

Uvod u CSS

C220



priručnik za polaznike 2014 Srce

TEČAJEVI **srca**



srce

Sveučilište u Zagrebu
Sveučilišni računski centar

Ovu inačicu priručnika izradio je autorski tim Srca u sastavu:

Autor: Edin Mujadžević

Recenzentica: Jasmina Plavac

Urednica: Gorana Kurtović

Lektorica: Jasna Novak Milić

TEČAJEVI Srca

Sveučilište u Zagrebu

Sveučilišni računski centar

Josipa Marohnića 5, 10000 Zagreb

edu@srce.hr

ISBN 978-953-7138-42-4 (meki uvez)

ISBN 978-953-7138-43-1 (PDF)

Verzija priručnika C220-20150219



Ovo djelo dano je na korištenje pod licencom Creative Commons Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna. Licenca je dostupna na stranici: <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Sadržaj

Uvod.....	1
1. Uvod i sintaksa CSS-a.....	3
1.1. O CSS-u	3
1.2. Uključivanje CSS-a u HTML-kôd.....	4
1.3. Sintaksa CSS-pravila	5
1.4. CSS-selektori	6
1.5. Prioritet CSS-pravila.....	13
1.6. Pseudoklase	14
1.7. Nasljeđivanje vrijednosti.....	18
1.8. Komentari	19
2. Oblikovanje teksta	21
2.1. Boja	21
2.2. Vrsta fonta	24
2.3. Veličina	24
2.4. Težina i stil	27
2.5. Poravnanje i razmak.....	28
2.6. Velika i mala slova.....	30
2.7. Ukrašavanje teksta.....	31
2.8. Skraćeni način navođenja svojstava fonta	33
2.9. Korištenje web-fontova.....	34
2.10. Vježba: Oblikovanje teksta	36
3. Oblikovanje elemenata.....	41
3.1. Boja pozadine.....	41
3.2. Prikaz elemenata.....	41
3.3. Širina i visina	43
3.4. Padding	46
3.5. Margina	48
3.6. Rubovi	50
3.7. Izračun dimenzija elementa	57
3.8. Resetiranje predefiniраниh vrijednosti.....	58
3.9. Pozadinska slika.....	59
3.10. Pozadinski prijelaz.....	65
3.11. Sjena	67
3.12. Vježba: Oblikovanje elemenata.....	70
4. Oblikovanje specifičnih elemenata.....	75
4.1. Linkovi	75
4.2. Polja za unos.....	76
4.3. Liste	82
4.4. Tablice.....	86
4.5. Slike.....	90

5. Pozicioniranje elemenata	95
5.1. Pozicioniranje pomoću elemenata inline-block.....	95
5.2. Pozicioniranje pomoću svojstva float.....	98
5.3. Model pozicioniranja	99
5.4. Pozicioniranje pomoću svojstva position	100
5.5. Vježba: Pozicioniranje i oblikovanje specifičnih elemenata.....	106

Uvod

U zadnjih su dvadesetak godina *web*-stranice postale sveprisutne i jedan su od glavnih medija komunikacije, prvo posredstvom osobnih računala, a zatim i pametnih telefona. Za razliku od standardnih medija, *web* nudi mogućnost objave sadržaja svakome, uz zanemariv ili nikakav financijski izdatak.

Namjena ovog tečaja je upoznavanje polaznika s osnovama CSS-a – tehnologije za oblikovanje *web*-stranica. Preduvjet pohađanju i razumijevanju ovog tečaja je poznavanje HTML-a – jezika za izradu sadržaja *web*-stranica.

Iako priručnik nastoji biti iscrpan, određeni broj rjeđe korištenih ili vrlo naprednih svojstva i mogućnosti CSS-a nije mogao biti uvršten u tečaj.

Također, CSS je tehnologija koja se neprestano razvija. U priručniku nisu navedena svojstva CSS-a i vrijednosti koje u trenutku pisanja nisu bila dio prihvaćenog standarda CSS-a.

Priručnik se sastoji od uvoda, pet poglavlja i tri praktične vježbe za koje su potrebni dodatni materijali.

U priručniku se rabe ovi simboli:

- kurzivom su istaknuti nazivi HTML-elemenata te CSS-svojstva, ključnih riječi i funkcija
- podebljano su označeni važni pojmovi te naziv CSS-svojstva kad se navodi prvi put
- u okvirima sa strane navedene su zanimljivosti i napomene.

Na kraju svakog poglavlja nalazi se pregled obrađenih cjelina.

1. Uvod i sintaksa CSS-a

CSS je jezik koji služi za oblikovanje *web*-stranica. Uz HTML, jezik pomoću kojeg se definiraju struktura i sadržaj *web*-stranica, CSS je osnovna tehnologija na kojoj se temelji današnji *web*.

1.1. O CSS-u

CSS je kratica za *Cascading Style Sheets*.

Pojam *style sheet* često se upotrebljava za datoteku koja sadrži CSS-kôd. Dakle, *style sheet* je datoteka koja definira stil, odnosno izgled *web*-stranice.

Riječ *cascading* označava kaskadnu primjenu CSS-pravila. CSS-pravilo može se napisati tako da bude primijenjeno na sve elemente ili samo na neke elemente ili tako da vrijedi samo za točno određeni element.

Prije pojave CSS-a oblikovanje izgleda *web*-stranice do određene razine bilo je moguće postići i u HTML-u. No, time se stvorio problem miješanja sadržaja i strukture s kôdom čija je jedina svrha bila prezentacija. HTML-kôd za definiranje izgleda morao se ponavljati iznova na svakom elementu i na svakoj stranici u *web*-sjedištu.

Pojavom CSS-a nastoji se riješiti taj problem. Glavna je ideja CSS-a odvajanje prezentacijskog kôda u zasebne datoteke i njegovo definiranje pomoću jednostavnih pravila koja se mogu odnositi na više elemenata odjednom.

Prva verzija CSS-a definirana je krajem 1996. No do usvajanja CSS-a od strane autora *web*-sadržaja i proizvođača *web*-preglednika prošlo je još dosta vremena. Vrlo dugo *web*-preglednici nisu dosljedno implementirali CSS-specifikaciju pa se autori nisu mogli pouzdati da će stranice izgledati približno isto u svim preglednicima. Razvijeni su brojni trikovi u CSS-u čija je namjena bila da isprave neočekivana ponašanja u nekim preglednicima. Današnja situacija je po tom pitanju puno bolja, iako se i danas savjetuje provjera izgleda stranice u što više preglednika.



Trenutačna verzija CSS-a je CSS 3, koji je donio brojne novitete i poboljšanja. Ta verzija još nije finalizirana, nego se i dalje neprestano razvija i nadograđuje. Razvojem CSS-a bavi se *World Wide Web Consortium* (W3C), tijelo zaduženo za razvoj *web*-standarda u suradnji s proizvođačima preglednika i zainteresiranim *web*-autorima.

1.2. Uključivanje CSS-a u HTML-kôd

CSS-kôd se tipično piše odvojeno od HTML-kôda, tj. u zasebnoj datoteci. Da bismo HTML dokument povezali s CSS-datotekom, koristimo HTML-element *link*:

```
<link rel="stylesheet" type="text/css"
      href="stilovi/boje.css" />
```

Kada se taj element koristi za uključivanje CSS-a (što je zapravo njegova glavna namjena), atribut *rel* mora imati vrijednost "stylesheet", a atribut *type* vrijednost "text/css".

Atribut *href* postavljamo na putanju do CSS-datoteke koju želimo uključiti. U ovom primjeru uključujemo datoteku *boje.css* koja se nalazi u mapi *stilovi*. Radi se o relativnoj putanji od HTML-datoteke do CSS-datoteke. Evo nekoliko najčešćih primjera uporabe relativne putanje:

Putanja	Opis
boje.css	HTML-datoteka i datoteka <i>boje.css</i> nalaze se u istoj mapi.
stilovi/boje.css	Datoteka <i>boje.css</i> nalazi se u mapi <i>stilovi</i> . HTML-datoteka i mapa <i>stilovi</i> nalaze se u istoj mapi.
../stilovi/boje.css	Datoteka <i>boje.css</i> nalazi se u mapi <i>stilovi</i> . Mapa u kojoj se nalazi HTML-datoteka i mapa <i>stilovi</i> nalaze se u istoj mapi.

Element *link* mora se uvijek nalaziti unutar HTML-elementa *head*.

Ako se HTML-stranica koristi većim brojem CSS-datoteka, uključit će se tako da se element *link* navede više puta.

Mogući su i načini pisanja CSS-kôda u samoj HTML-datoteci. Za to služi HTML-element *style*, u kojem se izravno mogu pisati CSS-pravila.

```
<style type="text/css">
  p
  {
    color: red;
  }
</style>
```

Element *style* uvijek se mora nalaziti unutar elementa *head*. Atribut *type* u tom se slučaju postavlja na vrijednost "text/css", a može se i izostaviti.

Drugi način na koji se CSS-kôd može izravno "ubaciti" u HTML je preko atributa *style*:

```
<p style="color:red;">
  Tko rano rani, dvije sreće grabi.
</p>
```

Atribut *style* može se staviti na gotovo svaki HTML-element. U ovom slučaju pišu se samo CSS-deklaracije, bez selektora, a te će se deklaracije primijeniti samo na taj HTML-element.

No iako se ponekad može činiti praktičnim, miješanje se CSS-a i HTML-a u istoj datoteci ne preporuča. Najbolji pristup je uvijek pisati CSS-kôd u zasebnoj datoteci. Time se ostvaruje ponovna iskoristivost kôda i sažetost – određeno CSS-pravilo dovoljno je napisati samo jednom, a ono će vrijediti u svim HTML-dokumentima. Također, kad je potrebno nešto promijeniti u nekom CSS-pravilu, dovoljno je to napraviti na jednom mjestu, a ne u svakoj HTML-datoteci zasebno.

1.3. Sintaksa CSS-pravila

CSS-kôd sastoji se od CSS-**pravila**. Evo primjera jednostavnog CSS-pravila:

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

Ovo pravilo postavlja svim *p* elementima boju teksta na crvenu.

Dio pravila koje određuje (odnosno *selektira*) elemente na koje se pravilo odnosi zove se **selektor**. Svako pravilo moramo započeti selektorom, a najjednostavniji selektor je upravo naziv elementa:

```
p
{
}
}
```

Nakon selektora dolaze vitičaste zagrade. CSS nije osjetljiv na prazan prostor pa vitičaste zagrade nije obavezno pisati u zasebnim redovima, ali se to preporuča radi čitljivosti.

Unutar vitičastih zagrada najprije se navodi **svojstvo** koje se postavlja. U ovom primjeru radi se o boji teksta (CSS-svojstvo *color*):

```
p
{
    color:
}
}
```

Nakon što se navede svojstvo koje se želi postaviti, dolazi dvotočka, a iza nje **vrijednost** na koju se postavlja to svojstvo (u ovom primjeru radi se o nazivu boje):

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

Svojstvo i vrijednost zajedno čine **deklaraciju**.

Unutar jednog pravila može biti više deklaracija:

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

Deklaracije obavezno moramo razdvojiti pomoću točke sa zarezom, a poželjno je svaku deklaraciju pisati u novom redu.

Iza posljednje deklaracije u CSS-pravilu nije obavezno navesti točku sa zarezom, ali se to ipak preporuča (radi eventualnog naknadnog dodavanja novih deklaracija unutar istog pravila).

1.4. CSS-selektori

Kao što je već rečeno, svako CSS-pravilo započinje selektorom, a selektor određuje za koji HTML-element vrijedi to pravilo. U selektoru se mogu koristiti nazivi, klase ili identifikatori elemenata.

Naziv elementa

Najjednostavniji selektor je upravo **naziv** elementa. Želi li se da se pravilo odnosi na *p* elemente, napisat će se:

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

Ako se želi da se pravilo odnosi na elemente *h1*, napisat će se:

```
h1
{
    color: blue;
}
```

Klasa

Često je potrebno da se CSS-pravilo primijeni samo na točno određene HTML-elemente. Da bi se to postiglo, u HTML-u te elemente treba označiti **klasom** određenog imena, upotrebom atributa *class*:

```
<p class="poslovica">
    Tko rano rani, dvije sreće grabi.
</p>
```

Kada se u selektoru navodi klasa, obavezno se prije imena klase mora staviti točka:

```
.poslovica
{
    color: red;
}
```

Navedeno pravilo vrijedit će samo za elemente koji imaju postavljenu klasu *poslovica*.

Identifikator

Osim pomoću klase, HTML-elementi mogu se označiti i identifikatorom pomoću atributa *id*.

```
<p id="jutro">
    Jutro je pametnije od večeri.
</p>
```

Kada se u selektoru koristi identifikator, prije identifikatora se mora navesti oznaka #, kao u primjeru:

```
#jutro
{
    color: red;
}
```

Ovo pravilo primijenit će se samo na element s identifikatorom *jutro*. Identifikator treba biti jedinstven – na istoj HTML-stranici ne bi trebalo dva puta upotrijebiti isti identifikator.

Kod klasa nema tog ograničenja – ista se klasa može upotrijebiti samo jednom, a možemo se upotrijebiti i više puta.

Imenovanje klasa i identifikatora

Prilikom imenovanja klasa i identifikatora treba se voditi određenim pravilima. Dopušteni znakovi su slova od A do Z, brojevi od 0 do 9, crtica (-) i donja crta (_). Naziv klase, odnosno identifikatora, ne smije započeti brojem (a ne preporučaju se ni crtica ili donja crta, jer je to rezervirano za posebnu upotrebu).

U nazivima klasa i identifikatora mogu se koristiti i velika i mala slova. Treba pripaziti na to da su nazivi klasa i identifikatora osjetljivi na velika i mala slova pa tako naziv *mojaKlasa* nije ekvivalentan nazivu *mojajklasa*.

Ako se naziv klase sastoji od dviju riječi, preporuča se da se one odvoje crticom (npr. *moja-klasa*), jer je to stil koji se također koristi i u nazivima CSS-svojtava (npr. *font-size*). Ako nam to ne odgovara, mogu se koristiti i varijante *mojaKlasa* ili *MojaKlasa*. U svakom slučaju, poželjno je dosljedno se držati odabranog stila imenovanja.

Također, prilikom imenovanja klasa i identifikatora poželjno je da imena budu jasna i razumljiva, ali da ipak ne budu preduga.

Kombiniranje naziva elementa, klase i identifikatora

Ako je potrebno, unutar selektora može se navesti i naziv elementa i klasa:

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

Gornje pravilo primijenit će se samo na *p* elemente s klasom *poslovica*, ali ne i na druge *p* elemente, niti na elemente *h1* koji eventualno imaju klasu *poslovica*.

Slična kombinacija moguća je i s identifikatorom:

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

Kombinacija koja se često može susresti su dvije (ili više) klasa u identifikatoru. Naime, na HTML-element može se postaviti više klasa,

Zanimljivosti i napomene

Iako se korištenje istog identifikatora više puta ne preporuča (jer takav HTML nije validan), ponekad se može susresti u praksi (jer većina preglednika to ipak podržava).

Zanimljivosti i napomene

Preporuka je potpuno izbjegavati donju crtu (_) u nazivima klasa i identifikatora, jer to nije podržano u nekim starijim preglednicima.

što je ponekad korisno (npr. kad se element želi dodatno stilizirati). Postavljanje više klasa na element u HTML-u izgleda ovako:

```
<p class="poslovica istina">
    Tko rano rani, dvije sreće grabi.
</p>
```

Unutar atributa *class* različite su klase odvojene razmakom. Taj element tako ima dvije klase – klasu *poslovica* i klasu *istina*.

Kad se u CSS-selektoru žele navesti obje klase, navode se bez razmaka. Važno je zapamtiti da se u selektoru prije naziva klase uvijek navodi točka:

```
.poslovica.istina
{
    color: red;
}
```

Ovo pravilo primijenit će se samo na elemente koji imaju i jednu i drugu klasu.

Ovo se pravilo može dodatno „pooštriti“ navođenjem da se ono odnosi samo na elemente *p*:

```
p.poslovica.istina
{
    color: red;
}
```

Navođenje više selektora u jednom pravilu

Dva (ili više) pravila koja sadrže iste deklaracije mogu se radi preglednosti napisati kao jedno pravilo. Pogledajmo ova dva pravila:

```
p
{
    color: red;
}

h1
{
    color: red;
}
```

Ta se dva pravila, s obzirom da sadrže iste deklaracije, mogu napisati kao jedno pravilo koje ima dva selektora odvojena zarezom:

```
p, h1
{
    color: red;
}
```

U sljedećem slučaju također se radi o dvama pravilima s istim deklaracijama. Kod prvog pravila selektor je naziv elementa, a kod drugog klasa.

```
p
{
    color: blue;
}

.poslovica
{
    color: blue;
}
```

I ovdje se ta dva pravila mogu napisati kao jedno pravilo s dva selektora:

```
p, .poslovica
{
    color: red;
}
```

Selektiranje pomoću roditeljskih elemenata

CSS-pravilo može se dodatno precizirati navođenjem roditeljskih elemenata.

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

Selektor za takvo pravilo može se čitati zdesna nalijevo. Dakle, to pravilo vrijedit će za elemente *strong*, ali samo za one koji se nalaze unutar elementa *p*.

Uz ovaj HTML-kôd:

```
<h1>Kako je <strong>Potjeh</strong> tražio
istinu</h1>

<p>
    <span>Bilo je to u <strong>vrlo</strong> davno
    doba.</span>

    Na jednoj krčevini u staroj bukovoj gori živio
    starac <strong>Vjest</strong> sa svoja tri unuka.
</p>
```

Gore navedeno CSS-pravilo dat će ovakav rezultat:

Kako je Potjeh tražio istinu

Bilo je to u **vrlo** davno doba. Na jednoj krčevini u staroj bukovoj gori živio starac **Vjest** sa svoja tri unuka.

To je pravilo primijenjeno na elemente *strong* koji se nalaze unutar elementa *p*, bez obzira je li im element *p* izravni roditelj (element *strong* s tekстом „Vjest“) ili se između njih nalazi jedna ili više razina (element *strong* s tekстом „vrlo“).

To pravilo, kao što je i očekivano, nije djelovalo na element *strong* koji se nalazi unutar elementa *h1* (element *strong* s tekстом „Potjeh“).

Ako bi se željelo da se to pravilo primijeni samo na element *strong* koji se nalazi unutar elementa *span* koji se nalazi unutar elementa *p* (u primjeru bi to bio element *strong* s tekстом „vrlo“), navest ćemo ovakav selektor:

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

Broj razina koje možemo navesti nije ograničen, ali u praksi se ne preporuča navoditi preveliki broj razina (zbog performansi i zbog mogućih pogrešaka).

Također, moguće je ograničiti da se to pravilo primijeni samo na elemente *strong* koji su izravna djeca elementa *p* (u ovom primjeru samo na element *strong* s tekстом „Vjest“):

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

Oznaka `>` u selektoru označava da se on odnosi na elemente *strong* koji su izravna djeca elementa *p*.

Umjesto naziva roditeljskog elementa, moguće je navesti njegovu klasu:

```
.poslovica strong
{
    color: red;
}
```

Ovaj se selektor odnosi na elemente *strong* koji se nalaze unutar elementa s klasom *.poslovica*.

Naravno, i naziv drugog elementa može se zamijeniti klasom. U ovom primjeru naveden je selektor koji se odnosi na elemente s klasom *.ime* koji se nalaze unutar elementa s klasom *.poslovica*.

```
.poslovica .ime
{
    color: red;
}
```

Zanimljivosti i napomene

Oznaka `>` u selektoru nije podržana u Internet Exploreru 6.

Selektiranje pomoću atributa

Osim prema klasama i identifikatorima, HTML-elementi mogu se selektirati i po drugim atributima.

Jedan od najčešćih slučajeva gdje je to korisno su elementi *input*. Ako se u HTML-kôdu nalazi element *input* tipa *text*, kao u ovom primjeru:

```
<input type="text" value="Unesite ime" />
```

sljedeće će se pravilo primijeniti samo na elemente *input* tipa *text*.

```
input[type="text"]  
{  
    color: red;  
}
```

Dakle, u selektoru je (u uglatim zagradama) potrebno navesti naziv atributa (*type*), a zatim i njegovu vrijednost (*text*).

Univerzalni selektor

U CSS-u postoji i posebni selektor, tzv. univerzalni selektor pomoću kojeg se mogu odabrati svi elementi.

```
*  
{  
    color: red;  
}
```

Oznaka za univerzalni selektor je * (zvjezdica). To pravilo postaviti će boju svih elemenata u HTML-dokumentu na crvenu.

Pregled dosad obrađenih selektora

U sljedećoj tablici dani su različiti primjeri selektora i objašnjenje za koje će elemente oni vrijediti:

Selektor	Elementi na koje će se pravilo primijeniti
<code>p</code>	svi elementi <i>p</i>
<code>h1</code>	svi elementi <i>h1</i>
<code>.poslovica</code>	svi elementi s klasom <i>poslovica</i>
<code>#jutro</code>	element s identifikatorom <i>jutro</i>
<code>p.poslovica</code>	svi elementi <i>p</i> s klasom <i>poslovica</i>
<code>p#jutro</code>	element <i>p</i> s identifikatorom <i>jutro</i>
<code>poslovica.istina</code>	svi elementi koji imaju i klasu <i>istina</i> i klasu <i>poslovica</i>
<code>p.poslovica.istina</code>	elementi <i>p</i> koji imaju i klasu <i>istina</i> i klasu <i>poslovica</i>
<code>p, h1</code>	svi elementi <i>p</i> i <i>h1</i>
<code>p strong</code>	svi elementi <i>strong</i> u elementu <i>p</i>
<code>p > strong</code>	elementi <i>strong</i> koji su izravna djeca elementa <i>p</i>
<code>.poslovica strong</code>	svi elementi <i>strong</i> u elementu s klasom <i>poslovica</i>
<code>.poslovica .ime</code>	svi elementi s klasom <i>ime</i> u elementu s klasom <i>poslovica</i>
<code>p span strong</code>	svi elementi <i>strong</i> u elementu <i>span</i> koji se nalaze u elementu <i>p</i>
<code>*</code>	svi elementi

1.5. Prioritet CSS-pravila

Ako se želi postaviti pravilo koje svim elementima *p* postavlja boju teksta na plavu, može se napisati:

```
p
{
    color: blue;
}
```

Ako se želi postaviti pravilo koje svim elementima s klasom *poslovica* postavlja boju teksta na crvenu, može se napisati:

```
.poslovica
{
    color: red;
}
```

Na taj HTML-element primijenit će se oba prethodno navedena CSS-pravila:

```
<p class="poslovica">
    Bez alata nema zanata.
</p>
```

Budući da oba CSS-pravila postavljaju svojstvo *color*, koje od njih će imati prioritet određeno je samim selektorom. CSS-pravilo čiji selektor ima veću **specifičnost** imat će i veći prioritet, te će ono odlučiti o vrijednosti svojstva.

Navođenje klase je specifičnije od navođenja naziva elementa pa će stoga drugo pravilo imati viši prioritet i boja teksta u elementu će biti crvena.

Još specifičnije od klase je navođenje identifikatora pa pravilo koje u selektoru navodi identifikator ima viši prioritet od oba ovdje navedena pravila.

A što kad se u selektoru nalazi kombinacija naziva elementa, klase i identifikatora i kad se u selektoru navode i roditeljski elementi? U takvim slučajevima može se izračunati specifičnost selektora da bi se znalo koje pravilo će imati prioritet. Specifičnost selektora računa se ovako:

- specifičnost naziva elementa iznosi 1
- specifičnost klase iznosi 10
- specifičnost identifikatora iznosi 100
- ako u selektoru postoji više naziva elemenata, klasa ili identifikatora, njihove vrijednosti se zbrajaju i tako se dobije ukupna specifičnost selektora.

Selektor *p span strong* ima specifičnost 3, jer sadrži tri naziva elementa.

Za selektor *.poslovica strong*, specifičnost iznosi 11, jer sadrži jednu klasu i jedan naziv elementa. Taj selektor ima veću specifičnost od prethodnog.

Zanimljivosti i napomene

Upravo zbog svoje visoke specifičnosti, ne preporuča se rabiti identifikatore u CSS-u, osim kad za to postoji poseban razlog. Čim se jedan identifikator nalazi u selektoru, taj selektor je „jači“ od svih drugih selektora koji sadrže samo klase pa to vrlo često ima neželjeni učinak.

Selektor *p#jutro* ima specifičnost 101, jer sadrži jedan identifikator i jedan naziv elementa i on ima najveću specifičnost od sva tri navedena selektora.

Ako dva selektora imaju istu specifičnost, veći će prioritet imati pravilo koje je navedeno poslije.

1.6. Pseudoklase

Pseudoklase su izrazi u CSS-u koji omogućuju selektiranje elemenata slično kao što to čine klase. U CSS-pravilima koriste se poput klasa, ali se za razliku od klasa ne postavljaju na elemente u HTML-u.

Pseudoklase vezane za položaj elementa

Za primjer se može uzeti HTML-kôd koji ima listu s tri stavke:

```
<ul>
  <li>Stribor</li>
  <li>Jagor</li>
  <li>Regoč</li>
  <li>Potjeh</li>
</ul>
```

Ako se želi samo na prvoj stavci postaviti boja teksta na crvenu, trebalo bi postaviti klasu na prvu stavku:

```
<ul>
  <li class="prva">Stribor</li>
  <li>Jagor</li>
  <li>Regoč</li>
  <li>Potjeh</li>
</ul>
```

Zatim je potrebno napisati CSS-pravilo koje će stavci s klasom *prva* postaviti boju na crvenu:

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

U ovakvim slučajevima, kada se želi primijeniti neko pravilo na temelju položaja elementa, nije potrebno uvijek napraviti novu klasu, nego je moguće koristiti se nekom od pseudoklasa vezanih uz položaj elementa.

Pseudoklasa ***first-child*** selektira element koji se nalazi prvi unutar roditeljskog elementa.

U gornjem primjeru moguće je izbaciti klasu iz HTML-kôda, a CSS-pravilo napisati ovako:

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

Zanimljivosti i napomene

Pseudoklase su podržane tek od Internet Explorera 9 (s iznimkom pseudoklasa *link*, *visited*, *hover* i *active*, koje su podržane i u starijim preglednicima).

Prilikom korištenja pseudoklase u selektoru treba navesti dvotočku (slično kao što se klasa u selektoru uvijek navodi s točkom).

To će pravilo postaviti boju teksta na crvenu samo prvoj stavci u listi:

- Stribor
- Jagor
- Regoč
- Potjeh

Ako se želi selektirati samo posljednja stavka liste, može se rabiti pseudoklasa ***last-child***. Ta pseudoklasa selektira element koji se nalazi posljednji unutar roditeljskog elementa.

```
li:last-child
{
    color: blue;
}
```

Sada se posljednjoj stavci liste boja teksta postavila na plavu.

- Stribor
- Jagor
- Regoč
- Potjeh

Ako se želi selektirati neka druga stavka u listi, može se koristiti pseudoklasa ***nth-child***. Toj pseudoklasi treba postaviti i parametar koji određuje na koji se element unutar roditeljskog elementa odnosi.

Ako se želi selektirati samo drugu stavku, napiše se:

```
li:nth-child(2)
{
    color: red;
}
```

To pravilo postavlja boju teksta na crvenu samo drugoj stavci liste:

- Stribor
- Jagor
- Regoč
- Potjeh

Ako se želi selektirati svaka druga stavka, umjesto 2 kao parametar pseudoklasi ***nth-child*** preda se $2n$:

```
li:nth-child(2n)
{
    color: red;
}
```

To pravilo postavlja boju teksta na crvenu svakoj drugoj stavci liste:

- Stribor
- Jagor
- Regoč
- Potjeh

Isti se efekt dobije ako se umjesto 2n napiše ključna riječ *even*:

```
li:nth-child(even)
{
    color: red;
}
```

To pravilo će također postaviti boju teksta na crvenu svakoj drugoj stavci liste, preciznije svakoj parnoj stavci.

Moguće je selektirati i svaku neparnu stavku, ako se umjesto ključne riječi *even* upotrijebi *odd*.

```
li:nth-child(odd)
{
    color: red;
}
```

To pravilo postavlja boju teksta na crvenu svakoj neparnoj stavci liste.

- Stribor
- Jagor
- Regoč
- Potjeh

Pseudoklase vezane uz korisničke akcije i elemente za unos

Pomoću ovih pseudoklasa moguće je različito stilizirati elemente ovisno o korisnikovim akcijama te tako dodatno naglasiti informaciju o tome koji je element korisnik odabrao ili koji je link već posjetio i time mu olakšati navigaciju i interakciju s *web*-stranicom.

Pseudoklasa ***link*** selektira odabire link (element *a*) koji korisnik još nije posjetio.

```
a:link
{
    color: blue;
}
```

Pseudoklasa ***visited*** selektira link koji je korisnik već posjetio:

```
a:visited
{
    color: purple;
}
```

Pseudoklasa **active** selektira element na koji je korisnik upravo kliknuo (npr. link ili dugme).

```
a:active
{
    color: red;
}
```

Pseudoklasa **hover** selektira element preko kojeg korisnik trenutačno prelazi mišem. To može biti link, ali i drugi element poput elemenata *div*, *span*, *p* ili *li*.

```
div:hover
{
    color: yellow;
}
```

Pseudoklasa **focus** selektira element koji je trenutačno u fokusu (npr. element za unos u kojem se nalazi kursor ili link).

```
input:focus
{
    color: red;
}
```

Ako neki element za unos (ili dugme) trenutačno nije omogućen, odnosno vidljiv je, ali iz nekog razloga ne dopuštamo unos teksta ili klik na njega, svakako to želimo vizualno poručiti korisniku.

Preglednici imaju svoj zadani stil kojim prikazuju onemogućene elemente, no taj se stil može promijeniti pomoću dviju pseudoklasa.

Pseudoklasa **enabled** selektira elemente koji su omogućeni (npr. linkovi, dugmad ili elementi za unos).

```
input:enabled
{
    color: black;
}
```

Pseudoklasa **disabled** selektira elemente koji nisu omogućeni.

```
input:disabled
{
    color: gray;
}
```

Na dva specijalna elementa za unos (kvačica i dugme za odabir) mogu se primijeniti dvije pseudoklase koje selektiraju element ovisno o njegovu stanju – je li označen (tj. je li stavljena kvačica) ili ne.

Pseudoklasa **checked** selektira elemente koji su označeni (elementi *input* tipa *checkbox* ili *radio*).

```
input:checked
{
    color: red;
}
```

Pseudoklasa ***indeterminate*** selektira elemente koji nisu označeni (također *input* elementi tipa *checkbox* ili *radio*).

```
input:indeterminate
{
    color: black;
}
```

Specifičnost pseudoklasa

Pseudoklase imaju istu specifičnost kao i klase (također nose vrijednost 10).

Selektor `li:first-child` imat će ukupnu specifičnost 11 (zbroj vrijednosti 1 od naziva elementa i vrijednosti 10 od pseudoklase).

1.7. Nasljeđivanje vrijednosti

Velik broj CSS-svojstava može se nasljeđivati. Ako na elementu nije postavljena vrijednost za neko od takvih svojstava, koristit će se vrijednost s roditeljskog elementa.

To omogućuje da se osnovna svojstva kao što su boja, veličina i vrsta fonta postave samo na jednom mjestu (najčešće na elementu *body*), umjesto da ih treba iznova definirati za svaki element.

```
body
{
    color: blue;
}
```

Ako je na elementu *body* postavljena boja teksta na plavu, ona će biti plava i na svim drugim elementima, bez da je treba eksplicitno postaviti, jer su svi elementi djeca (ili potomci) elementa *body*.

Naravno, na elementima za koje je to potrebno, moguće je postaviti različite vrijednosti:

```
p
{
    color: gray;
}
```

Ako se za neki od elemenata želi vratiti vrijednost svojstva na naslijeđenu, to se može postići uporabom ključne riječi *inherit*.

```
p.poslovica
{
    color: inherit;
}
```

Ključnu riječ *inherit* moguće je rabiti i na CSS-svojstvima čije se vrijednosti ne nasljeđuju. U tom se slučaju mijenja ponašanje CSS-svojstva i vrijednost se nasljeđuje s roditeljskog elementa, iako to inicijalno ne bi bilo tako.

1.8. Komentari

U slučaju kad se želi pojasniti namjena nekog CSS-pravila ili deklaracije, moguće je u CSS-kôd dodati komentare – proizvoljni tekst koji opisuje namjenu neke linije ili bloka kôda.

Komentari započinju slijedom znakova `/*`, a završavaju slijedom `*/`. Komentar se može nalaziti u jednom retku, ali se može protezati i kroz više njih.

Primjer komentiranja izgleda ovako:

```
p
{
    /* postavi boju */
    color: blue;
}
```

Preporuča se uvijek stavljati komentare kad namjena napisane CSS-deklaracije nije na prvi pogled jasna.

Komentari se također često koriste za grupiranje više CSS-pravila koji imaju zajedničku namjenu:

```
/* Definicija boja */
```

```
p
{
    color: blue;
}
h1
{
    color: blue;
}
```

Prilikom testiranja komentari se često koriste da bi se privremeno isključio dio kôda:

```
p
{
    /* color: blue; */
}
```

Kôd koji se nalazi unutar oznake za komentare neće se primijeniti, nego će biti ignoriran (jednako kao da se radi o običnom komentaru).

2. Oblikovanje teksta

CSS nudi brojne mogućnosti za oblikovanje teksta – od osnovnih kao što je odabir boje, veličine i vrste fonta, pa do onih rjeđe korištenih svojstava pomoću kojih se može upravljati proredima i razmakom između slova. U novije vrijeme CSS nudi i mogućnosti kao što je postavljanje sjene na slova te korištenje fontova koje korisnik ne mora imati instalirane na svom računalu, što je značajno unaprijedilo tipografiju na *webu*.

2.1. Boja

Boju teksta postavljamo pomoću svojstva **color**, s kojim smo se već susreli.

Ako vrijednost boje nije postavljena, ona će biti naslijeđena od roditeljskog elementa.

Vrijednost svojstva *color* moguće je postaviti na više načina – preko naziva boje te preko posebnog načina zapisivanja boje – heksadecimalnog kôda ili preko zapisa boja u modelima RGB ili HSL.

Naziv boje

U priručniku se dosad vrijednost svojstva *color* postavljala navođenjem engleskog naziva boje.

```
color: red;
```

U sljedećoj tablici navedeni su nazivi 17 osnovnih boja:

Naziv boje u CSS-u	Objašnjenje
<i>aqua</i>	jarka svijetloplava
<i>black</i>	crna
<i>blue</i>	plava
<i>fuchsia</i>	svijetloljubičasta
<i>gray</i>	siva
<i>green</i>	zelena
<i>lime</i>	jarka svijetlozelena
<i>maroon</i>	crvenosmeđa
<i>navy</i>	mornarski plava
<i>olive</i>	maslinasto zelena
<i>orange</i>	narandžasta
<i>purple</i>	ljubičasta
<i>red</i>	crvena

<i>silver</i>	srebrna
<i>teal</i>	tamnotirkizna
<i>white</i>	bijela
<i>yellow</i>	žuta

U CSS-u je na raspolaganju ukupno 140 boja koje se mogu koristiti pomoću svojeg imena.

Nazivi boja su vrlo praktični i jednostavni za korištenje, no tako se može koristiti ograničen broj boja.

Osim ključnih riječi za naziv boje, može se upotrijebiti i ključna riječ *inherit*, čime se boja postavlja na vrijednost roditeljskog elementa.

Heksadecimalni zapis boje

Najčešći način postavljanja boje u CSS-u je navođenjem heksadecimalnog zapisa boje.

```
color: #f5412a;
```

Takav se kôd sastoji od 6 heksadecimalnih znamenaka, a prije njega se obavezno navodi oznaka **#**. Kao znamenke se mogu pojaviti brojeke od 0 do 9 i slova a, b, c, d, e ili f.

Za prikaz boje na računalima koristi se **model boja RGB**, u kojem je svaka boja predstavljena bročanom vrijednosti svoje crvene (*red*), zelene (*green*) i plave (*blue*) komponente. Svaka komponenta može poprimiti vrijednost od 0 do 255. Tako se može zapisati više od 16 milijuna boja.

Heksadecimalni zapis boje sastoji se od vrijednost tih triju komponenti. Prve dvije znamenke heksadecimalnog kôda boje predstavljaju vrijednost crvene, druge dvije zelene, a posljednje dvije plave komponente boje.

Nedostatak je takvog načina zapisivanja boja da je teško čitljiv i da je nemoguće iz samog CSS-a prepoznati o kojoj se boji radi. Nasreću, za korištenje heksadecimalnog kôda boje nije potrebno poznavati njegov način izračuna, već je moguće kopirati heksadecimalni kôd iz nekog grafičkog programa ili *online* alata za odabir boje.

Zapisi boje RGB i RGBA

Osim pomoću heksadecimalnog zapisa, u CSS-u postoji mogućnost postavljanja boje izravnim navođenjem vrijednosti njezinih komponenti R, G i B. Kao što je već navedeno, svaka komponenta može poprimiti vrijednost od 0 do 255, a navode se u pozivu *rgb* funkcije:

```
color: rgb(245, 65, 42);
```

Osim veće čitljivosti, RGB-zapis boje ima još jednu prednost, a to je da se može proširiti dodavanjem još jedne komponente – prozirnosti. Takav zapis naziva se **RGBA-zapisom** u kojem komponenta *a* predstavlja prozirnost (*alpha channel*) boje. Moguće vrijednosti za prozirnost su

Zanimljivosti i napomene

Preporuča se uporaba malih slova u pisanju heksadecimalnih kôdova. Dakle, bolje je pisati #f5412a nego #F5412A, premda će oba načina pisanja funkcionirati. Razlog je dosljednost u pisanju – u cijelom CSS-u se preporuča uporaba isključivo malih slova.

Zanimljivosti i napomene

Postoji i skraćeni način zapisivanja heksadecimalnih kôdova boja u CSS-u. Ako se kôd sastoji od ponavljajućih znamenki, možemo ga zapisati sa samo 3 znamenke. Kôd #ff3300 u skraćenom zapisu postaje #f30.

Zanimljivosti i napomene

Pojam "*websafe* boje" nekada je bio vrlo bitan - radilo se o popisu 216 boja koje su nekadašnji monitori bili u stanju prikazati. Danas možemo zanemariti tu preporuku, i slobodno koristiti bilo koju heksadecimalnu, odnosno RGB boju.

decimalni brojevi od 0 do 1, koji predstavljaju postotak prozirnosti (0.5 znači 50%, dok 1 znači 100 %). Vrijednosti se navode u pozivu *rgba* funkcije:

```
color: rgba(245, 65, 42, 0.6);
```

U ovom primjeru postavlja se prozirnosti od 60 % na boju iz prethodnog primjera.

RGB-vrijednosti boja također se mogu kopirati iz grafičkih programa i koristiti u CSS-kôdu.

Zapisi boje HSL i HSLa

HSL model je još jedan od načina zapisivanja boje preko njezinih komponenti, s time da su komponente ovoga puta nijansa (*hue*), zasićenost (*saturation*) i *lightness* (svjetlo). Uporabom tog modela vrlo se lako mogu dobiti svjetliji i tamniji tonovi iste boje – mijenjanjem zasićenosti i svjetline, dok je to u modelu RGB nepraktično.

Kad se u CSS-u rabi model HSL, vrijednost prve komponente (nijanse) postavljamo na broj od 0 do 360, a vrijednosti drugih dviju komponenti (zasićenja i svjetline) na postotak od 0 do 100 % (ovoga puta se ne rabe decimalni brojevi nego postotci):

```
color: hsl(7, 91%, 56%);
```

Kao i kod modela RGB, i ovdje je moguće postaviti prozirnost kao dodatnu komponentu (*a*):

```
color: hsla(7, 91%, 56%, 0.6);
```

Potrebno je pripaziti na to da se prozirnost i dalje postavlja s decimalnim brojem od 0 do 1 (a ne s postotkom).

Model HSL ekvivalentan je modelu RGB, odnosno može prikazati isti broj boja, a moguća su preračunavanja iz jednog modela u drugi.

Treba biti pažljiv prilikom kopiranja HSL-vrijednosti iz grafičkih programa, jer se oni često koriste varijacijom tog modela (HSB-model) pa dobivena boja nije posve identična. Najbolje je koristiti se alatima specijaliziranim za odabir boje u HSL-zapisu.

Zanimljivosti i napomene

Zapisi boja RGB, RGBa, HSL i HSLa podržani su tek u novijim verzijama preglednika (IE9+, Firefox 3+, Opera 10+)

2.2. Vrsta fonta

Vrsta fonta postavlja se pomoću svojstva **font-family**. Kao vrijednost se navodi naziv fonta:

```
font-family: Helvetica;
```

Ako naziv fonta sadrži razmake (ili specijalne znakove), potrebno je navesti ga u navodnicima. Mogu se koristiti jednostruki ili dvostruki navodnici.

```
font-family: 'Helvetica Neue';
```

Ako font koji je naveden nije instaliran na korisnikovu računalu, preglednik će prikazati svoj predefinirani font (najčešće *Times New Roman*). Da bi se to izbjeglo, može se navesti jedan ili više zamjenskih fontova (odvojenih zarezom) koji će se upotrijebiti u slučaju ako prvi font nije instaliran:

```
font-family: Helvetica, Arial;
```

Ako nijedan od zamjenskih fontova nije instaliran, i dalje će se prikazati preglednikov predefinirani font. No moguće je odabrati barem koji će to tip fonta biti – sa serifima (najčešće *Times New Roman*) ili bez serifa (najčešće *Arial*). Na kraju popisa fontova navodi se ključna riječ:

```
font-family: Helvetica, Arial, sans-serif;
```

Na raspolaganju je pet ključnih riječi za tip fonta:

Ključna riječ	Objašnjenje
<i>serif</i>	font sa serifima (npr. Times New Roman)
<i>sans-serif</i>	font bez serifa (npr. Arial)
<i>monospace</i>	font s jednakom širinom svih slova (npr. Courier New)
<i>cursive</i>	font sličan rukopisu (npr. Comic Sans MS)
<i>fantasy</i>	dekorativni font prikladan za naslove (npr. Impact)

Vrsta fonta se, ako nije postavljena, također nasljeđuje od roditeljskog elementa.

2.3. Veličina

Veličina fonta postavlja se pomoću svojstva **font-size**. Vrijednost tog svojstva može se postaviti na više načina: pomoću ključnih riječi, pomoću apsolutnih jedinica (piksela i točke) te pomoću relativnih jedinica (postotci, *em* i *rem* jedinice).

Zanimljivosti i napomene

Naziv svojstva *font-family* odnosi se na „obitelj” fontova, npr. *Arial*. Unutar „obitelji” zapravo se nalazi više inačica fonta: *Helvetica Regular*, *Helvetica Bold*, *Helvetica Italic*, *Helvetica Narrow*...

Zanimljivosti i napomene

Kad god se neko CSS-svojstvo sastoji od dviju riječi (kao npr. *font-family*), one će biti odvojene crticom. Preporuča se rabiti crticu i u vlastitim nazivima klasa.

Zanimljivosti i napomene

Ne preporuča se oslanjati se na ključne riječi *cursive* i *fantasy*. Preglednici prikazuju vrlo različite fontove za *cursive*, a *fantasy* uopće nije podržan u nekim preglednicima (npr. Firefox).

Ključne riječi

Ovako izgleda postavljanje fonta navođenjem ključne riječi:

```
font-size: large;
```

U sljedećoj tablici dan je popis dostupnih ključnih riječi i primjeri njihove uporabe:

Ključna riječ	Objašnjenje	Primjer
<i>xx-small</i>	najmanja slova	<i>xx-small</i>
<i>x-small</i>	vrlo mala slova	<i>x-small</i>
<i>small</i>	manja slova	<i>small</i>
<i>medium</i>	srednja veličina slova	<i>medium</i>
<i>large</i>	veća slova	<i>large</i>
<i>x-large</i>	vrlo velika slova	<i>x-large</i>
<i>xx-large</i>	najveća slova	<i>xx-large</i>
<i>larger</i>	za jednu razinu veća slova od roditeljskog elementa	
<i>smaller</i>	za jednu razinu manja slova od roditeljskog elementa	

Veličina fonta se također, ako nije eksplicitno postavljena, nasljeđuje od roditeljskog elementa.

Pikseli

Veličina se fonta vrlo često definira pomoću piksela. Ovdje se radi o mjernoj jedinici koja je definirana veličinom pojedine točke (piksela) na ekranu računala. Ta se jedinica rabi kad je potrebno precizno definirati veličinu teksta (da bi u potpunosti odgovarala zadanom dizajnu).

Ovako izgleda postavljanje veličine fonta na vrijednost od 14 piksela:

```
font-size: 14px;
```

No veličina teksta koja je definirana u pikselima ipak će malčice varirati od preglednika do preglednika (jer se svaki preglednik koristi vlastitim algoritmom za renderiranje fonta).

Također, neki mobilni uređaji (npr. *iPhone* i neki *Android* uređaji) imaju vrlo visoku rezoluciju pa se na njima tekst veličine 12 piksela prikazuje kao vrlo malen, dok je inače normalno čitljiv na stolnim računalima.

Točke (points)

Točke su mjerna jedinica koja se rabi kod definiranja CSS-a za ispis *web*-stranice. Jedna točka (*point*) ispisana na papir zauzima 1/72 inča, odnosno 0.035 cm.

Zanimljivosti i napomene

U starijim preglednicima (*Internet Explorer 8* i stariji) korisnici nisu mogli povećati veličinu teksta postavljenu pomoću piksela i time sebi olakšati čitanje. Zbog toga se preporučivalo ne koristiti se pikselima, no u modernim preglednicima taj problem više ne postoji.

Zanimljivosti i napomene

Ako se uopće ne postavi veličina fonta, tekst će se prikazivati u predefiniranoj veličini koja je kod većine preglednika 16 piksela.

Zanimljivosti i napomene

Pikseli, točke te *em* i *rem* mjerne su jedinice CSS-a za duljinu. Osim za veličinu fonta, koriste se i za definiranje veličine pravokutnih elemenata, ali i na drugim mjestima gdje treba definirati određenu duljinu.

Postotci se u CSS-u također mogu koristiti za definiranje veličine fonta i pravokutnih elemenata, ali ih ipak ne možemo rabiti na svim mjestima gdje i jedinice za duljinu.

Tako se veličina fonta postavlja na vrijednost od 14 točaka:

```
font-size: 14pt;
```

Točke su namijenjene za korištenje samo u CSS-u za ispis i ne preporuča se rabiti ih za veličinu teksta koji će biti prikazan na *webu*, jer će rezultat jako varirati od preglednika do preglednika.

Postotci

Osim apsolutnim definiranjem veličine fonta pomoću piksela (ili točaka), uporabom se postotaka veličinu fonta može definirati relativno, tj. u odnosu na roditeljski element. Taj način postavljanja veličine fonta sličan je kao da se koristimo ključnim riječima *smaller* i *larger*, s tim da možemo preciznije definirati razliku u veličini.

Postavljanje veličine fonta na 75 % od veličine roditeljskog elementa:

```
font-size: 75%;
```

Prednost relativnog definiranja veličine fonta je u tome što je na jednom mjestu moguće promijeniti veličinu fonta za sve elemente. Često se na elementu *body* postavi početna veličina fonta, a za sve se druge elemente veličina fonta postavi relativno u odnosu na njihove roditeljske elemente. Tako se promjenom veličine fonta na elementu *body* proporcionalno mijenja veličina fonta na svim elementima.

Jedinica *em*

Jedinica *em* je relativna mjerna jedinica za duljinu koja se koristi slično kao i postotci. 1 *em* predstavlja veličinu fonta jednaku onoj roditeljskog elementa, 0.5 *em* predstavlja upola manju, a 2 *em* predstavlja dva puta veću veličinu fonta od roditeljskog elementa.

Ovako se postavlja veličina fonta na 0.75 (odnosno na 75 %) od veličine roditeljskog elementa:

```
font-size: 0.75em;
```

Kod postotaka i kod jedinice *em* pojavljuje se jedan problem: veličinu fonta uvijek treba zadati u odnosu na roditeljski element. To je problematično za praćenje kod elemenata koji se nalaze na različitoj razini. Ako se za element *body* definira veličina od 16 piksela, a potom se za elemente *p* i za *span* veličina definira kao 75 % od veličine roditeljskog elementa, na tim ćemo elementima imati veličinu fonta od 12 piksela. No ako unutar elementa *p* imamo ugniježđen element *span*, taj element će imati veličinu fonta 75 % od svojeg roditelja, odnosno 9 piksela, što najčešće nije željeni učinak.

Jedinica *rem*

Jedinica *rem* (*root em*) je rješenje gore opisanog problema. Radi se također o relativnoj mjernoj jedinici za duljinu, no veličina se fonta navodi u odnosu na korijenski (*root*) element (što je zapravo element *html*).

Dakle, nije potrebno voditi računa o tome koji se element nalazi unutar kojega i računati iznos razlike u postocima za svakoga, već se sve veličine definiraju u odnosu na veličinu fonta postavljenu na elementu *html*.

Zanimljivosti i napomene

Naziv jedinice *em* dolazi od slova M – ta jedinica originalno dolazi iz tipografije, a predstavljala je širinu velikog slova M.

Zanimljivosti i napomene

Jedinica *rem* nije podržana u starijim preglednicima (*Internet Explorer 8* i drugi stariji preglednici).

Ovako se postavlja veličina fonta na 0.75 (odnosno 75 %) od veličine fonta *html* elementa:

```
font-size: 0.75rem;
```

2.4. Težina i stil

Težina fonta

Težina (odnosno pojačanje) fonta postavlja se pomoću svojstva **font-weight**. Kao vrijednost može se navesti ključna riječ:

```
font-weight: bold;
```

Na raspolaganju su ove ključne riječi:

Ključna riječ	Objašnjenje
<i>normal</i>	standardna težina fonta
<i>bold</i>	pojačani (boldani) font
<i>bolder</i>	težina fonta je veća za jednu vrijednost u odnosu na roditeljski element (unutar dostupnih težina fonta)
<i>lighter</i>	težina fonta je manja za jednu vrijednost u odnosu na roditeljski element (unutar dostupnih težina fonta)

Ključna riječ *normal* koristit će se kad se želi poništiti drugo pravilo koje je postavilo težinu elementa na *bold*.

Vrijednost ovog svojstva se također nasljeđuje s roditeljskog elementa.

Osim ključnih riječi mogu se koristiti i numeričke vrijednosti:

```
font-weight: 700;
```

U tablici su dane sve vrijednosti koje se mogu postaviti za ovo svojstvo:

Vrijednost	Objašnjenje
100	slabije od <i>normal</i> za 3 težine
200	slabije od <i>normal</i> za 2 težine
300	slabije od <i>normal</i> za 1 težine
400	težina ekvivalentna <i>normal</i>
500	jače od <i>normal</i> za 1 težinu
600	jače od <i>normal</i> za 2 težinu
700	težina ekvivalentna <i>bold</i>
800	jače od <i>bold</i> za 1 težinu
900	jače od <i>bold</i> za 2 težine

Zanimljivosti i napomene

Ako postavimo vrijednost težine koja nije dostupna na korisnikovu računalu, bit će prikazana najbliža dostupna težina.

Da bi korisnik doista vidio tekst u postavljenoj težini, na njegovu računalu mora biti instalirana inačica fonta u toj težini. Većina fontova instalirana je samo u težinama *normal* (400) i *bold* (700). U praksi se stoga najčešće rabe samo ključne riječi *bold* i *normal*.

Stil fonta

Stil fonta postavlja se pomoću svojstva **font-style**. Kao vrijednost navodi se ključna riječ:

```
font-style: italic;
```

Na raspolaganju su ove ključne riječi:

Ključna riječ	Objašnjenje
<i>normal</i>	standardni stil fonta
<i>italic</i>	nakošeni stil fonta (kurziv)
<i>oblique</i>	nakošeni stil fonta

Slično kao i kod težina, moguće je rabiti samo stilove koje korisnik ima instalirane na svojem računalu. Većina fontova dolazi zajedno s inačicom *italic* pa se najčešće koristi ta ključna riječ.

Ključna riječ *normal* koristi se kad se želi poništiti drugo pravilo koje je postavilo stil elementa na *italic*.

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

2.5. Poravnanje i razmak

Poravnanje teksta

Za određivanje poravnanja teksta koristi se svojstvo **text-align**. Kao vrijednost moguće je postaviti jednu od sljedećih ključnih riječi:

Ključna riječ	Objašnjenje (i primjer)
<i>left</i>	tekst je poravnat uz lijevi rub elementa u kojem je sadržan
<i>right</i>	tekst je poravnat uz desni rub elementa u kojem je sadržan
<i>center</i>	tekst je centriran na sredini elementa u kojem je sadržan
<i>justify</i>	tekst je rastegnut, tako da je poravnat i uz lijevi i uz desni rub elementa u kojem je sadržan

Na primjer, na ovaj način postavlja se poravnanje teksta na lijevu stranu:

```
text-align: left;
```

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

Zanimljivosti i napomene

Italic fontovi obično su namjenski dizajnirani da izgledaju nakošeno, dok su fontovi *oblique* dobiveni zakošenjem originalnog fonta za određeni broj stupnjeva.

Zanimljivosti i napomene

Većina fontova dolazi instalirana i s inačicom *Bold Italic*, što čini mogućim da se istom elementu istovremeno postave i težina na *bold* i stil na *italic*.

Zanimljivosti i napomene

Za razliku od tiskanog teksta (npr. novina), na *webu* se rijetko susreću uski stupci i tekst kojem je poravnat i lijevi i desni rub (vrijednost *justify*). Na *webu* prevladava lijevo poravnati tekst, a ostali načini poravnanja koriste se samo u specifičnim situacijama.

Uvlačenje teksta

U tiskanom tekstu često se uvlači prvi red na svakom odlomku. Na *webu* to nije čest slučaj, ali je to moguće postići upotrebom svojstva ***text-indent***. To svojstvo kao vrijednost prima širinu za koju će tekst biti uvučen. Moguće je koristiti se mjernim jedinicama za duljinu u CSS-u: pikselima, točkama, jedinicama *em* i *rem*, kao i postotcima.

```
text-indent: 25px;
```

Kralj dade toga čovjeka pitati, da li bi mu htio dati svoga sina. On privoli. Kralj odvede njega u svoj dvor. Ubrzo mu omili, a kralj ga je ljubio kao rođeno dijete.

Ovom se svojstvu može postaviti i negativna vrijednost – u tom slučaju će tekst, umjesto da bude uvučen, biti izbačen ulijevo:

```
text-indent: -25px;
```

Kralj dade toga čovjeka pitati, da li bi mu htio dati svoga sina. On privoli. Kralj odvede njega u svoj dvor. Ubrzo mu omili, a kralj ga je ljubio kao rođeno dijete.

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

Razmak između slova

Pomoću svojstva ***letter-spacing*** moguće je podešavati razmak između pojedinih slova. Kao vrijednost predaje se širina razmaka koja se želi dodati na postojeći razmak između slova. Za mjernu jedinicu mogu se koristiti pikseli, točke, jedinice *em* ili *rem*, ali ne i postotci.

```
letter-spacing: 5px;
```

G d j e i m a d i m a i m a i v a t r e .

Osim navođenja iznosa širine, moguće je navesti i ključnu riječ *normal*, čime se razmak postavlja na standardni razmak u trenutačnom fontu. Tako se za neki element može isključiti djelovanje drugog CSS-pravila koje postavlja ovo svojstvo.

```
letter-spacing: normal;
```

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

Razmak između riječi

Također možemo podešavati i razmak između riječi, upotrebom svojstva ***word-spacing***. Kao vrijednost predajemo željenu širinu razmaka koju možemo definirati u pikselima, točkama, *em* ili *rem* jedinicama (ali ne i u postotcima).

```
word-spacing: 10px;
```

G d j e i m a d i m a i m a i v a t r e .

Na ovom se svojstvu također može navesti ključna riječ *normal*, čime se razmak postavlja na standardni razmak u trenutačnom fontu.

```
word-spacing: normal;
```

Zanimljivosti i napomene

Ako se definira širina za koju se tekst uvlači preko postotaka, radić će se o postotcima od širine elementa u kojem se nalazi tekst.

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

Visina retka

Visina retka u kojem se nalazi tekst može se mijenjati pomoću svojstva ***line-height***. Kao vrijednost postavlja se željeni iznos visine. Mogu se koristiti pikseli, točke, jedinice *em* ili *rem*, a također se mogu koristiti i postotci ili broj bez mjerne jedinice.

```
line-height: 30px;
```

Kralj dade toga čovjeka pitati, da li bi mu htio dati svoga sina. On privoli. Kralj odvede njega u svoj dvor. Ubrzo mu omili, a kralj ga je ljubio kao rođeno dijete.

Ako se koriste jedinice *em*, postotci ili broj bez mjerne jedinice, vrijednost visine se računa na temelju veličine fonta elementa u kojem se tekst nalazi. U ovom primjeru ona bi bila dvostruka od veličine fonta na roditeljskom elementu:

```
line-height: 2em;
```

Ista visina retka dobit će se i ako se izostavi jedinica *em*:

```
line-height: 2;
```

Kao i kod jedinica *em*, broj koji se navodi može biti decimalni broj:

```
line-height: 1.5;
```

Na ovom se svojstvu također može navesti ključna riječ *normal*, čime se visina retka postavlja na predefiniranu vrijednost, koja ovisi o pregledniku.

```
line-height: normal;
```

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

2.6. Velika i mala slova

Velika i mala slova

Iz estetskih razloga ponekad je potrebno sav tekst u nekom elementu (npr. naslovi, navigacija) ispisati velikim slovima. Da bi se to postiglo, ne mora se ručno mijenjati tekst u HTML-datoteci, već se može koristiti CSS-svojstvo ***text-transform***.

```
text-transform: uppercase;
```

Ključna riječ *uppercase* prikazat će sva slova velikima. U tablici je dan popis svih raspoloživih ključnih riječi (uz primjere).

Zanimljivosti i napomene

Postoji suptilna razlika između korištenja jedinice *em* i broja bez jedinice.

U slučaju kad se vrijednost svojstva *line-height* nasljeđuje od roditeljskog elementa, koriste li se jedinice *em* (ili postotci), one će se računati na temelju veličine fonta roditeljskog elementa, a koriste li se brojevi bez jedinice, oni će se računati na temelju veličine fonta elementa u kojem se tekst nalazi.

Ključna riječ	Objašnjenje	Primjer
<i>uppercase</i>	sva velika slova	SVI PUTOVI VODE U RIM.
<i>lowercase</i>	sva mala slova	svi putovi vode u rim.
<i>capitalize</i>	svaka riječ započinje velikim slovom	Svi Putovi Vode u Rim.
<i>none</i>	isključuje djelovanje drugog pravila	Svi putovi vode u Rim.

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

Prikaz teksta *small-caps*

Uz ovo svojstvo, postoji i svojstvo **font-variant**, kojim se postavlja specifičan prikaz teksta, u kojem su sva slova velika, ali su „prava” velika slova prikazana većima od drugih:

```
font-variant: small-caps;
```

Ovo se svojstvo može postaviti na jednu od dvije ključne riječi:

Ključna riječ	Objašnjenje	Primjer
<i>normal</i>	standardni prikaz fonta	Svi putovi vode u Rim.
<i>small-caps</i>	<i>small-caps</i> prikaz fonta	SVI PUTOVI VODE U RIM.

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

2.7. Ukrašavanje teksta

Crte

Pomoću CSS-a tekst se može podcrtati, precrtati ili se može staviti crta iznad teksta. Za to se koristi svojstvo **text-decoration**:

```
text-decoration: underline;
```

Kao vrijednost svojstva navodi se ključna riječ, a moguće su ove ključne riječi:

Ključna riječ	Objašnjenje	Primjer
<i>underline</i>	podcrtani tekst	<u>Svi putovi vode u Rim.</u>
<i>overline</i>	tekst s linijom iznad	<u>Svi putovi vode u Rim.</u>
<i>line-through</i>	precrtani tekst	Svi putovi vode u Rim.
<i>none</i>	isključuje djelovanje drugog pravila	Svi putovi vode u Rim.

Vrijednost se ovog svojstva ne nasljeđuje s roditeljskog elementa.

Zanimljivosti i napomene

Od navedenih ključnih riječi najčešće se koristi *underline* (za linkove, o čemu će biti više riječi u poglavlju 5).

Umjesto samo jedne, kod svojstva *text-decoration* možemo postaviti više ključnih riječi, npr:

```
text-decoration: underline overline;
```

U ovom primjeru postižemo kombinaciju dvaju efekata (crta ispod i crta iznad teksta):

Svi putovi vode u Rim.

Sjena

U novijim preglednicima tekstu se može dodati sjenu, što se postiže svojstvom ***text-shadow***:

```
text-shadow: gray 5px 5px 5px;
```

Vrijednost se ovog svojstva postavlja navođenjem četiriju parametara:

- boja sjene
- horizontalni pomak
- vertikalni pomak
- radijus zamućenja.

Boja sjene može se postaviti pomoću imena boje, pomoću heksadecimalnog ili kôdova RGBA i HSLa, a ostale tri vrijednosti u jedinicama duljine (pikselima i drugima).

Vodoravni i okomiti pomak određuju za koliko će sjena biti pomaknuta u odnosu na tekst.

Radijus zamućenja je duljina za koju je tekst raširen, što stvara efekt zamućenja. Taj se parametar može izostaviti i tada će njegova vrijednost biti 0.

Bez navođenja radijusa zamućenja dobivamo jednoliku boju sjene:

```
text-shadow: gray 5px 5px;
```

KLIN SE KLINOM IZBIJA.

No navede li se radijus zamućenja, sjena je zamućena, što može izgledati efektnije:

```
text-shadow: gray 5px 5px 5px;
```

KLIN SE KLINOM IZBIJA.

Na tekst se istovremeno može postaviti više sjena, ako se definicija svake sjene odvoji zarezom:

```
text-shadow: lightgray 3px 3px,
            gray 6px 6px,
            black 9px 9px;
```

Zanimljivosti i napomene

Svojstvo *text-shadow* podržano je u većini novijih preglednika, a Internet Explorer ga podržava tek od verzije 10.

Zanimljivosti i napomene

Parametri koji se navode kao vrijednost svojstva *text-shadow* mogu se navesti i ovim redoslijedom:

- horizontalni pomak
- vertikalni pomak
- radijus zamućenja
- boja.

I u ovom se načinu pisanja može izostaviti radijus.

KLIN SE KLINOM IZBIJA.

Ako se za pomak sjene navede negativna vrijednost, sjena će biti smještena s lijeve, odnosno s gornje strane slova. U ovom je primjeru dodana bijela sjena s lijeve i s gornje strane, čime je postignut efekt odsjaja:

```
text-shadow: lightgray -1px -1px,
            black 5px 5px;
```



Da bi se efekt odsjaja vidio, u primjeru je dodana i siva pozadina.

Za vrijednost tog svojstva može se postaviti i ključna riječ *none*. To se rabi kada se želi isključiti sjena koja je postavljena nekim drugim CSS-pravilom.

```
text-shadow: none;
```

Vrijednost se ovog svojstva također nasljeđuje s roditeljskog elementa.

2.8. Skraćeni način navođenja svojstava fonta

Osnovna svojstva za oblikovanje fonta mogu se postaviti na skraćeni način korištenjem samo jednog svojstva, svojstva **font**.

Svojstva su koja se mogu tako postaviti:

- *font-style*
- *font-variant*
- *font-weight*
- *font-size*
- *line-height*
- *font-family*.

Vrijednosti se navodi ovako:

```
font: style variant weight size/line-height family
```

Vrijednosti treba navesti gore navedenim redoslijedom, a neka se svojstva mogu i izostaviti. Obavezna su samo svojstva *font-size* i *font-family* (masno označena).

U ovom primjeru postaviti će se sva moguća svojstva:

```
font: italic small-caps bold 14px/20px Arial;
```

Sve su vrijednosti odvojene razmakom, osim vrijednosti svojstava *font-size* i *line-height*, koje su odvojene kosom crtom. Ako se izostavi vrijednost za *line-height*, izostavit će se i kosa crta:

```
font: italic small-caps bold 14px Arial;
```

Zanimljivosti i napomene

Ako se na jednom mjestu u CSS-u navede *font-weight: bold*, a zatim kasnije navede *font: 14px Arial*, deklaracija koja je došla kasnije imat će prednost i težina fonta bit će *normal* (jer je izostavljanjem u drugoj deklaraciji postavljena na *normal*).

Kao što je već rečeno, navesti treba samo svojstva *font-size* i *font-family*. Ovakvo bi izgledalo najjednostavnije korištenje svojstva *font*:

```
font: 14px Arial;
```

Vrijednosti koje nisu navedene, bit će postavljene na vrijednost *normal*.

Kao vrijednost za *font-family* moguće je navesti više naziva fontova i tada ih odvajamo zarezom (i opcionalno razmakom):

```
font: 14px Arial, Verdana, sans-serif;
```

2.9. Korištenje web-fontova

Korištenje fontova u CSS-u dugo je vremena bilo ograničeno samo na fontove koji su instalirani kod korisnika na računalu. Tu nažalost nije bilo mjesta za neke egzotičnije fontove, već su se dizajneri *web*-stranica morali ograničiti na tek desetak fontova koje su mogli koristiti relativno sigurno.

U posljednje vrijeme ta se situacija promijenila pa se danas, osim fontova instaliranih na korisnikov računalu, mogu koristiti i tzv. *web*-fontovi. Datoteka s fontom postavlja se na *web*-poslužitelj, a korisnikov preglednik preuzima tu datoteku i prikazuje tekst u željenom fontu.

Za uključivanje *web*-fonta potrebno je koristiti se specijalnim pravilom **@font-face**:

```
@font-face
{
    font-family: Fenix;
    src: url('Fenix.woff');
}
```

Tamo gdje se inače nalazi selektor navodi se znak @ i ključna riječ *font-face*, a u vitičastim se zagradama navode svojstva kojima definiramo font (u ovom primjeru samo *font-family* i *src*, a u složenijim primjerima moguća su i druga svojstva).

Svojstvom *font-family* definira se koji će se naziv koristiti za ovaj font dalje u CSS-kôdu (u ovom primjeru *Fenix*), a pomoću svojstva *src* navodi se putanja do datoteke fonta (*Fenix.woff*) kojom ćemo se koristiti. Ta se putanja predaje preko CSS-funkcije *url*.

Tamo gdje se želi navesti taj font, upotrijebit će se njegov naziv koji je prije naveden (uz obavezno navođenje zamjenskih fontova).

```
p
{
    font-family: Fenix, Garamond, serif;
}
```

Zanimljivosti i napomene

Fontovi su autorska djela, pa prilikom korištenja *web* fontova, obavezno treba voditi računa o licenci pod kojom se koristimo fontom. Ako je font instaliran na našem računalu, imamo samo licencu koja nam dozvoljava da ga koristimo na našem računalu.

Ako se font s našeg računala objavljuje na *webu* (kao font datoteka, što je nužno za korištenje *web* fontova), time se krše autorska prava, jer ga distribuiramo drugim korisnicima (posjetiteljima naše *web* stranice). Stoga, za takvo korištenje mora postojati posebna licencu.

U slučaju kad se žele koristiti inačice fonta *bold* ili *italic*, mogla bi se koristiti svojstva *font-style* i *font-weight*. No da bi rezultat izgledao podjednako dobro u svim preglednicima, bolji pristup je upotrijebiti zasebnu font-datoteku s inačicama fonta *bold* odnosno *italic*, koja se onda mora uključiti pomoću odvojenog pravila `@font-face`:

```
@font-face
{
  font-family: FenixBold;
  src: url('FenixBold.woff');
}
```

Za *web*-fontove postoji više tipova font-datoteka. Format WOFF (*Web Open Font Format*) standard je koji podržavaju najnoviji preglednici, no da bi *web*-fontovi radili za većinu korisnika, potrebno je font pripremiti u više različitih formata i sve ih navesti u pravilu `@font-face`.

U sljedećoj tablici popisani su mogući formati za *web*-fontove i njihova podržanost u različitim preglednicima:

Format	Preglednici koji ga podržavaju
EOT	Internet Explorer 8
TTF	Firefox 21, Chrome 21, Safari 5.1, Opera 12.1
OTF	Firefox 21, Chrome 21, Safari 5.1, Opera 12.1
SVG	Chrome 27, Safari 5.1, Opera 23, mobilni preglednici
WOFF	Internet Explorer 9, Firefox 21, Chrome 21, Safari 5.1, Opera 12.1, mobilni preglednici

Problem licence, kao i problem s velikim brojem potrebnih formata i pouzdanim CSS-kôdom koji se koristi trikovima da bi riješio probleme s nekim preglednicima, rješavaju posebna *web*-sjedišta namijenjena korištenju *web*-fontova.

Neka od tih *web*-sjedišta nude besplatne fontove, a neka uz pretplatu prodaju licencu za njihovo korištenje na *webu*. Prilikom preuzimanja fonta na većini njih dobijemo font-datoteke za sve preglednike i gotov CSS-kôd koji radi u svim preglednicima. Najpoznatija su *web*-sjedišta za tu namjenu www.fontsquirrel.com i www.google.com/fonts.

2.10. Vježba: Oblikovanje teksta

1. U mapi C220/Vjezbe/css otvorite datoteku *tekst.css*. Rezultat napisanih CSS-pravila možete vidjeti pregledavanjem datoteka *index.html* („Početna“) i *tecajevi.html* („Tečajevi“) u pregledniku.
2. Napišite pravilo za element *body*, kojim će se za vrstu fonta (*font-family*) postaviti font *Helvetica*. Primijetite da se promijenila vrsta fonta za sve elemente, jer su svi elementi djeca elementa *body* pa su naslijedili postavljenu vrijednost.
3. Kao zamjenske fontove, u slučaju da *Helvetica* nije instalirana, postavite *Arial* i preglednikov predefimirani font bez serifa.
4. Unutar pravila za element *body* dodajte i deklaraciju za postavljanje visine retka (*line-height*) na vrijednost 1.3.
5. Unutar pravila za element *body* postavite veličinu fonta (*font-size*) na 14 piksela.
6. Za elemente *h2* postavite veličinu fonta na 18 piksela. Također, svim elementima *h2* postavite boju na tamnocrvenu (*#c30f10*) i svojstvo *text-transform* tako da tekst unutar njih bude prikazan velikim slovima.
7. Za elemente *h3* postavite veličinu fonta na 16 piksela.
8. Za sve linkove (elemente *a*) postavite boju teksta na tamnocrvenu (*#c30f10*). Pomoću svojstva *text-decoration* maknite crtu kojom je podcrtan svaki link.
9. Veličinu fonta za linkove koji se nalaze u elementu s klasom *navigacija* postavite na 16 piksela. Također im postavite i svojstvo *text-transform* tako da tekst unutar njih bude prikazan velikim slovima.
10. Veličinu fonta za linkove koji se nalaze u elementu s klasom *podnozje* postavite na 13 piksela. Tim linkovima postavite boju teksta na svijetlosivu (*#e6e6e6*). Također postavite svojstvo *text-decoration* tako da njihov tekst bude podcrtan.
11. Elementu s klasom *selected* (odabrani link u navigaciji) također postavite boju teksta na svijetlosivu (*#e6e6e6*).



Dodatni zadatci

12. Veličinu fonta za element *p* koji se nalazi u elementu s klasom *zaglavlje* postavite na 12 piksela. Boju teksta u tom elementu postavite na tamnosivu (#6d6c6c). Stil teksta postavite na *italic*.
13. Za sve elemente *strong* postavite boju teksta na tamnosivu (#6d6c6c).
14. Za element *h3* koji se nalazi u elementu s klasom *kontakt* postavite boju teksta na tamnosivu (#6d6c6c). Tom elementu postavite i svojstvo *text-transform* tako da tekst unutar njega bude prikazan velikim slovima
15. Veličinu fonta za element s klasom *rss* postavite na 13 piksela.
16. Veličinu fonta u elementu s klasom *tecaj* (koji se koristi na stranici „Tečajevi“) postavite na 13 piksela.
17. Za element *h3* koji se nalazi u elementu s klasom *tecaj* postavite boju teksta na tamnocrvenu (#c30f10).
18. Za element *em* koji se nalazi u elementu s klasom *tecaj* postavite boju teksta na tamnosivu (#6d6c6c).
19. Definirajte pravilo *@font-face* za font *NeoSans*. (Datoteka fonta se nalazi u mapi C220/Vjezbe/fontovi, pa je putanja koju je potrebno navesti ova: `../fontovi/NeoSans.woff`). Taj font dodajte u deklaraciju kojom se postavlja vrsta fonta za element *body* kao prvi font po redu.
20. Definirajte pravilo *@font-face* za font *NeoSansMedium* (također se nalazi u mapi C220/Vjezbe/fontovi). Taj font postavite ćemo u pravilo za elemente *h2*, *h3* i *strong* te za linkove u elementu s klasom *navigacija*. Kao zamjenske fontove navest ćemo *Helvetica*, *Arial* i predefimirani font bez serifa.



3. Oblikovanje elemenata

Svi HTML-elementi zapravo su pravokutnici, odnosno zauzimaju pravokutnu površinu unutar ekrana preglednika. U ovom poglavlju objasnit će se kako se tim pravokutnicima postavljaju boja, veličina, obrub i druga vizualna svojstva.

3.1. Boja pozadine

Da bi HTML-elementi bili vidljivi, može im se postaviti boja (odnosno boja pozadine). Za postavljanje boje pozadine koristi se svojstvo **background-color**. Kao i kod svojstva *color*, vrijednosti se mogu postaviti na više načina. Može se koristiti naziv boje:

```
background-color: red;
```

Najčešće je korišten način postavljanja vrijednosti, kao i kod svojstva *color*, heksadecimalni kôd:

```
background-color: #f5412a;
```

Umjesto heksadecimalnog zapisa mogu se rabiti i RGB-vrijednosti:

```
background-color: rgb(245, 65, 42);
```

Ako je potrebno, moguće je postaviti i prozirnost i to tako da se navede vrijednost komponente *a* (kao decimalni broj između 0 i 1):

```
background-color: rgba(245, 65, 42, 0.5);
```

Također se za odabir boje može rabiti i model HSL:

```
background-color: hsl(7, 91%, 56%);
```

U njemu se također može postaviti prozirnost:

```
background-color: hsl(7, 91%, 56%, 0.5);
```

3.2. Prikaz elemenata

Prema načinu prikaza u HTML-u postoje dvije osnovne vrste elemenata.

Prva vrsta su **blok-elementi** (*block elements*). Oni zauzimaju cijeli redak u kojem se nalaze. Moguće im je odrediti visinu i širinu. Čak i ako im se smanji širina, oni će i dalje zauzimati cijeli red tako da sljedeći element mora započeti u novom redu. Primjeri takvih elemenata su *div*, *p* i *h1*.

div

p

h1

Druga vrsta su **linijski elementi** (*inline elements*). Ti elementi su prikazani unutar trenutačne linije teksta. Oni ne zauzimaju cijeli red nego onoliko prostora koliko zauzima njihov sadržaj (štoviše, njihova se širina i visina ne mogu postaviti). Primjeri su takvih elemenata *span*, *em* i *strong*:

span

em

strong

Pomoću CSS-svojstva **display** moguće je promijeniti tip prikaza na elementu. Postavi li se blok-elementima to svojstvo na *inline*, oni će se ponašati kao da su linijski elementi:

```
div, p, h1
{
    display: inline;
}
```

div

p

h1

Postavi li se linijskim elementima vrijednost svojstva *display* na *block*, oni će se ponašati kao da su blok-elementi:

```
span, em, strong
{
    display: block;
}
```

span

em

strong

Mijenjanje tipa prikaza na pojedinom elementu ponekad je potrebno, ako se neki element želi prikazati drugačije.

Zanimljiva vrijednost na koju se može postaviti svojstvo *display* je *inline-block*, čime se mijenja ponašanje blok-elementa tako da on više ne zauzima cijeli redak. Takav element zadržava druga svojstva blok-elementa – mogu mu se definirati visina i širina, što je zgodno za pozicioniranje elemenata, a o tome će više riječi biti u zasebnom poglavlju.

Osim blok-elemenata i linijskih elemenata, u HTML-u postoje i neki elementi koji prema tipu prikaza zapravo ne pripadaju ni jednoj od tih dviju skupina. To su stavke liste (elementi *li*) te tablice i elementi tablice.

Evo popisa vrijednosti na koje se može postaviti svojstvo *display*:

Ključna riječ	Objašnjenje
<i>none</i>	element neće biti vidljiv
<i>inline</i>	element se prikazuje kao linijski element
<i>block</i>	element se prikazuje kao blok-element
<i>inline-block</i>	element zadržava svojstva blok-elementa (moguće mu je postaviti širinu i visinu), no ne zauzima cijeli redak
<i>list-item</i>	element se prikazuje kao element <i>li</i>
<i>table</i>	element se prikazuje kao element <i>table</i> (slično blok-elementu)
<i>inline-table</i>	element se prikazuje kao element <i>table</i> (ali s obilježjima linijskog elementa)
<i>table-caption</i>	element se prikazuje kao element <i>caption</i>
<i>table-cell</i>	element se prikazuje kao element <i>td</i>
<i>table-column</i>	element se prikazuje kao element <i>col</i>
<i>table-column-group</i>	element se prikazuje kao element <i>colgroup</i>
<i>table-footer-group</i>	element se prikazuje kao element <i>tfoot</i>
<i>table-header-group</i>	element se prikazuje kao element <i>thead</i>
<i>table-row</i>	element se prikazuje kao element <i>tr</i>
<i>table-row-group</i>	element se prikazuje kao element <i>tbody</i>
<i>flex</i>	element se prikazuje kao blok-element u načinu pozicioniranja <i>flexbox</i>
<i>inline-flex</i>	element se prikazuje kao linijski element u načinu pozicioniranja <i>flexbox</i>

3.3. Širina i visina

Ako se širina i visina HTML-elementa ne postave eksplicitno, njegova širina i visina ovisit će ili o sadržaju koji se nalazi unutar njega ili o roditeljskom elementu unutar kojeg se nalaze.

Širina

Širina se može podešavati samo kod blok-elemenata (i elemenata *inline-block*). Ako im se ne postavi određena širina, blok-elementi će zauzeti svu širinu roditeljskog elementa.

S druge strane, širina je linijskog elementa određena njegovim sadržajem. Postavljanje širine na tim elementima neće imati nikakav učinak.

Blok-elementi inicijalno zauzimaju svu širinu roditeljskog elementa. Za definiranje širine elementa koristi se svojstvo ***width***. Vrijednost tog svojstva postavlja se u jednoj od mjernih jedinica duljine (pikseli, točke, jedinice *em* ili *rem*) ili u postotcima.

```
width: 400px;
```

Svojstvo *width* može se postaviti i na vrijednost *auto*, što je inicijalna vrijednost tog svojstva. Ta ključna riječ predstavlja vrijednost koju preglednik sam dodjeljuje elementu (kod blok-elemenata to će biti širina roditeljskog elementa). Ključna riječ *auto* eksplicitno se navodi samo u slučajevima kad se želi zaobići neko drugo CSS-pravilo koje je postavilo širinu na neku vrijednost.

Ako se vrijednost širine navede u postotcima, širina se računa kao postotak širine roditeljskog elementa. Postavljanjem širine elemenata u postotcima, može se postići da se ona se automatski prilagođava količini raspoloživog prostora.

```
width: 50%;
```

Smanjuje li se prozor preglednika, moguće je primijetiti da se i širina ovako definiranog elementa smanjuje.

Minimalna širina

No često se takvo smanjivanje širine elementa želi ograničiti. Pomoću svojstva ***min-width*** moguće je definirati minimalnu širinu koju element mora zauzimati:

```
min-width: 200px;
```

Vrijednost minimalne širine također se može postaviti u mjernim jedinicama za duljinu ili u postotcima. Inicijalna vrijednost za ovo svojstvo je 0.

Maksimalna širina

Osim minimalne širine elementu se može definirati i maksimalna širina koju smije zauzeti. Maksimalnu širinu postavljamo pomoću svojstva ***max-width***:

```
max-width: 400px;
```

Vrijednost maksimalne širine također se može postaviti u mjernim jedinicama za duljinu ili u postotcima.

Inicijalna vrijednost za ovo svojstvo je *none*, tj. maksimalna širina nije postavljena.

Zanimljivosti i napomene

Svojstva *min-width*, *max-width*, te *min-height* i *max-height* nisu podržana u Internet Exploreru 6.

Visina

Visinu također možemo podešavati samo kod blok-elemenata (i elemenata *inline-block*).

Visina se postavlja pomoću svojstva **height**. Vrijednost za to svojstvo također se može postaviti u mjernim jedinicama za duljinu ili u postotcima.

```
height: 200px;
```

Ako visina elementa nije postavljena, imat će vrijednost *auto*, što znači da će je preglednik sam izračunati na temelju visine sadržaja koji se nalazi unutar elementa (npr. teksta ili slike).

Ako je vrijednost visine postavljena u postotcima, visina će biti izračunata kao postotak visine roditeljskog elementa. Dakle, tako se može postići da se visina prilagođava količini raspoloživog prostora:

```
height: 50%;
```

Minimalna visina

Minimalnu visinu elementa moguće je, kao i minimalnu širinu, postaviti kada se želi ograničiti smanjivanje visine elementa zajedno s količinom raspoloživog prostora.

Na minimalna se visina najčešće koristi kad postoji dinamički sadržaj čija visina može varirati. U takvom se slučaju ne želi ograničiti visina elementa, nego se želi da se ona automatski prilagodi visini sadržaja. Najčešće je međutim poželjno da element ima neku minimalnu visinu u slučaju kad korisnik unese malu količinu teksta.

Tada se neće postaviti svojstvo *height* na elementu (čime će ono imati vrijednost *auto*), već će se postaviti samo minimalna visina pomoću svojstva **min-height**.

```
min-height: 100px;
```

Vrijednost minimalne visine također se može postaviti u mjernim jedinicama za duljinu ili u postotcima. Inicijalna vrijednost za to svojstvo je 0, a od ključnih se riječi može rabiti samo ključna riječ *inherit*.

Maksimalna visina

I kod visine se elementu može definirati maksimum koji smije zauzeti. Maksimalna se visina postavlja pomoću svojstva **max-height**.

```
max-height: 500px;
```

Vrijednost maksimalne visine može se također postaviti u mjernim jedinicama za duljinu ili u postotcima.

Inicijalna vrijednost za to svojstvo je *none*, tj. maksimalna širina nije postavljena.

3.4. Padding

Iz estetskih je razloga potreban određeni razmak između ruba elementa i teksta (ili drugog sadržaja) koji se nalazi u elementu. Taj razmak postavlja se korištenjem svojstva ***padding***.

Zanimljivosti i napomene

Postavi li se prevelik gornji *padding* kod linijskog (*inline*) elementa, to će uzrokovati pomicanje teksta izvan granica elementa.

Padding se može postaviti i blok-elementima i linijskim elementima.

Inicijalno *padding* na elementu nije postavljen (njegova vrijednost je 0):

Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo.

Svojstvu *padding* predaje se vrijednost s mjernom jedinicom za duljinu ili vrijednost u postotcima (u slučaju korištenja postotka veličina *paddinga* računa se na temelju širine roditeljskog elementa).

```
padding: 10px;
```

Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo.

Vrijednost svojstva *padding* ne može se naslijediti s roditeljskog elementa.

Ako se svojstvu *padding* navede samo jedna vrijednost, gornji, desni, donji i lijevi *padding* dobit će istu vrijednost.

Ako je potrebno postaviti različite veličine *paddinga* na ta četiri mjesta, navedu se četiri različite vrijednosti. Vrijednosti se navode u smjeru kazaljke na satu. Prva je vrijednost za gornji položaj, zatim desni pa donji i na kraju lijevi.

```
padding: 5px 10px 15px 30px;
```

Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo.

Mogu se navesti i tri vrijednosti. U tom se slučaju prva vrijednost opet odnosi na gornji položaj, druga na desni, a treća na donji. Vrijednost koja nedostaje (lijeva) bit će jednaka nasuprotnoj vrijednosti (desnoj):

```
padding: 5px 10px 15px;
```

Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo.

Ako se navedu samo dvije vrijednosti, prva vrijednost se odnosi na gornji položaj, a druga na desni. Donja će vrijednost biti jednaka gornjoj, a lijeva će biti jednaka desnoj:

```
padding: 5px 20px;
```

Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo.

Čini se potpuno nemogućim zapamtiti na koji se položaj odnosi koja vrijednost, ali ovdje pomaže mnemotehnička kratica **TRBL** (koju pamtimo pomoću engleske riječi *trouble*).

Slova TRBL predstavljaju pojedine položaje i pomažu da se zapamti njihov redosljed. Kada imamo četiri vrijednosti, prva vrijednost se odnosi na gornji položaj (**T**op), druga na desni (**R**ight), treća na donji (**B**ottom), a četvrta na lijevi (**L**eft). Zapravo, kreće se u smjeru kazaljke na satu, a započinje se s gornjim položajem.

U slučaju kad postoje tri vrijednosti, prva se ponovo odnosi na gornji položaj (**T**op), druga na desni (**R**ight), treća na donji (**B**ottom). Četvrta vrijednost (**L**eft) ne postoji, a kad vrijednost za lijevi položaj nije navedena, on će imati jednaku vrijednost kao i desni.

U slučaju kad postoje samo dvije vrijednosti, prva se odnosi na gornji položaj (**T**op), a druga na desni (**R**ight). Vrijednosti za **Bottom** i **Left** nisu navedene pa će lijevi položaj imati istu vrijednost kao i desni, a donji istu kao i gornji.

Svojestvo *padding* zapravo je skraćeni način postavljanja vrijednosti za četiri odvojena svojstva koja postoje za svaki pojedinačni položaj: ***padding-top***, ***padding-right***, ***padding-bottom***, ***padding-left***.

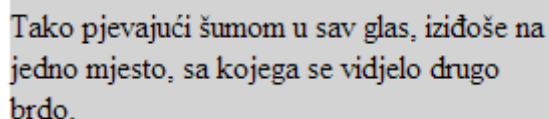
Primjer u kojem su postavljene četiri vrijednosti svojstvu *padding* mogao bi se napisati i ovako:

```
padding-top: 5px;
padding-right: 10px;
padding-bottom: 15px;
padding-left: 30px;
```

No ako se u ovakvom načinu pisanja izostave neki od položaja, oni neće imati postavljenu vrijednost (tj. imat će postavljenu inicijalnu vrijednost – 0).

```
padding-top: 5px;
padding-right: 10px;
```

Ako se navedu samo dvije gornje deklaracije, dobit će se ovakav rezultat (donji i lijevi položaj bit će bez *paddinga*):

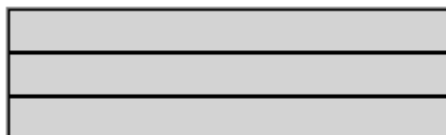
A screenshot showing the text "Tako pjevajući šumom u sav glas, iziđoše na jedno mjesto, sa kojega se vidjelo drugo brdo." The text is left-aligned and has no padding on the left or right sides, appearing to touch the edges of the container.

3.5. Margina

Između HTML-elemenata moguće je definirati razmak, odnosno marginu. Neki elementi (npr. elementi *p* i *h1*) imaju inicijalno postavljenu marginu, dok je drugi (elementi *div*) nemaju.

Margina se može postavljati i blok-elementima i linijskim elementima, s tim da je kod linijskih elemenata moguće postaviti samo lijevu i desnu marginu, dok postavljanje gornje i donje margine nema učinka.

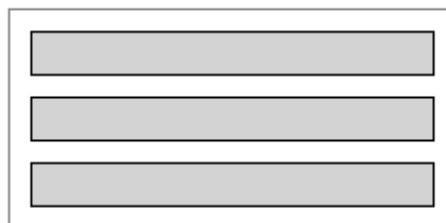
Elementi bez margina međusobno se dodiruju:



Margina elementa postavlja se pomoću svojstva ***margin***, kojem se kao vrijednost postavlja iznos u mjernim jedinicama duljine ili u postotcima. Ako se koriste postotci, oni se odnose na širinu roditeljskog elementa.

```
margin: 10px;
```

Postavljanjem margine dobivamo ovakav učinak:

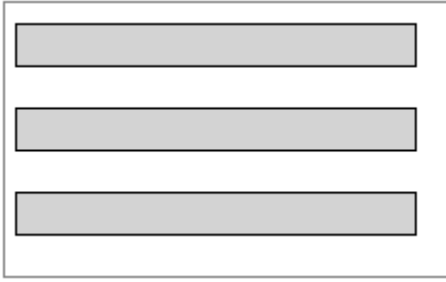


Oko svakog elementa sad postoji razmak od 10 piksela.

Prilikom postavljanja margine moguće je rabiti i ključnu riječ *auto*, koja znači da će preglednik sam dodijeliti vrijednost margine. Za gornju, desnu, donju i lijevu marginu moguće je navesti različite vrijednosti:

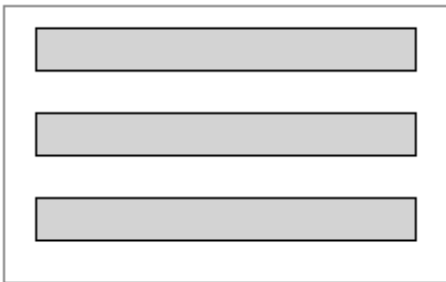
```
margin: 10px 15px 20px 5px;
```

Ovako navedene vrijednosti odnose se na gornju, desnu, donju i lijevu marginu (u smjeru kazaljke na satu, počevši od gornje margine).



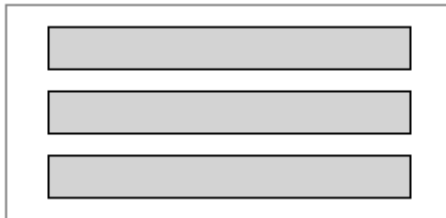
Ako se navedu samo tri vrijednosti, prva vrijednost će se odnositi na gornju marginu, druga na desnu, treća na donju, a lijeva će margina biti jednaka desnoj:

```
margin: 10px 15px 20px;
```



Ako se navedu samo dvije vrijednosti, prva vrijednost odnosit će se na gornju marginu, a druga na desnu. Donja margina bit će jednaka gornjoj, a lijeva će biti jednaka desnoj:

```
margin: 10px 20px;
```



Postoje i svojstva pomoću kojih se odvojeno mogu postaviti veličine pojedinih margina. To su svojstva ***margin-top***, ***margin-right***, ***margin-bottom*** i ***margin-left***.

Primjer u kojem su odvojeno postavljene četiri različite vrijednosti izgledao bi ovako:

```
margin-top: 10px;
margin-right: 15px;
margin-bottom: 20px;
margin-left: 5px;
```

Ako se izostavi neka od gornjih deklaracija, ta margina neće biti postavljena, nego će imati svoju inicijalnu vrijednost (koja će, ovisno o tipu elementa, biti 0 ili će imati neku vrijednost).

Zanimljivosti i napomene

Gornja se i donja margina susjednih elemenata spajaju. U primjeru su se spojile donja margina prvog elementa i gornja margina drugog elementa. Razmak između prvog i drugog elementa nije 30px (20px + 10px) nego samo 20px, tj. ostaje prisutan samo razmak od veće margine.

Kod lijeve i desne margine to nije slučaj. Kad bi postojao susjedni desni element, susjedne vertikalne margine ne bi se spojile, već bi ukupni razmak bio zbroj lijeve i desne margine.

3.6. Rubovi


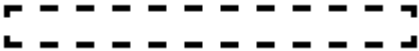
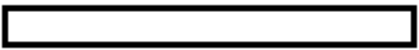





Na HTML-elementima može se pomoću CSS-a iscrtati rub. U tu se svrhu rabi više svojstava CSS-a s prefiksom *border-*.

Stil ruba

Izgled, odnosno stil ruba definira se pomoću svojstva ***border-style***. Kao vrijednost za to svojstvo navodi se ključna riječ. Najčešće se koristi ključna riječ *solid*, koja postavlja izgled ruba na punu crtu:

```
border-style: solid;
```

Ključne riječi koje se mogu rabiti dane su u tablici, zajedno s primjerom (ili objašnjenjem):

Ključna riječ	Primjer ili objašnjenje
<i>dotted</i>	
<i>dashed</i>	
<i>solid</i>	
<i>double</i>	
<i>groove</i>	
<i>ridge</i>	
<i>inset</i>	
<i>outset</i>	
<i>hidden</i>	rub se ne vidi (ali se njegova širina računa u dimenzije elementa)
<i>none</i>	bez ruba (inicijalna vrijednost)

Ako ovo svojstvo nije postavljeno, njegova inicijalna vrijednost je *none*. Drugim riječima, HTML-elementi inicijalno nemaju rub, dok im se ne postavi vrijednost tog svojstva na neku od ključnih riječi.

Vrijednost se ovog svojstva ne nasljeđuje od roditeljskog elementa.

Ako se svojstvu *border-style* postavi samo jedna vrijednost, svi će rubovi izgledati jednako.

No ako mu se postave dvije vrijednosti, prva će se odnositi na gornji i donji, a druga na lijevi i desni rub:

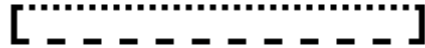
```
border-style: dotted solid;
```



U primjeru je vidljivo da su gornji i donji rub istočkani, dok su desni i lijevi rub prikazani punom crtom.

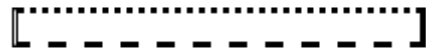
Ako se ovom svojstvu predaju tri vrijednosti, prva vrijednost odnosit će se na gornji, druga na desni i lijevi, a treća na donji rub.

```
border-style: dotted solid dashed;
```



Ako se predaju četiri vrijednosti, za svaki od rubova specificirat će se različita vrijednost. Prva vrijednost odnosit će se na gornji, druga na desni, treća na donji, a četvrta na lijevi rub.

```
border-style: dotted solid dashed double;
```



Svojstvo *border-style* zapravo je skraćeni način postavljanja vrijednosti za četiri odvojena svojstva koja postoje za svaki pojedinačni rub:

border-top-style, ***border-right-style***, ***border-bottom-style***, ***border-left-style***.

Posljednji primjer, u kojem su postavljene različite vrijednosti na svaki od četiri ruba, mogao bi se napisati i ovako:

```
border-top-style: dotted;
border-right-style: solid;
border-bottom-style: dashed;
border-left-style: double;
```

No ako se u ovakvom načinu pisanja izostave neke od vrijednosti, ti rubovi neće imati postavljenu vrijednost (tj. imat će inicijalnu vrijednost *none*).

```
border-top-style: dotted;
border-right-style: solid;
```

Kad bi bile navedene samo dvije gornje deklaracije, dobio bi se ovakav rezultat:



Širina ruba




Širina ruba postavlja se pomoću svojstva ***border-width***. Kao vrijednost može se predati ključna riječ ili vrijednost u nekoj od jedinica duljine (pikseli, jedinice *em* i *rem*, točke). Širina ruba ne može se navesti pomoću postotaka.

Ovako se postavlja širinu ruba pomoću ključne riječi:

```
border-width: thick;
```

Postavljanjem samo ovog svojstva rub elementa neće biti vidljiv (mora se postaviti i svojstvo *border-style*).

Ključne riječi koje se mogu rabiti dane su u tablici:

Ključna riječ	Primjer ili objašnjenje
<i>thin</i>	
<i>medium</i>	
<i>thick</i>	

Ako vrijednost nije postavljena, inicijalna je vrijednost ovog svojstva *medium*.

Vrijednost se ovog svojstva ne nasljeđuje od roditeljskog elementa.

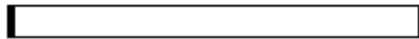
Širina koja će biti prikazana za ključne riječi *thin*, *medium* i *thick* u praksi može varirati od preglednika do preglednika pa je za najbolje rezultate bolje postaviti širinu ruba pomoću CSS-jedinica duljine.

```
border-width: 2px;
```

I kod ovog se svojstva mogu zadati različite vrijednosti na pojedinim rubovima.

Ako se postave dvije vrijednosti, prva će se vrijednost odnositi na gornji i donji, a druga na desni i lijevi rub:

```
border-width: 1px 4px;
```



Ako se postave tri vrijednosti, prva će se vrijednost odnositi na gornji, druga na desni i lijevi, a treća na donji rub:

```
border-width: 1px 4px 7px;
```



Ako se postave četiri vrijednosti, prva će se vrijednost odnositi na gornji, druga na desni, treća na donji, a četvrta na lijevi rub:

```
border-width: 1px 4px 7px 10px;
```



I svojstvo *border-width* zapravo je skraćeni način postavljanja vrijednosti za četiri odvojena svojstva koja postoje za svaki pojedinačni rub: ***border-top-width***, ***border-right-width***, ***border-bottom-width***, ***border-left-width***.

Posljednji primjer, u kojem su postavljene različite vrijednosti na svaki od četiri ruba, mogao bi se napisati i ovako:

```
border-top-width: 1px;
border-right-width: 4px;
border-bottom-width: 7px;
border-left-width: 10px;
```


No ako se u ovakvom načinu pisanja izostave neke od vrijednosti, ti rubovi neće imati postavljenu vrijednost (tj. imat će inicijalnu vrijednost *medium* pa će zbog toga ipak biti vidljivi).

```
border-top-width: 1px;
border-right-width: 4px;
```



Boja ruba

Boja ruba postavlja se pomoću svojstva ***border-color***. Kao i kod svojstava *color* i *background-color*, vrijednost se može postaviti pomoću naziva boje, heksadecimalnog ili kôda RGBa ili HSLa.

```
border-color: lightgray;
```



Postavljanjem samo ovog svojstva rub elementa neće biti vidljiv (mora se postaviti i svojstvo *border-style*).

I ovdje se mogu zadati različite vrijednosti na pojedinim rubovima.

Ako se postave dvije vrijednosti, prva će se vrijednost odnositi na gornji i donji, a druga na desni i lijevi rub:

```
border-color: lightgray black;
```



Ako se postave tri vrijednosti, prva će se vrijednost odnositi na gornji, druga na desni i lijevi, a treća na donji rub:

```
border-color: lightgray black gray;
```



Ako se postave četiri vrijednosti, prva će se vrijednost odnositi na gornji, druga na desni, treća na donji, a četvrta na lijevi rub:

```
border-color: lightgray black gray blue;
```



Svojstvo *border-color* također je skraćeni način postavljanja vrijednosti za četiri odvojena svojstva koja postoje za svaki pojedinačni rub: ***border-top-color***, ***border-right-color***, ***border-bottom-color***, ***border-left-color***.

Posljednji primjer, u kojem se postavljaju različite vrijednosti na svaki od četiri ruba, mogao bi se napisati i ovako:

```
border-top-color: lightgray;
border-right-color: black;
border-bottom-color: gray;
border-left-color: blue;
```

Ako se u ovakvom načinu pisanja izostave neke od vrijednosti, ti rubovi neće imati postavljenu vrijednost (tj. imat će inicijalnu vrijednost, a to je crna boja).

```
border-top-color: lightgray;
border-bottom-color: gray;
```



Skraćeni način pisanja pomoću svojstva *border*

Osim već spomenutih skraćenih načina pisanja, koristiti se može i svojstvo *border* pomoću kojeg se istovremeno mogu postaviti stil, širina i boja ruba.

```
border: 1px dashed gray;
```

Redoslijed navođenja vrijednosti nije važan pa bi ovaj primjer dao isti rezultat:

```
border: gray dashed 1px;
```



Kod svojstva *border* nije moguće navođenje različite vrijednosti za pojedine rubove, nego se moraju rabiti odvojena svojstva ***border-top***, ***border-right***, ***border-bottom*** i ***border-left***.

```
border-top: 1px dotted lightgray;
border-right: 4px solid black;
border-bottom: 7px dashed gray;
border-left: 10px double blue;
```



Ako se navedu samo prve dvije deklaracije, vrijednosti za druga dva ruba neće biti postavljene, nego će oni imati svoje inicijalne vrijednosti (tj. neće biti vidljivi).

```
border-top: 1px dotted lightgray;
border-right: 4px solid black;
```



Zakrivljenost uglova

HTML-elementi su pravokutnici, ali im se uglovi mogu zaobliti. Za postavljanje zaobljenih uglova koristi se svojstvo ***border-radius***, a vrijednost se može postaviti korištenjem mjerne jedinice duljine (pikseli, točke, jedinice *em* ili *rem*), a moguće je koristiti se i postotcima.

```
border-radius: 20px;
```



Zanimljivosti i napomene

U praksi se rjeđe koriste odvojena svojstva za stil, širinu i boju, nego se najčešće koristi zbirno svojstvo *border*. Kad je neki od rubova različit od drugih, koriste se svojstva *border-top*, *border-right*, *border-bottom* i *border-left*.

Zanimljivosti i napomene

Internet Explorer 8 i starije verzije tog preglednika ne podržavaju zakrivljene uglove.

Kao rezultat dobiva se jednoliko zakrivljenje po kružnici radijusa zadane duljine.



Moguće je kreirati i eliptično zakrivljenje, tako da se postave različite vrijednosti za vodoravni i okomiti radijus.



Vodoravni i okomiti radijus treba odvojiti pomoću kose crte:

```
border-radius: 20px / 10px;
```



Ako se umjesto mjernih jedinica rabe postotci, horizontalni radijus bit će izračunat kao postotak širine, a vertikalni radijus kao postotak visine elementa.

Pomoću svojstva *border-radius* moguće je definirati različite radijuse na pojedinim uglovima tako da se navedu četiri vrijednosti (bez kose crte).

```
border-radius: 30px 5px 20px 10px;
```



Prva vrijednost odnosi se na gornji lijevi ugao, a sljedeće vrijednosti se odnose na uglove koji slijede u smjeru kazaljke na satu: gornji desni, donji desni te donji lijevi ugao.

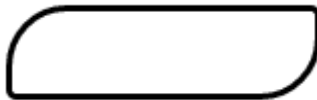
Prilikom navođenja vrijednosti radijusa mogu se navesti i samo tri vrijednosti. U tom slučaju prva vrijednost se odnosi na gornji lijevi ugao, druga na gornji desni, a treća na donji desni. Nedostaje vrijednost za donji lijevi ugao pa ona će biti jednaka vrijednosti za dijametralno suprotni ugao (gornji desni).

```
border-radius: 30px 5px 20px;
```



Ako se navedu samo dvije vrijednosti, prva vrijednost se odnosi na gornji lijevi ugao, a druga na gornji desni. Preostali uglovi imat će iste vrijednosti kao uglovi koji su im dijametralno suprotni.

```
border-radius: 30px 5px;
```



Ako se želi postići eliptično zakrivljenje, najprije se navedu vrijednosti za vodoravni radijus za sve uglove, a zatim, nakon kose crte, vrijednosti za okomiti radijus.

```
border-radius: 30px 5px 20px 10px /  
20px 15px 10px;
```



Zakrivljenost uglova može se postaviti i pomoću četiri odvojena svojstva koje se odnose na pojedine uglove: ***border-top-left-radius***, ***border-top-right-radius***, ***border-bottom-left-radius*** i ***border-bottom-right-radius***.

```
border-top-left-radius: 30px;  
border-top-right-radius: 5px;  
border-bottom-right-radius: 20px;  
border-bottom-left-radius: 10px;
```

Ako se navedu dvije vrijednosti, rezultat je eliptično zakrivljenje. Prva vrijednost predstavlja vodoravni, a druga okomiti radijus elipse (u ovom se slučaju ne rabi kosa crta za razdvajanje vrijednosti):

```
border-top-left-radius: 30px 20px;  
border-top-right-radius: 5px 15px;  
border-bottom-right-radius: 20px 10px;  
border-bottom-left-radius: 10px 15px;
```

Inicijalna vrijednost za ova svojstva je 0 (svi uglovi su pravokutni). Vrijednost se ne nasljeđuje s roditeljskog elementa.

Ako rub elementa nije postavljen (tj. ima inicijalnu vrijednost *none*), svojstvo ***border-radius*** svejedno će imati učinak na pozadinu elementa.

```
border-radius: 20px;  
background-color: lightgray;
```

Ove dvije deklaracije imat će ovakav učinak:



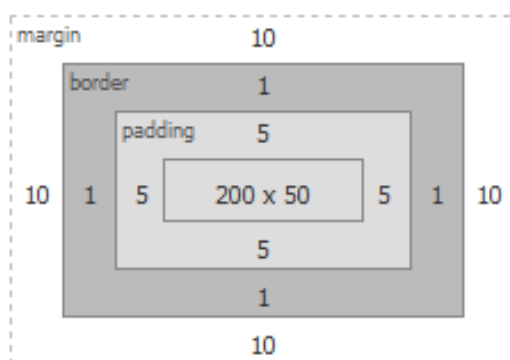
3.7. Izračun dimenzija elementa

Kad se računaju ukupne dimenzije HTML-elementa odnosno prostor koji će on zauzimati, osim postavljene visine i širine moraju se uzeti u obzir i *padding* i širina ruba elementa.

Za primjer će se rabiti ove dimenzije elementa:

```
width: 200px;
height: 50px;
padding: 5px;
border: 1px solid black;
margin: 10px;
```

Dimenzije elementa mogu se prikazati ovakvim dijagramom:



Vanjski pojas (omeđen izvana iscrtkanom linijom) prikazan na ilustraciji predstavlja margine elementa. Tamnosivi pojas predstavlja rub (*border*) elementa. Vanjski svijetlosivi pojas (do crte) predstavlja *padding* elementa, dok unutrašnji svijetlosivi pravokutnik predstavlja prostor gdje se nalazi sadržaj.

Prilikom postavljanja širine i visine elementa, potrebno je imati na umu da se svojstva *width* i *height* odnose samo na dio elementa gdje se nalazi sadržaj (unutrašnji svijetlosivi pravokutnik na ovoj ilustraciji).

Element dakle neće imati širinu od 200 piksela, nego toj širini treba dodati i širinu lijevog i desnog *paddinga* te lijevog i desnog ruba. Ukupna je širina vidljivog dijela:

$$1\text{px} + 5\text{px} + 200\text{px} + 5\text{px} + 1\text{px} = 212\text{px}$$

Visina elementa uključivat će rub i *padding*:

$$1\text{px} + 5\text{px} + 50\text{px} + 5\text{px} + 1\text{px} = 62\text{px}$$

Promjena načina izračuna

Način izračuna dimenzija elementa može se promijeniti pomoću svojstva ***box-sizing***. Ako se na nekom elementu postavi to svojstvo, može se promijeniti način izračuna dimenzija za taj element.

Zanimljivosti i napomene

Svojstvo *box-sizing* nije podržano u Internet Exploreru 7 pa ga ne možemo rabiti želimo li biti kompatibilni sa starijim preglednicima.

Zanimljivosti i napomene

Postoji i nekoliko gotovih CSS-*resetova* koji se mogu preuzeti i koristiti u vlastitim projektima. Najpoznatiji od njih je CSS *reset* Erica Meyera dostupan na meyerweb.com/eric/tools/css/reset.

Zanimljivosti i napomene

Da je deklaracija *line-height: 1* napisana unutar pravila koje počinje s univerzalnim selektorom, bila bi izgubljena mogućnost nasljeđivanja tog svojstva.

Kad bi postavili različiti *line-height* na nekom elementu, njegova djeca ne bi naslijedila tu vrijednost, već bi na njih bila primijenjena vrijednost definirana univerzalnim selektorom.

Time bi izgubili očekivano ponašanje, što nije poželjno.

Vrijednosti su koje se mogu predati ovom svojstvu:

Ključna riječ	Objašnjenje
<i>content-box</i>	svojstva <i>width</i> i <i>height</i> odnose se samo na širinu sadržaja
<i>border-box</i>	svojstva <i>width</i> i <i>height</i> odnose se na širinu sadržaja, <i>paddinga</i> i <i>margin</i> e

Inicijalna vrijednost ovog svojstva je *content-box*, odnosno način izračuna koji je već opisan. Želi li se da element iz gornjeg primjera bude zaista širok 200 piksela, potrebno mu je svojstvo *width* postaviti na taj iznos umanjeno za širinu ruba i *paddinga*, tj. 188 piksela).

Postavi li se vrijednost svojstva *box-sizing* na *border-box*, element s vrijednošću svojstva *width* od 200 piksela zaista će biti širok 200 piksela.

```
box-sizing: border-box;
```

3.8. Resetiranje predefiniranih vrijednosti

Preglednici sami definiraju početne vrijednosti za neke elemente. Neki HTML-elementi tako imaju postavljene inicijalne vrijednosti za margine (a neki nemaju, što ponekad dovodi do neočekivanih rezultata).

Preglednici također definiraju inicijalnu veličinu (i vrstu) fonta na nekim elementima te visinu retka (*line-height*).

Budući da se te predefinirane vrijednosti razlikuju između preglednika, a često ih zaboravimo sami promijeniti, nastaju sitne razlike u izgledu *web*-stranice ovisno o pregledniku kojim se koristimo.

Zbog toga na početku CSS-dokumenta treba navesti određena pravila (takozvani CSS-*reset*) kojim se takve vrijednosti postavljaju na inicijalne vrijednosti koje će biti ujednačene u svim preglednicima.

Najjednostavnija varijanta takvog CSS-*reseta* je postavljanje margine i *paddinga* svim elementima na nulu, što se može postići pomoću univerzalnog selektora (***):

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

Dodatno, bilo bi uputno postaviti visinu retka na prihvatljivu inicijalnu vrijednost:

```
body
{
    line-height: 1;
}
```

U ovom je primjeru veličina retka postavljena na vrijednost 1, čime će ona biti jednaka visini teksta. Umjesto na sve elemente, postavljena je

na element *body*, a drugi će elementi naslijediti tu vrijednost s elementa *body*.

3.9. Pozadinska slika

Postavljanje pozadinske slike

Osim boje u pozadinu elementa možemo staviti i sliku. Za postavljanje pozadinske slike koristi se svojstvo ***background-image***. Vrijednost se postavlja pozivom funkcije *url*, kojoj predajemo putanju do slike.

```
background-image: url('drvo.png');
```

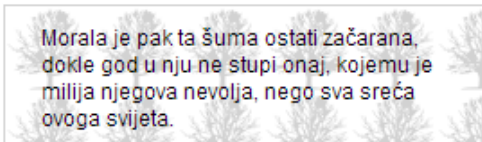
U mapi u kojoj se nalazi CSS-datoteka postoji i slika *drvo.png* koja izgleda ovako:



U slučaju da u toj CSS-datoteci postoji element *p* na kojem su već otprije postavljeni rub, visina i širina, gornja CSS-deklaracija dat će ovakav rezultat:



Pozadinska slika prikazuje se ispod sadržaja elementa. Ako se u element *p* doda tekst, rezultat će izgledati ovako:



Zanimljivosti i napomene

Kada se koristiti elementom *img*, a kada staviti sliku u pozadinu pomoću CSS-a?

Sjetimo se da je namjena CSS-a isključivo prezentacija, dok HTML-elementi definiraju strukturu i sadržaj stranice.

Slike koje zaista služe kao pozadinske slike, koje ne nose neku dodatnu informaciju, nego imaju samo estetsku funkciju, prikazivat ćemo na stranici kao pozadinske slike u CSS-u.

Slike koje su dio sadržaja stranice i koje nose informaciju prikazivat ćemo pomoću elementa *img*.

Ponavljanje pozadinske slike

Slika će se na početku ponavljati vodoravno i okomito dok ne popuni cijeli element. Da bi se promijenio način na koji se slika ponavlja, rabi se svojstvo **background-repeat**. Vrijednosti koje se mogu postaviti tom svojstvu su ključne riječi navedene u tablici:

Ključna riječ	Objašnjenje
<i>no-repeat</i>	slika se ne ponavlja
<i>repeat</i>	slika se ponavlja (inicijalna vrijednost)
<i>repeat-x</i>	slika se ponavlja vodoravno
<i>repeat-y</i>	slika se ponavlja okomito
<i>space</i>	slika se ponavlja, ali višak slike neće biti odrezan, nego će slike biti jednoliko razmaknute
<i>round</i>	slika se ponavlja, ali višak slike neće biti odrezan, nego će slike biti smanjene

Ovako se može postaviti ponavljanje slike bez da je višak slike odrezan:

```
background-repeat: round;
```



Svojstvu **background-repeat** mogu se predati i dvije vrijednosti. U tom se slučaju prva vrijednost odnosi na vodoravno, a druga na okomito ponavljanje. Ključne riječi koje se mogu koristiti u tom načinu pisanja su samo *no-repeat*, *repeat*, *round* i *space*.

```
background-repeat: space repeat;
```



Želi li se isključiti ponavljanje slike, može se navesti ključna riječ *no-repeat*:

```
background-repeat: no-repeat;
```

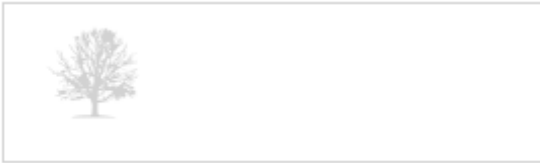


Položaj pozadinske slike

Pozadinska slika inicijalno je smeštena u gornjem lijevom uglu elementa. Ako je ponavljanje slike isključeno, moguće je promijeniti položaj slike pomoću svojstva **background-position**. Tom se svojstvu najčešće predaju dvije vrijednosti, pri čemu prva vrijednost predstavlja vodoravni, a druga okomiti pomak od početnog položaja (gornjeg ruba elementa). Vrijednosti se mogu postaviti u mjernim jedinicama duljine, u postotcima ili se mogu rabiti ključne riječi.

Okavako se pozadinska slika može pomaknuti 20 piksela udesno i 10 piksela prema dolje:

```
background-position: 20px 10px;
```



Na primjeru vodoravnog pomaka može se objasniti korištenje postotaka. Ako se vodoravni pomak postavi na 0 %, slika će se smjestiti uz lijevi rub, a ako se postavi na 100 %, slika će se smjestiti uz desni rub. Vrijednosti između 0 i 100 % raspodjeljuju se jednoliko između tih dvaju krajnjih položaja.

```
background-position: 40% 20%;
```

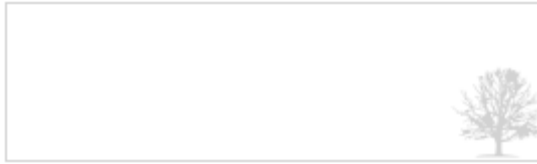


Također se mogu koristiti i ovdje navedene ključne riječi:

Ključna riječ	Objašnjenje
<i>bottom</i>	položaj uz donji rub (kod okomitog pomaka)
<i>center</i>	središnja položaj
<i>left</i>	položaj uz lijevi rub (kod vodoravnog pomaka)
<i>right</i>	položaj uz desni rub (kod vodoravnog pomaka)
<i>top</i>	položaj uz gornji rub (kod okomitog pomaka)

Ova kombinacija vrijednosti smješta sliku u donji desni kut:

```
background-position: right bottom;
```



Prvu je vrijednost moguće navesti na jedan, a drugu na drugi način:

```
background-position: right 70%;
```

Ako se navede samo jedna vrijednost, vodoravni i okomiti pomak bit će jednaki:

```
background-position: 50%;
```



Promjena veličine pozadinske slike

Pozadinska slika prema potrebi se može povećati ili smanjiti korištenjem svojstva **background-size**. Tom se svojstvu može postaviti vrijednost pomoću jedinica duljine, postotaka (koji se odnose na širinu, odnosno visinu elementa) ili pomoću ključnih riječi.

Ovako se može rastegnuti širina slike na pola širine elementa, a visina slike na ukupnu visinu elementa:

```
background-size: 50% 100%;
```



Ako se postave dvije vrijednosti, prva se odnosi na širinu, a druga na visinu. Ako se postavi samo jedna vrijednost, ona se odnosi na širinu, a visina će biti automatski izračunata tako da se zadrže proporcije slike.

```
background-size: 50%;
```



Ključne su riječi koje se mogu rabiti:

Ključna riječ	Objašnjenje
<i>auto</i>	visina, odnosno širina se automatski računa na temelju proporcija slike
<i>cover</i>	slika se povećava (uz zadržavanje proporcija) tako da cijeli element bude pokriven slikom
<i>contain</i>	slika se povećava ili smanjuje (uz zadržavanje proporcija) tako da se cijela slika nalazi unutar elementa

Deklaracija iz prethodnog primjera može se napisati i ovako:

```
background-size: 50% auto;
```

Ključna riječ *auto* jedina se može koristiti u načinu pisanja kad se navode dvije vrijednosti, a preostale ključne riječi (*cover*, *contain* i *inherit*) određuju i širinu i visinu pa se ne mogu rabiti na takav način.

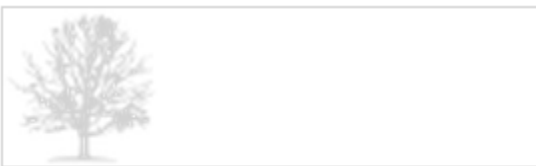
Pomoću ključne riječi *cover* dobiva se sljedeći učinak – slika je povećana tako da prekriva cijeli element, pri čemu su proporcije slike zadržane.

```
background-size: cover;
```



Korištenjem ključne riječi *contain* može se postići da cijela slika bude sadržana unutar elementa uz zadržavanje proporcija.

```
background-size: contain;
```



Postavljanje više slika

Na element se može postaviti više različitih pozadinskih slika. Pojedine slike odvajaju se zarezom:

```
background-image: url('drvo.png'), url('drvo2.png');
```

Ako se žele navesti različite vrijednosti za neka od svojstava koja počinju s *background-*, navodi se više svojstava odvojenih zarezom.

Zanimljivosti i napomene

Osim ovdje navedenih svojstava, pomoću svojstva *background* mogu se postaviti i vrijednosti za svojstva *background-attachment*, *background-clip* i *background-origin*, koja nisu obrađena u tečaju.

Najčešće se žele navesti različite vrijednosti za svojstvo *background-position* da se slike ne bi nalazile na istim položajima:

```
background-position: 10px 10px, 65px 10px;
```

**Skraćeni način pisanja**

Glavna svojstva za oblikovanje pozadine mogu se postaviti u skraćenom načinu pisanja pomoću svojstva **background**. To svojstvo objedinjuje ova pojedinačna svojstva:

- *background-color*
- *background-image*
- *background-repeat*
- *background-position*
- *background-size*.

Sva su svojstva opcionalna, tj. mogu se izostaviti. Redoslijed navođenja svojstava je proizvoljan, s time da svojstvo *background-size* mora biti navedeno poslije svojstva *background-position* i odvojeno kosom crtom (/).

```
background: color image repeat position/size
```

Primjer u kojem su navedena sva ta svojstva izgleda ovako:

```
background: gray url('drvo.png') no-repeat
           10px 5px / auto 90%;
```



Često se može vidjeti da se korištenjem svojstva *background* postavlja samo pozadinska boja (umjesto korištenjem svojstva *background-color*):

```
background: gray;
```

Ili, moguće je tim svojstvom postaviti samo pozadinsku sliku:

```
background: url('drvo.png');
```

Pomoću svojstva *background* također je moguće definirati više pozadinskih slika.

```
background: url('drvo.png') no-repeat 10px 5px,
            gray url('drvo2.png') no-repeat 70px 5px;
```

Važno je napomenuti da se u takvom slučaju boja može navesti samo prilikom navođenja posljednje pozadine.



3.10. Pozadinski prijelaz

Osim jednobojne pozadine, u pozadini elementa može se prikazati i prijelaz između dviju (ili više) boja.

Prijelaz se može definirati pomoću CSS-funkcije *linear-gradient*. Ta funkcija zapravo u pozadini generira sliku s prijelazom pa se koristi slično kao da se postavlja pozadinska slika – pomoću svojstva *background-image* (a ne, kako bi se moglo pomisliti, pomoću svojstva *background-color*).

```
background-image: linear-gradient(lightgray, gray);
```



U najjednostavnijoj varijanti funkcija *linear-gradient* prima dvije boje između kojih se radi prijelaz. Boje se mogu navesti pomoću ključne riječi, heksadecimalnog kôda ili kôdova RGBa i HSLa (što znači da se mogu rabiti i transparentne boje).

Ako se želi postaviti smjer prijelaza, potrebno ga je navesti kao prvi parametar funkcije pomoću ključne riječi:

```
background-image: linear-gradient(to right,
                                  lightgray, gray);
```

U ovom primjeru smjer prijelaza usmjeren je slijeva **nadesno**.



Zanimljivosti i napomene

Osim običnog linearnog prijelaza moguće je pomoću funkcije *repeating-linear-gradient* definirati i ponavljajući linearni prijelaz. Time se može dobiti efekt prugaste pozadine.

Također, može se definirati i radijalni prijelaz pomoću funkcije *radial-gradient* ili ponavljajući radijalni prijelaz pomoću funkcije *repeating-radial-gradient*.

Za postavljanje smjera prijelaza mogu se rabiti ove ključne riječi (odnosno izrazi):

Ključna riječ	Objašnjenje
<i>to top</i>	prijelaz je usmjeren od dna prema vrhu elementa
<i>to right</i>	prijelaz je usmjeren s lijeve na desnu stranu elementa
<i>to bottom</i>	prijelaz je usmjeren od vrha prema dnu elementa
<i>to left</i>	prijelaz je usmjeren s desne na lijevu stranu elementa
<i>to top left</i>	prijelaz je usmjeren od donjeg desnog prema gornjem lijevom kutu elementa
<i>to top right</i>	prijelaz je usmjeren od donjeg lijevog prema gornjem desnom kutu elementa
<i>to bottom left</i>	prijelaz je usmjeren od gornjeg desnog prema donjem lijevom kutu elementa
<i>to bottom right</i>	prijelaz je usmjeren od gornjeg lijevog prema donjem desnom kutu elementa

Ako se smjer prijelaza želi definirati preciznije nego što to dopuštaju ključne riječi, potrebno ga je navesti u stupnjevima (pomoću oznake za stupnjeve u CSS-u – *deg*). Dopuštene vrijednosti su od 0 do 360, a prijelaz usmjeren na desno (kao u prethodnom primjeru) postiže se postavljanjem vrijednosti od 90 stupnjeva:

```
background-image: linear-gradient(90deg, lightgray, gray);
```

Umjesto korištenja samo dviju boje za prijelaz, može ih se navesti i više:

```
background-image: linear-gradient(to right, lightgray, gray, black);
```



Tako se zapravo dodaju dodatne točke prijelaza. Položaj tih točaka se automatski određuje, no moguće ga je i postaviti navođenjem vrijednosti u jedinicama duljine ili u postotcima:

```
background-image: linear-gradient(to right, lightgray 20%, gray 70%, black);
```



Tako je prva točka prijelaza (između svjetlosive i sive) postavljena na 20 % širine elementa (gledano slijeva nadesno), a druga točka prijelaza (između sive i crne) na 70 % širine elementa.

Ako se elementu želi dodati i pozadinska slika i prijelaz, mogu se navesti dvije vrijednosti za svojstvo *background-image* odvojene zarezima. Budući da želimo da slika bude prikazana iznad prijelaza, navest ćemo je prvu:

```
background-image: url('drvo.png'),
                 linear-gradient(to right, lightgray, gray);

background-repeat: no-repeat;
```



Treba također podesiti i svojstvo *background-repeat*, da se slika ne bi ponavljala.

Za postavljanje prijelaza također se može koristiti skraćeni način pisanja pomoću svojstva *background*:

```
background: linear-gradient(to right, lightgray,
                              gray);
```

Ako se želi postaviti i pozadinska slika, pomoću svojstva *background* to se može napisati ovako:

```
background: url('drvo2.png') no-repeat,
            linear-gradient(to right, lightgray, gray);
```

3.11. Sjena

Kao što je moguće tekstu dodati sjenu (pomoću svojstva *text-shadow*), tako se ona može dodati i HTML-elementu pomoću svojstva ***box-shadow***. To svojstvo prima ove vrijednosti, koje treba navesti ovim redoslijedom:

- *inset* – ključna riječ za postavljanje unutrašnje sjene (opcionalni parametar)
- vodoravni pomak (vrijednost u pikselima ili drugim jedinicama za duljinu)
- okomiti pomak (vrijednost u pikselima ili drugim jedinicama za duljinu)
- radijus zamućenja (opcionalni parametar – vrijednost u pikselima ili drugim jedinicama za duljinu)
- radijus proširenja (opcionalni parametar – vrijednost u pikselima ili drugim jedinicama za duljinu)
- boja (navedena pomoću ključne riječi, heksadecimalnog kôda ili kôdova RGBa ili HSLa).

Evo najjednostavnijeg primjera dodavanja sjene elementu:

```
box-shadow: 5px 5px gray;
```

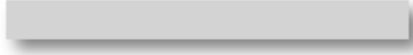


Zanimljivosti i napomene

Svojstvo *box-shadow* nije podržano u Internet Exploreru 8 i starijim verzijama *Internet Explorera*.

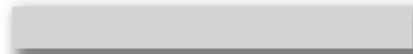
Radijus zamućenja, slično kao i kod sjene na tekstu, kontrolira koliko će sjena biti oštra ili mutna. Ako se izostavi, njegova će vrijednost biti 0 i sjena će biti potpuno oštra. Što se postavi veći radijus zamućenja, to će sjena biti mutnija.

```
box-shadow: 5px 5px 10px gray;
```



Radijus proširenja kontrolira koliki će prostor sjena zahvatiti. Ako se izostavi, njegova će vrijednost biti 0 i sjena će zauzimati samo inicijalnu veličinu (definiranu vodoravnim i okomitim pomakom).

```
box-shadow: 5px 5px 10px 5px gray;
```



Pomoću ključne riječi *none* može se isključiti sjena ako je postavljena nekim drugim pravilom:

```
box-shadow: none;
```



Ako se za vodoravni pomak navede negativna vrijednost, sjena će se postaviti iznad umjesto ispod elementa, a ako se navede negativna vrijednost za okomiti pomak, sjena će se postaviti lijevo umjesto desno od elementa.

```
box-shadow: -5px -5px gray;
```



Navođenjem ključne riječi *inset* sjena će biti postavljena unutar elementa (uz gornji i lijevi rub elementa). U ovom primjeru pokazano je postavljanje unutrašnje sjene:

```
box-shadow: inset 5px 5px 10px gray;
```



Kao i kod sjene na tekstu, moguće je navesti više sjena, odvojenih zarezom:

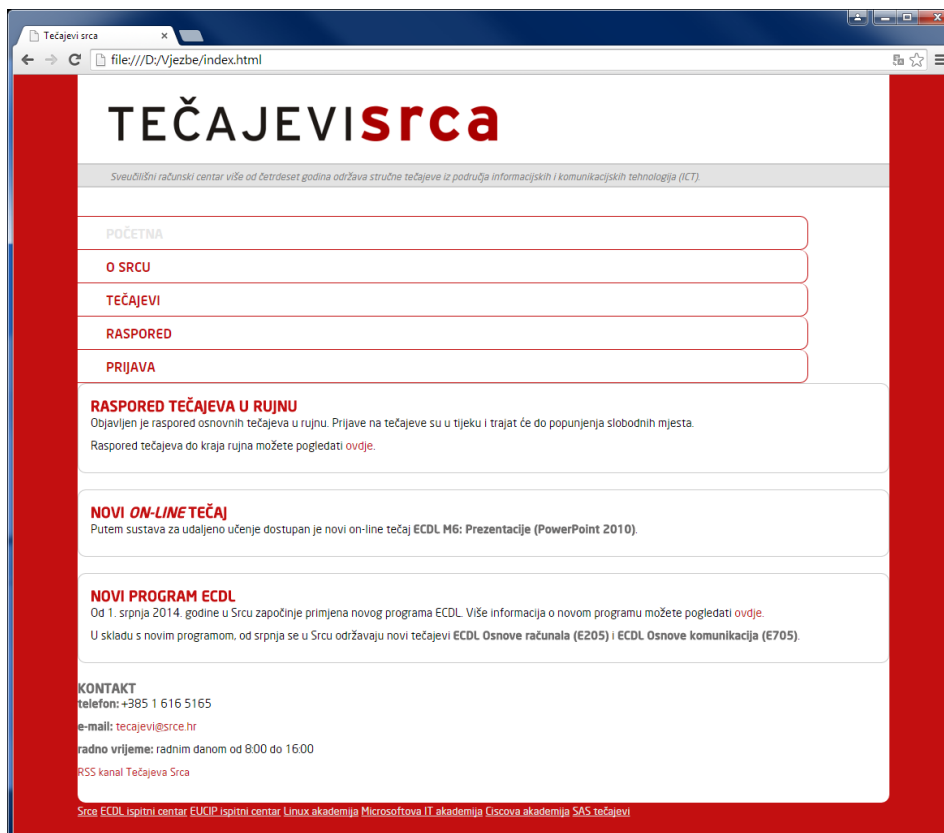
```
box-shadow: inset 5px 5px 10px white,  
            3px 3px 3px gray,  
            6px 6px 3px black;
```



3.12. Vježba: Oblikovanje elemenata

1. U mapi C220/Vjezbe/css otvorite datoteku *oblikovanje.css*. Rezultat napisanih CSS-pravila možete vidjeti pregledavanjem datoteka *index.html* („Početna“) i *tecajevi.html* („Tečajevi“) u pregledniku.
2. Pomoću univerzalnog selektora napišite osnovni CSS-*reset*, kako biste smanjili razlike u prikazu između različitih preglednika. Svim elementima postavite vrijednost za marginu i *padding* na 0. Uz to postavite svojstvo *box-sizing* na vrijednost *border-box*, da biste se mogli koristiti jednostavnijim načinom izračuna dimenzija elemenata.
3. Elementu *html* postavite boju pozadine na tamnocrvenu (`#c30f10`).
4. Elementu *body* postavite širinu na 1000 piksela i dodatno mu postavite obje horizontalne margine na vrijednost *auto*. Tako će glavni sadržaj uvijek biti centriran, bez obzira na veličinu prozora preglednika.
5. Svim elementima *p* postavite donju marginu na 10 piksela.
6. Elementu s klasom *zaglavlje* i elementu s klasom *glavno* postavite pozadinsku boju na bijelu.
7. Slici (elementu *img*) unutar elementa s klasom *zaglavlje* postavite gornju marginu na 20, lijevu također na 20, a donju na 10 piksela.
8. Za element *p* unutar elementa s klasom *zaglavlje* postavite pozadinsku boju na svijetlosivu (`#e3e3e3`). Visinu mu postavite na 30 piksela. Budući da smo postavili marginu na sve elemente *p*, a to nam u ovom slučaju ne odgovara, postavite mu vrijednost za donju marginu na 0. Postavite i lijevi *padding* na 40, a gornji na 7 piksela. Na kraju, gornji i donji rub elementa prikažite kao ravnu crtu debljine 2 piksela i sive boje (`#ccc`).
9. Na elementu s klasom *glavno* postavite gornji *padding* od 35 te donji *padding* od 30 piksela. Za donji lijevi i donji desni ugao postavite radijus zakrivljenosti od 10 piksela. Tom elementu postavite i minimalnu visinu – u slučaju da je njegov sadržaj manji, želimo da on bude visok barem 500 piksela.
10. Elementu s klasom *clanak* postavite *padding* (u svim smjerovima) od 15 piksela, a donju marginu mu postavite na 20 piksela. Postavite mu puni rub debljine 1 piksel i sive boje (`#ccc`) te mu postavite radijus zakrivljenja na 10 piksela.
11. Linku unutar elementa s klasom *navigacija* promijenite tip prikaza u blok-prikaz. Tako ćemo moći bolje oblikovati linkove u glavnoj navigaciji. Širinu mu postavite na 90 %, gornji i donji *padding* na 10 piksela, a lijevi i desni na 35 piksela. Donji i desni rub prikažite punom crtom debljine 1 piksela, tamnocrvene boje (`#c30f10`), a gornji desni i donji desni ugao zaoblite s radijusom od 10 piksela.
12. Ako se radi o prvom linku unutar elementa s klasom *navigacija* (što se može odabrati pomoću pseudoklase *first-child*), ipak je

potrebno prikazati i gornji rub kao punu crtu debljine 1 piksela tamnocrvene boje (#c30f10) pa to promijenite. Za ostale linkove u navigaciji gornji rub nije potreban, jer bi se inače preklapao s donjim rubom prethodnog elementa.



Dodatni zadatci

13. Ako se radi o linku na kojem se trenutno nalazimo, on će imati klasu *selected*. Dakle, potrebno je odabrati element *a* koji ima klasu *selected*, a nalazi se u elementu s klasom *navigacija*. U pozadini izradite prijelaz iz svijetlocrvene (#f7dcdc) u tamnocrvenu (#c30f10) pomoću svojstva *background-image* i funkcije *linear-gradient*. Uz to, kod odabranog linka povećajte širinu na 100 % da se dobije efekt „izvučenog“ dugmeta.
14. Elementu s klasom *istaknuto* postavite pozadinsku boju na svijetlocrvenu (#f7dcdc).
15. Elementu s klasom *kontakt* postavite svijetlosivu pozadinsku boju (#e8e8e8) i puni rub od 1 piksela sive boje (#ccc). Rub zaoblite s radijusom zakrivljenja od 10 piksela te postavite desnu i donju marginu, obje veličine 20 piksela.
16. Na elementu *h3*, koji se nalazi unutar elementa s klasom *kontakt*, postavite donji rub pomoću pune crte od 1 piksela sive boje (#ccc). Gornju i donju marginu postavite mu na 5 piksela, a lijevi *padding* na 35 piksela.
17. Za elemente *p* koji se nalaze unutar elementa s klasom *kontakt* postavite gornju i donju marginu na 5, a lijevu i desnu marginu na 15 piksela. Također mu postavite i lijevi *padding* od 20 piksela (zbog pozadinske slike).

18. Elementu s klasom *telefon* postavite za pozadinsku sliku datoteku *telefon.png* koja se nalazi u mapi *C200/Vjezbe/Slike*. Isključite ponavljanje pozadinske slike. Slično je potrebno napraviti i za elemente s klasom *email*, odnosno *radno-vrijeme*.
19. Element s klasom *rss* (koji je zapravo link) pretvorite u blok-element promjenom njegova svojstva *display*. Zatim mu postavite pozadinsku sliku (*rss.png*) kojoj također treba isključiti ponavljanje. Postavite mu lijevu marginu od 5 piksela te lijevi *padding* od 20 piksela (da bi se tekst odmakao od slike).
20. Za elemente *a* koji se nalaze unutar elementa s klasom *podnozje* postavite lijevi i desni *padding* od 10 piksela.
21. Elementu s klasom *tecaj* (koji se koristi na stranici „Tečajevi“) postavite puni rub od 1 piksela sive boje (#ccc) i zaoblite ga s radijusom zakrivljenja od 10 piksela.
22. Za element *h3* koji se nalazi u elementu s klasom *tecaj* postavite donju marginu na 5, a na sve ostale na 10 piksela.
23. Za element *em* koji se nalazi u elementu s klasom *tecaj* postavite lijevi *padding* na 10 piksela.
24. Za element *p* koji se nalazi u elementu s klasom *tecaj* postavite gornju marginu na 5, a preostale tri na 10 piksela.



4. Oblikovanje specifičnih elemenata

U ovom poglavlju upotrijebit će se dosad obrađena CSS-svojstva kako bi se oblikovali HTML-elementi s kojima se svakodnevno susrećemo, ali ipak imaju neke svoje posebnosti što se tiče oblikovanja. Pritom će se obraditi i neka, još neobrađena, CSS-svojstva.

4.1. Linkovi

Link, odnosno element `a` jedan je od osnovnih elemenata u HTML-u (štoviše, element na kojem se temelji čitav koncept hiperteksta). Inicijalno svi preglednici prikazuju linkove plavom bojom i podcrtano:

Ovo sve slušao i gledao [Bjesomar](#), vladar šumskih bjesova.

Kada se definira osnovni izgled teksta za *web*-stranicu, najčešće se to radi na elementu *body*. Svi drugi elementi naslijedit će svojstva fonta s elementa *body*. U ovom primjeru postaviti će se vrsta fonta, boja i početna veličina teksta:

```
body
{
    font-family: Arial, sans-serif;
    font-size: 14px;
    color: gray;
}
```

Ovo sve slušao i gledao [Bjesomar](#), vladar šumskih bjesova.

Iz primjera se vidi da je element `a` naslijedio vrstu i veličinu fonta s elementa *body*, no boja linka je i dalje ostala plava. Razlog tome je što se preglednikovo vlastito CSS-pravilo za postavljanje plave boje linka primijenjuje izravno na element `a` i mijenja vrijednost naslijeđenu s elementa *body*.

Da bi se promijenila boja linka, potrebno je da se u CSS-u *web*-stranice boja postavi izravno na element `a`:

```
a
{
    color: darkred;
}
```

Međutim, moguće je primijetiti da su linkovi ponekad ljubičasti, iako im je već postavljena željena boja. Naime, preglednik linkove koje je posjetitelj već posjetio prikazuje ljubičastom bojom.

To se ponašanje može promijeniti pomoću CSS-pseudoklase *visited*, koju smo obradili u prvom poglavlju. Ta pseudoklasa primjenjuje se na već posjećene linkove.

Ako se želi da i posjećeni i neposjećeni linkovi imaju istu boju, moguće je promijeniti gornje pravilo tako da se ono eksplicitno odnosi i na posjećene linkove:

```
a, a:visited
{
    color: darkred;
}
```

Pomoću pseudoklase *hover* može se postići da izgled linka bude promijenjen u trenutku kad korisnik prijeđe mišem preko linka. Tako korisniku dodatno dajemo do znanja da je taj element stranice aktivan, odnosno da se na njega može kliknuti mišem.

Najčešće se koriste ovi efekti: promjena boje linka, promjena pozadinske boje linka i postavljanje/uklanjanje crte ispod teksta.

```
a: hover
{
    color: orange;
    background-color: lightgray;
    text-decoration: none;
}
```

Prema tipu prikaza element *a* je linijski (*inline*) element, tj. zauzima samo onoliko mjesta koliko je velik njegov sadržaj. Zbog toga je i prostor koji je osjetljiv na klik razmjerno mali.

Kada se radi o linkovima koji su izdvojeni iz teksta (npr. linkovi u navigaciji), moguće je povećati prostor na koji korisnik može kliknuti mišem tako da se element pretvori u blok-element (ili element *inline-block*) i da mu se postave odgovarajuća visina i širina.

```
a
{
    display: block;
    width: 200px;
    height: 50px;
}
```

Ako se radi o slici koja je ujedno i link (ako unutar elementa *a* postoji element *img*), neki preglednici (npr. starije verzije Internet Explorera) prikazat će plavi rub oko slike. Da bi se on uklonio, potrebno je postaviti rub takve slike na *none*:

```
a img
{
    border-style: none;
}
```

4.2. Polja za unos

Polja za unos podataka preglednici prikazuju vrlo različito. Postizanje željenog izgleda polja za unos, koji će pritom biti jednak u svim preglednicima, ponekad je zahtjevno, a kod nekih tipova polja (npr. polje za prijenos datoteke) zapravo i nije moguće.

No nekima od najčešće korištenih polja za unos ipak je moguće donekle prilagoditi izgled u većini preglednika.

Za primjer će se rabiti jednostavan obrazac za unos podataka kao u ovom primjeru:

Postavite pitanje

HTML-kôd ovog obrasca (u pojednostavljenom obliku) mogao bi izgledati ovako:

```
<form>
  Postavite pitanje
  <input type="text" placeholder="Vaše ime" />
  <textarea
    placeholder="Vaše pitanje?">
  </textarea>
  <select>
    <option>Vaša dob</option>
    <option>do 20 godina</option>
    <option>20 - 30 godina</option>
    <option>30 - 40 godina</option>
    <option>preko 40 godina</option>
  </select>
  <button>Pošalji</button>
</form>
```

Ako se osnovna svojstva teksta definiraju na elementu *body*, polja za unos zadržat će vlastitu veličinu, boju i vrstu fonta.

```
body
{
  font-family: Verdana, sans-serif;
  font-size: 14px;
  color: navy;
}
```

Postavite pitanje

Zanimljivosti i napomene

Inicijalni tekst u poljima za unos teksta (postavljen pomoću atributa *placeholder*) prikazuje se u većini preglednika sivom bojom. Boja se tog teksta može promijeniti samo u nekim preglednicima.

Također, boja se teksta u listi za odabir (*select*-elementu) može promijeniti samo u nekim preglednicima.

Zanimljivosti i napomene

Za slučaj da postoji više različitih elemenata *input*, pravilo treba ograničiti samo na polje za unos teksta pomoću selektora *input[type=text]*.

Uzrok je ovoga u tome što preglednici postavljaju vlastite inicijalne vrijednosti za svojstva fonta na poljima za unos. Žele li se na tim poljima postaviti ista svojstva fonta kao i u ostatku stranice, može se eksplicitno reći elementima za unos da naslijede ta svojstva s roditeljskog elementa:

```
input[type="text"], textarea, select, button
{
    font-family: inherit;
    font-size: inherit;
    color: inherit;
}
```

Postavite pitanje

Prednost je takvog pristupa da su vrijednosti za vrstu fonta, veličinu i boju navedene samo na jednom mjestu, što omogućava lakše izmjene. Želimo li na poljima za unos imati npr. različitu veličinu ili boju teksta, možemo postaviti konkretne vrijednosti tim svojstvima.

Polja za unos su linijski elementi pa u slučaju da se želi da svaki bude prikazan u svojem redu, treba ih pretvoriti u blok-elemente. Kako bi se ujednačio razmak između pojedinih polja u svim preglednicima, potrebno je postaviti i okomitu marginu na elemente.

```
input[type="text"], textarea, select, button
{
    display: block;
    margin: 10px 0px;
}
```

Postavite pitanje

Ako se želi ujednačiti širina elemenata, može im se jednostavno postaviti širina na željenu vrijednost pomoću svojstva *width*. No u nekim preglednicima (npr. *Internet Explorer* 9 i starije verzije) to neće rezultirati posve jednakom širinom zbog načina kako preglednici inicijalno računaju širinu:

Postavite pitanje

Taj se problem može riješiti tako da se pomoću svojstva *box-sizing* promijeni način izračuna širine:

```
input[type="text"], textarea, select, button
{
    width: 200px;
    box-sizing: border-box;
}
```

Postavite pitanje

Poljima za unos može se postaviti i željena visina te *padding*. Pri tome je obično potrebno elementu za unos višelinijskog teksta (*textarea*) postaviti veću visinu:

```
input[type="text"], select, button
{
    height: 30px;
    padding: 3px 5px;
}
textarea
{
    height: 60px;
    padding: 3px 5px;
}
```

Postavite pitanje

Da bi polja za unos imala istu boju ruba u svim preglednicima, potrebno ju je eksplicitno postaviti. Također je moguće zaobliti rubove te dodati unutarnju sjenu:

```
input[type="text"], select, textarea
{
    border: 1px solid darkgray;
    border-radius: 5px;
    box-shadow: inset 2px 2px 5px lightgray;
}
```

Postavite pitanje

Element *textarea* neće biti sasvim zaobljen u *Internet Exploreru* zbog strelica za pomicanje koje se nalaze s desne strane.

Strelice se mogu sakriti (barem inicijalno, dok korisnik ne unese tekst veći od polja) pomoću svojstva **overflow**, kojim se vrijednost postavlja na *auto* da bi se strelice pojavile samo ako su zaista potrebne.

```
textarea
{
    overflow: auto;
}
```

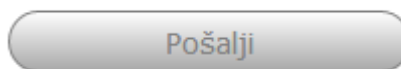
Dugme za slanje podataka također se može uljepšati zaobljivanjem i stavljanjem prijelaza u pozadinu:

```
button
{
  border: 1px solid gray;
  border-radius: 20px;
  background-image: linear-gradient
                    (lightgray, darkgray, gray);
  color: white;
}
```

Postavite pitanje

Prilikom klika mišem na dugme, preglednici obično promijene njegov izgled tako da dočaraju reakciju dugmeta na klik. Budući da je inicijalni izgled dugmeta u ovom slučaju promijenjen, izgubljena je ta dodatna funkcionalnost pa je potrebno promijeniti izgled pritisnutog dugmeta. Pritisnuto dugme može se selektirati pomoću pseudoklase *active*. U ovom primjeru promijenit će se boja prijelaza i boja slova:

```
button:active
{
  background-image:linear-gradient
                    (white, lightgray, darkgray);
  color:gray;
}
```



Preglednici najčešće označavaju trenutno odabrani element dodavanjem plavog obruba oko njega. Korištenjem pseudoklase *:focus*, moguće je odrediti kako će se vizualno označiti taj element.

```
input[type="text"]:focus, textarea:focus,
select:focus
{
  background-color: rgba(0,0,0,0.1);
}
```

U primjeru ćemo postaviti svjetlosivu transparentnu boju (kako bi se i dalje vidio tekst upisan u atribut *placeholder*) u pozadinu odabranog elementa.

Da bi se maknuo plavi obrub koji preglednik sam postavlja oko odabranog elementa, potrebno je upotrijebiti svojstvo **outline**. Ono služi za postavljanje obruba, a slično je svojstvu *border* za postavljanje ruba. Za razliku od ruba, obrub se nalazi izvan elementa i ne računa se u njegove dimenzije. Svojstvo *outline* skraćeni je način pisanja za odvojena svojstva *outline-style*, *outline-width* i *outline-color*, koja primaju iste vrijednosti kao i svojstva *border-style*, *border-width* i *border-color*.

Postavljanjem vrijednosti svojstva *outline* na *none* uklonit će se taj plavi obrub.

```
input[type="text"]:focus, textarea:focus,
select:focus, button:focus
{
    outline: none;
}
```

Postavite pitanje

The image shows a web form with four elements: a text input field labeled 'Vaše ime', a text area labeled 'Vaše pitanje?', a dropdown menu labeled 'Vaša dob', and a 'Pošalji' button. All these elements have a prominent blue outline around them, which is the default browser focus style.

4.3. Liste

Liste se u HTML-u često koriste kad se želi navesti niz međusobno povezanih elemenata. Inicijalni način na koji se lista prikazuje definira preglednik, no CSS omogućuje potpunu kontrolu nad tim kako će lista biti prikazana.

HTML-kôd za običnu, nenumeriranu listu izgleda ovako:

```
<ul>
  <li>Regoč</li>
  <li>Jagor</li>
  <li>Šuma sriborova</li>
</ul>
```

Takvu listu preglednik će prikazati tako da uz svaku stavku prikaže kao oznaku krug u boji teksta:

- Regoč
- Jagor
- Šuma Striborova

Izgled oznake stavke može se promijeniti pomoću CSS-svojstva **list-style-type**, kojoj se kao vrijednost predaje ključna riječ. Važno je zapamtiti da se to svojstvo postavlja na samu listu (u ovom primjeru element *ul*), a ne na stavku liste (element *li*).

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

- Regoč
- Jagor
- Šuma Striborova

Ključne riječi koje se mogu rabiti dane su u tablici:

Ključna riječ	Objašnjenje
<i>disc</i>	• krug (inicijalna vrijednost kod nenumerirane liste)
<i>circle</i>	◦ kružnica
<i>square</i>	▪ kvadrat
<i>decimal</i>	1. decimalni brojevi (inicijalna vrijednost kod numerirane liste) 2. numerirane liste)
<i>decimal-leading-zero</i>	01. decimalni brojevi s početnom nulom 02.
<i>lower-roman</i>	i. rimski brojevi (malim slovima) ii.
<i>upper-roman</i>	I. rimski brojevi II.
<i>lower-greek</i>	α. grčka slova β.
<i>lower-alpha</i>	a. mala slova (latinica) b.
<i>lower-latin</i>	isto kao i <i>lower-alpha</i>
<i>upper-alpha</i>	A. velika slova (latinica) B.
<i>upper-latin</i>	isto kao i <i>upper-alpha</i>
<i>armenian</i>	Ա. armenski brojevi Բ.
<i>georgian</i>	ს. gruzijski brojevi ბ.
<i>none</i>	bez oznake

Prve tri vrijednosti odnose se samo na nenumerirane liste (element *ul*), a sljedećih se 11 vrijednosti odnosi samo na numerirane liste (element *ol*).

Korištenjem svojstva *list-style-type* moguće je nenumeriranu listu prikazati kao numeriranu i obrnuto, ali to nije poželjno.

Ako nam oznaka stavke ne odgovara, možemo umjesto nje upotrijebiti vlastitu sliku pomoću svojstva ***list-style-image***. Kao vrijednost treba postaviti poziv funkcije *url*, kojoj se predaje putanja do slike:

```
ul
{
  list-style-image: url('Star.png');
}
```

Rezultat je slika umjesto standardne oznake stavke:

- ★ Regoč
- ★ Jagor
- ★ Šuma Striborova

Ako je postavljena slika za oznaku stavke pomoću ovog svojstva, ona će biti upotrijebljena čak i ako je postavljeno svojstvo *list-style-type*.

Osim poziva funkcije *url*, kao vrijednost ovom svojstvu može se predati i ključna riječi *none*.

Ključna riječ	Objašnjenje
<i>none</i>	ne koristi se slika kao oznaka stavke

Inicijalna vrijednost ovog svojstva je *none*.

Pomoću svojstva ***list-style-position*** moguće je (donekle) podešavati položaj oznake. Kao vrijednost se tom svojstvu mogu postaviti ove ključne riječi:

Ključna riječ	Objašnjenje
<i>outside</i>	oznaka je izvan prostora koji zauzima tekst
<i>inside</i>	oznaka je unutar prostora koji zauzima tekst

Djelovanje tog svojstva vidljivo je kad imamo tekst koji se prelama u sljedeći red. Inicijalna vrijednost tog svojstva je *outside*, pri čemu lista izgleda ovako:

- Regoč
- Jagor
- Šuma Striborova
- Bratac Jaglenac i sestrice
Rutvica

Kad bi se postavila vrijednost *inside*, oznaka stavke bi se nalazila unutar prostora u kojem se nalazi tekst pa bi zadnja stavka, koja se prelama u novi red, izgledala drugačije.

```
ul
{
    list-style-position: inside;
}
```

- Regoč
- Jagor
- Šuma Striborova
- Bratac Jaglenac i sestrice
Rutvica

Ta tri svojstva vezana za liste mogu se napisati na skraćeni način pomoću svojstva ***list-style***. Kao vrijednost se navodi vrijednosti za svojstva *list-style-type*, *list-style-image* i *list-style-position* u proizvoljnom redoslijedu, a neka se mogu i izostaviti.

Ovako se sva tri svojstva postavljaju odjednom:

```
ul
{
    list-style: circle url('Star.png') inside;
}
```

Istovremeno postavljanje svojstava *list-style-image* i *list-style-type* u načelu nema previše smisla (jer prvo svojstvo poništava drugo) pa se najčešće postavljaju samo dva svojstva:

```
ul
{
    list-style: url('Star.png') inside;
}
```

Prema tipu prikaza liste su blok-elementi, tj. inicijalno zauzimaju cijeli redak. Stavka liste, tj. element *li* također zauzima cijeli redak, no ona ima vlastiti tip prikaza – *list-item*, koji je sličan blok-elementu, ali prikazuje i oznaku stavke.

Ako se promijeni tip prikaza elementu *li* (u želji da se dobije vodoravna lista, oznaka liste više neće biti vidljiva):

```
li
{
    display: inline;
}
```

Regoč Jagor Šuma Striborova

Liste se, budući da predstavljaju niz povezanih elemenata, često koriste za definiranje navigacijskog izbornika *web*-stranice. Unutar stavki liste, umjesto običnog teksta u tom se slučaju stavljaju elementi *a* koji vode na pojedine stranice.

Time navigacija dobiva poželjnu strukturu liste, ali najčešće se ne želi zadržati inicijalni izgled liste s oznakama stavki. Promjenom tipa prikaza na elementu *li* i dodatnim stiliziranjem moguće je listu oblikovati po želji.

Evo primjera liste koja je pretvorena u navigacijski izbornik:



```
li
{
    display: inline-block;
    height: 20px;
    width: 150px;
    padding: 10px;
    text-align: center;
    color: white;
    border-left: 1px solid lightgray;
    border-right: 1px solid gray;
    background-image: linear-gradient
        (lightgray, darkgray, gray);
}
```

Zanimljivosti i napomene

Da ne bi bili vidljivi razmaci između elemenata *inline-block*, potrebno je elemente *li* u HTML-u napisati bez razmaka, dakle
`RegočJagorŠuma Striborova`.

4.4. Tablice

Zanimljivosti i napomene

Tablice su se nekada rabile i za kreiranje rasporeda stranice, zbog toga što je pomoću njih bilo lako podijeliti stranicu na stupce. Takva je praksa bila opravdana u vrijeme dok u CSS-u nije postojalo dobro rješenje za taj problem, no danas se nikako ne preporuča uporaba tablice da bi se postigao željeno pozicioniranje elemenata.

Tablice su HTML-elementi čija je namjena organizacija sadržaja u tabličnom obliku. U načelu se tu uvijek radi o podacima koje je prirodno prikazati kao retke u tablici.

Za primjer će se koristiti tablica s podacima o knjigama za djecu i njihovim cijenama:

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

HTML-kôd ovakve tablice izgleda ovako:

```
<table>
  <tr>
    <th>Naslov</th>
    <th>Autor</th>
    <th>Cijena (kn)</th>
  </tr>
  <tr>
    <td>Vuk koji je htio obići svijet</td>
    <td>Orianne Lallemand</td>
    <td>39,90</td>
  </tr>
  <tr>
    <td>Zec Mrkvus Veliki</td>
    <td>Kęstutis Kasparavičius</td>
    <td>94,50</td>
  </tr>
  <tr>
    <td>Matilda i zagonetne žabe</td>
    <td>Simeon Marinković</td>
    <td>65,00</td>
  </tr>
</table>
```

Preglednici inicijalno prikazuju tablicu bez rubova, a tekst unutar ćelija zaglavlja (elementi *th*) bit će centriran i mastan. Tablica iz primjera izgleda ovako:

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Da bi tablica bila preglednija, treba prikazati crte koje omeđuju retke i stupce. To se postiže tako da se postavi rub na ćelije (elemente *td* i *th*):

```
td, th
{
    border: solid 1px gray;
}
```

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Preglednici inicijalno postavljaju određeni razmak između ćelija, kao što se može vidjeti na gornjoj slici. Pomoću svojstva ***border-spacing*** može se upravljati veličinom tog razmaka. To se svojstvo postavlja na elementu *table*, a prima vrijednost definiranu u nekoj od jedinica duljine:

```
table
{
    border-spacing: 5px;
}
```

Ako se tom svojstvu predaju dvije vrijednosti, prva će se vrijednost odnositi na vodoravni razmak (između stupaca), a druga na okomiti razmak (između redaka):

```
table
{
    border-spacing: 5px 3px;
}
```

Kao vrijednost se tom svojstvu može postaviti i ključna riječ *inherit*, čime će se naslijediti vrijednost s roditeljskog elementa.

Ako se razmak između ćelija želi potpuno ukloniti, svojstvo *border-spacing* može se postaviti na nulu:

```
table
{
    border-spacing: 0;
}
```

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Prostor između ćelija je nestao, no zbog dodirivanja susjednih rubova neke linije izgledaju deblje.

Umjesto upotrebom svojstva *border-spacing*, rubovi se pojedinih ćelija mogu spojiti pomoću svojstva ***border-collapse***, ako se njegova vrijednost postavi na ključnu riječ *collapse*:

```
table
{
    border-collapse: collapse;
}
```

Tako će se susjedni rubovi spojiti u jednu crtu i neće izgledati deblje.

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Svojstvo *border-collapse* može se postaviti na ove vrijednosti:

Ključna riječ	Objašnjenje
<i>collapse</i>	rubovi susjednih ćelija se spajaju u jednu crtu
<i>separate</i>	rubovi susjednih ćelija su dvije odvojene crte (inicijalna vrijednost)

Zanimljivosti i napomene

Ako je svojstvo *border-collapse* postavljeno na vrijednost *collapse*, vrijednost svojstva *border-spacing* se ignorira, tako da ga u tom slučaju nije ni potrebno postavljati.

Čitljivost teksta u tablici može se poboljšati postavljanjem *paddinga* na ćelije:

```
td, th
{
    padding: 10px 5px;
}
```

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Veličina tablice i ćelija unutar nje automatski se prilagođavaju sadržaju. Ako je potrebno, moguće je postaviti širinu ili visinu na tablicu ili na pojedine ćelije čime se definira širina stupca, odnosno visina retka.

Pretpostavimo da se želi postaviti širina tablice na 600 piksela, a širina prvog stupca na 50 % širine tablice. Za podešavanje širine stupca može se pomoću pseudoklase *first-child* selektirati prva ćelija u retku:

```
table
{
    width: 600px;
}
td:first-child
{
    width: 50%;
}
```

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Želi li se tekst u ćeliji poravnati vodoravno, to se može postići pomoću svojstva *text-align*. Obično se želi postići da numerički iznosi, kao što je npr. cijena, budu poravnati udesno pa je potrebno odabrati zadnju ćeliju unutar retka:

```
td:last-child
{
    text-align: right;
}
```

Za okomito poravnanje teksta u ćelijama tablice koristi se svojstvo ***vertical-align***. Kada se primijenjuje na ćelije tablice, mogu se koristiti ove ključne riječi:

Ključna riječ	Objašnjenje
<i>top</i>	tekst je okomito poravnat uz gornji rub ćelije
<i>middle</i>	tekst se nalazi u sredini ćelije (inicijalna vrijednost)
<i>bottom</i>	tekst je okomito poravnat uz donji rub ćelije

Tekst u ćelijama inicijalno je okomito centriran, ali to se može promijeniti pomoću svojstva *vertical-align*:

```
td, th
{
    vertical-align: top;
}
```

Zanimljivosti i napomene

Svojstvo *text-align* koristi se za horizontalno poravnanje teksta i unutar ćelija tablica i unutar blok elemenata (npr. *div* ili *p* elemenata). No, za razliku od tog svojstva, svojstvo *vertical-align* ne može se koristiti za vertikalno poravnanje teksta unutar blok elemenata.

Svojstvo *vertical-align* ima dvije namjene – za vertikalno poravnanje teksta unutar ćelije tablice, i dodatno, za promjenu vertikalne pozicije linijskog elementa (npr. *span*, *sub* ili *sup* elementa) unutar linije teksta. U ovom posljednjem slučaju mogu se koristiti i dodatne ključne riječi.

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

Za poboljšanje preglednosti tablice moguće je postaviti drugačiju boju pozadine u ćelijama zaglavlja:

```
th
{
    background-color: darkgray;
}
```

Boja pozadine može se postaviti na ćeliji (kao u gornjem primjeru), ali i na cijelom retku.

Da bi se lakše vizualno odvojili pojedini retci, može se parnim i neparnim retcima postaviti različita boja pozadine. To se može postići pomoću pseudoklase *nth-child*, koju treba navesti uz parametar *odd* da bi se selektirao svaki parni redak:

```
tr:nth-child(odd)
{
    background-color: #ebebeb;
}
```

Naslov	Autor	Cijena (kn)
Vuk koji je htio obići svijet	Orianne Lallemand	39,90
Zec Mrkvus Veliki	Kęstutis Kasparavičius	94,50
Matilda i zagonetne žabe	Simeon Marinković	65,00

4.5. Slike

Slike se u HTML-u dodaju pomoću elementa *img*. Na *web*-stranicama najčešće se prikazuju uz tekst i postoji više načina na koji se slika može uklopiti u tekst.

Kao primjer može poslužiti ovaj HTML-kôd:

```
<p>
    
    Zašao neki momak u šumu Striborovu
    ...
</p>
```

Prema tipu prikaza element *img* je linijski element pa se slika inicijalno prikazuje u jednom retku teksta:

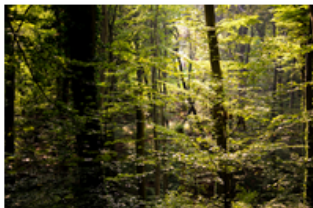


Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda

Da bi se slika smjestila unutar teksta, koristi se svojstvo **float**. To svojstvo izbacuje element iz normalnog toka dokumenta i postavlja ga uz lijevi (ili desni) rub roditeljskog elementa. Tekst i drugi *inline*-elementi koji se nalaze pored elementa kojem je postavljeno svojstvo *float* postavljaju se i prelamaju oko njega.

Ovako se slika postavlja uz lijevi rub, a tekst teče oko slike i prelama se oko nje:

```
img
{
    float: left;
}
```



Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda dobra, ali i naopaka - svakome po zaslugi. Morala je pak ta šuma ostati začarana, dokle god u nju ne stupi onaj, kojemu je milija njegova nevolja, nego sva sreća ovoga svijeta. Nasjekao dakle onaj momak drva i sjeo na panj, da počine, jer bijaše lijep zimski dan. Ali iz panja iziđe pred njega zmija i stade se umiljavati oko njega. Ono pak ne bijaše prava zmija, nego bijaše ljudska duša, radi grijeha i zlobe ukleta, a mogao ju je osloboditi samo onaj, koji bi se s njom vjenčao.

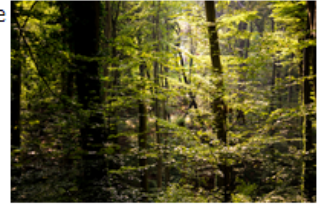
Svojstvo *float* može poprimiti ove vrijednosti:

Ključna riječ	Objašnjenje
<i>left</i>	element je postavljen uz lijevi rub roditeljskog elementa
<i>right</i>	element je postavljen uz desni rub roditeljskog elementa
<i>none</i>	element je u normalnom toku dokumenta (inicijalna vrijednost)

Bez obzira na to što se element *img* u HTML-kôdu nalazi prije teksta, ako se vrijednost svojstva *float* postavi na *right*, tekst se prikazuje na lijevoj strani, a slika na desnoj:

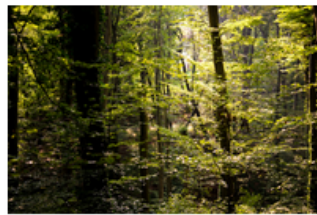
```
img
{
    float: right;
}
```


Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda dobra, ali i naopaka - svakome po zasluži. Morala je pak ta šuma ostati začarana, dokle god u nju ne stupi onaj, kojemu je milija njegova nevolja, nego sva sreća ovoga svijeta. Nasjekao dakle onaj momak drva i sjeo na panj, da počine, jer bijaše lijep zimski dan. Ali iz panja iziđe pred njega zmija i stade se umiljavati oko njega. Ono pak ne bijaše prava zmija, nego bijaše ljudska duša, radi grijeha i zlobe ukleta, a mogao ju je osloboditi samo onaj, koji bi se s njom vjenčao.



Da bi se slika malo odmaknula od teksta, mogu joj se dodati margine:

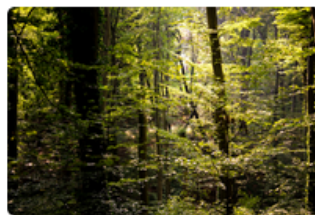
```
img
{
    float: left;
    margin: 5px 15px 10px 0;
}
```



Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda dobra, ali i naopaka - svakome po zasluži. Morala je pak ta šuma ostati začarana, dokle god u nju ne stupi onaj, kojemu je milija njegova nevolja, nego sva sreća ovoga svijeta. Nasjekao dakle onaj momak drva i sjeo na panj, da počine, jer bijaše lijep zimski dan. Ali iz panja iziđe pred njega zmija i stade se umiljavati oko njega. Ono pak ne bijaše prava zmija, nego bijaše ljudska duša, radi grijeha i zlobe ukleta, a mogao ju je osloboditi samo onaj, koji bi se s njom vjenčao.

Ako se slika želi dodatno oblikovati, moguće joj je npr. dodati zaobljene rubove:

```
img
{
    float: left;
    margin: 5px 15px 10px 0;
    border-radius: 5px;
}
```



Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda dobra, ali i naopaka - svakome po zasluži. Morala je pak ta šuma ostati začarana, dokle god u nju ne stupi onaj, kojemu je milija njegova nevolja, nego sva sreća ovoga svijeta. Nasjekao dakle onaj momak drva i sjeo na panj, da počine, jer bijaše lijep zimski dan. Ali iz panja iziđe pred njega zmija i stade se umiljavati oko njega. Ono pak ne bijaše prava zmija, nego bijaše ljudska duša, radi grijeha i zlobe ukleta, a mogao ju je osloboditi samo onaj, koji bi se s njom vjenčao.

Ako se slika želi dodatno istaknuti, može joj se dodati rub. Da bi on bio vidljiviji, može se odvojiti od slike pomoću *paddinga*:

```
img
{
    float: left;
    margin: 5px 15px 10px 0;
    border-radius: 5px;
    border: 1px solid gray;
    padding: 5px;
}
```



Zašao neki momak u šumu Striborovu, a nije znao da je ono šuma začarana i da se u njoj svakojaka čuda zbivaju. Zbivala se u njoj čuda dobra, ali i naopaka - svakome po zasluži. Morala je pak ta šuma ostati začarana, dokle god u nju ne stupi onaj, kojemu je milija njegova nevolja, nego sva sreća ovoga svijeta. Nasjekao dakle onaj momak drva i sjeo na panj, da počine, jer bijaše lijep zimski dan. Ali iz panja iziđe pred njega zmija i stade se umiljavati oko njega. Ono pak ne bijaše prava zmija, nego bijaše ljudska duša, radi grijeha i zlobe ukleta, a mogao ju je osloboditi samo onaj, koji bi se s njom vjenčao.

5. Pozicioniranje elemenata

Pozicioniranje HTML-elemenata pomoću CSS-a zapravo je najteži dio CSS-a, a glavni razlog tome je što dosad CSS nije na jednostavan način omogućivao pozicioniranje kakvo su *web*-dizajneri željeli, odnosno trebali.

Željeni raspored HTML-elemenata na *web*-stranici može se postići na više različitih načina, no složeniji raspored ponekad zahtijeva i uporabu nekih trikova.

U trenutku pisanja ovog tečaja razvijaju se nove mogućnosti pozicioniranja u CSS-u – modeli *flexbox* i *grid*, čime će konačno pozicioniranje u CSS-u biti riješeno na zadovoljavajući način.

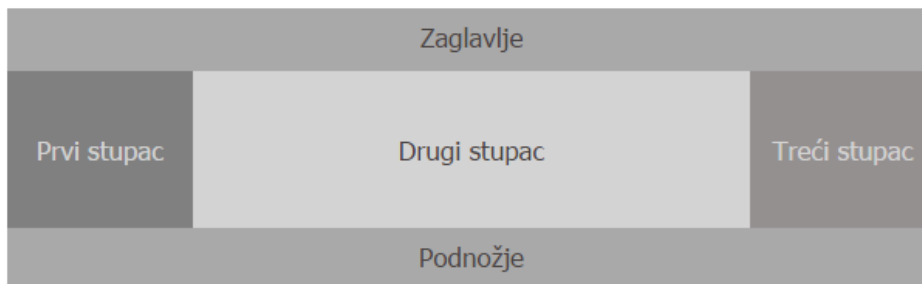
5.1. Pozicioniranje pomoću elemenata *inline-block*

Blok-elementi zauzimaju cijeli redak, čak i ako im se postavi manja širina. Budući da oni uvijek uzrokuju prebacivanje sljedećeg elementa u novi red, nije moguće postaviti dva blok-elementa u istom retku.

Za razliku od njih, linijski elementi neće uzrokovati prebacivanje u novi red, ali njima se ne mogu podešavati širina i visina.

Postavljanjem svojstva *display* na vrijednost *inline-block*, elementu je moguće zadati širinu i visinu, a on neće uzrokovati prelazak u novi red. Tako se u istom redu može imati više elemenata, tj. taj se red može podijeliti na više stupaca.

Jedan od najčešće korištenih rasporeda na *web*-stranicama sastoji se od zaglavlja, središnjeg dijela koji je podijeljen u tri stupca i podnožja elementa.



HTML-kôd takve *web*-stranice mogao bi izgledati ovako:

```
<div class="zaglavlje">
  Zaglavlje
</div>

<div class="prvi">
  Prvi stupac
</div>

<div class="drugi">
  Drugi stupac
</div>
```

Zanimljivosti i napomene

Inline-block elementi podržani su tek od Internet Explorera 8, pa ih se ne može koristiti za raspored stranice u ranijim preglednicima.

```
<div class="treći">
  Treći stupac
</div>

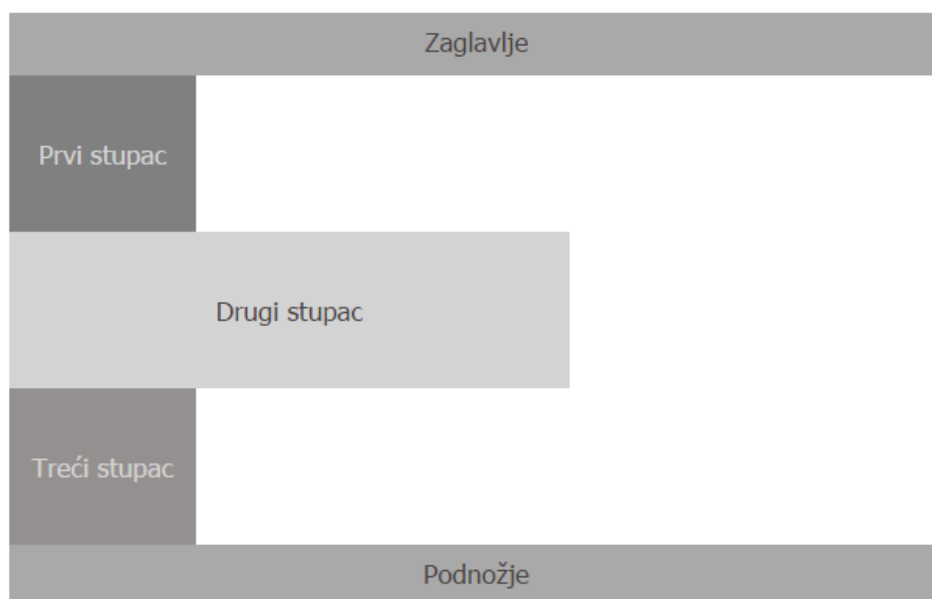
<div class="podnozje">
  Podnožje
</div>
```

U ovom se slučaju elementi koji imaju klasu prvi, drugi i treći žele prikazati u istom stupcu. Da bi se to postiglo, najprije im treba postaviti širinu: prvi i treći stupac imat će širinu od 20 %, a drugi stupac zauzimat će preostalih 60 % prostora.

```
.prvi, .treći
{
  width: 20%;
}

.drugi
{
  width: 60%;
}
```

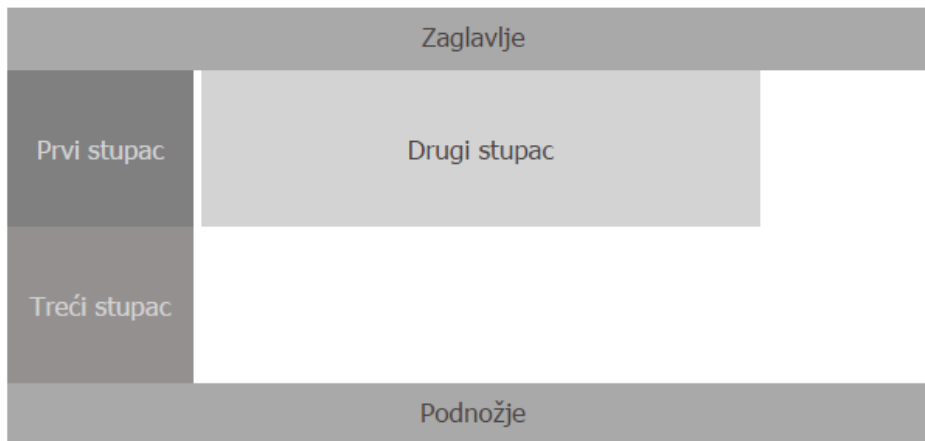
Svaki blok-element zauzima cijeli redak pa se, unatoč tome što mu je postavljena širina, dobiva ovakav rezultat:



Elementima koji se žele smjestiti kao stupci treba promijeniti tip prikaza u *inline-block*:

```
.prvi, .drugi, .treći
{
  display: inline-block;
}
```

No rezultat koji se dobije ipak nije zadovoljavajuć:



Prvi i drugi stupac zaista se nalaze u istom redu, no čini se da se između stupaca nalazi određeni razmak pa stoga nije bilo dovoljno mjesta da bi i treći stupac bio u istom retku.

Ovaj se razmak pojavljuje zbog toga što se elementi *inline-block* ipak djelomično ponašaju kao linijski elementi pa kada postoji bilo kakav razmak između njih u HTML-kôdu (pa makar i prelazak u novi red), taj je razmak vidljiv u pregledniku.

Da bi se izbjegao taj razmak, HTML-kôd se mora pisati tako da nakon završetka jednog elementa *div* odmah započinje sljedeći:

```
<div class="prvi">
  Prvi stupac
</div><div class="drugi">
  Drugi stupac
</div><div class="treći">
  Treći stupac
</div>
```

Tako se postiže željeni raspored elemenata (uz malu žrtvu preglednosti HTML-kôda).

U slučaju da se (donekle) želi zadržati preglednost HTML-kôda, moguće je osigurati da ne bude razmaka umetanjem HTML-ovih oznaka za komentare između elemenata *div*, kao u ovom primjeru:

```
<div class="prvi">
  Prvi stupac
</div><!--
--><div class="drugi">
  Drugi stupac
</div><!--
--><div class="treći">
  Treći stupac
</div>
```

5.2. Pozicioniranje pomoću svojstva float

Osnovna namjena svojstva *float* je prelamanje teksta oko slike, kao što je prikazano u prošlom poglavlju.

No svojstvo *float* može se rabiti i za postizanje željenog rasporeda elemenata na stranici. Blok-elementi kojima se postavi svojstvo *float* neće uzrokovati prelazak u novi red, nego će ih biti moguće smjestiti jednog do drugog u istom retku.

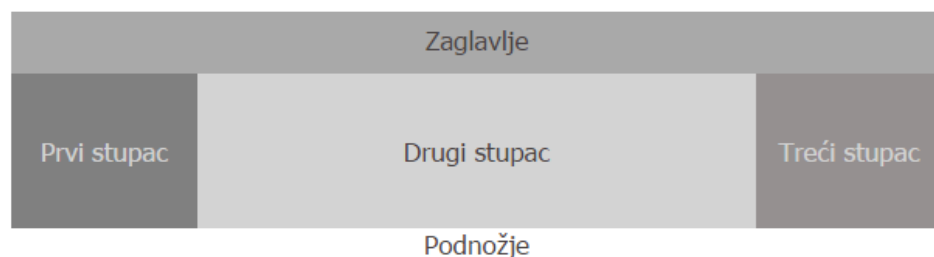
Podsjetimo se, element kojem se postavi svojstvo *float* na vrijednost *left* bit će pozicioniran uz lijevi rub roditeljskog elementa. Ako se i sljedećem elementu postavi svojstvo *float* na *left*, on će se pozicionirati uz rub prethodnog elementa.

Na elemente koji se želi pozicionirati kao stupci postaviti će se stoga svojstvo *float* na vrijednost *left*:

```
.prvi, .drugi, .treći
{
    float: left;
}
```

Kod korištenja svojstva *float* ne treba se brinuti o razmaku između elemenata u HTML-kôdu, ali postoje drugi problemi koji se mogu pojaviti. Elementi koji slijede iza elemenata sa svojstvom *float* mogu se ponašati neočekivano.

U ovom primjeru, *div* s klasom *podnožje* pozicionirao se uz donji rub elementa *div* s klasom *zaglavlje* i više nije vidljiv, a njegov je tekst ostao ispod elemenata sa svojstvom *float*.



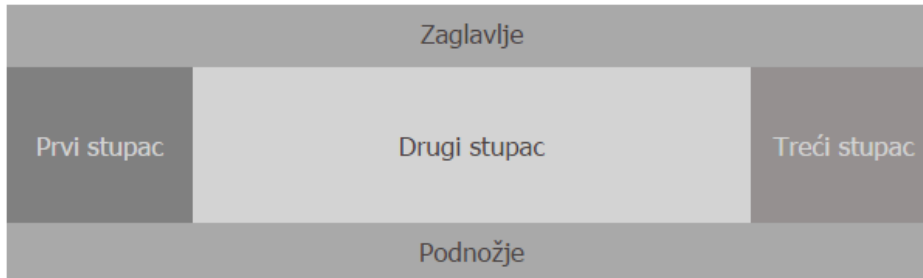
Da bi se izbjegli takvi problemi, u većini je slučajeva dovoljno postaviti svojstvo **clear** na prvi sljedeći element koji dolazi iza elemenata sa svojstvom *float*. Svojstvom *clear* isključuje se djelovanje svojstva *float* na elemente koji slijede i ponovno se nastavlja s uobičajenim pozicioniranjem elemenata na stranici.

Svojstvo *clear* može se postaviti na ove ključne riječi:

Ključna riječ	Objašnjenje
<i>left</i>	prestaje <i>floating</i> elemenata na lijevu stranu
<i>right</i>	prestaje <i>floating</i> elemenata na desnu stranu
<i>both</i>	prestaje <i>floating</i> elemenata na ijednu stranu

Postavljanjem svojstva *clear* na vrijednost *left* na elementu s klasom *podnozje* dobit će se željeni raspored elemenata na stranici.

```
.podnozje
{
    clear: left;
}
```



5.3. Model pozicioniranja *flexbox*

Model *flexbox* novi je model pozicioniranja koji je tek u posljednje vrijeme standardizirao W3C, a podržan je u novim verzijama svih glavnih preglednika.

Namjera tog modela riješiti probleme i nedostatke koje imaju druge metode pozicioniranja.

Budući da je taj model složen i da nudi brojne mogućnosti, u ovom se priručniku neće obraditi u potpunosti, nego će se samo pokazati kako se pomoću njega može postići raspored elemenata prikazan u prethodnim primjerima.

Jedan od problema koji dosadašnje metode pozicioniranja imaju je da visina stupaca neće biti jednaka, osim ako se eksplicitno ne postavi na željenu vrijednost. Ako količina sadržaja u stupcima varira, nije moguće postaviti fiksnu visinu pa će neki stupci uvijek biti kraći od drugih. Postoji više složenih tehnika koje rješavaju taj problem, no kod modela *flexbox* stupci će uvijek imati istu visinu (osim ako im se eksplicitno ne postavi različita visina).

Da bi se dobili stupci pomoću modela *flexbox*, elementi koji se žele pozicionirati kao stupci trebaju biti sadržani unutar zasebnog roditeljskog elementa:

```
<div class="zaglavlje">
    Zaglavlje
</div>

<div class="sadrzaj">
    <div class="prvi">
        Prvi stupac
    </div>

    <div class="drugi">
        Drugi stupac
    </div>
```

Zanimljivosti i napomene

Uz *flexbox* model, u pripremi je još jedan novi model pozicioniranja – *grid* model, no on u trenutku pisanja ovog tečaja još nije standardiziran, a podržan je samo u Internet Exploreru 11.

```

        <div class="treći">
            Treći stupac
        </div>
</div>
<div class="podnozje">
    Podnožje
</div>

```

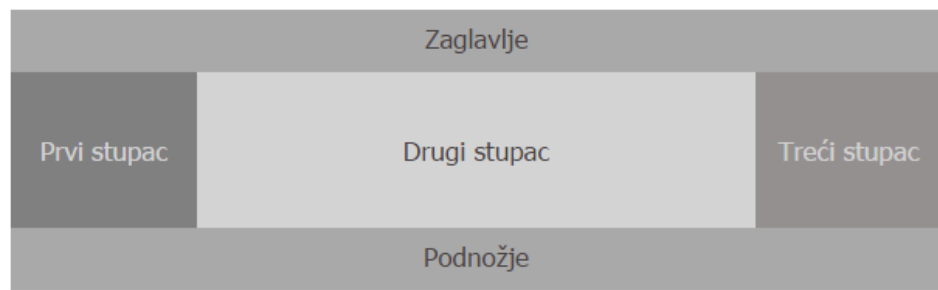
Sve što je potrebno napraviti je tom roditeljskom elementu postaviti svojstvo *display* na vrijednost *flex*:

```

.sadrzaj
{
    display: flex;
}

```

Tako je postavljen tip prikaza prema modelu *flexbox* za taj element i za elemente unutar njega, koji se (ako se ne definira drugačije) ponašaju kao stupci.



5.4. Pozicioniranje pomoću svojstva *position*

HTML-elementi u prozoru se preglednika prikazuju redoslijedom kojim su navedeni u HTML-kôdu. Blok-elementi (npr. elementi *div* i *p*) zauzimaju cijeli redak i prebacuju sljedeći element u novi red. Linijski elementi (npr. elementi *span* i *em*) ne zauzimaju cijeli redak, već samo onoliko mjesta koliko zauzima njihov sadržaj.

To je normalni slijed (ili tijek) elemenata u HTML-u. Elementi slijede jedan iza drugoga istim redoslijedom kojim su napisani. No pomoću CSS-svojstva ***position*** moguće je izbaciti željeni element iz tog slijeda i staviti ga na drugi položaj na stranici, koji ne mora imati veze s njegovim redoslijedom u HTML-kôdu.

Svojstvo *position* može se postaviti na četiri osnovne vrijednosti: *static*, *relative*, *absolute* i *fixed*.

Vrijednost *static*

To je inicijalna vrijednost koju ima svojstvo *position*, ako nije promijenjeno u neku od ostalih vrijednosti. Ključna riječ *static* označava da se element nalazi unutar normalnog slijeda elemenata.

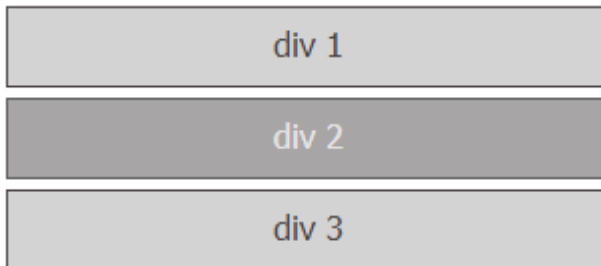
```
position: static;
```


U praksi će se ta vrijednost eksplicitno navoditi samo kad se želi isključiti druga vrijednost (postavljena pomoću drugog CSS-pravila).

Za primjer može poslužiti HTML-kôd koji se sastoji od elementa *div*:

```
<div>div 1</div>
<div class="drugi">div 2</div>
<div>div 3</div>
```

Ako ta tri elementa *div* imaju inicijalnu vrijednost svojstva *position* (dakle *static*), pozicionirat će se ovako:



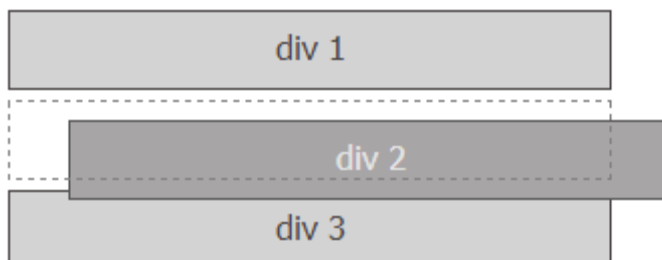
Vrijednost *relative*

Postavljanjem vrijednosti *relative* za svojstvo *position* dopušta se da element bude pomaknut u odnosu na mjesto koje bi inače zauzimao. Pomak elementa definira se pomoću CSS-svojstava ***top***, ***right***, ***bottom*** i ***left***.

```
.drugi
{
    position: relative;
    top: 10px;
    left: 30px;
}
```

Tim svojstvima postavlja se iznos pomaka u jedinicama duljine ili u postocima (postotak se odnosi na širinu elementa kod svojstava *right* i *left*, a kod svojstava *top* i *bottom* na visinu elementa.) Osim vrijednosti, tom se svojstvu može postaviti i ključna riječ *auto*.

Deklaracijom *top: 10px* element se pomiče **odozgo** za 10 piksela prema dolje, a deklaracijom *left: 30 px* element se pomiče **slijeva** za 30 piksela udesno.



Za definiranje pomaka dovoljno je navesti samo dva svojstva: *top* i *left* ili *bottom* i *right*.

Zanimljivosti i napomene

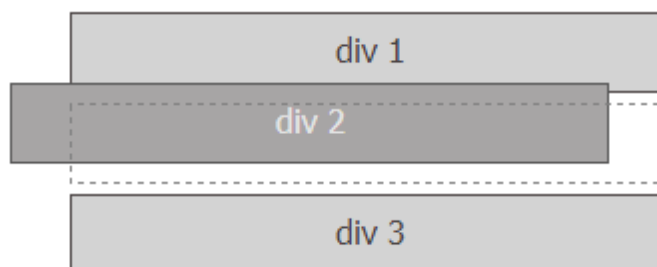
Svojstvima *top*, *right*, *bottom* i *left* mogu se predati i negativne vrijednosti, što uzrokuje pomak u suprotnom smjeru. Deklaracija *left:-30px* pomaknut će element za 30 piksela zdesna ulijevo, što je isti rezultat koji se dobije navođenjem *right:30px*.

Zanimljivosti i napomene

Postavljanje svojstava *top*, *right*, *bottom* i *left* na elementu kod kojeg svojstvo *position* ima vrijednost *static* neće imati nikakav učinak.

Navođenjem *bottom: 10px* element se pomiče **odozdo** za 10 piksela prema gore, a navođenjem *right: 30px* se element pomiče **zdesna** za 30 piksela ulijevo.

```
.drugi
{
    position: relative;
    bottom: 10px;
    right: 30px;
}
```



Važno je napomenuti da kad se koristimo vrijednošću *relative*, prostor koji bi pomaknuti element inače zauzimao (na slici prikazan crtkano) ostaje sačuvan, a element koji slijedi iza njega ne pomiče se sa svojeg originalnog položaja da bi popunio taj slobodni prostor. Kod preostalih vrijednosti svojstva *position* (*absolute* i *fixed*) to nije tako.

Vrijednost *absolute*

Korištenjem vrijednosti *absolute* element se također pomiče iz slijeda elemenata na stranici, a pomoću svojstava *top*, *right*, *bottom* ili *left* određuje se pomak elementa u odnosu na ishodišni element.

Ishodišni element u odnosu na koji se element pomiče inicijalno je element *body*. Ako je potrebno, kao ishodišni se element može postaviti i drugi element.

```
.drugi
{
    position: absolute;
    top: 25px;
    left: 180px;
}
```



Pomoću svojstva *top* navodi se udaljenost za koju se element pomiče u odnosu na **gornji** rub ishodišnog elementa, a pomoću svojstva *left* u odnosu na **lijevi** rub ishodišnog elementa.

Ako se rabi svojstvo *bottom*, element se pomiče za navedenu udaljenost u odnosu na **donji** rub elementa, a ako se koristi svojstvo *right*, u odnosu na **desni** rub elementa. Ako je ishodišni element *body*, pomak

će biti u odnosu na donji, odnosno desni rub prozora preglednika (jer element *body* zauzima upravo prostor preglednika).

U primjeru se vidi da kad se rabi vrijednost *absolute*, prostor koji je zauzimao pomaknuti element nije sačuvan, nego se na njega smjestio sljedeći element u nizu.

Kao ishodišni element, umjesto elementa *body*, može se postaviti neki od roditeljskih elemenata našeg elementa koji se želi pomaknuti. Postavlja se tako da se roditeljevo svojstvo *position* postavi na *relative*.

HTML-kôd izmijenit će se tako da se unutar drugog elementa *div* stavi novi element *div*:

```
<div>div 1</div>
<div class="drugi">
  div 2
  <div>div 4</div>
</div>
<div>div 3</div>
```

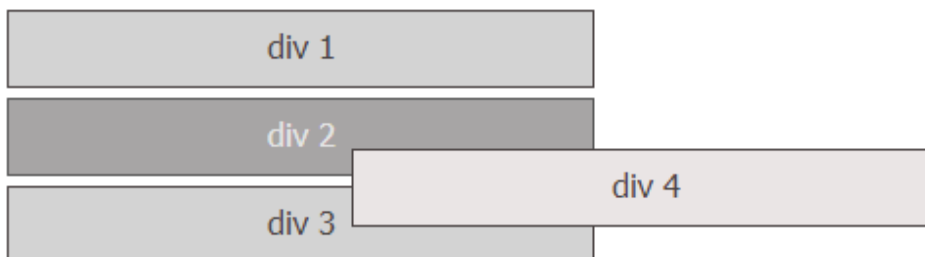
Drugom elementu *div* postaviti će se svojstvo *position* na *relative*, čime će se on postaviti za ishodišni element:

```
.drugi
{
  position: relative;
}
```

Novom elementu *div* (koji je dijete drugog) postaviti će se svojstvo *position* na *absolute* i definirati mu se pomak u odnosu na drugi element pomoću svojstava *top* i *left*:

```
.drugi div
{
  position: absolute;
  top: 20px;
  left: 170px;
}
```

Rezultat će izgledati ovako:



Vrijednost *fixed*

Vrijednost *fixed* slična je vrijednosti *absolute*. I ovdje se element pomiče u odnosu na ishodišni element, s tom razlikom da ishodišni element može biti samo element *body*.

Kao i kod vrijednosti *absolute*, i ovdje prostor koji bi inače zauzimao pomaknuti element neće biti sačuvan.

Zanimljivosti i napomene

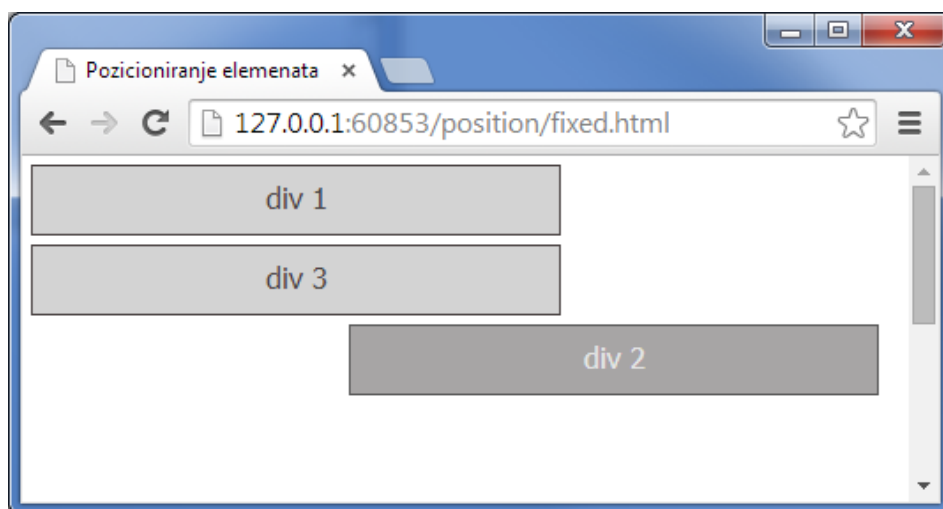
Vrijednost *fixed* podržana je tek u relativno novijim preglednicima: tek od Internet Explorera 8, Firefoxa 30 i Chromea 27.

Razlika je u tome što je element pozicioniran na ovaj način „fiksiran“ na svojem položaju i za razliku od ostalih elemenata ostaje na istom mjestu čak i kad korisnik okomito pomiče stranicu u pregledniku.

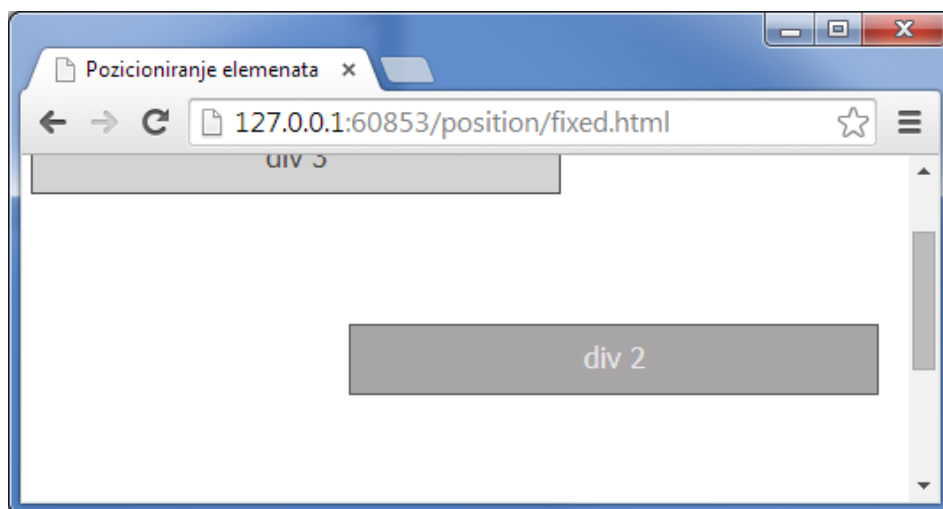
Ako se tako pozicionira element *div* s klasom *drugi*:

```
.drugi
{
    position: fixed;
    top: 90px;
    left: 180px;
}
```

element će se inicijalno pozicionirati ovako:



Pomakne li se dio stranice koji vidimo u prozoru preglednika prema dolje, element „div 2“ ostat će fiksiran na istom mjestu:



Na kraju, evo još jednom pregleda ključnih riječi na koje se može postaviti svojstvo *position*:

Ključna riječ	Objašnjenje
<i>static</i>	element se nalazi u normalnom slijedu elemenata
<i>relative</i>	element je pomaknut u odnosu na svoj inicijalni položaj u normalnom slijedu elemenata
<i>absolute</i>	element se pomiče u odnosu na element <i>body</i> ili u odnosu na najbliži roditeljski element za koji svojstvo <i>position</i> ima vrijednost <i>relative</i> (ako takav postoji)
<i>fixed</i>	element se pomiče u odnosu na element <i>body</i> , a prilikom okomitog pomicanja stranice u pregledniku ostaje na istom mjestu

5.5. Vježba: Pozicioniranje i oblikovanje specifičnih elemenata

Pozicioniranje

1. U mapi C220/Vjezbe/css otvorite datoteku *pozicioniranje.css*. Rezultat napisanih CSS-pravila može se vidjeti pregledavanjem datoteka *index.html* („Početna“) i *tecajevi.html* („Tečajevi“) u pregledniku. Za stranicu „Početna“ želimo dobiti ovakav rezultat:



2. Elemente koji imaju klasu *lijevi*, *srednji* i *desni* potrebno je prikazati kao stupce pa im postavite svojstvo *display* na vrijednost *inline-block*. Nakon te promjene svojstvo *vertical-align* postavite im na vrijednost *top*, kako bi sadržaj u njima bio okomito poravnat uz vrh elementa.
3. Elementu s klasom *lijevi* postavite širinu na 20 %, elementu s klasom *srednji* na 55 %, a elementu s klasom *desni* na 25 %.
4. Elementima s klasom *lijevi* i *srednji* potrebno je još postaviti desni *padding* na 20 piksela (da se njihov sadržaj ne bi dodirivao).

Linkovi

5. U mapi C220/Vjezbe/css otvorite datoteku *oblikovanje.css*. Ovdje se mogu dodati željena pravila za ponašanje linkova (pored postojećih definicija za boju i veličinu teksta u linkovima). Za sve linkove postavite pojavljivanje crte ispod teksta na korisnikov prelazak mišem koristeći se pseudoklasom *hover* i svojstvom *text-decoration*.

6. Za linkove koji se nalaze u elementu s klasom *podnozje* postavite obrnuto ponašanje (skrivanje crte), jer su ti linkovi već podcrtani. I njih možete selektirati pomoću pseudo klase *hover*.

Slike

7. U datoteku *oblikovanje.css*, možemo dodati i željeno pravilo za izgled slike (koje će biti primijenjeno na sliku na stranici *srce.html*, odnosno „O Srcu“). Za sliku (element *img*) koji se nalazi u elementu s klasom *clanak* postavite svojstvo *float* na vrijednost *left*, kako bi se tekst prelomio oko slike. Također postavite desnu marginu od 10 piksela da se odvoji slika od teksta te puni rub debljine 1 piksela i sive boje (#ccc) koji treba zaobliti s radijusom zakrivljenja od 5 piksela. Rub odvojite od slike pomoću *paddinga* od 3 piksela te dodajte sjenu pomoću svojstva *box-shadow* s pomakom od 3px, radijusom zamućenja od 5px i neka bude sive boje (#ccc).

Dodatni zadatci

Pozicioniranje

8. Na stranici „Tečajevi“ želimo postići malo drugačiji raspored elemenata. Elemente s klasom *tecaj* želimo slagati u redove od po dva elementa:



9. To se lako može postići pomoću prikaza *inline-block* pa najprije (u datoteci *pozicioniranje.css*) postavite svojstvo *display* na *inline-block* za elemente s klasom *tecaj*. Uz to, i ovdje treba postaviti svojstvo *vertical-align* na vrijednost *top*. Postavite minimalnu visinu

elementa na 170 piksela da bi svi elementi bili jednake visine (u slučaju da neki element nema dovoljno sadržaja). Širinu elementa postavite na 250 piksela tako da dva elementa stanu u dostupni prostor. Dodajte i donju marginu od 20 piksela kako bi postojao vodoravni razmak između elemenata.

10. Okomiti razmak između elemenata može se dobiti postavljanjem desne margine na elemente s klasom *tecaj*. No marginu treba staviti samo na elementima koji se nalaze na lijevoj strani (jer na desnoj strani već postoji razmak). Radi se zapravo o svakom drugom elementu pa je moguće upotrijebiti pseudoklasu *nth-child(even)*, pomoću koje možemo selektirati svaki parni element (ne svaki neparni jer brojanje počinje od elementa *h2*, koji se također nalazi u istom roditeljskom elementu *div*). Pomoću te pseudoklase postavite desnu marginu od 20 piksela na elemente s klasom *tecaj*, ali samo na one koji se nalaze na lijevoj strani (tj. na svaki parni element).

Obrasci

11. U mapi C220/Vježbe/css otvorite datoteku *obraci.css*. Rezultat se može vidjeti pregledavanjem datoteke *prijava.html* („Prijava“) u pregledniku.
12. Da bismo pozicionirali elemente *input* svaki u svoj redak, najprije svim elementima *input* promijenite svojstvo *display* u *block*. Time će i elementi *label* biti prebačeni svaki u svoj red. Osim toga, postavite im širinu na 200 piksela, visinu na 25 piksela, *padding* na 3 piksela i donju marginu na 10 piksela (da bi se napravio razmak između elemenata). Na elementima *input* postavite još i tamnosivi (#6d6c6c) rub debljine 1 piksela i zaoblite ga s radijusom od 5 piksela.
13. Na elementima *label* postavite drugu boju (npr. tamnosivu – #6d6c6c) te im postavite vrstu fonta na *NeoSansMedium* (ako ste prije uključili *web-fontove*).
14. Za dugme treba definirati specifičan izgled. Postavite mu pozadinski prijelaz iz svijetlocrvene (#f7dcdc) u tamnocrvenu (#c30f10), različitu boju ruba (npr. tamnocrvenu) te mu podesite visinu na 35, širinu na 120. Dodatno ga pozicionirajte udesno postavljajući mu lijevu marginu na 80 piksela.

Tablice

15. U mapi C220/Vježbe/css otvorite datoteku *tablice.css*. Rezultat se može vidjeti pregledavanjem datoteke *raspored.html* („Raspored“) u pregledniku.
16. Najprije treba definirati osnovno svojstvo tablice da nema razmaka između ćelija i da su susjedni rubovi spojeni u jednu crtu, tako da za element *table* postavite svojstvo *border-collapse* na vrijednost *collapse*.
17. Na elementima *td* i *th* definirajte rub kao punu crtu od 1 piksela, tamnosive boje (#6d6c6c). Dodatno postavite i *padding* (okomiti od 5 i vodoravni od 3 piksela) te veličinu fonta na 12px.

18. Na elementu *th* postavite boju fonta na tamnosivu (#6d6c6c) te vrstu fonta na *NeoSansMedium* (ako ste prije uključili *web-fontove*). Zatim postavite boju pozadine na svijetlocrvenu (#f7dcdc) te visinu elementa na 30 piksela.
19. U stupcu „Trajanje“ tekst u ćeliji poravnajte udesno, a zatim dodajte desni *padding* od 5 piksela. Četvrtu ćeliju u retku stupac selektirajte pomoću pseudoklase *nth-child(4)*.
20. Svaki drugi redak (najbolje svaki neparni) ispunite različitom pozadinskom bojom da bi se povećala preglednost tablice. Pozadinsku boju postavite i na samom retku (tj. elementu *tr*), a svaki neparni redak selektirajte pomoću pseudoklase *nth-child(odd)*.