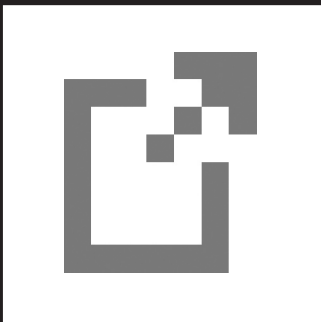


4 OBLIKOVANJE VEZA



[This is an external link](#) 

| | |
|-------------------------|-------------------------------------|
| Andy Budd's Blogography | <input checked="" type="checkbox"/> |
| Stuff and Nonsense | <input type="checkbox"/> |
| Hicks Design | <input type="checkbox"/> |
| Clagnut | <input type="checkbox"/> |
| HTML Dog | <input checked="" type="checkbox"/> |
| Adactio | <input type="checkbox"/> |
| All In The Head | <input type="checkbox"/> |
| Mark Boulton | <input type="checkbox"/> |
| Ian Lloyd | <input type="checkbox"/> |

Skromna veza do sidra (engl. *anchor link*) temelj je World Wide Weba. To je mehanizam koji omogućuje međusobno povezivanje Web stranica, a korisnicima istraživanje i kretanje kroz njih. Podrazumijevano oblikovanje veza prilično je nenadahnuo, no s malo CSS-a možete učiniti zadivljujuće stvari.

U ovom ćete poglavlju učiti o:

- Redoslijedu selektora veza temeljenom na kaskadi
- Izradi stiliziranog podvlačenja veza
- Izradi vanjskih veza upotrebom selektora atributa
- Izradi veza koje se ponašaju kao gumbi
- Izradi stilova posjećenih veza
- Izradi pravih CSS podsjetnika za alate

Jednostavno oblikovanje veza

Najlakši način oblikovanja veze je upotreba selektora tipa sidra. Na primjer, ovo će pravilo sva sidra obojati crveno:

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

Međutim, sidra mogu služiti kao unutarnje reference baš kao i vanjske veze pa upotreba selektora tipa nije uvijek idealna. Kao primjer uzmite ovu situaciju. Prvo sidro sadrži identifikator fragmenta pa kad korisnik pritisne to sidro, stranica će se pomaknuti do drugog imenovanog sidra:

```
<p><a href="#mainContent">Skip to main content</a></p>
...
<h1><a name="mainContent">Welcome</a></h1>
```

Iako biste vjerojatno željeli da samo veza bude oblikovana crvenom bojom, sadržaj naslova također će biti oblikovan crvenom bojom. Da bi se to izbjeglo, CSS pruža dva posebna selektora koji se nazivaju selektorima pseudoklase veze. Selektor pseudoklase `:link` koristi se za ciljanje veza koje nisu bile posjećene, a selektor pseudoklase `:visited` za ciljanje posjećenih veza. Tako će u ovom primjeru sve neposjećene veze biti plave, a sve posjećene veze zelene:

```
a:link {color: blue;} /* Neposjećene veze su plave */
a:visited {color: green;} /* Posjećene veze su zelene */
```

Ostala dva selektora koja možete upotrijebiti za oblikovanje veza su dinamički selektori pseudoklase `:hover` i `:active`. Dinamički selektor pseudoklase `:hover` služi za ciljanje elemenata kad se iznad njih nalazi pokazivač miša, a dinamički selektor pseudoklase `:active` cilja elemente kad su aktivirani. U slučaju veza, do aktiviranja dolazi kad se veza pritisne. Tako će u ovom primjeru veza postati crvena kada se iznad nje postavi pokazivač miša ili kada se pritisne:

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

Jedna od prvih stvari za koju većina dizajnera nauči koristiti ove selektore jest isključivanje podvlačenja veza, a zatim njegovo ponovno uključivanje kada se iznad veze posatvi pokazivač miša ili kada se pritisnu. To se može učiniti postavljanjem svojstva `text-decoration` na `none` za neposječene i posječene veze, a na `underline` za veze iznad kojih je pokazivač i aktivne veze:

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

U prethodnom je primjeru redoslijed selektora vrlo važan. Ukoliko je redoslijed obrnut, stil za oblikovanje kada je pokazivač iznad veze i aktivni stil neće raditi:

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

Razlog tome je kaskada. U poglavlju 1 naučili ste da, kada dva pravila imaju jednaku specifičnost, prednost ima pravilo koje je definirano posljednje. U ovoj situaciji oba pravila imaju jednaku specifičnost pa će stilovi `:link` i `:visited` premostiti stilove `:hover` i `:active`. Da biste bili sigurni kako se to neće dogoditi, preporučljivo je primjenjivati stilove veza sljedećim redoslijedom:

```
a:link, a:visited, a:hover, a:active
```

Ovaj redoslijed lako se može zapamtiti s pomoću mnemotehničkog izraza `LoVe:HAtE`, gdje *L* znači `link` (veza), *V* `visited` (posječena), *H* `hover` (kada je pokazivač miša iznad veze), a *A* `active` (aktivna).

Zabava s podvlačenjem

S gledišta upotrebljivosti i pristupačnosti, važno je da se veze ne razlikuju samo po boji. Razlog je u tome što je mnogim ljudima s oslabljenim vidom teško razlikovati slabo kontrastne boje, posebice kod sitnog teksta. Na primjer, ljudi koji ne razaznaju boje ne mogu razlikovati određene kombinacije boja pri sličnim razinama svjetloće ili zasićenja. Zato su veze podrazumjevano podvučene.

Dizajnerima se obično ne sviđa podvlačenje veza jer ono stranici dodaje preveliku težinu i vizualnu zbrku. Odlučite li ukloniti podvlačenje veza, možda ćete umjesto njega odabrati podebljavanje veza. Na taj će način stranica izgledati manje natrpana, a veze će se i dalje isticati:

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

Podvlačenje možete ponovno primijeniti kad se iznad veze postavi pokazivač miša ili se ona aktivira, što ističe njihovu mogućnost interakcije:

```
a:hover, a:active {
  text-decoration: underline;
}
```

Moguće je i manje upadljivo podvući veze zadavanjem rubova. U sljedećem je primjeru podrazumijevana crta uklonjena i zamijenjena manje upadljivom isprekidanom crtom. Kada se iznad veze postavi pokazivač miša ili kada je posjetitelj pritisne, ona se pretvori u punu crtu da bi se korisniku pružila vizualna informacija da se nešto dogodilo:

```
a:link, a:visited {
  text-decoration: none;
  border-bottom: 1px dotted #000;
}

a:hover, a:active {
  border-bottom-style: solid;
}
```

Maštovito podvlačenje veza

Možete postići neke vrlo zanimljive efekte upotrebom slika za podvlačenje veza. Na primjer, izradio sam vrlo jednostavnu grafiku za podvlačenje koja se sastoji od poprečnih crtica (slika 4-1).



Slika 4-1. Jednostavna slika za podvlačenje.

Tu sliku možete primijeniti na veze upotrebom sljedećeg koda:

```
a:link, a:visited {
  color:#666;
  text-decoration: none;
  background: url(images/underline1.gif) repeat-x left bottom;
}
```

Nastalu oblikovanu vezu možete vidjeti na slici 4-2.



Slika 4-2. Prilagođeno podvlačenje veza.

Ne morate se zaustaviti na stilu veze i stilu posjećene veze. U ovom sam primjeru izradio animirani GIF za stanje kada se iznad veze nalazi pokazivač miša i aktivno stanje, koji primjenjujem upotrebom sljedećeg CSS-a:

```
a:hover, a:active {
  background-image: url(images/underline1-hover.gif);
}
```

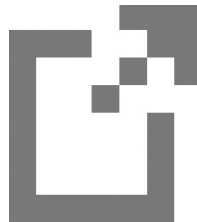
Kad postavite pokazivač miša iznad veze ili kada ju pritisnete, čini se da se poprečne crtice pomiču slijeva nadesno, ostavljajući zanimljiv dojam pulsiranja ili neprekidnog kretanja. Ne podržavaju svi preglednici animirane pozadinske slike, no takvi će obično prikazati prvu sličicu animacije, osiguravajući da se efekt lijepo degradira u starijim preglednicima.

Ne zaboravite da animacije treba oprezno koristiti jer nekim korisnicima mogu izazvati probleme. Ako ste u nedoumici, uvijek se sjetite provjeriti Web Content Accessibility Guidelines (WCAG 1.0) na www.w3.org/TR/WAI-WEBCONTENT/.

Isticanje različitih tipova veza

Na mnogim lokacijama teško je razlučiti pokazuje li veza prema drugoj stranici na toj lokaciji ili prema sasvim drugoj lokaciji. Pritisnemo vezu očekujući da će nas dovesti do druge stranice na trenutnoj lokaciji, a budemo prebačeni na neku drugu neočekivanu stranicu. Da bi riješile ovaj problem, mnoge lokacije otvaraju vanjske veze u novom prozoru. Međutim, to nije dobra zamisao jer korisniku oduzima nadzor i potencijalno zasipa njegovu radnu površinu neželjenim prozorima.

Najbolje je rješenje nekako ukazati na vanjske veze te dopustiti korisniku da sam odluči hoće li napustiti lokaciju, otvoriti vezu u novom prozoru ili, što je u današnje vrijeme vjerojatnije, na novoj kartici u istom prozoru preglednika. To možete postići dodavanjem male ikone pored vanjskih veza. Lokacije kao što je wikipedia.com to već rade pa se počinje rađati konvencija ikona za veze prema vanjskim lokacijama: okvir sa strelicom (slika 4-3).



Slika 4-3. Ikona veze koja vodi do vanjske lokacije.

Ovo je najlakše učiniti dodavanjem klase vanjskim vezama, a zatim primjenjivanjem ikone kao pozadinske slike. U ovom sam primjeru napravio prostor za ikonu tako što sam vezi dodijelio malo popune s desne strane, a zatim sam primijenio ikonu kao pozadinsku sliku desno iznad veze (pogledajte sliku 4-4).

```
.external {
  background: url(images/externalLink.gif) no-repeat right top;
  padding-right: 10px;
}
```

This is an external link 

Slika 4-4. Oblikovanje vanjske veze.

Iako ova metoda funkcionira, to nije posebno mudar ili elegantan postupak jer klasu morate ručno dodati svakoj vanjskoj vezi. Kako bi bilo kad bi postojao način da CSS umjesto vas utvrdi je li nešto vanjska veza? Pa, zapravo i postoji: upotrebom selektora atributa.

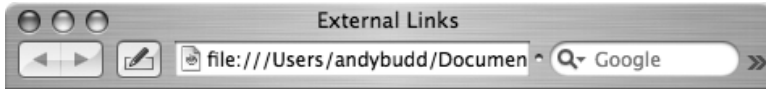
Kao što ste naučili u poglavlju 1, selektori atributa omogućuju ciljanje elementa na temelju postojanja ili vrijednosti atributa. CSS 3 proširuje tu mogućnost selektorima atributa koji pronalaze podudarnosti s podnizovima znakova. Kao što im i ime govori, ovi selektori omogućuju ciljanje elementa pronalaženjem podudarnosti između dijela vrijednosti atributa i odabranog teksta. CSS 3 još nije službena specifikacija pa će upotreba ovih naprednih selektora vjerojatno razbiti kod. Međutim, brojni standardizirani preglednici, kao što su Firefox i Safari, već podržavaju ove CSS 3 selektore pa je mogućnost da budu odbačeni iz konačne specifikacije vrlo mala.

Ova tehnika djeluje tako da se prvo ciljaju veze koje započinju s tekstom `http:` koristeći selektor atributa `[att^=val]`:

```
a[href^="http:"] {
    background: url(images/externalLink.gif) no-repeat right top;
    padding-right: 10px;
}
```

To bi trebalo istaknuti sve vanjske veze. No, uključit će i unutarnje veze upotrebom apsolutnih umjesto relativnih URL adresa. Da biste ovo izbjegli, morate ponovno postaviti veze s vlastitom lokacijom uklaňanjem ikone vanjske veze. To ćete učiniti pronalaženjem veza koje pokazuju prema imenu vaše domene, uklaňanjem ikone vanjske veze te ponovnim postavljanjem desne popune (pogledajte sliku 4-5).

```
a[href^="http://www.yoursite.com"], a[href^="http://yoursite.com"] {
    background-image: none;
    padding-right: 0;
}
```



This is an external link

And here is a reasonably long line of text containing an absolute internal link, some text, an an external link, some more text, a relative internal link and then some more text.

Slika 4-5. Stranica koja pokazuje vanjske veze oblikovane različito od unutarnjih.

Većina standardiziranih preglednika podržava ovu tehniku, ali prijašnji preglednici kao što su IE 6 i stariji jednostavno će je zanemariti.

Ako želite, možete proširiti ovu tehniku da istaknete i veze za elektroničku poštu. U ovom primjeru dodajem malu ikonu e-pošte na sve veze za kontakt:

```
a[href^="mailto:"] {
  background: url(images/email.png) no-repeat right top;
  padding-right: 10px;
}
```

Mogli biste istaknuti čak i nestandardne protokole, kao što je je AIM protokol za razmjenu instant poruka, malom ikonom AIM čovječuljka (pogledajte sliku 4-6):

```
a[href^="aim:"] {
  background: url(images/im.png) no-repeat right top;
  padding-right: 10px;
}
```

```
<a href="aim:goim?screenname=andybudd">instant message</a>
```

Contact me by [email](#) 

Send me an [instant message](#)  using AIM/iChat.

Slika 4-6. Stilovi veze za e-poštu i instant poruke

Isticanje dokumenata i kanala koji se mogu preuzeti

Do još jedne uobičajene frustracije dolazi kada pritisnete vezu misleći da će vas odvesti na stranicu, a umjesto toga započne preuzimanje PDF ili Microsoft Word dokumenta. Srećom, CSS može pomoći pri razlikovanju i ovih tipova veza. To se postiže upotrebom selektora atributa [att\$=val] koji cilja attribute koji završavaju osobitom vrijednošću, kao što su .pdf ili .doc:

```
a[href$=".pdf"] {
  background: url(images/pdfLink.gif) no-repeat right top;
  padding-right: 10px;
}

a[href$=".doc"] {
  background: url(images/wordLink.gif) no-repeat right top;
  padding-right: 10px;
}
```

Tako na sličan nači možete istaknuti veze prema Word dokumentima ili PDF-ovima s pomoću njihovih posebnih ikona, upozoravajući korisnike da su one veze za preuzimanje, i ne vode prema drugoj stranici.

Na kraju, mnoge Web stranice nude RSS kanale. Zamisao je da korisnici kopiraju veze kanala u programe za praćenje. Međutim, ako slučajno pritisnete jednu od tih veza može vas odvesti do stranice s naizgled besmislenim podacima. Da bi se izbjegla moguća zabuna, mogli biste veze do RSS kanala istaknuti s posebnom RSS ikonom:

```
a[href$=".rss"], a[href$=".rdf"] {
    background: url(images/feedLink.gif) no-repeat right top;
    padding-right: 10px;
}
```

Sve ove tehnike mogu poboljšati korisnikov doživljaj vaših stranica. Upozoravanjem korisnika na veze prema vanjskim stranicama ili dokumentima za preuzimanje obavještavate ih što točno mogu očekivati kad pritisnu vezu te tako izbjeci nepotrebno vraćanje unatrag i frustracije.

Na žalost, IE 6 i stariji ne podržavaju selektor atributa. Srećom, sličan učinak možete postići dodavanjem klase svakom elementu koristeći JavaScript i DOM. Jedan od najboljih načina da to učinite je s pomoću odlične funkcije getElementBySelector Simona Willisona. Više detalja možete pronaći na adresi <http://tinyurl.com/dmao4>.

Izrada gumba i prijelaza

Sidra su redni elementi, što znači da se aktiviraju samo kad pritisnete na sadržaj veze. Međutim, ima slučajeva kad ćete poželjeti izraditi efekt koji više nalikuje gumbu s većim područjem za pritiskanje. To možete učiniti postavljanjem svojstva sidra display na block i nakon toga mijenjanjem width, height i ostalih svojstava da izradite stil i aktivno područje kakvo želite.

```
a {
    display: block;
    width: 6em; /* dimenzije potrebne za IE5.x/Win */
    padding: 0.2em;
    line-height: 1.4;
    background-color: #94B8E9;
    border: 1px solid black;
    color: #000;
    text-decoration: none;
    text-align: center;
}
```

Nastala veza sada bi trebala izgledati kao na slici 4-7.



Slika 4-7. Veza oblikovana poput gumba.

Kad se veza prikazuje kao element na razini bloka, pritiskanje bilo gdje unutar bloka aktivirat će vezu.

Pogledate li CSS kôd, vidjet ćete da je `width` izričito postavljeno u jedinicama `em`. Po svojoj prirodi, elementi na razini bloka šire se da bi popunili raspoloživu širinu pa ako je širina njihovih roditeljskih elemenata veća od potrebne širine veze, na vezu biste morali primijeniti željenu širinu. To bi vjerojatno bio slučaj kad biste željeli upotrijebiti tako oblikovanu vezu u glavnom području za sadržaj na stranici. Međutim, ukoliko će oblikovane veze biti smještene u bočnom stupcu, na primjer, vjerojatno biste postavili samo širinu tog stupca i ne biste brinuli o širini veza.

Na žalost, IE 5.x u Windowsima ima pogrešku zbog koje, ako nisu zadana svojstva `width` ili `height`, samo tekst veze postaje aktivan, iako je svojstvo `display` postavljeno na `block`. U prethodnom primjeru kontroliram visinu gumba upotrebom `line-height` pa je neophodna eksplicitna širina da bi se cijelo područje moglo pritiskati u IE-u 5.x za Windowse.

Možda se pitate zašto umjesto `height` koristim `line-height` za kontroliranje visine gumba. Pa, to je zapravo koristan mali trik za okomito centriranje teksta gumba. Kad biste postavili `height`, vjerojatno biste morali upotrijebiti popunu da potisnete tekst prema dolje i imitirate okomito centriranje. Međutim, tekst je uvijek okomito centriran u okviru reda pa će se upotrebom `line-height` tekst uvijek nalaziti u sredini okvira. Ipak, postoji jedan nedostatak. Ako se tekst u okviru prelama u dva reda, gumb će biti dvostruko viši nego što biste željeli. Jedini način da to izbjegnute je da zadate veličinu gumba i teksta tako da se tekst ne prelama, ili se barem ne prelama sve dok se veličina teksta prekomjerno ne uveća.

Jednostavni prijelazi

U lošim starim vremenima, za izradu efekata prijelaza koristile su se velike i pretjerano komplicirane JavaScript funkcije. Srećom, upotreba pseudoklase `:hover` omogućuje izradu efekata prijelaza bez potrebe za JavaScriptom. Prethodni primjer možete proširiti da bi uključivao vrlo jednostavan efekt prijelaza samo zadavanjem boje pozadine i teksta veze kad se iznad nje postavi pokazivač miša (slika 4-8):

```
a:hover {
  background-color: #369;
  color: #fff;
}
```



Slika 4-8. Stil za stanje kada se iznad veze nalazi pokazivač miša prikazuje aktivno područje.

Mijenjanje pozadinskih boja djelotvorno je kod jednostavnih gumba, no kod kompliciranijih gumba najbolje je upotrijebiti pozadinske slike. Za slijedeći sam primjer izradio dvije slike gumba, jednu za obično stanje i jednu za stanje kada se iznad veze nalazi pokazivač miša (pogledajte sliku 4-9). Ako želite, možete dodati i aktivno stanje koje bi se izazvalo upotrebom dinamičke pseudoklase `:active`.



Slika 4-9. Slike za normalno stanje gumba i stanje kada se iznad gumba nalazi pokazivač miša.

Kôd za ovaj primjer sličan je onome za prethodni primjer. Glavna je razlika u tome što se umjesto pozadinskih boja koriste pozadinske slike.

```

a:link, a:visited {
    display: block;
    width: 200px;
    height: 40px;
    line-height: 40px;
    color: #000;
    text-decoration: none;
    background: #94B8E9 url(images/button.gif) no-repeat left top;
    text-indent: 50px;
}

a:hover {
    background: #369 url(images/button-over.gif) no-repeat left top;
    color: #fff;
}
    
```

Ovaj primjer koristi gumbe fiksne širine i visine pa je to razlog što sam postavio izričite dimenzije u pikselima u CSS-u. Međutim, ništa vas ne zadržava da izradite pretjerano velike slike gumbi ili upotrijebite kombinaciju pozadinskih boja i slika za izradu tekućeg ili elastičnog gumba.

Prijelazi stila Pixy

Glavni nedostatak kod metode s više slika je mali zastoj dok preglednici po prvi puta učitavaju sliku koja se koristi za stanje kada se iznad gumba nalazi pokazivač miša. To može izazvati nepoželjni dojam trzanja i uzrokovati da gumbi malo sporije reagiraju. Moguće je unaprijed učitati slike, tako da ih se primijeni kao pozadinu na roditeljski element. Međutim, postoji i drugi način. Umjesto izmjenjivanja više pozadinskih slika, upotrijebite samo

jednu sliku i mijenjajte njezinu pozadinsku poziciju. Dodatna korist od upotrebe jedne slike je smanjivanje broja zahtjeva poslužitelju kao i mogućnost čuvanja svih stanja gumba na jednom mjestu. Ova je metoda poznata kao Pixy metoda, nazvana prema nadimku njenog autora, Petra Staničeka.

Započnite izradom slike spojenih gumba (pogledajte sliku 4-10). U ovom slučaju koristim samo normalno stanje i stanje kad je pokazivač iznad gumba, ali vi možete uključiti i aktivno i posječeno stanje, ako želite.



Slika 4-10. Oba stanja gumba kao jedna slika.

4

Kôd je gotovo identičan prethodnom primjeru. Međutim, ovaj put sliku prijelaza poravnate ulijevo za normalno stanje veze, a zatim je pomičete udesno za stanje kad se iznad veze nalazi pokazivač.

```
a:link, a:visited {
    display: block;
    width: 200px;
    height: 40px;
    line-height: 40px;
    color: #000;
    text-decoration: none;
    background: #94B8E9 url(images/pixy-rollover.gif) no-repeat left top;
    text-indent: 50px;
}

a:hover {
    background-color: #369;
    background-position: right top;
    color: #fff;
}
```

Na žalost, IE u Windowsima još uvijek vraća poziv poslužitelju zahtijevajući novu sliku, iako se samo mijenja poravnanje slike. To uzrokuje neznatno treperenje, što može smetati. Da biste izbjegli treperenje, stanje prijelaza morate primijeniti na roditeljski element veze, primjerice na odlomak koji je sadrži.

```
p {
    background: #94B8E9 url(images/pixy-rollover.gif) no-repeat right top;
}
```

Slika će i dalje na trenutak nestati dok se ponovno učitava. Međutim, tijekom tog vremena, ispod nje će se pojaviti ista slika, skrivajući treperenje.

Stilovi posjećenih veza

Dizajneri i programeri često zaborave na stil posjećene veze pa posjećene veze oblikuju jednako kao i neposjećene. Međutim, poseban stil posjećene veze korisnicima može pomoći pri snalaženju, pokazujući im koje su stranice ili lokacije već posjetili da bi izbjegli nepotrebno vraćanje na stranice koje su već posjetili. Stilovi posjećenih veza mogu pretrpati glavno područje za sadržaj stranice pa ih razumno koristite. Međutim, one dolaze do izražaja kad se koriste u bočnom stupcu ili u sporednom izborniku za kretanje.

Vrlo jednostavan stil posjećenih veza možete izraditi dodavanjem okvira za potvrdu svakoj posjećenoj vezi:

```
a:visited {
  padding-right: 20px;
  background: url(check.gif) right middle;
}
```

Nadalje, recimo da u bočnom stupcu imate popis veza prema vanjskim lokacijama:

```
<ul>
  <li><a href="http://www.andybudd.com/">Andy Budd's Blogography</a></li>
  <li><a href="http://allthatmalarkey.co.uk/">Stuff and Nonsense</a></li>
  <li><a href="http://www.hicksdesign.co.uk/">Hicks Design</a></li>
  <li><a href="http://www.clagnut.com/">Clagnut</a></li>
  <li><a href="http://www.htmldog.com/">HTML Dog</a></li>
  <li><a href="http://adactio.com/journal/">Adactio</a></li>
  <li><a href="http://www.allinthehead.com/">All In The Head</a></li>
  <li><a href="http://www.markboulton.co.uk/">Mark Boulton</a></li>
  <li><a href="http://www.ian-lloyd.com/">Ian Lloyd</a></li>
</ul>
```

Upotrebom Pixy metode o kojoj ste učili ranije u ovom poglavlju, mogli biste izraditi jednu sliku za neposjećeno i posjećeno stanje (pogledajte sliku 4-11). Ako želite, možete dodati stanje kada se iznad veze nalazi pokazivač miša i aktivno stanje.



Slika 4-11. Grafike posjećene i neposjećene veze u jednoj slici.

Zatim biste primijenili pozadinsku sliku na isti način kao i prije. Za sada nemojte brinuti o oblikovanju popisa jer ću popise detaljno obraditi u sljedećem poglavlju. Najvažnije stvari na koje treba obratiti pažnju su oblikovanje pozadinske slike na sidru i posjećeno stanje.

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

li {
  margin: 5px 0;
}
```

```

li a {
  display: block;
  width: 300px;
  height: 30px;
  line-height: 30px;
  color: #000;
  text-decoration: none;
  background: #94B8E9 url(images/visited.gif) no-repeat left top;
  text-indent: 10px;
}

li a:visited {
  background-position: right top;
}

```

4

Nastali popis veza možete vidjeti na slici 4-12. Svaka lokacija koju ste posjetili imat će oznaku potvrde pored imena, pružajući vrijednu informaciju da ste tamo već bili. Simon Collison u svojoj analizi slučaja demonstrira kako se ova koncepcija može praktično upotrijebiti.



Slika 4-12. Popis vanjskih veza prikazuje posjećene lokacije s pomoću oznaka za potvrdu.

Pravi CSS podsjetnici za alate

Podsjetnici za alate (engl. *tool tips*) su mali žuti okviri za tekst koji se u nekim preglednicima pojavljuju kada pokazivač miša postavite iznad elemenata s oznakama naslova. Nekoliko dizajnera izradilo je vlastite prilagođene stilizirane podsjetnike za alate koristeći kombinaciju JavaScripta i CSS-a. Međutim, moguće je izraditi prave CSS podsjetnike alata upotrebom CSS tehnika pozicioniranja. Da bi pravilno funkcionirala, ova tehnika zahtijeva moderni standardizirani preglednik kao što je Firefox. Prema tome, to nije tehnika koju biste koristili u svakodnevnom radu. Ipak, ona demonstrira moć naprednog CSS-a te pruža naznaku onoga što će biti moguće kad CSS bude bolje podržan.

Kao i kod svih primjera u ovoj knjizi, morate započeti s dobro strukturiranim i jasnim (X)HTML-om:

```
<p>
<a href="http://www.andybudd.com/" class="tooltip">
Andy Budd<span> (This website rocks) </span></a>
is a web developer based in Brighton England.
</p>
```

Vezi sam dodijelio klasu tooltip da bi se razlikovala od drugih veza. Unutar veze sam dodao tekst koji želim prikazati kao tekst veze, a slijedi ga tekst podsjetnika za alat obuhvaćen elementom span. Tekst podsjetnika za alat okružio sam zagradama tako da rečenica ima smisla i kad su stilovi isključeni.

Prva stvar koju morate učiniti je postaviti svojstvo sidra position na relative. To omogućuje apsolutno pozicioniranje sadržaja elementa span, u odnosu na poziciju njegovog roditeljskog sidra. Ne želite da se tekst podsjetnika prikaže na početku pa njegovo svojstvo display trebate postaviti na none:

```
a.tooltip {
  position: relative;
}

a.tooltip span {
  display: none;
}
```

Želite da se sadržaj elementa span pojavljuje tek kad se iznad sidra postavi pokazivač miša. To ćete postići postavljanjem svojstva display elementa span na block, no samo onda kad je iznad veze postavljen pokazivač miša. Ako biste sada testirali kôd, postavljanje pokazivača miša iznad veze rezultiralo bi pojavljivanjem teksta iz elementa span pored veze.

Da biste pozicionirali sadržaj elementa span ispod i desno od sidra, morate postaviti svojstvo position elementa span na absolute te ga pozicionirati 1 em od vrha sidra i 2 em od lijeve strane.

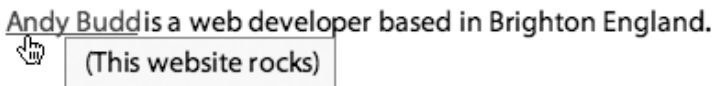
Ne zaboravite: apsolutno pozicionirani element pozicioniran je u odnosu na svog najbližeg pozicioniranog pretka ili, ako je to nemoguće, na korijenski element. U ovom smo primjeru pozicionirali sidro pa je span pozicioniran u odnosu na njega.

```
a.tooltip:hover span {
  display: block;
  position: absolute;
  top: 1em;
  left: 2em;
}
```

I to je veći dio tehnike. Još je samo preostalo dodati oblikovanje koje će učiniti da span više liči na podsjetnik za alat. To možete učiniti ako elementu span dodijelite popunu, rub i pozadinsku boju:

```
a.tooltip:hover span {
  display:block;
  position:absolute;
  top:1em;
  left:2em;
  padding: 0.2em 0.6em;
  border:1px solid #996633;
  background-color:#FFF666;
  color:#000;
}
```

Pregled u Firefoxu trebao bi pokazati nešto kao na slici 4-13.



Slika 4-13. Pravi CSS podsjetnik alata.

Na žalost, ova tehnika ne radi pravilno u Internet Exploreru 5.x za Windows. Čini se da IE ima problema s oblikovanjem elemenata unutar veza sidra koje koriste dinamičku pseudo-klasu. Međutim, evo ispravka:

```
a.tooltip:hover {
  font-size: 100%; /* Fixes bug in IE5.x/Win */
}
```

Postavljanje veličine pisma na 100% na sidru iznad kojeg se nalazi pokazivač miša na neki način uzrokuje da Internet Explorer u Windowsima ispravno oblikuje sadržani span. Znam da je čudno, ali takav vam je IE.

Šteta je što ova tehnika ne radi u Safariju, ali još nisam uspio pronaći rješenje kao što sam učinio za Internet Explorer u Windowsima.

Sažetak

U ovom ste poglavlju naučili kako oblikovati veze na različite načine. Sada znate kako oblikovati veze ovisno o stranici ili datoteci do koje vode, možete postići da se veze ponašaju kao gumbi i znate izraditi efekte prijelaza upotrebom boja ili slika. Možete čak izrađivati napredne efekte poput pravih CSS podsjetnika alata.

U sljedećem ćete poglavlju naučiti kako manipulirati popisima te, koristeći informacije koje ste usvojili u ovom poglavlju, kako izraditi popise za kretanje kroz stranice, prave CSS slikovne mape i daljinske prijelaze. Neka zabava počne.

