koriste `rem` jedinice, veličina njegovog teksta iznosi 8 px, s obzirom na to da je relativna veličini teksta `html` elementa, koja iznosi 16px.

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 definisati korišćenjem `color` svojstva CSS opisa. Vrednost se može izraziti korišćenjem svih onih pristupa koji su opisani u jednoj od prethodnih lekcija ovog modula.

HEX:

```
h1 {  
  color: #00FF00;  
}
```

Imenovana boja:

```
h1 {  
  color: green;  
}
```

RGB:

```
h1 {  
  color: rgb(0,255,0);  
}
```

HSL:

```
h1 {  
  color: hsl(120, 100%, 50%);  
}
```

U svim prikazanim primerima ilustrovano je postavljanje boje teksta `h1` naslova na zelenu.

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:

```
#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:

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

```
p {  
    text-indent: 50px;  
}
```

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

```
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:

```
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 sledećim primerom:

```
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`:

```
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 tako nečeg koristi se svojstvo `text-decoration`, koje može imati sledeće vrednosti:

- none;
- underline;
- overline;
- line-through.

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

```
a
{
  text-decoration: none;
}
```

U navedenom primeru stilizovan je prikaz teksta jednog linka. U primeru nije slučajno uzeto upravo stilizovanje jednog linka, s obzirom na to 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:

```
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:

```
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.

Korišćenje fontova koji ne postoje na korisničkom računaru

Svi dosadašnji primeri ilustrovali su upotrebu fontova koji se nalaze na klijentskom računaru. Ipak, unutar HTML dokumenata moguće je koristiti i fontove koji se ne nalaze na korisničkom uređaju. Takav pristup veoma je koristan ukoliko je u HTML dokumentu potrebno upotrebiti neke fontove koji nisu standardni i koji ne pripadaju onoj listi sigurnih web fontova.

Dva najčešća pristupa za korišćenje fontova kojih nema na klijentskom sistemu su:

- samostalno hostovanje fonta fizičkim smeštanjem fajlova fonta na istu lokaciju na kojoj se nalazi sajt ili aplikacija;
- korišćenje nekih od web fontova.

Samostalno hostovanje fonta

Samostalno hostovanje fonta podrazumeva smeštanje fajlova fonta na istu lokaciju na kojoj se nalazi i web sajt ili aplikacija. Da bi se jedan takav font koristio, lokaciju na kojoj se nalazi potrebno je definisati korišćenjem specijalnog CSS koda:

```
@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` zapravo je 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. 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 `tff`, `otf` ili `svg`. Neophodno je znati da različiti browseri podržavaju fontove različitih formata (tabela 17.1), te je stoga najbolje posedovati jedan isti font u nekoliko oblika, baš kao što je to učinjeno i u primeru (EOT za starije verzije Internet Explorera, a WOFF za ostale).

Format	Podržanost
EOT (Embedded OpenType)	IE
WOFF (Web Open Font Format)	IE, Edge, Firefox, Chrome, Safari, Opera, iOS, Android
SVG	Safari, iOS
TTF (TrueType), OTF (OpenType)	Edge, Firefox, Chrome, Safari, Opera, iOS, Android
WOFF 2.0 (Web Open Font Format)	Edge, Firefox, Chrome, Safari, Opera, iOS, Android

Tabela 17.1. Različiti formati fontova koje browseri mogu razumeti

Font definisan korišćenjem konstrukcije `@font-face` može se iskoristiti na nekom od HTML elemenata na identičan način kao i fontovi koji postoje na uređaju klijenta:

```
p {  
  font-family: 'My Nicer Font';  
}
```

Korišćenje web fontova

Još jedan od načina za korišćenje fontova koji nisu smešteni na klijentskom uređaju podrazumeva korišćenje takozvanih web fontova. To je pristup koji na današnjem modernom webu poseduje veliku popularnost, pa čak i sam Google poseduje repozitorijum sa velikim brojem takvih fontova, koje je moguće besplatno koristiti (<https://fonts.google.com/>). Njihova upotreba zasniva se na uključivanju određenog fonta u HTML dokument korišćenjem `link` elementa:

```
<link href="https://fonts.googleapis.com/css?family=Roboto"  
rel="stylesheet">
```

Prikazana linija ilustruje uključivanje Roboto fonta u HTML dokument. Uključivanje se obavlja korišćenjem `link` elementa, koji je potrebno smestiti unutar zaglavlja dokumenta (head sekcija).

Ovako uključen web font može se iskoristiti na nekom od HTML elemenata, kao i svaki drugi font do sada:

```
p {  
    font-family: 'Roboto';  
}
```

Rezime

- Tipografija je tehnika za prikaz teksta na razumljiv i čitljiv način.
- Serif fontove karakterišu detalji na krajevima linija.
- Sans Serif fontovi su oni fontovi koje odlikuje odsustvo detalja koje poseduju Serif fontovi.
- Monospace su oni fontovi kod kojih svi karakteri zauzimaju isti horizontalni prostor.
- Izbor fonta za prikaz teksta nekog elementa vrši se postavljanjem `font-family` svojstva.
- Izbor veličine teksta postiže se korišćenjem svojstva `font-size`.
- Boja teksta u HTML dokumentu može se promeniti korišćenjem `color` svojstva CSS opisa.
- Boldovan tekst moguće je definisati korišćenjem CSS svojstva `font-weight`.
- Da bi se određeni tekst formatirao kao *italic*, moguće je koristiti svojstvo `font-style`.
- Za horizontalno poravnanje teksta koristi se svojstvo `text-align`.
- Poravnanje po vertikalnoj osi postiže se korišćenjem svojstva `vertical-align`.
- Uvlačenje teksta moguće je postići korišćenjem svojstva `text-indent`.
- Za kontrolisanje razmaka između karaktera koristi se svojstvo `letter-spacing`, dok se za definisanje razmaka između reči koristi svojstvo `word-spacing`.
- Za definisanje visine jedne linije teksta koristi se svojstvo `line-height`.
- Postavljanje linije koja se može naći ispod, iznad ili na sredini teksta postiže se korišćenjem svojstva `text-decoration`.
- Za transformaciju teksta, odnosno za konvertovanje malih slova u velika i obrnuto, koristi se svojstvo `text-transform`.