
Stilizovanje teksta i druge osnovne tehnike

U ovom poglavlju proučavamo primenu CSS-a za stilizovanje teksta, predočavamo mnoge osnovne tehnike CSS-a i odgovaramo na neka od često postavljanih pitanja o ovoj tehnologiji. Ukoliko se prvi put srećete sa CSS-om, kroz navedene primere upoznaćete razna svojstva i njihovu primenu, i steći ćete čvrstu osnovu koju možete nadograđivati tako što ćete samostalno eksperimentisati. Upućeni ma u CSS ovo poglavlje biće efikasan podsetnik u trenucima kada ne mogu da se sete kako da ostvare određeni efekat.

Primeri iz ovog poglavlja dobro su podržani u raznovrsnim čitačima i verzijama; naravno, i dalje je važno da isprobate kôd u različitim čitačima. Iako možda postoje male neusaglašenosti sa starijim čitačima ili uopšte nema podrške za ove tehnike, nijedno od predstavljenih rešenja ne bi trebalo da izazove ozbiljne probleme.

Kako da zamenim oznake `` CSS stilovima?

Jednom, davno, pre nego što je CSS imao široku podršku, Web dizajneri su tekst na stranama stilizovali pomoću oznake ``. Međutim, čitači – počevši od verzije 4 – podržavaju stilizovanje teksta pomoću CSS-a, pa više nema ubedljivog razloga da se i dalje koriste oznake ``.

Evo primera: ako ste koristili oznake ``, morali ste da zadate stil za svaki pasus na strani na sledeći način:

```
<p><font color="#800080" face="Verdana, Geneva, Arial, Helvetica, sans-serif">These stuffed peppers are lovely as a starter, or as a side dish for a Chinese meal. They also go down well as part of a buffet and even children seem to like them.</font></p>
```

Rešenje

Pomoću CSS-a, možete u kaskadnom opisu stila da zadate boju i font za pasus. Da bismo uklonili oznake `` iz gornjeg primera, svojstvu `color` elementa `p` dodelili smo vrednost `#800080`, a svojstvu `font-family` zadali smo vrednost `Verdana, Geneva, Arial, Helvetica, sans-serif`:

basicfont.css (odlomak)

```
p {
  color: #800080;
  font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
}
```

Svaki naredni tekst koji budete dodavali dokumentu između oznaka `<p>` imaće ovakav stil – nije potreban nikakav dodatni HTML kôd. To će vam olakšati život ukoliko klijent iznenada odluči da promeni font Verdana u Times na 100 strana Web lokacije!

Da li da za veličinu fonta koristim piksele, tačke, em ili nešto drugo?

U CSS-u možete zadati veličinu teksta pomoću svojstva `font-size` na sledeći način:

```
font-size: 12px;
```

U prethodnom slučaju, veličinu smo zadali u pikselima, ali vrednost svojstva `font-size` može se navesti u raznim drugim jedinicama. Da biste odabrali odgovarajuću mernu jedinicu, morate da znate kakve su relativne prednosti svake od njih.

Rešenje

Zadavanje veličine fonta u mernim jedinicama

U tabeli 2.1 navedene su jedinice koje možete koristiti za izražavanje veličine fonta.

Tabela 2.1. Dostupne jedinice za zadavanje veličine fonta

Identifikator jedinice	Odgovarajuća jedinica
pt	tačka
pc	pajk
px	piksel
em	em
ex	eks
%	procenat

Razmotrimo redom svaku jedinicu.

Tačke i pajkovi

```
p {
  font-size: 10pt;
}
```

Trebalo bi da izbegavate da pomoću *tačaka* i *pajkova* stilizujete tekst za prikazivanje na ekranu. Ova jedinica je izvrsna za zadavanje veličine fonta pri pripremi za štampu, jer je tačka kao merna jedinica i osmišljena u tu svrhu. Tačka ima nepromenljivu veličinu od 1/72 inča (0,35 mm), dok je pajk jednak jednoj šestini inča (4,2 mm). Štampani dokument čiji su fontovi definisani pomoću ovih jedinica, prikazuju se upravo onako kako ste hteli – 4,2 milimetra uvek predstavlja istu meru bez obzira na to da li štampate na strani A4 ili na bilbordu. Međutim, računari ne mogu tačno da predvide fizičku veličinu u kojoj će se elementi prikazati na monitoru, te (loše) nagađaju veličinu tačke ili pajka, a rezultat se razlikuje od jedne platforme do druge.

Ukoliko pravite opis stila za štampanje, ili dokument koji je namenjen za štampu – ne za prikaz na ekranu – trebalo bi da koristite tačke i pajkove. Međutim, preporučujemo da ih izbegavate u Web dizajnu.

Pikseli

```
p {
  font-size: 12px;
}
```

Mnogi dizajneri vole da navode veličinu fonta u *pikselima*, jer ova jedinica pomaže da se očuva konsistentnost prikaza teksta u različitim čitačima i na različitim platformama. Međutim, kada se koriste mere u pikselima, zanemaruju se sva korisnička podešavanja čitača, pa korisnik u mnogim čitačima ne može izmeniti veličinu slova koju je dizajner zadao u pikselima. To je ozbiljan problem za posetioce Weba koji moraju da uvećaju tekst kako bi ga jasno videli.

Iako pikseli možda deluju kao najlakša opcija za zadavanje veličine fonta, ukoliko se može koristiti druga metoda, izbegavajte piksele, posebno za velike blokove teksta. Ako pravite dokument za štampu, ili opis stila za štampu, trebalo bi da potpuno izbegavate piksele. Pikseli nemaju značenje u svetu štampe; kao i kada se za prikazivanje na ekranu koristi tačka, ukoliko aplikacijama namenjenim za štampu zadajete veličinu fonta u pikselima, one će pokušati da pogode u kojoj veličini bi font trebalo da se prikaže na papiru – s pogrešnim ishodom.

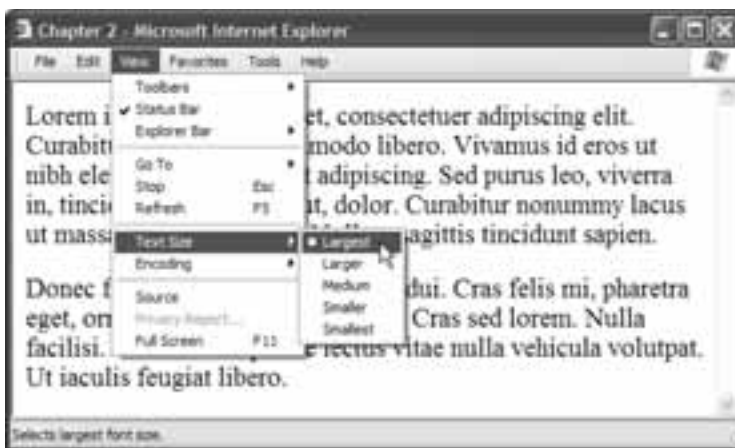
Em

Mera *em* je relativna jedinica mere za veličinu fonta: jedan em je jednak visini velikog slova M u podrazumevanoj veličini fonta. U kaskadnim opisima stilova, jedan em je jednak podrazumevanoj veličini fonta, odnosno veličini fonta roditeljskog elementa ukoliko je zadata veličina različita od podrazumevane.

Ako upotrebljavate em (ili bilo koju drugu relativnu mernu jedinicu) za zadavanje veličine svih fontova, korisnici će moći da izmene veličinu teksta tako da odgovara veličini koju su podesili u svojim čitačima Weba. Kao primer, napisaćemo deklaraciju kojom se zadaje da se tekst u elementu `p` prikazuje u veličini od `1em`:

```
p {
  font-size: 1em;
}
```

Kada korisnik Internet Explorera 6 u kome je za veličinu teksta odabrana opcija **Medium** poseti stranu, videće tekst pasusa prikazan na slici 2.1.



Slika 2.1. Prikaz u prozoru čitača kada svojstvo `font-size` ima vrednost `1em`, a veličina teksta je **Medium**

Ukoliko korisnik odabere **Largest** za veličinu teksta, tekst u fontu veličine `1em` prikazaće se kao na slici 2.2.

Tačno je da ćete, ukoliko koristite jedinicu `em` za zadavanje veličine fonta, imati manje uticaja na način na koji korisnici vide dokument. Međutim, ovakav pristup znači da, na primer, korisnici kojima je neophodan veoma krupan font, mogu da čitaju sadržaj vašeg dokumenta – što je, verovatno, razlog zbog kog stavljate taj tekst na Web stranu.

Vrednosti u jedinicama `em` mogu se zadavati pomoću decimalnih brojeva. Na primer, da biste prikazali tekst za 10% manji od podrazumevane veličine koju je postavio korisnik (ili veličine fonta njegovog roditeljskog elementa), primenite sledeće pravilo:

```
p {
  font-size: 0.9em;
}
```



Slika 2.2. Prikaz u prozoru čitača kada svojstvo font-size ima vrednost 1em, a veličina teksta je **Largest**

Ukoliko želite da prikazete tekst za 10% veći od podrazumevane ili nasleđene veličine, upotrebite ovo pravilo:

```
p {
  font-size: 1.1em;
}
```

Ex

Ex je relativna jedinica mere koja odgovara visini malog slova *x* u podrazumevanoj veličini fonta. Teorijski, ako zadate da svojstvo `font-size` pasusa ima vrednost `1ex`, velika slova u tekstu biće visoka kao što bi bilo visoko malo slovo *x* kada veličina fonta ne bi bila zadata (a mala slova će biti proporcionalno manja od velikih).

Nažalost, moderni čitači još uvek ne podržavaju tipografske odlike neophodne da bi se precizno odredila veličina teksta – obično nagađaju koliko otprilike iznosi ova jedinica. Zato se u trenutku pisanja ovih redova jedinica *ex* retko koristi.

Procenti

```
p {
  font-size: 100%;
}
```

Kao u slučaju jedinica *em* i *ex*, kada se veličine fontova zadaju u *procentima*, poštuje se korisnikov izbor veličine teksta, i korisnik ih može menjati. Ako zadate da veličina fonta za element *p* bude 100%, tekst će se prikazati s podrazumevanim korisničkim podešavanjima za veličinu fonta (isto bi bilo kada bi se za veličinu fonta odabrao `1em`). Ukoliko smanjite procenat, smanjiće se i tekst:

```
p {
  font-size: 90%;
}
```

Povećate li procenat, i tekst će se uvećati:

```
p {  
  font-size: 150%;  
}
```

Zadavanje veličina fontova pomoću ključnih reči

Za zadavanje veličine teksta, umesto numeričkih vrednosti možete koristiti apsolutne i relativne rezervisane reči (engl. *keywords*).

Apsolutne rezervisane reči

Veličina teksta u CSS-u može se zadati pomoću bilo koje od sledećih sedam apsolutnih rezervisanih reči:

- `xx-small`
- `x-small`
- `small`
- `medium`
- `large`
- `x-large`
- `xx-large`

Te rezervisane reči su definisane relativno jedna u odnosu na drugu i čitači ih implementiraju na različite načine. U većini čitača, font veličine `medium` prikazuje se u istoj veličini kao nestilizovan tekst, pri čemu se veličina definisana ostalim rezervisanim rečima određuje prema tome, u različitom stepenu. Međutim, u Internet Exploreru 5 (i verziji 6, zavisno od tipa dokumenta), font veličine `small` isti je kao nestilizovan tekst.

Mere zadate pomoću ključnih reči smatraju se apsolutnim jer se ne nasleđuju od roditeljskog elementa. Ipak, za razliku od apsolutnih vrednosti za visinu – na primer, u pikselima i tačkama – uz ovakve mere dozvoljeno je menjanje veličine teksta u čitaču i poštuju se korisnikova podešavanja. Glavni problem s korišćenjem apsolutnih rezervisanih reči jeste činjenica da, na primer, tekst veličine `x-small` može biti sasvim čitljiv u jednom čitaču, dok će u drugom slova biti veoma sitna.

Relativne rezervisane reči

Veličina teksta zadata pomoću relativnih rezervisanih reči – `larger` i `smaller` – određuje se prema roditeljskom elementu na isti način na koji je preciziraju jedinice `em` i procenat. Ukoliko veličinu elementa `<p>` zadate pomoću apsolutne rezervisane reči `small`, a želite da se istaknut tekst prikazuje veći od običnog teksta, mogli biste dodati naredni opis stila:

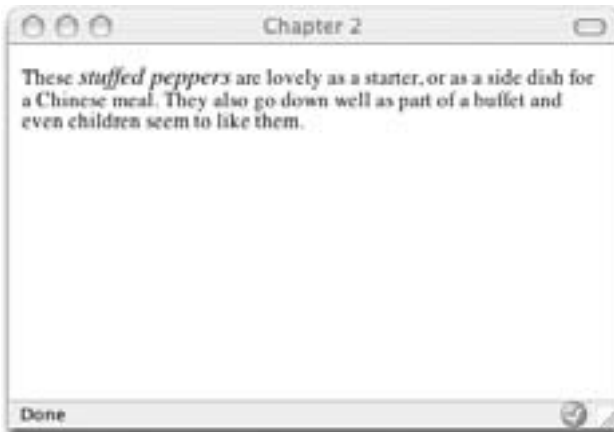
relative.css

```
p {
  font-size: small;
}
em {
  font-size: larger;
}
```

Sledeći kôd bi se prikazao kao na slici 2.3, jer je tekst između oznaka `` i `` veći od roditeljskog elementa `p`:

relative.html (odlomak)

```
<p>These <em>stuffed peppers</em> are lovely as a starter, or as a side dish
for a Chinese meal. They also go down well as part of a buffet and even
children seem to like them.</p>
```



Slika 2.3. Istaknut tekst je veći od teksta pasusa u kome se nalazi

Relativno zadavanje veličine i nasleđivanje

Ukoliko pribegavate relativnom zadavanju veličine, zapamtite da će taj element naslediti početnu veličinu od roditeljskog elementa, a da se njegova veličina potom prilagođava prema dodatnim podešavanjima. To je jasno u dizajnerskim rešenjima u kojima su elementi ugnežđeni na jednostavan način; međutim, ovakav obrazac nasleđivanja može postati problematičan na stranicama sa ugnežđenim tabelama u kojima roditeljski element nije uvek očigledan – tu su putevi nasleđivanja uistinu čudni! Naredni primer ilustruje takvu situaciju.

Moj opis stila sadrži pravilo kojim se određuje da se tekst u elementu `td` prikazuje u veličini 80%. To je nešto manje od podrazumevanih veličina fontova korisnika, ali će oni moći da izmene veličinu teksta:

nesting.css

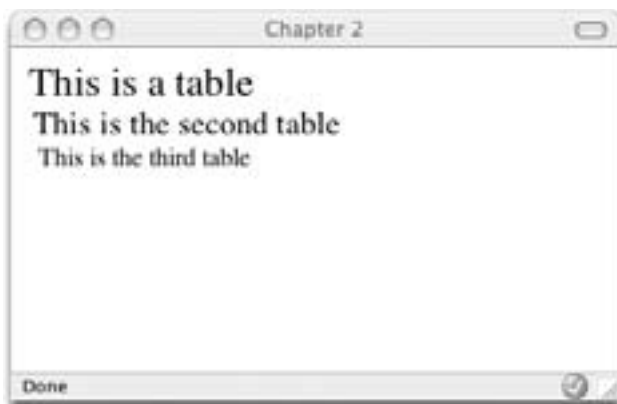
```
td {  
    font-size: 80%;  
}
```

Na strani bez ugnežđenih tabela, dosledno će se prikazati nešto sitniji tekst. Međutim, na strani sa ugnežđenom tabelom, kao u narednom odlomku koda, veličina teksta u svakoj ugnežđenoj tabeli biće 80% veličine fonta table koja je sadrži:

nesting.html (odlomak)

```
<table>  
  <tr>  
    <td>This is a table  
      <table>  
        <tr>  
          <td>This is the second table  
            <table>  
              <tr>  
                <td>This is the third table</td>  
              </tr>  
            </table>  
          </td>  
        </tr>  
      </table>  
    </td>  
  </tr>  
</table>
```

Rezultat se vidi na slici 2.4. Kao što vidite, tekst se progresivno smanjuje u svakoj narednoj ugnežđenoj tabeli.



Slika 2.4. Relativno zadavanje veličine fonta u ugnežđenim tabelama

Diskusija

Kad razmatrate kako da zadate veličinu teksta, najbolje je ono rešenje koje omogućava svim korisnicima da je menjaju i osigurava da tekst odgovara parametrima čitača. Relativno zadate veličine fonta verovatno će se pravilno prikazivati na stranama dizajniranim pomoću CSS-a i na stranama s jednostavnim tabelama, ali će možda biti teško primenljive na složene ugneždene tabele zbog načina na koji elementi nasleđuju veličinu. Međutim, da biste ostvarili makar osnovni nivo dostupnosti Web strana, neophodno je da omogućite korisnicima da menjaju veličinu fonta u stepenu koji im odgovara.

Ukoliko stranu dizajnirate tako da je moguće menjati veličinu teksta na njoj, izbeći ćete i problem čest u čitačima u kojima je dozvoljeno menjanje veličine piksela na stranicama čiji su dizajneri pretpostavili da će – zadajući veličinu fonta u pikselima – moći da fiksiraju visine elemenata u kojima se tekst nalazi, ili da postave tekst preko slike zadate visine. Ovakav pristup je efikasan u Internet Exploreru koji ne menja veličinu teksta zadatu u pikselima, ali može napraviti veliku pometnju u Firefoxu, u kome se visine polja s tekстом nikako ne mogu znati, pa se tekst lako može izliti van polja.

Kako da postignem da se tekst prikazuje u određenom fontu?

Rešenje

Font teksta zadaćete pomoću svojstva `font-family` na sledeći način:

```
p {  
  font-family: Verdana;  
}
```

Diskusija

CSS omogućava da pored određenih fontova (na primer, Verdana ili Times) odaberete generičke (opšte) porodice fontova:

- serif
- sans-serif
- monospace
- cursive
- fantasy

Kada zadajete font, morate imati na umu da korisnici verovatno nemaju iste fontove kao vi na svom računaru. Ukoliko odaberete font koji oni nemaju, tekst će se prikazati u podrazumevanim fontovima njihovih čitača, bez obzira na ono što ste zadali.

Da biste to izbegli, možete navesti generička imena fontova i prepustiti korisničkim sistemima da odaberu font. Na primer, ukoliko želite da se dokument prikazuje u beserifnom fontu kakav je Arial, mogli biste da upotrebite naredno pravilo stila:

```
p {
  font-family: sans-serif;
}
```

Ipak, verovatno ćete hteti veću kontrolu nad time kako se Web strane prikazuju – što je i moguće. Možete zadati i imena pojedinačnih fontova i generičke fontove u istom deklarativnom bloku. Razmotrimo, na primer, naredno pravilo stila za element p:

```
p {
  font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
}
```

Ovde smo zadali da se koristi font Verdana ukoliko je instaliran na sistemu; u suprotnom, čitaču je naloženo da proveri da li je instaliran font Geneva; ukoliko nema ni tog fonta, računar će potražiti Arial, a potom font Helvetica. Ako nije dan nije dostupan, čitač će upotrebiti podrazumevani beserifni font na sistemu.

Fontovi koji su prilično pouzdani:

Windows	Arial, Lucida, Impact, Times New Roman, Courier New, Tahoma, Comic Sans, Verdana, Georgia, Garamond
Mac	Helvetica, Futura, Bodoni, Times, Palatino, Courier, Gill Sans, Geneva, Baskerville, Andale Mono

Ovaj spisak otkriva zašto smo odabrali fontove navedene u našem pravilu stila: na početku navodimo naš prvi izbor, uobičajan Windowsov font (Verdana), potom dodajemo sličan font za Mac (Geneva); iza njih slede fontovi koji bi trebalo da se koriste ukoliko nijedan od ova dva fonta nije dostupan.

Kako da uklonim linije za podvlačenje iz veza?

Široko prihvaćen podrazumevani vizuelni pokazatelj da tekst na Web strani predstavlja vezu (engl. *link*) ka drugom dokumentu jeste to što je podvučen i što se prikazuje u drugačijoj boji od ostatka teksta. Ipak, u nekim slučajevima ćete hteti da uklonite liniju ispod tekstualne veze.

Rešenje

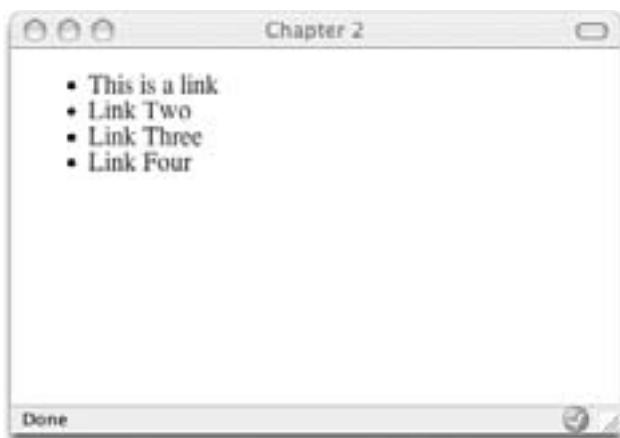
Pomoću svojstva `text-decoration` uklanjamo linije za podvlačenje iz teksta veza. Čitači podrazumevano dodeljuju vrednost `underline` svojstvu `text-decoration` elementa `a`. Ukoliko želite da uklonite te linije, dodelite svojstvu `text-decoration` veze vrednost `none`:

```
text-decoration: none;
```

Evo CSS datoteke koja stvara efekat sa slike 2.5:

textdecoration.css

```
a:link, a:visited {
    text-decoration: none; }
```



Slika 2.5. Pomoću svojstva `text-decoration` prave se veze koje nisu podvučene

Diskusija

Svojstvo `text-decoration` može imati i druge vrednosti osim `underline` i `none`:

- `overline`
- `line-through`
- `blink`

Ove vrednosti možete kombinovati. Na primer, ukoliko hoćete da i ispod i iznad određene veze bude linija – kao na slici 2.6 – primenili biste naredno pravilo stila:

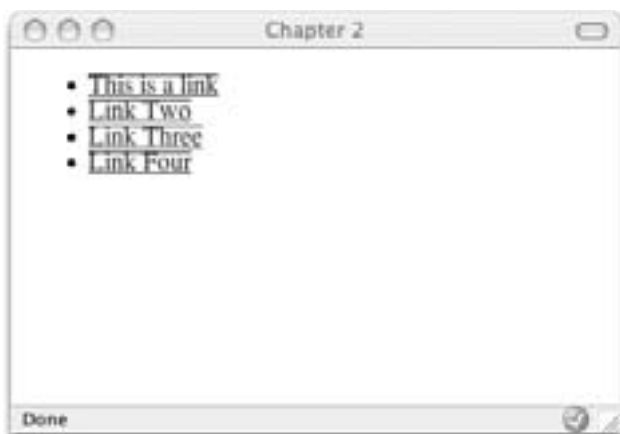
textdecoration2.css

```
a:link, a:visited {
    text-decoration: underline overline;
}
```



Izbegavajte da koristite linije koje mogu navesti na pogrešan zaključak

Pomoću svojstva `text-decoration` možete podvući tekst koji ne predstavlja vezu, ali oprezno s tim. Podvlačenje veza je toliko široko prihvaćena konvencija da su korisnici skloni da pomisle kako je *svaki* podvučen tekst veza ka drugom dokumentu.



Slika 2.6. Kombinovanje vrednosti svojstva `text-decoration` da bi i iznad i ispod veze bila linija

Kada nije dobro uklanjati linije za podvlačenje?

Podvučene veze su standardna konvencija koju poštuju svi Web čitači, pa i korisnici očekuju da veze budu podvučene. Ukoliko veze u tekstu nisu podvučene, korisnici će možda teško uvideti da su te reči zapravo veze, a ne samo istaknut tekst. Savetujem da ne uklanjate crte ispod veza u tekstu. Veze možete stilizovanjem ulepšati na druge načine, i retko kad je neophodno (a možda nikad i nije potrebno) ukloniti linije ispod njih.

Potpuno druga stvar su veze u okviru menija, ili u drugim okolnostima u kojima je očigledno da je tekst veza – na primer, kada je tekst stilizovan pomoću CSS-a tako da podseća na grafičko dugme. Ukoliko želite, uklonite linije ispod ovakvih veza, jer bi na osnovu njihovog konteksta trebalo da bude očigledno da se radi o vezama.

Kako da napravim vezu koja menja boju kada se pokazivač miša pomeri nad njom?

Promenom boje veze kada se preko nje postavi pokazivač miša postiže se atraktivan efekat. To se može efikasno uraditi s navigacionim menijima napravljenim pomoću CSS-a, ali i sa vezama unutar običnog teksta u pasusima.

Rešenje

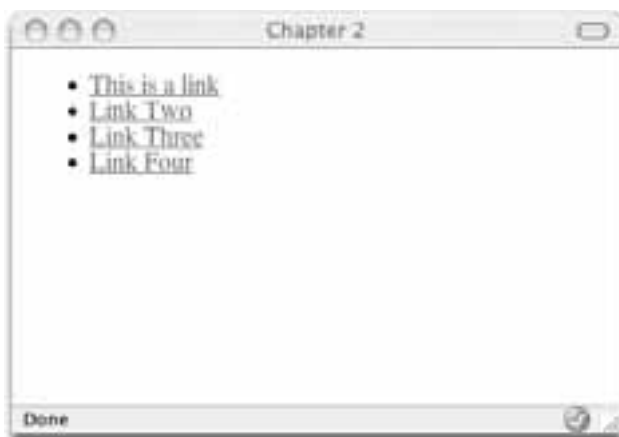
Ovaj efekat možete postići tako što ćete dinamičke pseudoklase `:hover` i `:active` elementa veze stilizovati drugačije od drugih njegovih pseudoklasa.

Razmotrimo jedan primer. Evo tipičnog pravila stila koje primenjuje iste deklaracije na sve pseudoklase elementa veze:

textdecoration3.css

```
a:link, a:visited, a:hover, a:active {
    text-decoration: underline;
    color: #6A5ACD;
    background-color: transparent;
}
```

Kada se primeni ovaj opis stila, veze će se prikazivati podvučene i biće plave (boja #6A5ACD), kao što je prikazano na slici 2.7.

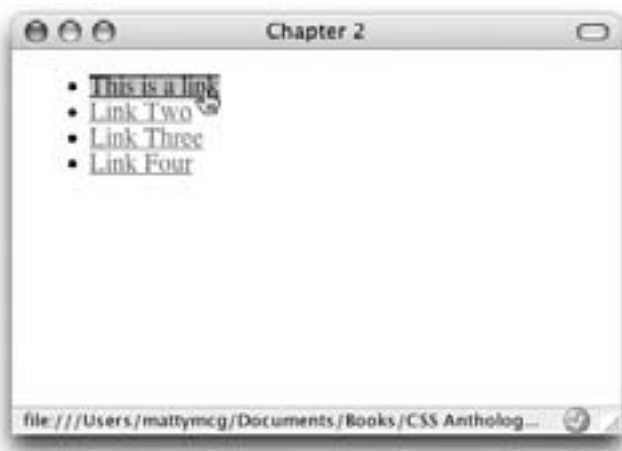


Slika 2.7. Korišćenje iste deklaracije za sve pseudoklase veza

Da bismo drugačije stilizovali pseudoklase `:hover` i `:active`, moramo ih ukloniti iz deklaracije u kojoj su bile sa ostalim pseudoklasama i dodeliti im zasebnu deklaraciju. U narednom kaskadnom opisu stila, odlučila sam da podvučenoj vezi dodam i liniju iznad. Takođe, zadala sam boju pozadine, a za tekst veze predvide-la sam tamniju boju; slika 2.8 prikazuje kako ovi stilovi izgledaju u čitaču:

textdecoration4.css

```
a:link, a:visited {
    text-decoration: underline;
    color: #6A5ACD;
    background-color: transparent;
}
a:hover, a:active {
    text-decoration: underline overline;
    color: #191970;
    background-color: #C9C3ED;
}
```



Slika 2.8. Pomeranje kursora preko veze na koju je primenjen stil `hover`

Verovatno ste već shvatili da i ostale pseudoklase elementa `anchor` možete zasebno stilizovati. Možda biste naročito hteli da primenite drugačiji stil na veze koje su korisnici posetili. Za to je dovoljno da zasebno stilizujete pseudoklasu `:visited`.

Pazite da prilikom stilizovanja pseudoklasa ne promenite veličinu ili težinu (podebljanost) teksta. Ukoliko uradite tako, izgledaće kao da strana nije poravnata, jer će sadržaj oko veze morati da se pomeri kako bi se tekst veze ispod pokazivača miša prikazao uvećan ili podebljan.



Redosled deklaracija pseudoklasa

Pseudoklase veza bi trebalo da se deklariraju sledećim redom: `link`, `visited`, `hover`, `active`. U suprotnom, možda se neće ponašati kao što hoćete. Jedan od načina da zapamtite pravilan redosled jeste mnemonički: **LoVeHAte**.

Kako da na istoj strani prikažem dva različita stila veze?

U prethodnom rešenju je objašnjeno kako se stilizuju različiti selektori elementa veze, ali šta ako želite da u istom dokumentu primenite više stilova veze? Recimo, hoćete da veze u navigacionom meniju ne budu podvučene, ali i da se veze unutar teksta na stranama lako prepoznaju. Ili, možda, deo dokumenta ima tamnu pozadinu, pa na tom mestu morate da koristite veze svetle boje.

Rešenje

Pravljenje različitih stilova za veze na istoj strani pokazaćemo na primeru u kome smo već stilizovali obične veze:

linktypes.css (odlomak)

```
a:link, a:visited {
  text-decoration: underline;
  color: #6A5ACD;
  background-color: transparent;
}

a:hover, a:active {
  text-decoration: underline overline;
  color: #191970;
  background-color: #C9C3ED;
}
```

Ovo će biti podrazumevani stilovi veze – tj. uobičajan način prikazivanja veza u vašim dokumentima. Prvo pravilo nalaže da veze budu plave, pa ako deo strane ima plavu pozadinu, veze u toj oblasti neće biti čitljive. Potrebno je da za tu oblast napravimo drugi skup stilova.

Prvo ćemo napraviti atribut `class` ili `id` za element koji će da sadrži veze različitih boja. Ukoliko je kontejner već stilizovan pomoću CSS-a, moguće je da već ima atribut `class` ili `id` koji možemo upotrebiti. Pretpostavimo da dokument sadrži sledeći HTML kôd:

linktypes.html (odlomak)

```
<div class="boxout">
  <p>Visit our <a href="store.html">online store</a>, for all your
  widget needs.</p>
</div>
```

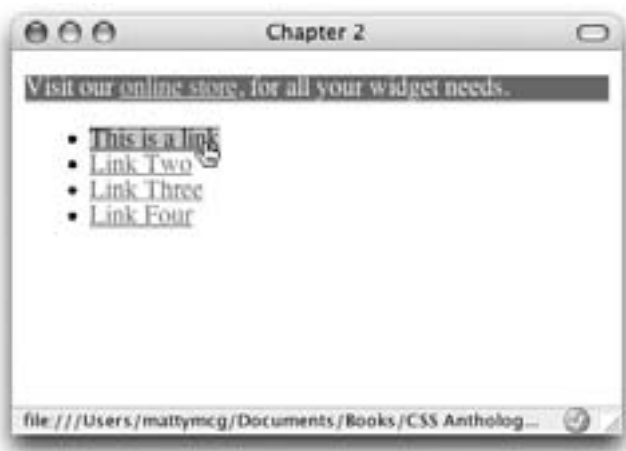
Moramo da napišemo pravilo stila koje će važiti za sve veze u elementu klase `boxout`:

linktypes.css (odlomak)

```
.boxout {
  color: #FFFFFF;
  background-color: #6A5ACD;
}
```

```
.boxout a:link, .boxout a:visited {
  text-decoration: underline;
  color: #E4E2F6;
  background-color: transparent;
}
.boxout a:hover, .boxout a:active {
  background-color: #C9C3ED;
  color: #191970;
}
```

Slika 2.9 pokazuje sledeće: ovo pravilo nalaže da se prvi stil pridružuje svim vezama u dokumentu izuzev onima u elementu `boxout` – te veze će biti svetlije boje.



Slika 2.9. Korišćenje dva različita stila za veze u dokumentu

Kako da prvu stavku u listi stilizujem drugačije od ostalih?

Dizajnerima je često potrebno da prvu stavku iz nekog skupa – poput stavki liste ili broj pasusa u kontejneru – stilizuju drugačije od ostatka skupa. Jedan od načina da se to postigne jeste da se prvoj stavki dodeli klasa koja će se stilizovati drugačije od ostalih stavki. Međutim, postoji elegantniji način da se ovaj efekat postigne u modernim čitačima: pomoću selektora pseudoklase `first-child`.

Rešenje

Evo jednostavnog skupa u obliku neuređene liste:

firstchild.html (odlomak)

```
<ul>
  <li>Brie</li>
  <li>Cheddar</li>
  <li>Red Leicester</li>
  <li>Shropshire Blue</li>
</ul>
```

Korišćenje selektora `first-child`

Da bismo izmenili boju prve stavke u listi bez posledica po njegove susede, upotrebicemo selektor `first-child`. To omogućava da obeležimo prvi element u elementu `ul`, kao što je prikazano na slici 2.10:

firstchild.css (odlomak)

```
li:first-child {
  color: red;
}
```



Slika 2.10. Prikazivanje prve tekstualne stavke crvenom bojom

Nažalost, Internet Explorer 6 ne podržava atribut `first-child`. Stoga, sve dok broj korisnika koji pomoću ovog čitača pristupaju vašoj Web lokaciji ne bude zanemarljiv, moraćete da pribegavate drugoj metodi za postizanje ovog efekta. Jedan način je pomoću selektora klase.

Korišćenje selektora klase

Da bismo postigli opisani efekat u Internet Exploreru 6, dodajemo atribut `class` ili `id` elementu koji želimo drugačije da stilizujemo. U narednom primeru upotrebićemo atribut `class`:

firstchildwithclass.html (odlomak)

```
<ul>
  <li class="unusual">Brie</li>
  <li>Cheddar</li>
  <li>Red Leicester</li>
  <li>Shropshire Blue</li>
</ul>
```

Posle toga pišemo pravilo stila kojim ćemo realizovati željeni efekat:

firstchildwithclass.css (odlomak)

```
li.unusual {
  color: red;
}
```

Kako da obojim pozadinu naslova?

CSS omogućava da dodamo boju pozadini proizvoljnog elementa, uključujući i naslov.

Rešenje

Sledeće CSS pravilo odnosi se na sve naslove prvog nivoa u dokumentu. Rezultat je prikazan na slici 2.11.

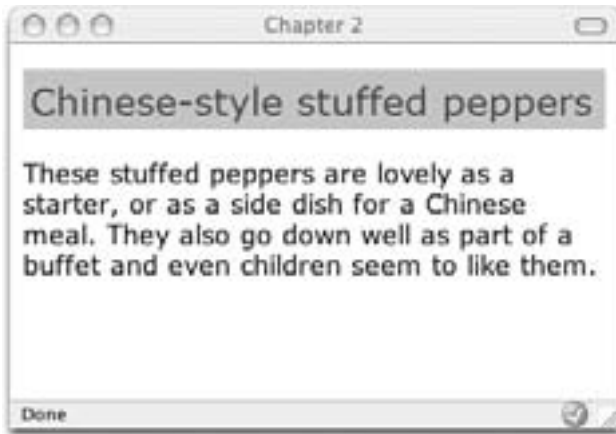
headingcolor.css (odlomak)

```
h1 {
  background-color: #ADD8E6;
  color: #256579;
  font: 1.6em Verdana, Geneva, Arial, Helvetica, sans-serif;
  padding: 0.2em;
}
```



Napravite mesta za boju!

Kad dodajete pozadinu naslovu, možda biste mogli da povećate razmak između teksta naslova i ivice obojene pozadine kao što sam ja uradila u ovom primeru.



Slika 2.11. Prikazivanje naslova sa obojenom pozadinom

Kako da podvučem naslove?

Rešenje

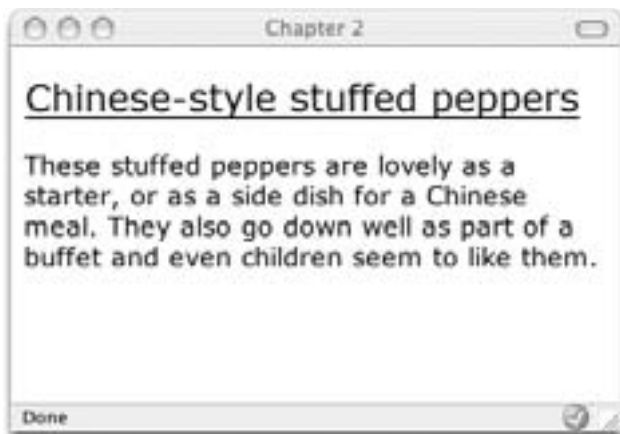
Tekst možete podvući na dva načina. Najjednostavnije je upotrebiti svojstvo `text-decoration` s kojim smo se već susreli u ovom poglavlju, u odeljku „Kako da uklonim linije za podvlačenje iz veza“. Ova metoda omogućava da tekst podvučete linijom iste boje kakav je i font, kao što pokazuje ovaj kôd i slika 2.12:

headingunderline.css (odlomak)

```
h1 {
  font: 1.6em Verdana, Geneva, Arial, Helvetica, sans-serif;
  text-decoration: underline;
}
```

Efekat podvlačenja možete postići i dodavanjem donje ivice naslovu. Ovakvo rešenje, čiji je rezultat prikazan na slici 2.13, fleksibilnije je po tome što pomoću slobodnog prostora možete razdvojiti liniju za podvlačenje od naslova i što linije možete obojiti drugačije od teksta.

Manje je verovatno da će neko pomisliti da je naslov na koji se primenjuje ovaj efekat zapravo veza koja je podvučena pomoću svojstva `text-decoration`. Međutim, ovaj efekat se može unekoliko razlikovati od jednog čitača do drugog, te

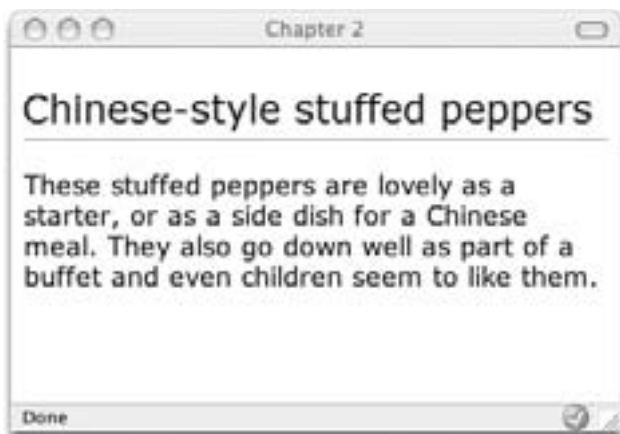


Slika 2.12. Dodavanje linije za podvlačenje naslovu pomoću svojstva `text-decoration`

ćete morati da prvo proverite izgleda li prihvatljivo na čitačima koji bi vaši posetioци mogli da koriste. Evo potrebnog pravila stila:

`headingunderline2.css`

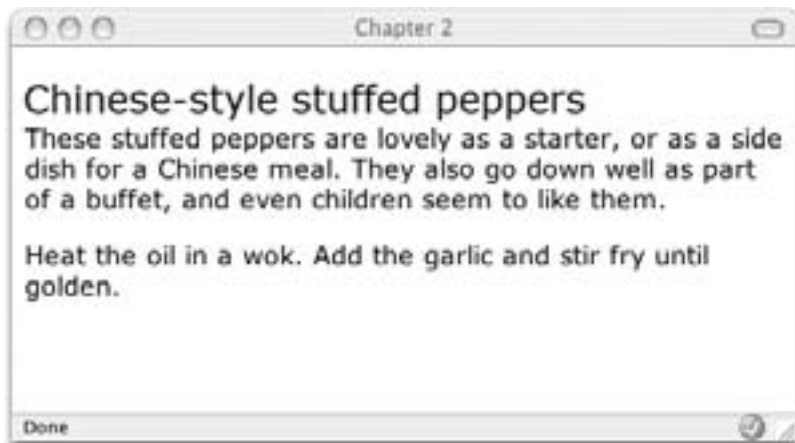
```
h1 {
  font: 1.6em Verdana, Geneva, Arial, Helvetica, sans-serif;
  padding: 0.2em;
  border-bottom: 1px solid #aaaaaa;
}
```



Slika 2.13. Efekat podvlačenja dobijen pomoću donje ivice

Kako da uklonim veliki razmak između elementa h1 i sledećeg pasusa?

Čitači podrazumevano prikazuju belinu između svih naslova i pasusa. Tu belinu čine podrazumevane gornje i donje margine koje čitači primenjuju na ove elemente. Margina uz naslov na slici 2.14 predstavlja podrazumevanu vrednost i može se ukloniti pomoću CSS-a.



Slika 2.14. Podrazumevano rastojanje između naslova i pasusa

Rešenje

Da biste uklonili razmak između naslova i pasusa koji sledi, prvo morate da sklonite donju marginu naslova i gornju marginu pasusa. U modernim čitačima – uključujući Internet Explorer 7 – to možemo uraditi pomoću CSS-a, uz korišćenje *selektora susednog elementa* (engl. *adjacent selector*). Međutim, za isti efekat u starijim čitačima, moramo da primenimo tehnike koje su bolje podržane.

Korišćenje selektora susednog elementa

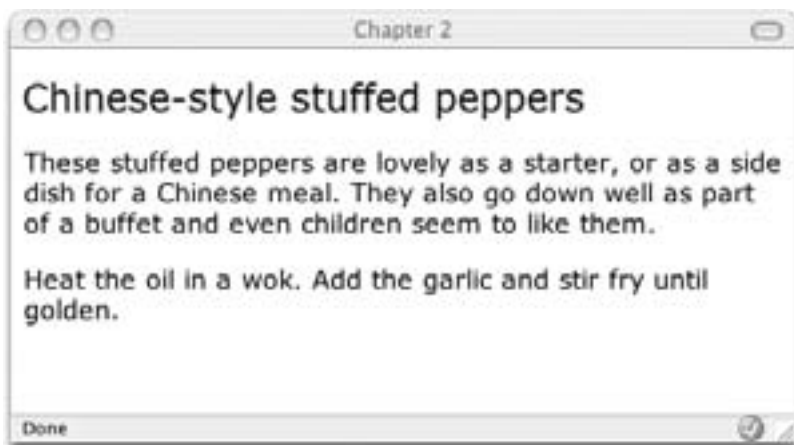
Pomoću selektora susednog elementa možete da obeležite element koji sledi iza drugog elementa, ukoliko imaju zajedničkog roditelja. Zapravo, pomoću selektora susednih elemenata možete zadati element koji je postavljen iza nekoliko drugih elemenata, ne samo iza jednog – element na koji se primenjuje stil uvek je *poslednji element u lancu*. Ukoliko vas ovo zbunjuje, ne brinite: biće vam mnogo jasnije na konkretnom primeru.

Narednim pravilima stila uklanja se gornja margina iz svakog pasusa koji sledi neposredno iza naslova prvog nivoa. Obratite pažnju na to da gornja margina nije uklonjena iz elementa h1 – samo iz pasusa iza njega:

headingnospace.css (odlomak)

```
h1 {  
    font: 1.6em Verdana, Geneva, Arial, Helvetica, sans-serif;  
    margin-bottom: 0;  
}  
h1+p {  
    margin-top: 0;  
}
```

Slika 2.15 prikazuje kako stranica izgleda pošto se primeni ovo pravilo.



Slika 2.15. Izmenjen izgled naslova pomoću selektora susednog elementa

Vidi se da prvi pasus iza elementa `h1` više nema gornju marginu; međutim, svi naredni pasusi su zadržali gornje margine.

Već sam pomenula da tehnika sa selektorima susednih elemenata funkcioniše samo u novijim čitačima – na primer, Internet Explorer 7 je jedina verzija tog programa koja podržava selektore susednih elemenata. U pojedinim slučajevima, možda ćete zaključiti da je za korisnike starijih čitača prihvatljivo da vide razmak između naslova i teksta. Ali, ukoliko želite da uklonite margine sa strane koja će se pregledati i starijim čitačima, imate nekoliko mogućnosti.

Selektore klase možete upotrebiti kao u odeljku „Kako da na istoj strani prikazem dva različita stila veze?“ da određenoj klasi dodelite marginu 0. Ukoliko ste pogledali to rešenje, lako ćete primeniti ovaj pristup. Druga mogućnost je da za naslov zadate negativnu marginu, što je objašnjeno u nastavku.

Primenjivanje negativne margine

U CSS-u, margine mogu imati pozitivnu ili negativnu vrednost. Međutim, razmak (engl. *padding*) može imati samo pozitivnu vrednost.

Prostor između naslova i prvog pasusa možete ukloniti i tako što ćete naslovu pridružiti negativnu marginu. Narednim pravilom stila postiže se sličan efekat kao na slici 2.15:

```
h1 {
  font: 1.6em Verdana, Geneva, Arial, Helvetica, sans-serif;
  margin-bottom: -0.6em;
}
```

Kako da istaknem tekst na strani?

Pre pojave CSS-a, pomoću oznaka `` mogle su se istaći važne stavke na strani ili označiti fraze koje su korisnici upotreбили za nalaženje dokumenta u pretraživaču (engl. *search engine*). CSS je u velikoj meri olakšao ovaj proces.

Rešenje

CSS omogućava da napravite klasu za stil koji služi za isticanje teksta i da taj stil primenite na oznake `` između kojih ćete staviti istaknut tekst. Na primer, u narednom pasusu smo jednu frazu stavili između oznaka `` koje primenjuju klasu `hilite`:

hilite.html (odlomak)

```
<p>These <span class="hilite">stuffed peppers</span> are lovely
as a starter, or as a side dish for a Chinese meal. They also
go down well as part of a buffet, and even children seem to
like them.</p>
```

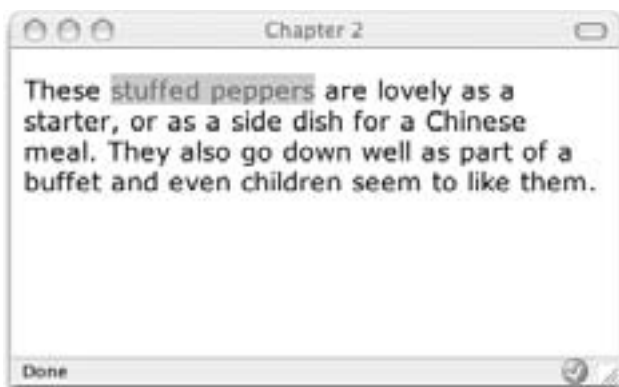
Klasa `hilite` je prikazana u sledećem listingu; istaknuti deo će se prikazati kao na slici 2.16:

hilite.css (odlomak)

```
.hilite {
  background-color: #FFFFCC; color: #B22222;
}
```

Kako da izmenim visinu reda (prored) u tekstu?

Jedna od velikih prednosti CSS-a nad oznakama `` jeste to što omogućava mnogo bolju kontrolu izgleda teksta na strani. U narednom rešenju izmieniće-mo prored (engl. *leading*) teksta u dokumentu.



Slika 2.16. Tekst istaknut pomoću CSS-a

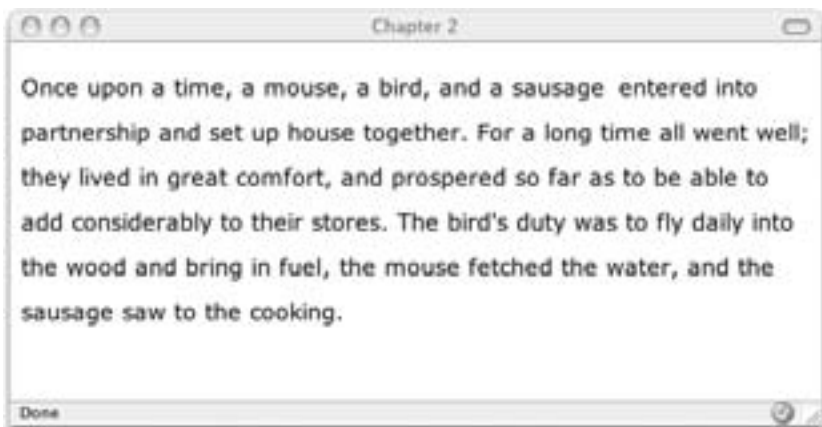
Rešenje

Ukoliko je podrazumevani prored u tekstu na strani premali, izmenite ga pomoću svojstva `line-height`:

leading.css

```
p {
  font: 0.9em Verdana, Geneva, Arial, Helvetica, sans-serif;
  line-height: 2.0;
}
```

Rezultat je prikazan na slici 2.17.



Slika 2.17. Izmena proreda pomoću svojstva `line-height`

Prosto kao pasulj! Nemojte previše da proredite tekst da ne postane teško čitljiv.



Bez jedinica?

Primitili ste da u ovom primeru nismo naveli merne jedinice – vrednost 2.0 je odnos (engl. *ratio*). *Možete* zadati vrednost svojstva `line-height` u standardnim CSS-ovim mernim jedinicama poput `em` i piksela, ali ćete tako narušiti vezu između visine reda i veličine fonta za elemente `decu`.

Na primer, da je u prethodnom primeru postojao element `span` kojim je zadata velika vrednost za `font-size`, visina reda bi se povećala srazmerno tome, tako da ostane isti odnos, jer je svojstvu `line-height` pasusa dodeljena numerička vrednost 2.0. Međutim, da je svojstvu `line-height` dodeljena vrednost `2em` ili `200%`, element `span` bi nasledio konkretnu visinu reda, a ne odnos, i to što je font veliki ne bi uticalo na prored. Možda ste upravo to i hteli.

Kako da obostrano poravnam tekst?

Prilikom obostranog poravnavanja teksta, menjate razmak između reči tako da i leva i desna margina budu ravne. Ovaj efekat možete lako postići pomoću CSS-a.

Rešenje

Pasus ćete poravnati pomoću svojstva `text-align` na sledeći način:

```
justify.css
```

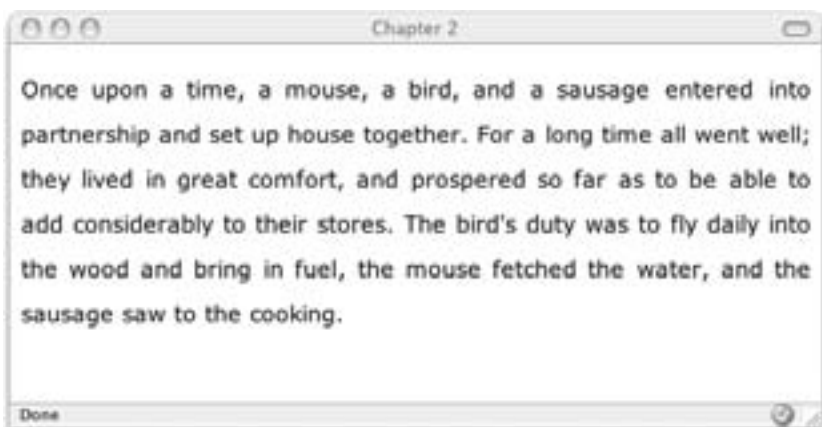
```
p {
  text-align: justify;
  font: 1em Verdana, Geneva, Arial, Helvetica, sans-serif;
  line-height: 2.0;
}
```

Slika 2.18 pokazuje rezultat dodeljivanja vrednosti `justify` svojstvu `text-align`.

Diskusija

Ostale vrednosti za svojstvo `text-align` su:

- right** poravnava tekst udesno u elementu u kome se nalazi
- left** poravnava tekst ulevo u elementu u kome se nalazi
- center** centrirava tekst u elementu u kome se nalazi



Slika 2.18. Poravnavanje teksta pomoću svojstva `text-align`

Kako da stilizujem horizontalnu liniju?

U načelu, trebalo bi da izbegavate da strani dodajete HTML elemente koji su čisto prezentacioni, poput horizontalne linije (`hr`). Dokument koji je strukturiran semantički jednostavnije se održava, brže se učitava i optimizovan je za indeksiranje u pretraživačima. Efekat sličan onom koji se dobija pomoću elementa `hr`, obično se može postići dodavanjem ivica (engl. *borders*) postojećim elementima.

Međutim, ponekad je primena elementa `<hr>` najbolji način da se postigne željeni efekat ili je taj element neophodan kako bi se nestilizovan sadržaj prikazao u starijim čitačima koji ne podržavaju CSS.

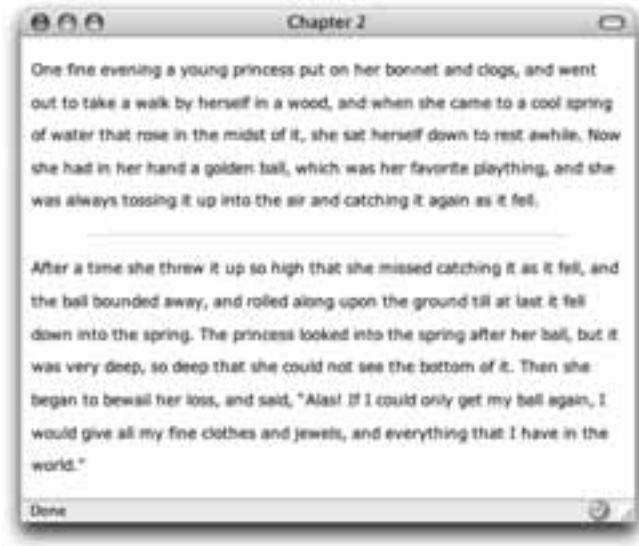
Rešenje

Pomoću CSS-a možete izmeniti boju, visinu i debljinu horizontalne linije. Međutim, obratite pažnju na neke neusklađenosti između čitača. Na primer, u donjem primeru su upotrebljene iste vrednosti za svojstva `color` i `background-color` jer čitači zasnovani na Mozilli boje liniju pomoću svojstva `background-color`, dok Internet Explorer za to koristi svojstvo `color`:

hrstyle.css (odlomak)

```
hr {
  border: none;
  background-color: #ADD8E6;
  color: #ADD8E6;
  height: 1px;
  width: 80%;
}
```

Rezultat primene ovog pravila vidi se na slici 2.19.



Slika 2.19. Promena boje, visine i debljine horizontalne linije

Kako da uvučem tekst?

Rešenje

Tekst ćemo uvući tako što ćemo na element u kome se tekst nalazi (kontejner) primeniti pravilo kojim se zadaje vrednost svojstva `padding-left`, na sledeći način:

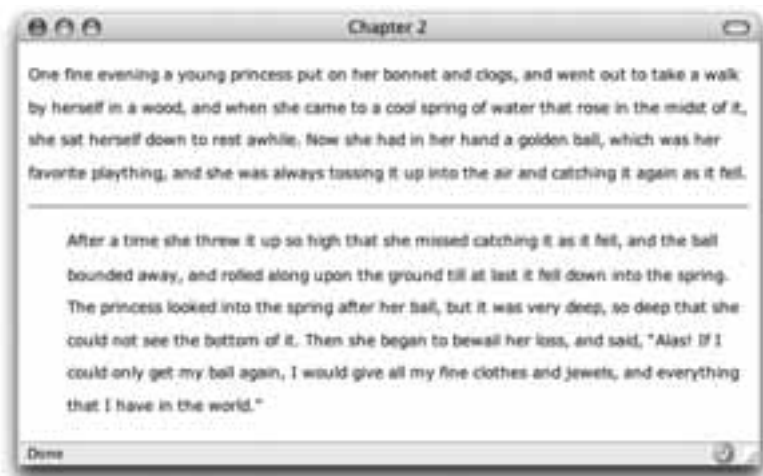
indent.html (odlomak)

```
<p class="indent">After a time she threw it up so high that she missed catching it as it fell, and the ball bounded away, and rolled along upon the ground till at last it fell down into the spring. The princess looked into the spring after her ball, but it was very deep, so deep that she could not see the bottom of it. Then she began to bewail her loss, and said, "Alas! If I could only get my ball again, I would give all my fine clothes and jewels, and everything that I have in the world."</p>
```

indent.css (odlomak)

```
.indent {
  padding-left: 30px;
}
```

Uvučeni pasus je prikazan na slici 2.20.



Slika 2.20. Uvlačenje teksta pomoću CSS-a

Diskusija

Tekst ne bi trebalo da uvlačite pomoću HTML oznake `<blockquote>` sem ukoliko se radi o citatu. U vizuelnim okruženjima za izradu Web strana – kao što je Adobe Dreamweaver – za uvlačenje teksta koristi se `blockquote`, ali ipak nemojte tako da radite; umesto toga, napišite CSS pravilo za uvlačenje odgovarajućih blokova kao što je prikazano gore.

Oznaka `<blockquote>` namenjena je za označavanje citata, i uređaji poput čitača ekrana za slabovide predstavice ovakav tekst na način koji će pomoći takvim korisnicima da shvate da je ono što čuju citat. Ukoliko oznaku `<blockquote>` upotrebite za uvlačenje običnih pasusa, zbunićete korisnike koji će taj sadržaj čuti kao da se radi o citatu.



Samo jedan red

Srodna tehnika omogućava da se uvuče samo prvi red svakog pasusa. Dovoljno je da zadate vrednost CSS svojstvu `text-indent` pasusa ili klase za pasuse koje hoćete da prikazete na ovaj način:

indent2.css

```
p {
  text-indent: 20px;
}
```

Kako da centriram tekst?

Rešenje

Tekst ili bilo koji element centriraćete tako što ćete svojstvu `text-align` dodeliti vrednost `center`:

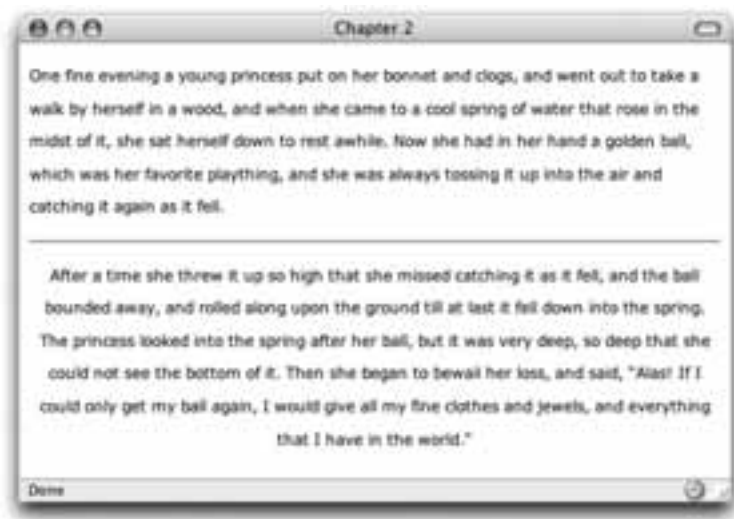
center.html (odlomak)

```
<p class="centered">After a time she threw it up so high that she missed catching it as it fell, and the ball bounded away, and rolled along upon the ground till at last it fell down into the spring. The princess looked into the spring after her ball, but it was very deep, so deep that she could not see the bottom of it. Then she began to bewail her loss, and said, "Alas! If I could only get my ball again, I would give all my fine clothes and jewels, and everything that I have in the world."</p>
```

center.css (odlomak)

```
.centered {  
    text-align: center;  
}
```

Na slici 2.21 vidi se rezultat primene ovog pravila.



Slika 2.21. Centriranje teksta pomoću svojstva `text-align`

Kako da pomoću CSS-a postignem da sva slova u tekstu budu velika?

Rešenje

Pomoću svojstva `text-transform` možete zadati da sva slova u tekstu budu velika i izvesti druge transformacije:

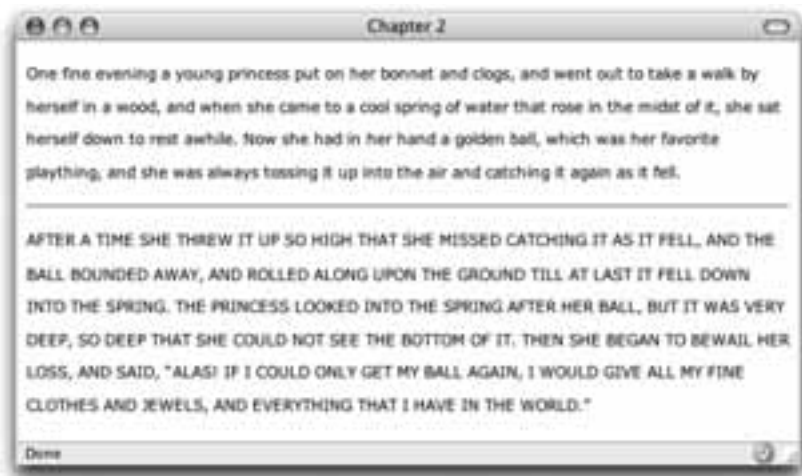
uppercase.html (odlomak)

```
<p class="transform">After a time she threw it up so high that she missed catching it as it fell, and the ball bounded away, and rolled along upon the ground till at last it fell down into the spring. The princess looked into the spring after her ball, but it was very deep, so deep that she could not see the bottom of it. Then she began to bewail her loss, and said, "Alas! If I could only get my ball again, I would give all my fine clothes and jewels, and everything that I have in the world."</p>
```

uppercase.css (odlomak)

```
transform {  
    text-transform: uppercase;  
}
```

Rezultat je prikazan na slici 2.22.



Slika 2.22. Prikazivanje teksta velikim slovima pomoću svojstva `text-transform`

Diskusija

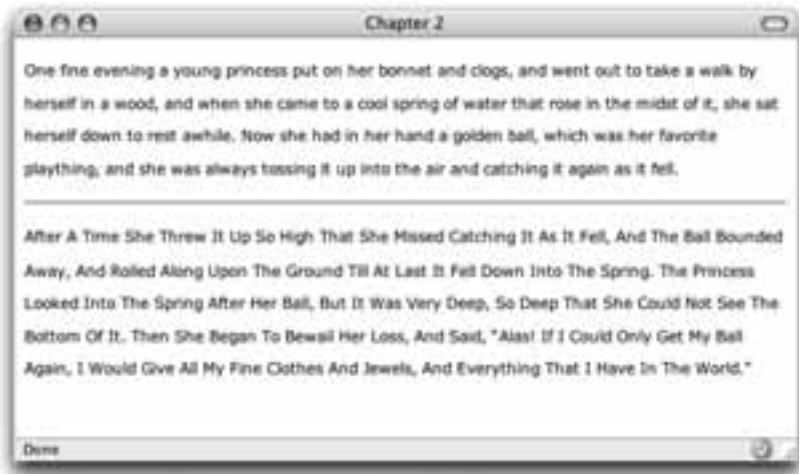
I ostale moguće vrednosti svojstva `text-transform` mogu biti korisne. Vrednost `capitalize` znači da će prvo slovo svake reči biti veliko, kao što je prikazano na slici 2.23:

capitalize.css (odlomak)

```
transform {
  text-transform: capitalize;
}
```

Ostale moguće vrednosti svojstva `text-transform` su:

- lowercase
- none (podrazumevana vrednost)



Slika 2.23. Svojstvo `text-transform` za pretvaranje prvog slova svake reči u veliko

Kako da izmenim ili uklonim znakove za nabranjanje?

Rešenje

Stil znaka za nabranjanje (engl. *bullet*) u neuređenoj listi (engl. *unordered list*) izmehićete ako promenite vrednost svojstva `list-style-type`. Evo HTML koda liste:

listtype.html (odlomak)

```
<ul>
  <li>list item one</li>
  <li>list item two</li>
  <li>list item three</li>
</ul>
```

Ukoliko želite četvrtaste znakove za nabranjanje poput onih sa slike 2.24, svojstvu `list-style-type` morate dodeliti vrednost `square`:

listtype.css

```
ul {  
  list-style-type: square;  
}
```



Slika 2.24. Četvrtasti znakovi za nabranjanje stavki liste

Diskusija

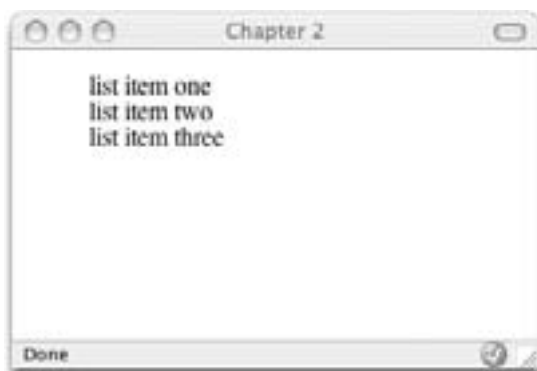
Svojstvo `list-style-type` može da ima i sledeće vrednosti:

- `disc`
- `circle`
- `decimal-leading-zero`
- `decimal`
- `lower-roman`
- `upper-roman`
- `lower-greek`
- `lower-alpha`
- `lower-latin`
- `upper-alpha`
- `upper-latin`
- `Hebrew`
- `Armenian`
- `Georgian`
- `none`

Ne podržavaju svi čitači sve ove moguće vrednosti; čitači koji ne podržavaju određeni tip znaka za nabranjanje, prikazaće umesto njega podrazumevani znak. CSS Test Suite za stilove liste* omogućava da pregledate različite vrste znakova za nabranjanje i da proverite podržava li ih vaš čitač. Ukoliko svojstvu `list-style-type` zadate vrednost `none`, znakovi za nabranjanje se neće prikazivati na strani mada će lista biti uvučena kao da su tu (slika 2.25):

listtype2.css

```
ul {  
  list-style-type: none;  
}
```



Slika 2.25. Prikazivanje liste bez znakova za nabranjanje

Kako da kao znak za nabranjanje stavki liste upotrebim sliku?

Rešenje

Ako hoćete da kao znak za nabranjanje upotrebite sliku, napravite odgovarajuću sliku, pa stilizujte znakove za nabranjanje pomoću svojstva `list-style-image` umesto svojstva `list-style-type`. Svojstvu `list-style-image` možete dodeliti URL adresu koja sadrži putanju do izabrane slike:

listimage.css

```
ul {  
  list-style-image: url(bullet.gif);  
}
```

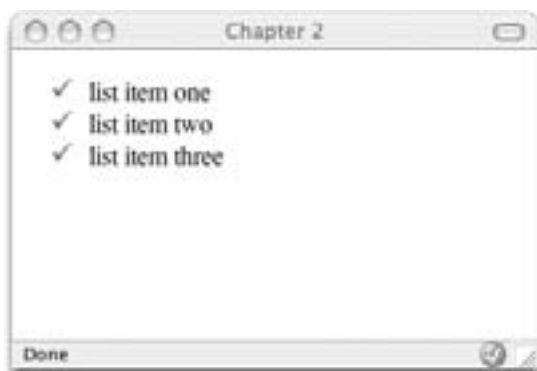
* <http://meyerweb.com/eric/css/tests/css2/sec12-06-02a.htm>

Slika 2.26 prikazuje kako se ovaj efekat može iskoristiti za ulepšavanje liste.



Stilizovanje znakova za nabranje pojedinačnih stavki liste

Svojstvo `list-style-image` primenjuje se na pojedinačne elemente liste (``). Međutim, ukoliko svojstvo `list-style-image` primenite na celu listu (na element `` ili ``), naslediće ga svaka stavka liste. Ipak, imate mogućnost da za svaku stavku liste pojedinačno zadate ovo svojstvo (tako što ćete stavkama pridružiti selektor `<class>` ili `<id>`), pa će svaka stavka imati drugačiju sliku kao znak za nabranje.



Slika 2.26. Slika kao znak za nabranje u listi

Kako da sprečim uvlačenje liste po levoj margini?

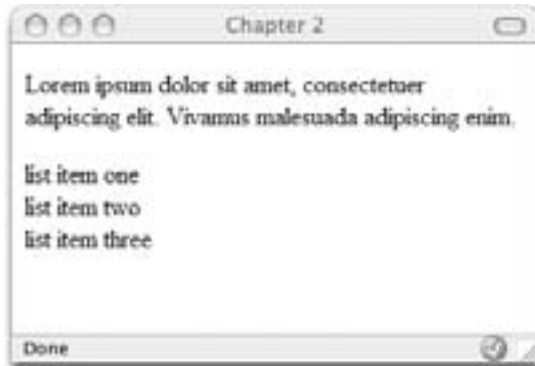
Ukoliko ste svojstvu `list-style-type` dodelili vrednost `none`, možda ćete hteti i da uklonite ili smanjite podrazumevanu levu marginu koju čitač dodeljuje listi.

Rešenje

Ako želite da sasvim poništite uvlačenje i da lista bude poravnata ulevo sa, na primer, prethodnim pasusom, kao na slici 2.26, primenite pravilo stila slično narednom:

listnomargin.css

```
ul {  
  list-style-type: none;  
  padding-left: 0;  
  margin-left: 0;  
}
```



Slika 2.27. Lista bez uvlake i znakova za nabranjanje

Diskusija

Stavke liste možete i uvući na potrebno rastojanje. Sadržaj ćete uvući za nekoliko piksela na sledeći način:

listsmallmargin.css

```
ul {
  list-style-type: none;
  padding-left: 5px;
  margin-left: 0;
}
```

Kako da listu prikažem horizontalno?

Stavke liste se podrazumevano prikazuju kao blokovi, pa će se svaka nova stavka prikazati u novom redu. Međutim, u pojedinim situacijama, deo sadržaja na strani biće lista u strukturnom smislu, ali možda nećete želeći da se prikazuje kao lista – dobar primer za to je skup navigacionih veza (engl. *navigation links*). Kako da takve stavke liste prikažete horizontalno?

Rešenje

Lista će se prikazivati horizontalno ukoliko svojstvu `display` elementa `li` dodelite vrednost `inline` na sledeći način:

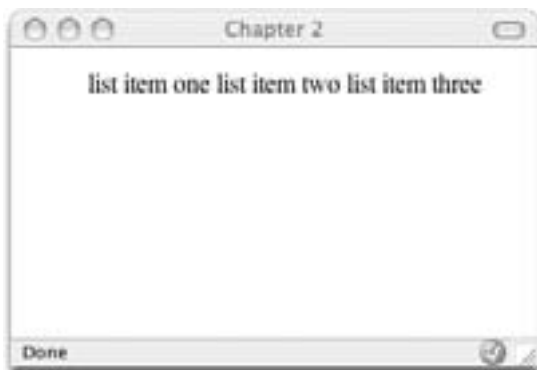
listinline.html (odlomak)

```
<ul class="horiz">
  <li>list item one</li>
  <li>list item two</li>
  <li>list item three</li>
</ul>
```

listinline.css

```
ul.horiz li {  
    display: inline;  
}
```

Rezultat primene ovog stila prikazan je na slici 2.28.



Slika 2.28. Prikazivanje liste horizontalno

Kako da dodam komentare CSS datoteci?

CSS datotekama možete da dodate komentare, što se i preporučuje. Komentari možda nisu potrebni jednostavnim CSS datotekama koje sadrže tek nekoliko pravila za stilizovanje teksta. Međutim, kada na Web lokaciju počnete da primenjujete veliki broj pravila i opisa stilova, komentari će biti veoma korisni! Bez njih možete protračiti mnogo vremena loveći odgovarajuće klase, utvrđujući šta tačno radi svaka klasa i pokušavajući da otkrijete u kom se opisu stila nalazi.

Rešenje

CSS, baš kao i JavaScript, podržava višeredne komentare (engl. `multiline comments`) u stilu jezika C. Dakle, ako određeni deo koda hoćete da pretvorite u komentar, stavite ga između narednih sekvenci znakova:

```
/*  
    :  
*/
```

Preporučujemo da dodate komentar bar na vrh svakog opisa stila kako biste objasnili šta sadrži. Primer je sledeći komentar:

```
/* Ovo je podrazumevani stil za sav tekst na ovoj Web lokaciji */
```

Kako da uklonim margine sa strane ne dodajući attribute oznaci <body>?

Pre nego što je CSS bio široko prihvaćen, Web dizajneri su podrazumevani razmak između dokumenta i prozora čitača uklanjali tako što su dodavali atribut oznaci <body> na sledeći način:

```
<body topmargin="0" leftmargin="0" marginheight="0"
marginwidth="0">
```

Rešenje

Navedeni atributi oznake <body> sada su zastareli. Trebalo bi ih zameniti narednim pravilima stila definisanim za element* body:

```
body {
margin: 0;
padding: 0;
}
```



Temeljno uklanjanje margina

Pojedini čitači (poput starijih verzija Opere) primenjuju svojstva `margin` i `padding` na element `html` umesto na element `body`. Dakle, trebalo bi da za svaki slučaj uvrstite i element `html` u pravilo stila:

zeropagegutter.css (odlomak)

```
html, body {
margin: 0;
padding: 0;
}
```

Kako da uklonim podrazumevane margine i razmake čitača za sve elemente?

Prikaz nestilizovanog elementa u čitaču rezultat je internog opisa stila čitača. Često su razlike u prikazu nestilizovane strane u različitim čitačima posledica toga što čitači nemaju iste interne opise stilova.

Rešenje

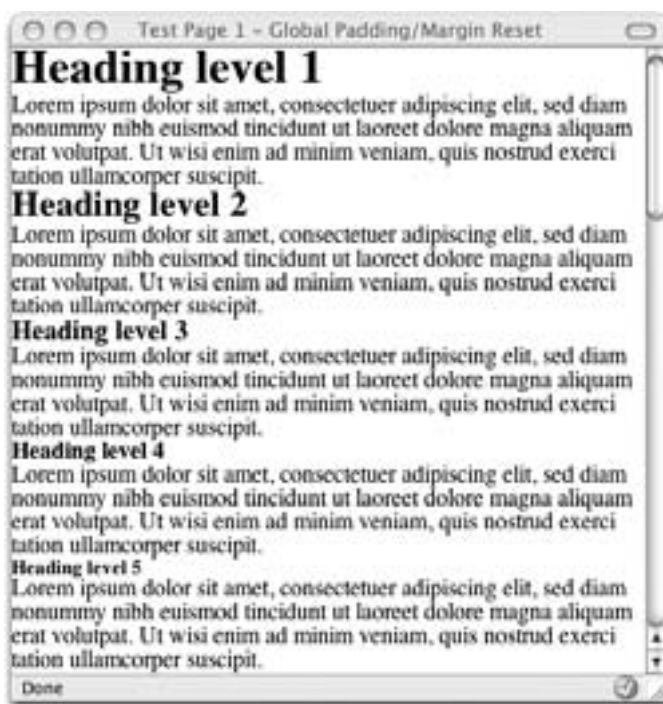
Jedan način da rešite ovaj problem jeste da pre nego što napravite svoje stilove uklonite podrazumevane margine i razmake iz svih elementa.

* <http://archivist.incutio.com/viewlist/css-discuss/46074/>

Narednim pravilom se razmaci i margine *svih* elemenata svode na nulu. Kao posledica toga, svi elementi na strani – pasusi, naslovi, liste i ostalo – prikazaće se nerazmaknuti od suseda, kao na slici 2.29:

zeropagemargin.css (odlomak)

```
* {
  margin: 0;
  padding: 0
}
```



Slika 2.29. Uklanjanje podrazumevanih margina i razmaka iz svih elemenata na strani

Diskusija

U ovom pravilu koristi se univerzalni selektor – * – da bi se uklonile margine i razmaci iz *svega*, što je tehnika poznata kao *globalno resetovanje belina* (engl. *global whitespace reset*).^{*} Ukoliko radite na složenom dizajnu, ovo je možda najbolji način da počnete projekat.

* <http://leftjustified.net/journal/2004/10/19/global-ws-reset/>

Međutim, posle toga ćete morati da dodate margine i razmake svakom elementu koji budete koristili. To je posebno važno za neke elemente obrazaca (engl. *forms*) koji se zbog primene ovog pravila stila mogu prikazivati tako da budu sasvim beskorisni!

Ipak, ako je dizajn jednostavniji, uklanjanje razmaka iz svakog elementa obično je nepotrebno i samo će napraviti više posla, jer ćete morati da dodate razmake i margine svakom elementu (pasusima, citatima, listama itd.). Dobra alternativa za jednostavno dizajnirane strane jeste da uklonite margine i razmake samo iz skupa odabranih elemenata. Naredno pravilo stila pokazuje kako to funkcioniše; njime se uklanjaju razmaci iz naslova i elemenata liste:

```
h1, h2, h3, h4, h5, h6, ul, ol {  
  margin: 0;  
  padding: 0;  
}
```

Sažetak

U ovom poglavlju smo razjasnili neka pitanja koja često postavljaju programeri bez mnogo iskustva sa CSS-om – pitanja koja se odnose na stilizovanje teksta na strani i upravljanje njime. Kombinovanjem opisanih tehnika možete napraviti atraktivne efekte čiji će se prikaz na odgovarajući način izmeniti u čitačima koji ne podržavaju CSS.

