

# Lekcija 1 – osnove HTML-a

## Uvod

Odlučio sam početi pisati ovu malu školu iz nekoliko razloga:

1. Zato što sam skužio da nisam objasnio dobro neke stvari u prva dva [tutorijala](#).
2. Zato što želim napraviti jedan sveobuhvatni tečaj na ovim stranicama počevši od samih osnova pisanja koda pa do grafičkog oblikovanja web stranica u Photoshopu.

Lekcije će izlaziti barem jednom tjedno i bit će napravljene kao obični tekstualni ili kao video tutoriali, zavisi o temi lekcije. Naprimjer za ovu lekciju nije potreban video tutorial zato što se ovdje radi većinom o teoriji i mislim da je lakše pratiti ovako napisano nego kao video tutorial.

Nadalje, pokušat ću ovu školu napraviti tako da je poprilično jednostavna za pratit i bez prevelikog ulaska u tehničke detalje, pošto i inače u stvarnom životu predajem neke informatičke tečajeve shvatio sam da ljudi puno bolje nauče ako nešto rade nego ako samo čitaju ili slušaju o nekoj temi. Naravno ohrabrujem svako detaljiziranje na svoju ruku. Probajte prateći ovaj tutorial, raditi vježbe – ako ih tako mogu nazvati – koje i ja radim.

## Struktura HTML dokumenta

Za početak ćemo uzeti jedan html dokument i njega ćemo malo rasčlaniti da vidimo očemu se tu radi:

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3  <html xmlns="http://www.w3.org/1999/xhtml">
4  <head>
5  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
6  <title>Naslov stranice</title>
7  </head>
8  <body>
9  <h1>Pozdrav. Ovo je moja prva html stranica.</h1>
10 </body>
11 </html>
12
```

Što je ovo iznad? Kako dolazimo do toga? Znači, neki dokument nazivamo html dokumentom kada ima nastavak .html ili .htm. Znači da bi napravili ovu vježbu, napravite dokument u vašem omiljenom text editoru (može i obični notepad) prekopirajte gornji kod i nazovite dokument prva.html. Nakon toga ga pokrenite (dvoklik) i trebali bi dobiti nešto [ovako](#).

Na donjoj slici možete vidjeti glavne dijelove html dokumenta:

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2
3 <html xmlns="http://www.w3.org/1999/xhtml">
4
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
7 <title>Naslov stranice</title>
8 </head>
9
10 <body>
11
12 <h1>Pozdrav. Ovo je moja prva html stranica.</h1>
13
14 </body>
15 </html>
16

```

Gornja slika prikazuje glavnu podjelu html dokumenta. Sada ću malo opisati te dijelove:

1. Pod brojem jedan imamo doctype – doctype služi da bi browser bolje generirao stranicu. Bilo bi najbolje da si uvijek kopirate ovakav doctype u zaglavlje svih html dokumenata. Browser će generirati stranicu i bez doctypa ali velika je vjerovatnost da ju neće generirati na način na koji mi to očekujemo. Zasad toliko o doctypu rko želi znati više neka posjeti ovaj [link](#).

2. Broj 2 počinje sa < html xmlns=... a završava sa < / html >. Ova formacija u kodu se naziva TAG. Tag je najjednostavnije rečeno označivač koji govori gdje neki dio html-a počinje a gdje završava. Kao što vidite većina tagova počinje sa < unutar zagrade se nalazi ime taga nakon kojeg mogu slijediti neka svojstva i onda početak taga zatvorimo sa >. Kraj taga opet počinje sa < nakon toga stavljamo kosu crtu / koja označava kraj taga i nakon toga upisujemo ime taga, ovdje ne idu nikakva dodatna svojstva i naravno opet zatvaramo sa >. < html > tag označava početak html koda i bilo bi jako dobro da se nalazi u svakom vašem html dokumentu.

3. Primjetite da se broj 3 i 4 nalaze unutar html taga, što znači da su to njegovi elementi. Sljedeći tag na koji nailazimo je head tag. **Head** tag predstavlja zaglavlje vaše stranice, primjetite da se unutar head taga nalazi još nekoliko dodatnih tagova (iako ih inače ima puno više – ali o tome kasnije). Znači imamo:

```

1 <meta http-equiv="Content-Type" content="text/html;
  charset=iso-8859-1" />

```

Ovaj tzv. **meta tag** opisuje koji charset (set slova) će se koristiti na stranici, u ovom primjeru to će biti zapadnoeuropski, ako hoćemo dobro prikazivati naša slova trebali bi upisati **cacharset=iso-8859-2**. Primjetite da meta tag nema zatvarajućeg elemnta, zanci ne postoji nešto kao < / meta > nego se zatvara sa "/>". Nakon meta taga imamo **title** tag:

```

1 <title>Naslov stranice</title>

```

Title tag određuje naslov stranice. Što god da upišete između < title > i < / title > će biti naslov stranice koji će se prikazivati na vrhu vašeg omiljenog browsera. Ispod title taga imamo < / head > koji označava završetak head taga odnosno zaglavlja stranice.

Vrlo je bitno da shvatite da head tag ne predstavlja zaglavlje stranice u nekom grafičkom smislu. Zanci da se head tag praktički neće vidjeti na stranici, osim naslova na vrhu browsera. On samo sadrži podatke koji su potrebni za ispravno funkcioniranje stranice. Ali o tome više kasnije.

4. I na kraju dolazimo do **body** taga. Body je jako bitan zato što označava gdje počinje i završava ono što korisnik web stranice vidi. Uglavnom sve što budete željeli da vam se prikaže na web stranici morate staviti u body tag.

Što znači između < body > i < / body >. Unutar body taga imamo tag koji se naziva h1 – h1 sa h1 označavamo da je sve što se nalazi između početka i završetka taga naslov.

## h tagovi

Zašto je podnaslov h tagovi? Zato što naravno imamo više h tagova. Oni se razlikuju u broju koji dolazi poslije slova h. Znači imamo ih nekoliko: h1, h2, h3 itd. Ali da nebi previše pričali o tome, proširite sada vaš prvi html dokument da izgleda ovako:

```
1
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5  <html xmlns="http://www.w3.org/1999/xhtml">
6
7  <head>
8  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
9  <title>Naslov stranice</title>
10 </head>
11
12 <body>
13 <h1>Pozdrav. Ovo je moja prva html stranica.</h1>
14 <h2>Pozdrav. Ovo je moja prva html stranica.</h2>
15 <h3>Pozdrav. Ovo je moja prva html stranica.</h3>
16 <h4>Pozdrav. Ovo je moja prva html stranica.</h4>
17 <h5>Pozdrav. Ovo je moja prva html stranica.</h5>
18 <h6>Pozdrav. Ovo je moja prva html stranica.</h6>
19
20 </body>
21 </html>
```

i trebali bi dobiti ovako [nešto](#).

Znači što je broj iza slova h veći to će slova na prikazu biti manja. To je super, ali je isto tako i netočno i nepotrebno, zato što ćemo kasnije naučiti da mi možemo odrediti veličinu slova h tagova pomoću css-a. Za sada je bitno shvatiti da h tagovi određuju konstrukciju stranice, odnosno, određuju koliko je ono što je između h taga upisano važno ili koliko bi se trebalo isticati. Možemo na to gledati i ovako, h1 je glavni naslov nekog teksta, h2 je neki podnaslov u tom tekstu, h3 je podnaslov u podnaslovu itd. Možemo mi napraviti stranicu ne koristeći ni jedan h tag, ali to je loša praksa. Pošto ako je stranica dobro konstruirana web tražilice će ju bolje prepoznavati i prije će ju unijeti u svoju bazu.

Još jedna bitna stvar je to da bi stranica trebala dobro izgledati i bez stilizacije (css dokumenta), a u tome h tagovi pomažu da se zna što je najvažnije na stranici, što je manje važno itd.

## p tag

**P** tag nam služi za određivanje gdje počinje i završava pojedini paragraf odnosno odlomak teksta. Za primjer možete izmijeniti vaš .html dokument da izgleda ovako:

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
2
3  <html xmlns="http://www.w3.org/1999/xhtml">
4
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```

6 <title>Naslov stranice</title>
7 </head>
8
9 <body>
10 <h1>Tekst ispod ovoga je paragraf.</h1>
11
12 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque id diam et n
13 litora torquent per conubia nostra, per inceptos hymenaeos. Duis gravida. Curabitur a
14 tristique nibh. Donec nec massa. Pellentesque habitant morbi tristique senectus et net
15
16 <p>Praesent purus tortor, scelerisque tristique, ullamcorper ut, imperdiet et, felis.
17 Aenean sed orci. Pellentesque auctor arcu sit amet nisi egestas dapibus. Maecenas sed
18 nonummy in, ornare et, mi. Donec sit amet magna at libero volutpat consequat. Cras au
19 Praesent aliquam libero nec mauris. Morbi porttitor hendrerit enim. Nam sapien diam, o
20 ullamcorper, tortor sem egestas urna, a hendrerit leo sem ac justo.</p>
21
22 </body>
23 </html>

```

i trebali bi dobiti [ovo](#).

Znači napravili smo 2 odlomka. Možemo vidjeti da su 2 pošto su i fizički odvojena. P tagove koristimo kada želimo da nam neki tekst predstavlja odlomak. P tagovi se isto kao i h tagovi kasnije mogu oblikovati pomoću css-a da bi izgledali onako kako mi to želimo i isto su bitni za samo konstrukciju stranice.

## ul i li tagovi

**ul** i **li** su tagovi koji se koriste u kombinaciji. **li** tagovi se uvijek moraju nalaziti unutar **ul** tagova. Za primjer ćemo dopisati ispod drugog odlomka sljedeće:

```

1 <ul>
2   <li>Prvi navod</li>
3   <li>Drugi navod</li>
4   <li>Treći navod</li>
5
6 </ul>

```

Rezultat možete pogledati [ovdje](#).

Dakle ul i li tagovi nam služe za prikaz ne uređene liste. ul je unordered-list (neuređena lista) ali li je list-item (član liste). Moramo paziti da li tagovi uvijek budu između <ul> i </ul>. <ul> tag kaže da započinjemo listu, zatim <li> tag kaže da započinjemo opis člana liste, zatim </li> tag govori da smo završili sa upisom člana i onda nam na kraju </ul> tag govori da smo završili sa upisom liste.

## img tag

**img** tag je jedan od rijetkih tagova koji se ne zatvara sa npr. </img> nego se zatvara sa />. Ovaj nam tag služi za prikazivanje slika. Da bi pokazali kako radi ovaj tag, trebamo napraviti male promjene u mapi u kojoj ste spremili vaš .html dokument. Najprije napravite mapu koju ćete nazvati "slike" ili "images" ili kako već želite, ta mapa će nam služiti da u njoj držimo slike koje ćemo stavljati na našu stranicu. U tu mapu spremite neku sliku po vašem izboru i promijenite joj ime u slika.jpg ili slika.gif ako koristite .gif sliku, tako da lakše možete pratiti ovaj primjer. Kada ste to sve obavili upišite sljedeće ispod neuređene liste:

```

1 

```

**Ovako** izgleda taj primjer sa slikom koju sam ja stavio.

Ajmo sad malo objasniti tu liniju koda:

`.

## div tag

**Div** tag je zadnji tag kojeg ćemo obraditi u ovoj lekciji. Njegova primjena vam najvjerojatnije neće biti baš najjasnija dok ne pređemo na drugu lekciju i počnemo učiti CSS pomoću kojega ćemo oblikovati ove tagove koje smo naučili. Dodajte ovo ispod slike:

```
1 <div>Ovo je moj prvi div tag</div>
```

Probajte pokrenuti dokumnet `prva.html` i vidjet ćete da nam se pojavio samo tekst "Ovo je moj prvi div tag", i ništa se drugo posebno nije dogodilo. To je zato što **div** tag opisuje blok, odnosno neku određenu cijelinu na stranici. Da bi se shvatila puna moć **div** tagova, moramo se upoznati sa CSS-om i u njemu oblikovati `div` tag. Zato ovdje više neću opisivati `div` tagove jer neće imati smisla ako ga ne vidite u punoj snazi, a to nam slijedi u idućoj lekciji.

## Zaključak

Ovih nekoliko tagova koje sam naveo gore su vam sasvim dovoljni da napravite svoju prvu web stranicu, što ćemo nastaviti raditi u lekciji 2, kada se počinjemo baviti CSS-om i oblikovanjem tagova koje smo danas naučili. Od sljedeće lekcije počinjemo praviti naš web site. Korak po korak, tako da ćemo do kraja ovog tečaja imati potpuno funkcionalan web site. Pišite ako vam što nije jasno.

## Lekcija 2 – osnove CSS-a

Zadnji puta smo stali na [ovome](#). Znači napravili smo kostur naše stranice. Odnosno definirali smo strukturu stranice, kao što vidite, stranica baš i ne izgleda nešto posebno. Da bi ju malo uljepšali moramo se koristiti CSS-om. Ako nam je HTML kostur, onda bi CSS bio meso koje ćemo staviti na taj kostur, u dizajnerskom smislu, naravno.

CSS je skraćenica za Cascading Style Sheets. Za ovo baš i nema nekog pravog prijevoda na naš jezik, barem ne nekog smislenog, možda sam ja u krivu, ako netko zna neki dobar prijevod neka ga svakako upiše u kao komentar na ovaj članak. Tako da ćemo mi koristiti ovu skraćenicu.

CSS služi da bi sve one tagove koje smo naveli u prošloj lekciji doveli u red, da bi ih oblikovali da budu oku ugodniji i na kraju upotrebljiviji. Postoji nekoliko načina kako možemo raditi sa CSS-om, ja ću u ovim lekcijama prikazati samo jedan, i to onaj koji ja najčešće koristim i za koji mislim da je najbolji.

### Pozivanje CSS-a

Mi ćemo u ovoj maloj školi raditi sa odvojenim CSS datotekama. To znači da ćemo imati jednu datoteku u kojoj ćemo definirati strukturu stranice (HTML) i jednu datoteku u kojoj ćemo definirati izgled stranice (CSS). Za početak moramo napraviti datoteku u koju ćemo upisivati CSS kod. Otvorite svoj omiljeni text editor, otvorite novi dokument u njemu i snimite ga kao `stil.css`. Znači, CSS datoteke imaju nastavak `.css` (zamisli!). Kada ste to napravili, otvorite `.html` datoteku iz prve lekcije (`prva.html`), u nju ćemo ubaciti sljedeći kod:

```
1 <link href="stil.css" rel="stylesheet" type="text/css" />
```

Ovaj kod ubacite odmah iznad `</head>` taga. Znači mora biti unutar `head` taga, odnosno prije njegova završetka. Pomoću ovog koda smo rekli HTML dokumentu da za svoje oblikovanje koristi dokument koji smo nazvali `stil.css`.

### CSS pravila

Kako HTML ima svoja pravila pri pisanju koda, isto tako ima i CSS. Upišite ovo u `stil.css`:

```
1 body{background:#ccc;}
```

Pokrenite `prva.html` i trebali dobiti nešto [ovako](#).

Što se desilo? Kao što vidite podloga stranice se promjenila u sivu boju. Zašto? Zato što smo rekli da tag `body` ima sivu pozadinu. CSS je zapravo vrlo jednostavan glede naziva svojstava.

**Svojstvo** je naziv koji ćemo od sada koristiti za nazivanje "komandi" u CSS-u. U gornjem slučaju, svojstvo se zove **background**

Svojstva u CSS-u može bez problema razumjeti svatko tko donekle razumije engleski jezik, pošto su svojstva nazivana prema engleskim riječima. Background na engleskom znači pozadina. Ono što stavljamo ispred vitičaste zagrade (`{}`) je objekt u HTML-u na koji želimo primjeniti neko svojstvo. Na gornjem primjeru smo dakle na `body` tag primjenili svojstvo pozadine, kojem smo dali vrijednost `#ccc`, odnosno upisali smo heksadecimalnu vrijednost sive boje. Ja ću u ovim lekcijama koristiti isključivo heksadecimalne vrijednosti boja, no ajmo čisto primjera radi pogledati kako smo ovo još mogli napisati:

```
1 body{background:#cccccc;} /*ovo je produžena verzija*/
```

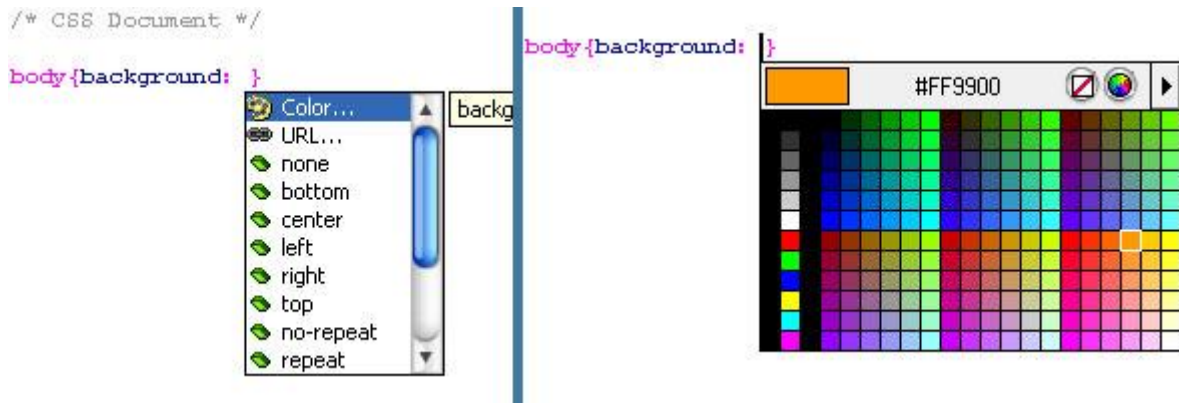
```
2 body{background:gray;} /*kao što vidite možemo koristiti i engleske nazive za boju npr. red, white, black itd.*/
```

```
3 body{background:rgb(204, 204, 204);} /*možemo koristiti i RGB vrijednosti neke boje*/
```

Ispod se nalaze heksadecimalne vrijednosti nekih osnovnih boja:

#000000  
#333333  
#666666  
#999999  
#CCCCCC  
#FFFFFF  
#FF0000  
#00FF00  
#0000FF  
#FFFF00  
#00FFFF  
#FF00FF

Ako koristite Dreamweaver za pisanje svog koda, onda će vam odabir boja biti poprilično lagan, pošto Dreamweaver pri upisu koda izbacuje meni s bojama i nakon što odaberete boju on sam upisuje njenu heksadecimalnu vrijednost.



Ako nemate Dreamweaver ili ga iz nekog razloga ne želite koristiti, možete se poslužiti nekim grafičkim programom poput Photoshopa i iz njega izvući i kopirati traženu heksadecimalnu vrijednost. Većina takvih programa ima prikaz boja u različitim vrijednosnim sustavima. A ako nema, zbilja nebi trebali koristiti takav program.



No, za sada dosta o bojama. Raščlanit ćemo malo onaj gornji primjer našeg prvog CSS unosa, dakle, kada smo napisali *body* odredili smo da browser ono što poslije toga upišemo primjeni *nobody* tag, nakon toga smo otvorili vitičastu zagradu {, unutar vitičastih zagrada upisujemo sva svojstva koja želimo dodijeliti nekom elementu, nakon toga smo upisali ime svojstva – u našem slučaju **background**, i iza njega stavili dvotočje : čime smo označili početak unosa vrijednosti svojstva, upisali smo samu vrijednost – #ccc i zatvorili to svojstvo sa ;, nakon toga smo zatvorili to sve sa vitičastom zagradom } i time označili kraj unosa svojstava u *body* tag. Znači pri pisanju našeg CSS dokumenta moramo poštovati neka pravila. Najbitnije je otvaranje i zatvaranje vitičastih zagrada, a zatim zatvaranje svojstava sa točka zarez. Nije bitno dali ćete sva svojstva pisati u istom redu

```
1 #fwd{margin:3px 0 0 0;height:24px; width:300px;font-size:10px; color:#999999;}
```

ili ćete svako pisati u posebnom redu

```
1 #menu ul li a:link, #menu ul li a:visited, li.close{
2     display:block;
3     height:77px;
4     padding:20px 20px 0 20px;
5     float:left;
6     font:normal 26px Arial, Helvetica, sans-serif;
7     color:#003986;
8     border:none;
9 }
```

to ovisi o vama i sa vremenom ćete naučiti što vam je lakše. Ja pišem CSS zavisno o tome koliko ću svojstava imati za određeni element, ako će ih biti malo onda sve u istom redu, a ako će ih biti više svako u posebnom redu. Razmaci isto nisu toliko bitni. Vidjet ćete kasnije da je CSS vrlo fleksibilan glede načina pisanja koda.

Da bi CSS radio kako treba svojstva moraju biti zatvorena vitičastim zagradama i na kraju svagog svojstva mora doći točka-zarez. Ako vam nešto ne radi uvijek prvo pogledajte dali vam je kod ispravno napisan a tek onda tražite greške na drugim mjestima. Greške u upisu koda su najčešće greške.

Nadalje, pri pisanju CSS koda ponekad je poželjno komentiranje, komentari su dijelovi CSS-a koje browser ne čita ali služe nama da bi se lakše snalazili u onome što smo napisali, ja sam već koristio komentare kada sam



vam pokazao kako možete još napisati da nam pozadina bude sive boje. Komentar počinjemo sa `/*` a završavamo sa `*/` kao npr.:

```
1 /*ovo je komentar*/
```

## Osnovna CSS svojstva

### BORDER

Postoje neka osnovna CSS svojstva, koja ćemo obraditi u ovoj lekciji. Za početak ćemo promjeniti boju naše pozadine u bijelu. Sivu sam iskoristio samo kao primjer. Znači izmjenit vaš CSS dokument da izgleda ovako:

```
1 body {background:#fff;}
```

Sada bi trebali dobiti nešto [ovako](#).

Znači stranica se vratila u prijašnje stanje, opet smo upisali da heksadecimalni kod boje ima skraćenu vrijednost.

Kada se heksadecimalni kodovi sastoje od 6 jednakih znamenki ili slova, dovoljno je da upišemo samo prva 3 slova ili znamenke.

Ok, sada ćemo se upoznati sa još jednim svojstvom, to svojstvo se naziva **border** ili ti po naški, obrub. Ovo svojstvo ćemo primjeniti na naše odlomke, odnosno na *p* tagove. U vaš CSS dodajte ovaj kod ispod onoga što smo dosad upisali:

```
1 p{border:1px solid #ff0000;}
```

I trebali bi dobiti ovako [nešto](#).

Mi smo sa ovim kodom browseru rekli da želimo da nam svi *p* tagovi na stranici imaju obrub debljine 1px, punu liniju (solid) i da budu crvene boje (#ff0000). U većini tutoriala o CSS su vam svojstva pokazana u svom produženom obliku. Ja sam odlučio odmah prikazivati skraćeni oblik. Pošto mislim da to nije toliko teško za shvatiti. Produženi oblik bi ovako izgledao:

```
1 p{
2     border-width:1px;
3     border-style: solid;
4     border-color:#FF0000;
5 }
```

Znači dobili bi totalno istu stvar a napisali bi puno više koda. Bitno je za zapamtiti, da svako CSS svojstvo mora imati neke svoje vrijednosti. U gornjem primjeru vidimo da **border** ima tri osnovne vrijednosti, a to su, debljina obruba, vrsta obruba i boja obruba. Napomenuo bih da je vrsta crte definitivno najzanimljivija vrijednost ovog svojstva, probajte se s time malo igrati, znači, probajte u gornji kod upisati nešto od ovoga i pogledajte što će se desiti: dashed, dotted, double, groove, ridge. Svakako isprobajte dashed, to je iscrtkana linija, za ove ostale ne mogu garantirati pošto se u nekim browserima ili ne prikazuje dobro ili se uopće ne prikazuje, istočkana linija odnosno dotted će se dobro prikazivati u Firefoxu, ali u Internet Exploreru neće.

Ljepota CSS-a je u sljedećem, recimo da želimo da nam naši odlomci imaju obrube samo gore i dolje, da budu debljine 4px, iscrtkane i zelene boje, to bi napisali ovako:

```
1 p {
```

```
2     border-top: 4px dashed #00ff00;
3     border-bottom: 4px dashed #00ff00;
4 }
```

I dobit ćemo **ovo**.

Primjetie opet kako je CSS zapravo jednostavan glede svojih komandi, zanči ako upišete **samoborder**, to podrazumjeva da ćete imati obrub sa svih strana, no ako upišete **border-top** to znači da želite napraviti samo gornji obrub, top znači gore na engleskom, tako da uz **border-top** imate još **border-bottom** (doljni obrub), **border-left** (lijevi obrub) i **border-right** (desni obrub).

#### Zadatak:

Vaš zadatak će sada biti da napravite da odlomci, odnosno *p* tagovi imaju gornji i donji obrub u crvenoj boji, da budu debljine 2px i da budu pune linije. Lijevi i desni obrubi će biti crtkani, debljine 5px i plave boje. Riješenje ovog zadatka će biti u idućoj lekciji.

## WIDTH I HEIGHT

Ovo su možda 2 najjednostavnija svojstva u CSS-u width na engleskom znači širina, a height znači visina koja se određuje najčešće u pikselima, ali ima i drugih načina o kojima ćemo kasnije. Recimo da želimo da nam širina svih odlomaka bude 300px a visina 500px jednostavno ćemo dodati ove dvije linije u naš CSS kod pod definiciju *p* taga:

```
1 width:300px;
2 height:500px;
```

Znači da nam sve ukupna definicija treba izgledati ovako:

```
1 p {
2     border-top: 4px dashed #00ff00;
3     border-bottom: 4px dashed #00ff00;
4     width:300px;
5     height:500px;
6 }
```

i to će izgledati ovako.

Upisivanjem ove dvije linije smo rekli da želimo da nam širina odlomka bude 300px a visina 500px. Vrlo jednostavno.

Ako širina nekog HTML elementa nije određena, taj elemnt automatski poprima širinu stranice ili širinu nekog drugog elementa u kojem se nalazi. Visina će ovisiti o sadržaju koji se nalazi u tom elementu.

Eto to bi bilo to, što se tiče ove lekcije, znam da za sada ima dosta teorije i da jedva čekate da počnete raditi prave stvari, ali prvo moramo svladati neke osnove. U sljedećoj lekciji nastavljamo sa CSS svojstvima, probajte do tada riješiti onaj zadatak. Ako šta nije jasno, upišite to u komentare, napokon sam napravio da sistem komentiranja radi kako treba

## Lekcija 3 – osnove CSS-a

U prošloj lekciji zadatak je bio da napravite da odlomci, odnosno `p` tagovi imaju gornji i donji obrub u crvenoj boji, da budu debljine 2px i da budu pune linije. Lijevi i desni obrubi će biti crtkani, debljine 5px i plave boje.

Riješenje ovog zadatka je vrlo jednostavno:

```
1  p{
2  border-bottom: 2px solid #FF0000;
3  border-top: 2px solid #FF0000;
4  border-left: 5px dashed #0000FF;
5  border-right: 5px dashed #0000FF;
6  }
```

Možete kliknuti [ovdje](#) da vidite to rješenje. Sada kad smo to riješili, nastavljamo dalje, još uvijek imamo stranicu koja nam se zove **prva.html** i css koji nam se zove **stil.css**

Da bi imali u svim primjerima isto za dalje, u HTML-u nemojte ništa dirati, a u CSS-u obrišite sve, znači neka vam CSS bude prazan dokument. Sada ćemo nastaviti sa daljnjim upoznavanjem CSS-a i njegovih *svojstava*.

### BACKGROUND – PROŠIRENO

Prije nego počnemo sa ovom vježbom skinite si prvo [ovu .zip datoteku](#), raspakirajte ju, i spremite **pozadina-1.jpg** u mapu `images`.

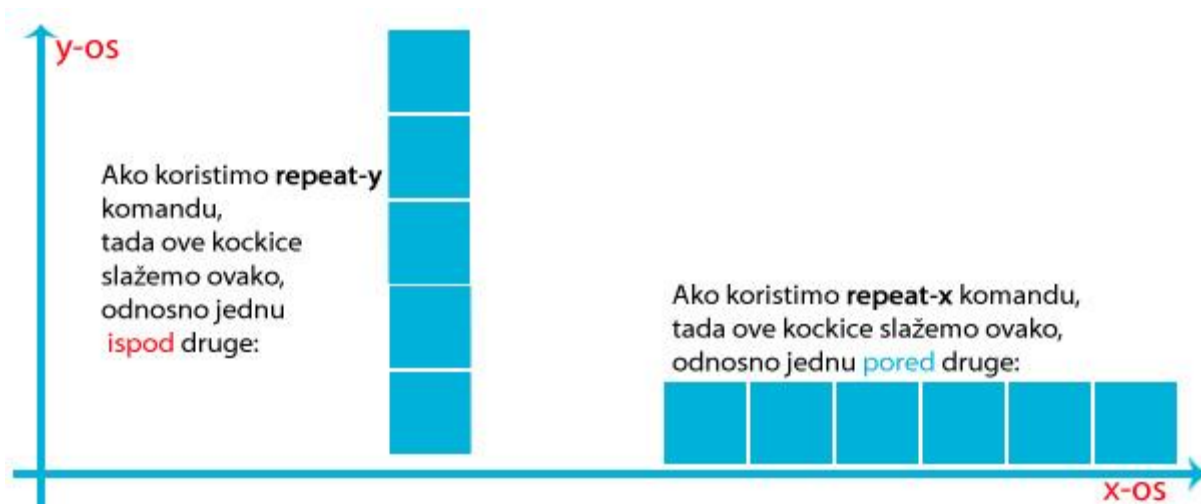
U prošloj sam lekciji objasnio kako možete pomoću svojstva `background` promjeniti boju pozadine, znači upišete `background:#FF0000`; i boja pozadine elementa u koji upisujete ovo svojstvo će se promjeniti (u ovom slučaju u crvenu boju.)

Tag `body` je isto element na stranici poput `p` taga ili `div` taga. Jedina je razlika što se svi ostali tagovi koji sačinjavaju strukturu stranice moraju nalaziti unutar okvira `body` taga da bi stranica ispravno radila.

Ako želimo promjeniti pozadinu stranice, moramo upisati svojstvo za pozadinu u `body` tag. No ovaj puta ćemo to malo drukčije napraviti, nego što smo to zadnji put. U CSS upišite sljedeći kod.

```
1  body{ background: url(images/pozadina-1.jpg) repeat-x top
   left; }
```

[Pogledajte](#) što biste trebali dobiti. Malo ćemo proučiti što smo to točno napravili. Prvo smo rekli na koji element želimo primjeniti neko svojstvo, u našem slučaju, `body`. Zatim smo rekli da želimo definirati svojstvo pozadine odnosno `background`. I sad dolazimo do novih stvari, sa `url` smo rekli browseru da počne tražiti neku sliku, i u zagradi smo mu rekli gdje se ta slika nalazi. Znači slika se nalazi u mapi **images** i zove se **pozadina-1.jpg**. Ako ste pogledali sliku koju sam dao da skinete, možda ste primjetili da je slika jako uska (samo 2px) i da je visoka kakvih 500px. I sad je pitanje, kako to da je ta slika koja je široka svega 2px uspjela prekriti cijelu stranicu. Tu dolazi naredba `repeat-x`. `Repeat-x` govori browseru da ponavlja sliku **pozadina-1.jpg** po X osi ekrana. Uvijek pišite `background` kao u ovom primjeru. I svakako nakon zatvorene zagrade napravite razmak, pa onda `repeat`. Jer sam primjetio da Internet Explorer ponekad ima problema sa time. Znači `url`, pa pozicija slike.



Znači browser je uzeo našu sliku i “umnožio” je toliko puta koliko je potrebno da prekrije s njom cijeli ekran. Kada smo napisali *top* to je značilo da browser treba početi prikazivati sliku na vrhu *body* elementa, što je u ovom slučaju i vrh stranice, i to sa lijeve strane. Ali znači sliku je ponavljao samo po x-osi ekrana, kao što vidite donji dio ekrana je ostao bijele boje. Ako bi željeli da se pozadina ponavlja i po x i po y osi trebamo upisati samo *repeat* umjesto *repeat-x* ili *repeat-y*. U našem primjeru mi to nećemo napraviti, nego ćemo ostaviti sve kako je, ali ćemo napraviti da nam stranica izgleda malo bolje, znači da nema tog bijelog dijela, nego da cijela pozadina izgleda neprekinuto, i da izgleda kao jedna cijelina. Da bi to učinili moramo samo dodati još boju sa kojom završava naša slika, ta boja ima heksadecimalnu vrijednost #00B2D9. Tako da ćemo u nas kod samo dodati tu boju, to bi trebalo ovako na kraju izgledat:

```
1 body{ background: #00B2D9 url(images/pozadina-1.jpg) repeat-
  x top left; }
```

Trebali bi dobiti [ovo](#). Znači samo smo dodali boju koja nam je tako reći popunila onaj bijeli dio stranice.

Za zadatak isprobajte kako bi izgledalo da umjesto *repeat-x* upišete *repeat-y*. Probajte skoristiti i neke druge slike, za pozadinu. Uglavnom poigrajte se malo sa ovim svojstvom da bi ga lakše naučili.

## FONTOVI I SLOVA

Naravno za web stranicu moramo koristiti i određene fontove a i trebaju nam neka svojstva pomoću kojih ćemo definirati izgled istih. Pošto nam se tekstovi nalaze u *p* tagovima, onda ćemo sljedeća svojstva primjenjivati na njih, naravno.

U ovom slučaju ću vam prvo pokazati kako ćete svojstva za fontove napisati produženo pa ćemo onda napraviti skraćeno. Recimo da želimo napraviti sljedeće, želimo da slova u našim odlomcima, budu Arial, da budu veličine 14px, da imaju razmak između redaka (prored) od 1,5 redak. I da budu bijele boje. Duži oblik pisanja bi bio:

```
1 p {
2   font-family: Arial, Helvetica, sans-serif;
3   font-size:14px;
4   line-height:1.5;
5   color:#FFF;
6 }
```

Rezultat pogledajte [ovdje](#). Sa *font-family* smo rekli browseru da koristi font Arial ako ga korisnik ima instaliranog na računalu, ako nema neka koristi Helveticu, ako nema ni Helveticu neka koristi bilo koji font iz sans-serif obitelji fontova.

U *font-family* možete upisati bilo koji font. Ali problem je u tome da korisnik mora taj font imati instaliran na svojem računalu. Zato se za *font-family* koriste fontovi za koje se zna da ih većina korisnika ima, pošto dolaze sa većinom operativnih sustava, a to su: Arial, Helvetica, Times New Roman, i ponekad Tahoma, Trebuchet MS, Courier, Geneva i Georgia.

Nakon *font-family* smo imali *font-size* što je naravno veličina fonta u pikselima. Tu se nema što puno objašnjavati, jedino bih naglasio da pikseli nisu jedini način određivanja veličine slova, koriste se još neke druge mjerne jedinice, koje ćemo kasnije objasniti.

Sa *line-height* određujemo pored, odnosno razmak između redaka, taj razmak možemo odrediti ovako kao što sam ja napisao 1.5, to znači da će razmak biti jedan i po redak. A mogli smo ga odrediti i u pikselima. Iako preporučam određivanje u redcima zato što je 1.5 redak nakakav standardni pored.

Na kraju nam je ostalo svojstvo *color*. *Color* je naravno boja slova koju isto kao i kod *backgrounda* i *bordera* možete definirati heksadecimalnim kodom, rgb kodom ili engleskom riječi za boju koju želite. Pretpostavljam da svojstvo *color* ima takav naziv, a ne *font-color*, iz razloga što je prva verzija CSS-a bila većinom orijentirana prema tekstu. Čak i u Dreamweaveru, kada želite u HTML pridružiti CSS datoteku idete na Text pa CSS Styles. Čak i u verziji CS3 uređivanje CSS-a se još uvijek nalazi kao pod izbornik u glavnom izborniku Text.

Pokazao sam vam kako izgleda kada svojstva fonta upisujete produženo, to skraćeno izgleda ovako:

```
1 p { font:normal 14px/1.5 Arial, Helvetica, sans-serif;
    color:#FFF; }
```

Znači sve smo to uspjeli staviti u jedan red, iako smo ipak *color* morali posebno odvojiti. E sad, pogledajte da na početku imamo vrijednost *normal*. Ova vrijednost nam označava da debljina, odnosno težina fonta bude normalna, da smo htjeli podebljana slova tu bi upisali *bold* umjesto *normal*. A u onom dužem pisanju bi to definirali kao **font-weight** svojstvo.

Pošto fontovi imaju mali milijun svojstava, ja ću pokušati navesti one koji bi vam mogli biti korisni u donjoj tablici, moram reći da fontovi imaju i više od ovih svojstava koje ću ja dolje nabrojati, ali nažalost dosta od njih rade samo u nekim browserima, a neki ne rade ni u jednom browseru. Ali ih naravno nećemo sve isprobati. To će biti vaš zadatak da samostalno napravite.

Svojstvo	Opis	Vrijednosti
		<b>Fontovi su:</b> Arial, Helvetica, Tahoma, Genova, Georgia, Times New Roman, Trebuchet MS.
font-family	Određuje koji će se font ili obitelj fontova koristiti.	<b>Obitelji fontova su:</b> sans-serif, serif, monospace
font-size	Određuje veličinu fonta u pikselima, u postotcima ili em.	broj <b>px</b> , broj <b>%</b> , broj <b>em</b>
font-style	Određuju stil fonta	<i>italic</i> , <i>oblique</i> , normal. Italic i oblique daju isti rezultat, odnosno ukošena slova
font-variant	Određuje varijantu fonta	normal, small-caps (kada napišete small-caps to znači da će vam sva slova biti velika, ali će biti smanjena, dok će npr. slova na početku rečenice biti veća od ostatka, iako će i ona biti velika.)

font-weight	Određuje težinu fonta, odnosno dali su i koliko su slova podebljana	bold, bolder, lighter, normal, brojevi od 100 do 900. Iako iz mog iskustva mogu reći da se ovdje jedino isplati koristiti <b>bold</b> i normal. Pošto oni jedini daju nekakav učinak, sve ostalo se ne primjeti.
letter-spacing	Razmak između slova. Vrlo korisno svojstvo ako se želite igrati sa fontovima.	Upišete razmak između slova u pikselima, npr. letter-spacing: 5px;
line-height	Prored u pikselima ili brojevima redaka	Upišete ili broj piksela i broj redaka razmaka. npr. line-height: 1.5;
text-decoration	Ukrašavanje fonta	blink (blinkanje), underline (podvučeno), line-through (linija kroz sredinu), overline (linija iznad teksta)
text-align	Ovo je jedan od najvažnijih svojstava, znači poravnanje teksta	left (lijevo), center (po sredini), right(desno), justify(poravnato sa obje strane), ove vrijednosti se odnose naravno na objekt u kojem se tekst nalazi. Ako je tekst u pelementu, onda će se poravnanje vršiti u odnosu na taj element.
text-transform	Transformira tekst	capitalize (promjenit će tekst tako da će svaka rečenica imati veliko početno slovo), uppercase (pretvorit će sav tekst u velika slova), lowercase (pretvorit će sav tekst u mala slova)

## ZAKLJUČAK

U ovoj smo lekciji obradili još 2 CSS svojstva. Što je mislim sasvim dovoljno za nekakvo upoznavanje sa CSS-om. Ovo ne znači da CSS više nećemo raditi, čak naprotiv, tek smo počeli. Ali od idućeg puta ovi tutoriali će biti većinom u video formatu, zato što bih morao pisati ogromne tekstove da bi vam dalje objašnjavao šta i kako, a to mi se neda. Pošto stvari postaju sve kompliciranije. Znači idući puta nastavljamo sa CSS-om ali u video formatu, a vi se dotad igrajte sa ovime što smo do sada obradili. Najbolje ćete CSS naučiti ako stalno budete isprobavali njegova svojstva i njegove mogućnosti. Naravno, sva pitanja, komentare, pohvale i kritike pošaljite putem komentara.

## Dizajniranje bez tablica za početnike #2 – Izbornici

Evo kao što sam obećao napravio sam video tutorial za izbornike i CSS. Ispričavam se odmah na kvaliteti zvuka i možda na još nekim problemima unutar videa. Imajte razumjevanja, ipak mi je ovo prvi ovakav tutorial.

Pošto ovdje kvaliteta i nije baš neka, možete ovaj video skinuti tako da odete na <http://www.keepvid.com> i zalijepite ovu adresu <http://video.google.com/videoplay?docid=1425392176111052062&hl=en> i tamo ćete moći izbirati dali ga želite skinuti u .flv .avi ili ipod formatu. Ja preporučujem .avi pošto ćete onda imati najbolju kvalitetu slike.

Nekoliko napomena na početku. Ovaj tutorial će biti malo kraći od onog prijašnjeg. Zašto? Zato što će uz njega ići i video tutorial, kojeg ću pokušati napraviti i postaviti još danas. Video tutorial ću raditi zato da bi vam bilo lakše shvatiti neke stvari, a i isto tako da mogu objasniti neke povezanosti i pravila puno brže i efikasnije nego da to radim u pisanom obliku. Ako prijem takvog tutoriala bude dobar nastavit ću raditi samo video tutoriale, pogledao sam i 74% ljudi koji su došli na prijašnji tutorial imaju neku vrstu brze veze na internet tako da mislim da to neće biti problem. Isto tako video tutoriali će se moći i skinuti da ga možete kasnije lakše ponovno pregledati. Ajmo sad na tutorial.

### Početak

Ovaj tutorial je nastavak na [prvi tableless tutorial](#) koji sam napravio. Znači ako niste odradili prvi tutorial, odradite ga ili [skinite gotovo dijelo](#).

Za ovaj tutorial će nam trebati još neke dodatne, većinom, grafičke datoteke koje ćete moći skinuti kada dođemo do tog dijela.

Ja sam preimenovao datoteke *tableless.css* i *tableless.html* u **izbornik.css** i **izbornik.html**. Naravno, čim napravite neku takvu promjenu morate promijeniti i pozive na datoteke, u ovom slučaju poziv na css datoteku u html datoteci, inače nam ova vježba neće uspjeti. Dakle promijenite ovu liniju koda koja vam se nalazi pri početku .html datoteke:

```
1 <link href="tableless_design_za_pocetnike.css" rel="stylesheet"
1 type="text/css" />
```

u ovo:

```
1 <link href="izbornik.css" rel="stylesheet" type="text/css" />
```

Naravno ako nećete promijeniti imena datoteka iz prošlog tutoriala, ovaj gornji korak vam je nepotreban.

Napraviti ćemo 3 izbornika. Jedan će biti horizontalni, jedan vertikalni a na jednom ćemo pokazati kako koristiti slike kao pozadine vaših izbora i kako te slike mjenjati kada miš pređe preko njih.

**PRIVATNE LEKCIJE IZ WEB DIZAJNA**  
prve 2 prijave dobivaju popust od 20% [PRIJAVITE SE](#)

### Izbornik #1 – Vertikalni

Sada na našoj prvoj stranici već imamo jedan vertikalni izbornik, ali on je malo prejednostavan, znači čisti tekst koji se podvuče kada kada pređemo preko njega mišem. Ono što ćemo napraviti sada, je da ga malo uljepšamo. Da bi izbornik mogli pravilno pozivati dodat ćemo još jedan DIV u naš .html dokument. Ali tako da "omotamo" već postojeće elemente na stranici. Ti elementi su ul i li tagovi. Napravite ovako dodajte ovu liniju ispred <ul > taga:

```
1 <div id="izbornik">
```

i naravno

```
1 </div>
```

ispod `</ul>` taga. Sada bi to trebalo izgledati ovako:

```
1
2 <div id="izbornik"><!--početak izbornika-->
3
4     <ul>
5         <li><a href="#">Početna</a></li>
6         <li><a href="#">Blog</a></li>
7         <li><a href="#">Tutoriali</a></li>
8         <li><a href="#">Kontakt</a></li>
9     </ul>
10 </div><!--kraj izbornika-->
```

Ovo smo napravili da bi lakše i smisljeno mogli pozivati elemente izbornika u CSS-u. Sada ćemo otići u `.css` dokument (izbornik.css u mom slučaju) i definirati neka svojstva DIV elementa izbornik. Pošto nam se u `.html` dokumentu DIV izbornik nalazi unutar elementa lijevo, bilo bi dobro da i u `.css` dokumentu počnemo pisati ispod selektora za lijevi DIV:

```
1 #izbornik{
2     width:200px;
3     float:left;
4 }
```

Znači samo smo odredili da će nam širina biti 200px i da izbornik pluta lijevo. Sada ćemo dodati sljedeći selektor. Pazite, ovo NE dodajete selektoru gore nego pravite novi, koji mora izgledati ovako:

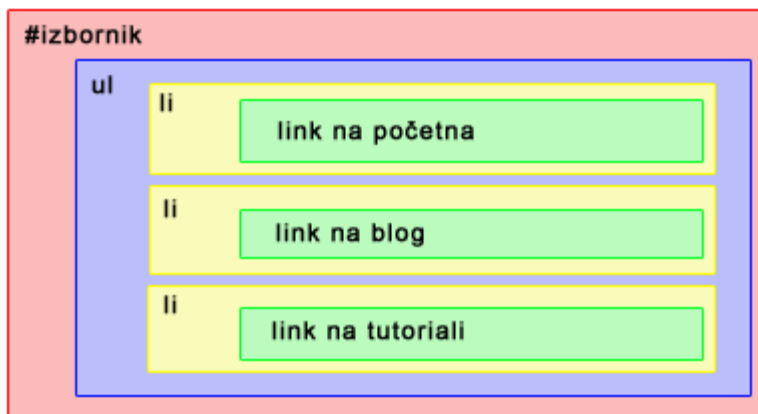
```
1 #izbornik ul{
2     width:200px;
3     padding:0px;
4     margin:0px;
5 }
```

Ovo je sada jedna nova stvar. Šta nam znači linija `#izbornik ul`? Ova linija govori pregledniku da se svojstva u ovom selektoru primjenjuju na `ul` tag koji se nalazi u DIV tagu *izbornik*. Ovakav način pisanja je vrlo koristan ako ćete imati još `ul` i `li` tagova koji će se nalaziti u drugom DIV tagu i koji neće služiti kao izbornik. Da smo napisali samo `ul`... to bi značilo da će se svojstva primjenjivati na sve `ul` tagove koji postoje u dokumentu. Znači mi smo ga sad izdvojili i reki da se ovakva svojstva koriste samo na `ul` tagove unutar *izbornik* DIVa. Ostala svojstva su vam nadam se jasna, širina:200px, padding 0 i margin 0. Ovaj selektor nam u osnovi i nije toliko bitan ali sam ga iskoristio da bih vas uveo u ovaj način pisanja. Ajmo dalje. Sada opet ovo NEMOJTE dodavati već postojećem selektoru nego napravite novi koji će izgledati ovako:

```
1 #izbornik ul li a:link, a:visited{
2
3     width:190px;
4     height:20px;
5     display:block;
6     background: #000000;
7     padding-left: 10px;
8
9 }
```

Što ovo sad znači? Prva linija je slična kao iz prethodnog primjera. `#izbornik ul li a:link, a:visited`, znači da će se ova svojstva primjenjivati na `a` tagove (odnosno linkove) koji se nalaze u `li` tagu koji se nadalje nalaze u `ul` tagu i koji se nalaze u DIVu *izbornik*. Nadam se da će vam ova sličica malo razjasniti situaciju:





sa **#izbornik ul li a:link, a:visited {...** zapravo određujemo kako će se ponašati samo zeleni blokovi na prikazu gore

Pošto su nam **a** tagovi linkovi, vidjeli smo u prvom tutorialu da oni mogu imati određene načine ponašanja, zato smo ovdje odmah naveli kako želimo da se ponašaju obični linkovi (**a:link**) i posjećeni linkovi (**a:visited**). Širinu sam stavio na 190px zato što sam kasnije dodao lijevi padding od 10px pa je  $190px + 10px = 200px$ , background je naravno pozadina, koja je u ovom slučaju crne boje. Ono što nam je ovdje zanimljivo je svojstvo **display:block** – u doslovnom prijevodu znači prikaži blok, što je ujedno i najjednostavniji opis ovog svojstva. Znači rekli smo mu da nam link prikazuje kao blok koji će biti širine 200px i visine 20px. Ovo moramo napraviti pošto linkovi sami po sebi nisu blokovi.

Ok, ajmo dalje, sada ćemo pomoću **css**-a reći kako želimo da nam se linkovi ponašaju kada miš prijeđe preko istih. I to ćemo napraviti vrlo jednostavno pomoću – kako smo već u prvom tutorialu vidjeli – **a:hover** pseudo-selektora. Napišite novi selektor sa sljedećim svojstvima:

```

1 #izbornik ul li a:hover{
2
3         background: #FFFFFF;
4         color:#000000;
5     }

```

Ako smo sve napravili kako treba, sada kada pređemo mišem preko izbornika trebao bi promijeniti boju u bijelu. A slova da budu crne boje. Primjetit ćete da će vam se kada pređete mišem preko slova pojaviti crta ispod njih. To je zato što smo na početku **.css** dokumenta na **a:hover** odredili to svojstvo (**text-decoration:underline**) i sada je ovaj selektor naslijedio to svojstvo (o nasljeđivanju ću više pričati u video tutorialu). Ako ne želite da vam se pojavljuje ta crta onda samo poništite svojstvo **text-decoration** i to tako da u gornji selektor dodate liniju **text-decoration:none**. Sada bi vam sav kod koji smo upisali trebao izgledati ovako:

```

1 #izbornik{
2     width:200px;
3     float:left;
4 }
5
6 #izbornik ul{
7     width:200px;
8     padding:0px;
9     margin:0px;
10 }
11
12 #izbornik ul li a:link, a:visited{
13     width:190px;
14     height:20px;
15     display:block;
16     background: #000000;

```

```

16         padding-left: 10px;
17
18     }
19
20 #izbornik ul li a:hover{
21     background: #FFFFFF;
22     color:#000000;
23     text-decoration:none;
24 }
25
26
27

```

A stranica bi vam trebala izgledati otprilike ovako (kliknite na sliku za veći prikaz):



## Horizontalni izbornici

Sada ćemo izraditi horizontalni izbornik, i za njega ćemo koristiti isti kod koji smo do sad napisali, ali ćemo ga malo modificirati. Za početak ćemo morati izbornik premjestiti na drugu lokaciju, jednostavno zato što u DIVu *lijevo* nemamo mjesta da prikažemo cijeli izbornik. Zato ćete ovaj dio koda:

```

1   <div id="izbornik"><!--početak izbornika-->
2
3       <ul>
4           <li><a href="#">Početna</a></li>
5           <li><a href="#">Blog</a></li>
6           <li><a href="#">Tutoriali</a></li>
7           <li><a href="#">Kontakt</a></li>
8       </ul>
9
10      </div><!--kraj izbornika-->

```

Kopirati i zatim obrisati te prebaciti u DIV *desno* pa će vam kod izgledati otprilike ovako:

```

1   <div id="desno">
2
3       <div id="izbornik"><!--početak izbornika-->
4
5           <ul>
6               <li><a href="#">Početna</a></li>
7               <li><a href="#">Blog</a></li>
8               <li><a href="#">Tutoriali</a></li>
9               <li><a href="#">Kontakt</a></li>
10          </ul>
11
12      </div><!--kraj izbornika-->
13
14          <h2>Tableless Design Za Početnike</h2>

```

```
14         <p>Lorem ipsum dolor sit amet,  
15 consetetur.....  
16
```

A stranica bi vam trebala izgledati otprilike ovako (kliknite na sliku za veći prikaz):



Ovo nam naravno nikako ne odgovara. Pošto, za početak, zapravo taj izbornik nije horizontalan nego je samo premješten u drugi DIV. Sada da bi napravili pravi moramo se prebaciti u .css dokument i tamo doraditi CSS da nam prikazuje horizontalni izbornik. Prvo ćemo zakomentirati i promijeniti sljedeće linije: u selektoru `#izbornik` zakomentirajte `width:200px`, u selektoru `#izbornik ul` zakomentirajte isto `width:200px` – ovo smo napravili zato što nam je ovdje određena fiksna širina elemenata, a kada radimo horizontalne izbornike ne želimo da nam širina bude fiksna.

Nadalje u selektoru `#izbornik ul li a:link, a:visited` promijenite `width:200px` u `width:100px`. Ovo radimo zato što nam je DIV *desno* širok samo 480px i kad bi ostavili da nam širina bude 200px, ukupna širina bi nam bila 800px, pošto imamo 4 izbora, pa je  $4 \times 200 = 800$ px. Zato upišemo da nam širina bude 100 pa je onda  $4 \times 100 = 400$  i na to dodamo još padding sa lijeve strane što je za svaki izbor 10px znači dobivamo ukupnu širinu od 440px. Znači kod bi vam nakon tih promjena trebao izgledati ovako:

```
1  
2 #izbornik{  
3     /*width:200px;*/  
4  
5     }  
6  
7 #izbornik ul{  
8     /*width:200px;*/  
9     padding:0px;  
10    margin:0px;  
11    }  
12  
13 #izbornik ul li a:link, a:visited{  
14  
15     width:100px;  
16     height:20px;  
17     display:block;  
18     background: #000000;  
19     padding-left: 10px;  
20 }  
21  
22 #izbornik ul li a:hover{  
23  
24     background: #FFFFFF;  
25     color:#000000;  
26     text-decoration:none;  
27     }
```

I što se promjenilo? Pa ništa zapravo osim širine izbornika koji je sada 100px. Sada slijedi prava čarolija. Da bi nam sad izbornik postao horizontalan samo ubacimo bilo gdje u selektor **#izbornik ul li a:link, a:visited** ovu liniju:

```
1 float:left;
```

pa će vam kod izgledati ovako:

```
1
2 #izbornik ul li a:link, a:visited{
3
4     width:100px;
5     height:20px;
6     display:block;
7     background: #000000;
8     padding-left: 10px;
9     float:left;
10 }
```

I sad ako radite u Firefoxu sve izgleda ok.... No postoji tu jedan browser koji nama, web dizajnerima iz dana u dan zagorčava život. Govorim naravno o Internet Exploreru. Evo ovako bi sad naš izbornik trebao izgledati u Firefoxu (a i u većini drugih browsera koji nisu IE):



A ovako će izgledati u Internet Exploreru:



Kao što vidite razlika je poprilično velika. Ako pogledate ovu stranicu u oba preglednika vidjet ćete da postoje neke sitne razlike u pozicioniranju elemenata. To nam i nije toliki problem. Ali ovo je, kada dizajniramo stranice trebali bi ih raditi tako da ih većina popularnih browsera može pravilno pročitati. Pošto većina ljudi još uvijek koristi IE ovu razliku moramo ispraviti. To ćemo napraviti dosta jednostavno. Ispod selektora **#izbornik ul** dodajte drugi selektor koji će se zvati **#izbornik ul li** i u njega upišite svojstvo **float:left**, znači ovako će to izledati:

```
1 #izbornik ul{
2     /*width:200px;*/
3     padding:0px;
4     margin:0px;
5
6     }
7
8     #izbornik ul li{
9         float:left;
```

10  
11  
12

}

I to bi trebalo riješiti taj problem, sada bi izbornik trebao izgledati isto i u IE i u FF. E, sad kada smo riješili to, ajmo pogledat što se desilo zapravo. Kako je vertikalni izbornik postao horizontalni. Znači uz one sitne promjene smo dodali, `float:left` u **#izbornik ul li a:link, a:visited**. `float:left` je zapravo izvršio transformaciju izbornika iz vertikalnog u horizontalni. Zato što je link i li elementima rekao da plutaju i to na lijevu stranu, a kad neki element pluta to znači da se neće prebaciti u drugi red dok god ima mjesta u elementu u kojem se nalazi. Možda će vam ova sličica bolje razjasniti što se desilo:



Ok to bi bilo sve što se tiče horizontalnog izbornika. Naravno ima se tu još puno toga za reći ali ne želim sada duljiti. Sada prelazimo na tzv. fancy izbornike koji će se samo nastaviti na ovaj i poboljšati ga.

## “Fancy” izbornici

Da bi ovaj izbornik napravili da bude “fancy” moramo dodati samo nekoliko linija koda, poslužiti se lukavošću i jednom vrlo jednostavnom sličicom. Sličicu možete desni klik -> Save image as... ispod:



Ovo je nekakav klasičan izbornik koji ima dva stanja, jedan dok je miš negdje okolo i drugo stanje kada se miš nalazi preko izbora. E sad, ako ste prije radili izbornike u dreamweaveru imali ste onu swap image funkciju, gdje odaberete jednu sliku za prvo stanje i jednu za drugo stanje i dobivate isti efekt koji ću vam ja sad tu pokazati s razliko što će vam DW još dodati gomilu JavaScripta i tako vam onečistiti kod.

U čem je stvar sa gornjom sličicom? Nju ćemo koristiti kao pozadinu za naš izbornik, jedna polovica sličice predstavlja prvo stanje a druga polovica stanje kada pređemo mišem. Pošto smo definirali da su nam izbornici visoki 20px napravio sam ovu sličicu da bude visoka duplo – 40px. Kada stavimo da pozadina izbornika bude ova sličica mi ćemo moći vidjeti samo njezinih prvih 20px a ostatak će ostati sakriven pošto je izbornik visine samo 20px. Znači u selektor **#izbornik ul li a:link, a:visited** ćemo postaviti da na pozadina bude ova sličica (u mom slučaju se zove izbornik-bg.gif), samo ovom linijom zamjenite svojstvo `background`:

```
1 background:url(images/izbornik-bg.gif) repeat-x;
```

i sada kada pogledate stranicu vidjet ćete da je pozadina vaših izbora prvih 20px sličice. Kada pređete preko nekog izbora sve i dalje ostaje isto. Ok, ajmo sad to riješit u selektoru **#izbornik ul li a:hover** obrišite svojstvo `background` i umjesto njega dodajte ovu liniju:

```
1 background-position:bottom;
```

i VOILA! Sada kada pređete preko nekog izbora pozadina se pretvori u donji dio sličice. Ovo se dešava zato što smo sa `background-position:bottom` rekli da nam tu sličicu prikazuje od dna.

Sada bi vaš izbornik trebao izgledati ovako:



A kod bi vam trebao izgledati ovako:

```
1
2
3 #izbornik{
4     /*width:200px;*/
5
6     }
7
8 #izbornik ul{
9     /*width:200px;*/
10    padding:0px;
11    margin:0px;
12
13    }
14
15 #izbornik ul li{
16
17     float:left;
18
19    }
20
21 #izbornik ul li a:link, a:visited{
22
23     width:100px;
24     height:20px;
25     display:block;
26     background:url(images/izbornik-bg.gif)
27     repeat-x;
28     padding-left: 10px;
29     float:left;
30
31    }
32
33 #izbornik ul li a:hover{
34     background-position:bottom;
35
36     color:#000000;
37     text-decoration:none;
38    }
```

Ovo je jako dobra praksa izrade izbornika, zato što imate samo jednu sličicu za oba stanja i preglednici je odmah učitavaju, tako da se neće dešavati da korisnik dođe na neki izbornik i onda treba pričekat da mu se učita sličica za drugo stanje, ovako je sličica odmah cijela učitana. I za razliku od DW-a imat ćete izbornik koji će raditi na većini browsera nove generacije a nećete imati ni liniju JavaScript koda.

Eto to bi bilo to što se tiče izbornika za ovaj tutorial. Šaljite mi komentare za ovaj i prijedloge za iduće tutoriale.

## WordPress 05 – Izrada teme – Drugi dio

U ovoj epizodi nastavljamo sa prilagodbom već gotovog HTML+CSS predloška u funkcionalnu WordPress temu. Dodati ćemo još neke dodatne funkcionalnosti u našu temu i napraviti da bude skoro u potpunosti upotrebljiva.

### U OVOM TUTORIALU ĆETE NAUČITI:

- Kako razdijeliti index.php datoteku u 3 sastavna dijela
- Kako dodati izbornik (menu) u WordPress 3
- Kako dodati regiju u koju ćete stavljati widgete

### DOWNLOAD TEME SA DO SAD NAPRAVLJENIM PROMJENAMA

[Download](#)

#### **Prikaz naslova u title tagu**

```
1 <?php bloginfo('name'); ?> <?php if ( is_single() ) { ?> &amp;amp;amp;amp;raquo; ;
  <?php } ?> <?php wp_title(); ?>
```

#### **Prikaz naslova web stranice i link na početnu**

```
1 <h1><a href="<?php echo get_option('home'); ?>/"><?php
  bloginfo('name'); ?></a></h1>
```

#### **Pozivi na header.php, sidebar.php i footer.php**

```
1 <?php get_header(); ?> , <?php get_sidebar(); ?> , <?php
  get_footer(); ?>
```

#### **Obavezni pozivi koji idu u header.php i footer.php**

```
1 <?php wp_head(); ?> , <?php wp_footer(); ?>
```

#### **Definiranje izbornika u functions.php**

```
1 if (function_exists('add_theme_support')) {
2     add_theme_support('menus');
3 }
```

#### **Definiranje pozicije izbornika u header.php**

```
1 <?php wp_nav_menu('menu=first'); ?>
```

#### **Definiranje sidebara u functions.php**

```
1 if ( function_exists('register_sidebar') )
2     register_sidebar(array(
3         'name' => 'SidebarJedan',
```

```
3     'before_widget' => '<li>',
4     'after_widget' => '</li>',
5     'before_title' => '<h2>',
6     'after_title' => '</h2>',
7     ));
8
```

### ***Definiranje pozicije sidebara u sidebar.php***

```
1 <ul>
2     <?php if ( !function_exists('dynamic_sidebar') ||
3     !dynamic_sidebar(SidebarJedan) ) : ?><?php endif; ?>
4 </ul>
```

Ostali Tutorijali na linku: <http://www.xtorials.com/tutorials/wordpress-04-izrada-teme-prvi-dio>