

AV4-primjeri

CSS boje i pozadina

- U prošlim poglavljima smo naučili upravljati svojstvima fonta i teksta, a u ovom poglavlju ćemo naučiti kako upravljati njihovim bojama i pozadinom kako bismo naše stranice učinili ljepšima. Jer ako dobro složimo boje i pozadine naša će stranice izgledati mnogo profesionalnije.
 - color
 - background-color
 - background-image
 - background-repeat
 - background-attachment
 - background-position
 - background

color

- Svojstvo koje postavlja boju teksta u zadanom elementu, no ponekad može označavati i boju samog elementa kao na primjer kod oznake `<hr>`. Najbolje ćemo to demonstrirati na primjeru:

```
<html>
<head>
<style type="text/css">
p.plavi {color:blue;}
p.crveni { color:red; }
hr { color:green; }
</style></head>
<body>
<p class="plavi">Ovo je odlomak sa plavim tekstom.
</p>
<hr>
<p class="crveni"> Ovo je odlomak sa crvenim tekstom.
</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je odlomak sa plavim tekstom.

Ovo je odlomak sa crvenim tekstom.

background-color

- Postavlja boju pozadine u danom elementu i ispunjava čitavu pozadinu elementa, čak i u slučaju ako mi stavimo da imamo odmak teksta od ruba elementa boja će ispunjavati čitav element a ne samo dio gdje je tekst. U sljedećem primjeru stavili smo da je odmak teksta od ruba elementa 10 piksela i vidimo da boja ispunjava čitavu pozadinu elementa.

```
<html>
<head>
<style type="text/css">
div.odmak {color:blue; background-color:yellow; padding:10px;
           border:2px solid blue; width:500px; }
div.bezodmaka { color:red; background-color:green;
               border:2px solid red; width:500px; }
</style></head>
<body>
<div class="odmak">Ovo je odlomak sa plavim tekstom i odmakom od
ruba od 10 px.
</div> <br><br>
<div class="bezodmaka"> Ovo je odlomak sa crvenim tekstom i bez
odmaka od ruba elementa.</div>
</body>
</html>
```

- Slijedi rezultat u web pregledniku:

Ovo je odlomak sa plavim tekstom i odmakom od ruba od 10 px.

Ovo je odlomak sa crvenim tekstom i bez odmaka od ruba elementa.

background-image

- Svojstvo koje u zadani element postavlja sliku kao pozadinu. Ovo svojstvo se kombinira sa još nekim svojstvima kako bismo mogli definirati način na koji će se definirati položaj slike u pozadini. Tako je moguće odrediti da slika bude fiksna i da se ne pomiče sa sadržajem ili da se ponavlja po nekoj od osi. Slijede svojstva koja se koriste za definiranje položaja slike u pozadini.

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif '); }
p {color:red; font-size:16pt; }
</style></head>
<body>
<p>Po defaultu slika se ponavlja i vertikalno i horizontalno.</p>
</body>
</html>
```

- Slijedi rezultat u web pregledniku iz kojeg vidimo da se slika ponavlja i horizontalno i vertikalno, ako mi nismo definirali drugačije:



background-repeat

- Svojstvo pomoću kojeg određujemo ponavljanje slike u pozadini i može imati sljedeće vrijednosti: repeat (slika se ponavlja), no-repeat (slika se ne ponavlja), repeat-x (slika se ponavlja po horizontalnoj osi) i repeat-y (slika se ponavlja po vertikalnoj osi).
- Slijedi primjer ponavljanja slike po horizontalnoj osi:

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif '); background-repeat:repeat-x; }
p { color:red; font-size:16pt; }
</style></head>
<body>
<p>Primjer u kojem se slika ponavlja horizontalno.</p>
</body>
</html>
```

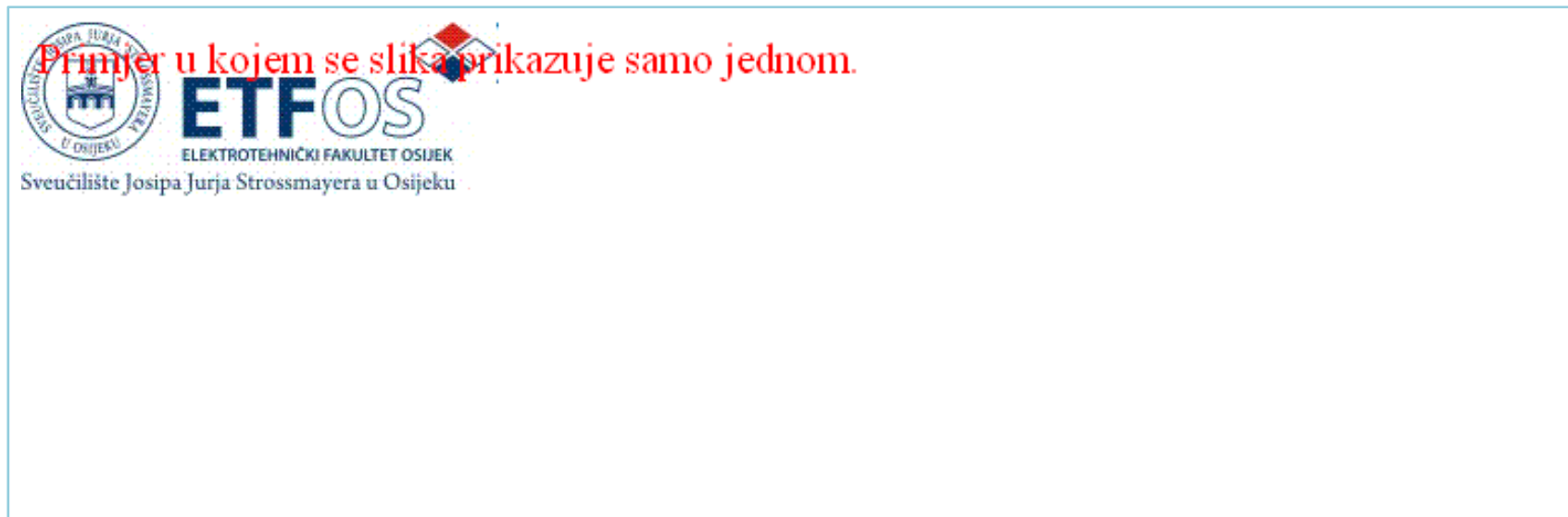
- Slijedi rezultat iz kojeg vidimo da se slika ponavlja samo horizontalno:



- Primjer kada želimo da se slika pokaže samo jednom:

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif ');
      background-repeat:no-repeat; }
p {color:red; font-size:16pt; }
</style>
</head>
<body>
<p>Primjer u kojem se slika prikazuje samo jednom.</p>
</body>
</html>
```

- Slijedi rezultat:



background-attachment

- svojstvo koje određuje pomicanje pozadinske slike sa elementom i može imati dvije vrijednosti: scroll (slika se pomiče sa elementom) i fixed (slika se ne pomiče). Primjer u kojem želimo postaviti fiksnu sliku u pozadinu:

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif ');
      background-repeat:no-repeat;
      background-attachment:fixed; }
p {color:red; font-size:14pt; }
</style></head>
<body>
<p>U ovom slučaju ako imamo velike količine teksta i kad listamo
stranicu prema dolje slika uvijek ostaje na svome fiksnom
položaju.</p>
</body>
</html>
```

background-position

- ako nismo drugačije definirali slika se po definiranim postavkama uvijek postavlja u gornji lijevi kut elementa u kojem se nalazi. Ovo svojstvo nam omogućava da postavimo početni položaj u kojem će se nalaziti slika i svojstvo može poprimiti nekoliko vrijednosti: left, right, top, bottom, center. Isto tako možemo zadati parametre u pikselima koliko će slika biti udaljena od gornjeg lijevog kuta ili čak u postotcima. Slijedi primjer postavljanja početnog položaja slike u centar:

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif ');
      background-repeat:no-repeat;
      background-attachment:fixed;
      background-position:center; }
p {color:red; font-size:14pt; }
</style></head>
<body>
<p>Početni položaj slike je u centru.</p>
<body><html>
```

- Slijedi rezultat:

Početni položaj slike je u centru.



- Primjer određivanja položaja slike u odnosu na gornji lijevi kut elementa pomoću postotaka:

```
<html>
<head>
<style type="text/css">
body { background-image:url('etfos.gif ');
      background-repeat:no-repeat;
      background-attachment:fixed;
      background-position:40% 20%; }
p { color:red; font-size:14pt; }
</style>
</head>
<body>
<p>Početni položaj slike određen pomoću postotaka. 40% je
udaljenosti po horizontalnoj osi, <br> a 20% po vertikalnoj osi
u odnosu na gornji lijevi kut.</p>
</body>
</html>
```


- Slijedi rezultat:

Početni položaj slike određen pomoću postotaka. 40% je udaljenosti po horizontalnoj osi, a 20% po vertikalnoj osi u odnosu na gornji lijevi kut.

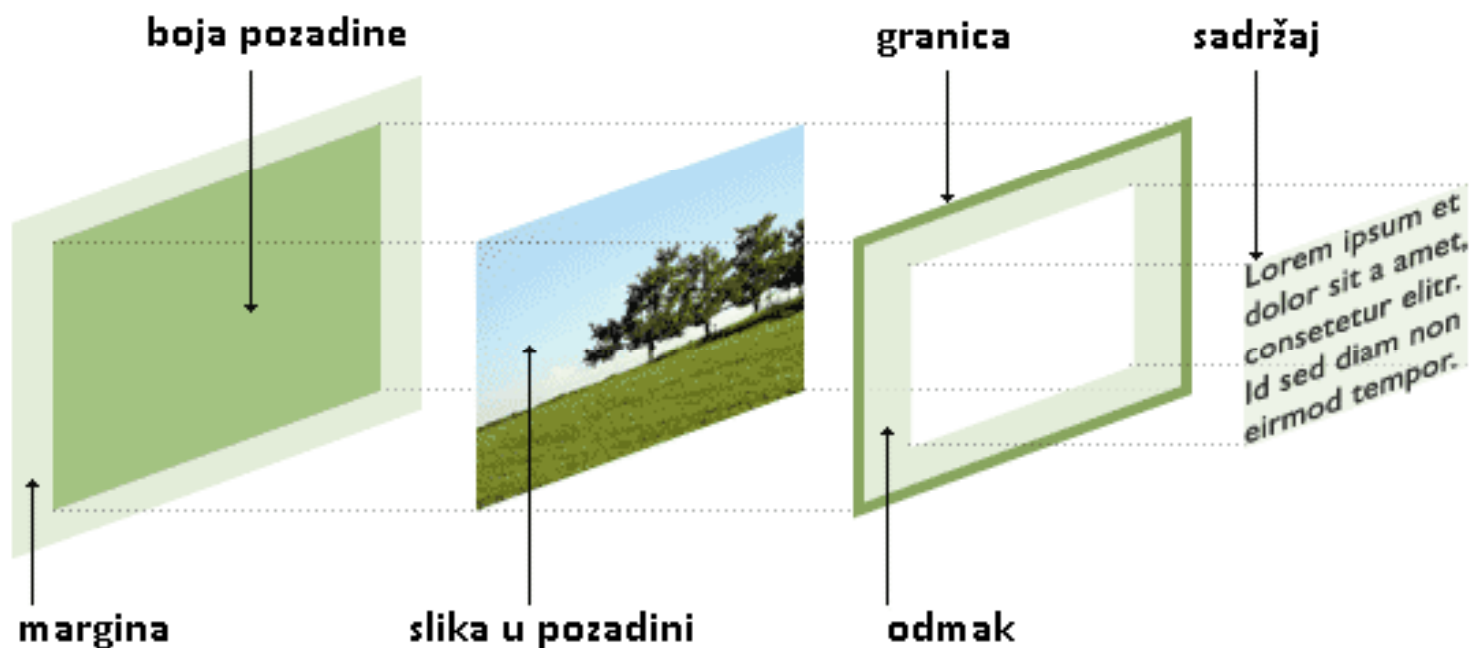


background

- ovo je skraćeno svojstvo u kojem možemo napisati sva gore navedena svojstva i nije bitan redoslijed navođenja svojstava. Kad pokrenete idući primjer vidjet ćete da slika ostaje nepomična na vrhu iako listate tekst:

[illegible]

CSS kutija i njena svojstva



CSS Margine

margin-top - gornja margina koja predstavlja udaljenost od elementa koji se nalazi iznad.

margin-right - desna margina koja predstavlja udaljenost od elementa koji se nalazi s desne strane.

margin-bottom - donja margina koja predstavlja udaljenost od elementa koji se nalazi ispod.

margin-left - lijeva margina koja predstavlja udaljenost od elementa koji se nalazi s lijeve strane.

margin - skraćeno svojstvo pomoću kojeg možemo navesti sve margine odjednom.

```
<html>
<head>
<style type="text/css">
p { background-color:red; }
p.margine{ background-color:red;
          margin-top:70px;
          margin-right:100px;
          margin-bottom:80px;
          margin-left:50px; }

</style>
</head>
<body>
<p> Odlomak koji nema postavljene margine.</p>
<p class="margine"> Odlomak koji ima postavljene margine.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Odlomak koji nema postavljene margine.

Odlomak koji ima postavljene margine.

- Da smo gornji primjer pisali pomoću skraćenog svojstva margin pisali bi na sljedeći način:

```
<html>
<head>
<style type="text/css">
p { background-color:red; }
p.margine{ background-color:red;
           margin:70px 100px 80px 50px; }
</style>
</head>
<body>
<p> Odlomak koji nema postavljene margine.</p>
<p class="margine"> Odlomak koji ima postavljene margine.</p>
</body>
</html>
```

- Dobijemo isti rezultat kao u gornjem primjeru. No bitno je navesti da se margine navode sljedećim redoslijedom. Prvo se navede gornja margina (70px), pa desna (100px), pa donja (80px) i na kraju lijeva (50px).

CSS odmaci

padding-top - odmak od gornje granice,

padding-right - odmak od desne granice,

padding-bottom - odmak od donje granice,

padding-left - odmak od lijeve granice,

padding - skraćeno pisanje svih odmakova.

```
<html>
<head>
<style type="text/css">
p { background-color:red; }
p.odlomak{ background-color:red;
padding-top:30px;
padding-right:60px;
padding-bottom:25px;
padding-left:40px; }

</style>
</head>
<body>
<p> Odlomak koji nema postavljene odmake.</p>
<p class="odlomak"> Odlomak koji ima postavljene odmake.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Odlomak koji nema postavljene odmake.

Odlomak koji ima postavljene odmake.

- Da smo gornji primjer pisali pomoću skraćenog svojstva padding pisali bi na sljedeći način:

```
<html>
<head>
<style type="text/css">
p { background-color:red; }
p.odlomak{ background-color:red;
           padding: 30px 60px 25px 40px; }
</style>
</head>
<body>
<p> Odlomak koji nema postavljene odmake.</p>
<p class="odlomak"> Odlomak koji ima postavljene odmake.</p>
</body>
</html>
```

- Bitno je navesti da se odmaci navode sljedećim redoslijedom. Prvo se navede gornja odmak (30px), pa desni (60px), pa donji (25px) i na kraju lijevi (40px). No ukoliko je odmak jednak sa svih strana, na primjer 20px možemo pisati:

```
p.odlomak{ background-color:red;
           padding: 20px
```

CSS granice

- Svojstva širine granice:

border-top-width - širina gornje granice,

border-right-width - širina desne granice,

border-bottom-width - širina donje granice,

border-left-width - širina lijeve granice.

```
<html>
<head>
<style type="text/css">
div.granica{background-color:red;
            border-top-width:10px;
            border-right-width:15px;
            border-bottom-width:20px;
            border-left-width:25px;
            border-style:solid;
            border-color:#00FF00;
            width:500px }
</style>
</head>
<body>
<div class="granica">Element koji ima postavljene granice.<br>
Postavili smo da se granica elementa povećava kako krenemo od gornje granice.<br>
Pozadina elementa je crvene boje a granice zelene boje.
</div>
</body>
</html>
```

- Kao rezultat dobijemo:

Element koji ima postavljene granice.

Postavili smo da se granica elementa povećava kako krenemo od gornje granice.

Pozadina elementa je crvene boje a granice zelene boje.

- Kao i prethodna svojstva, i ovo se svojstvo dijeli na četiri položaja - gore, desno, dolje i lijevo. Ako želimo jednaku debljinu granice od 10px sa sve četiri strane onda pišemo:

```
<html>
<head>
<style type="text/css">
div.granica{ background-color:red;
              border-width:10px;
              border-style:solid;
              border-color:#00FF00;
              width:500px }

</style>
</head>
<body>
<div class="granica">Element koji ima postavljene granice.<br>
Postavili smo da je granica elementa jednaka sa svih strana.<br>
Pozadina elementa je crvene boje a granice zelene boje.
</div>
</body>
</html>
```

- Svojstva boje granice:

border-top-color - boja gornje granice.

border-right-color - boja desne granice.

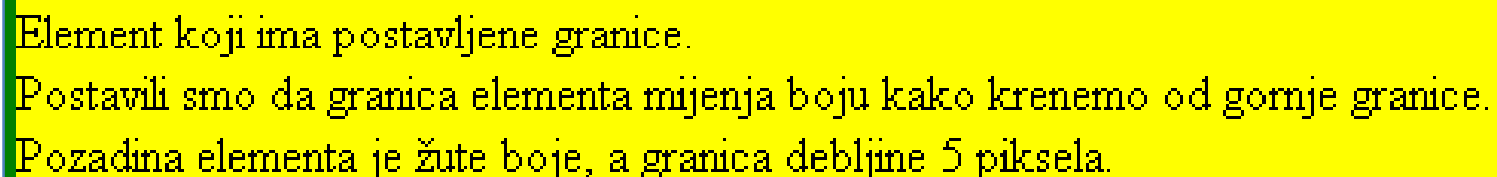
border-bottom-color - boja donje granice.

border-left-color - boja lijeve granice.

- Pogledajmo primjer:

```
<html>
<head>
<style type="text/css">
div.granica{ background-color:yellow;
              border-top-color:red;
              border-right-color:blue;
              border-bottom-color:black;
              border-left-color:green;
              border-style:solid;
              border-width:5px;
              width:500px }
</style>
</head>
<body>
<div class="granica">Element koji ima postavljene granice.<br>
Postavili smo da granica elementa mijenja boju kako krenemo od gornje
granice.<br>
Pozadina elementa je žute boje, a granica debljine 5 piksela.
</div>
</body>
</html>
```

- Kao rezultat dobijemo:



Element koji ima postavljene granice.
Postavili smo da granica elementa mijenja boju kako krenemo od gornje granice.
Pozadina elementa je žute boje, a granica debljine 5 piksela.

border-top-style - izgled gornje granice.

border-right-style - izgled desne granice.

border-bottom-style - izgled donje granice.

border-left-style - izgled lijeve granice.

```
<html>
<head>
<style type="text/css">
div.granica{ background-color:yellow;
              border-top-style:ridge;
              border-right-style:hidden;
              border-bottom-style:dashed;
              border-left-style:solid;
              border-color:green;
              border-width:5px;
              width:500px }

</style>
</head>
<body>
<div class="granica">Element koji ima postavljene granice.<br>
Postavili smo da granica elementa ima zelenu boju, ali da se izgled granice
mijenja kako <br>krenemo od gornje granice. Pozadina elementa je žute boje,
a granica debljine 5 piksela.
</div>
</body>
</html>
```


- Dobijemo rezultat:

Element koji ima postavljene granice.

Postavili smo da granica elementa ima zelenu boju, ali da se izgled granice mijenja kako

krenemo od gornje granice. Pozadina elementa je žute boje, a granica debljine 5 piksela.



- A ako nam je izgled jednak sa svih strana granice možemo pisati:

```
<html>
<head>
<style type="text/css">
div.granica{ background-color:yellow;
               border-style:ridge;
               border-color:green;
               border-width:5px;
               width:500px }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="granica">Element koji ima postavljene granice.<br>
```

Postavili smo da granica elementa ima zelenu boju, ali da je izgled granice jednak sa svih strana.
Pozadina elementa je žute boje, a granica debljine 5 piksela.

```
</div>
```

```
</body>
```

```
</html>
```

- Sliiedi rezultat :

Element koji ima postavljene granice.

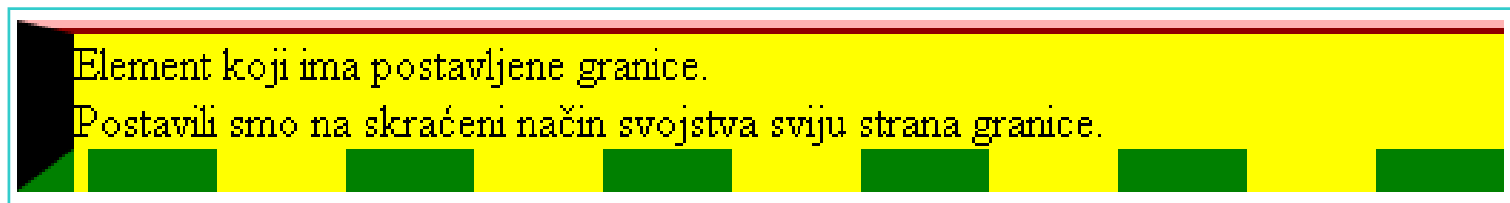
Postavili smo da granica elementa ima zelenu boju, ali da je izgled granice jednak sa svih strana.

. Pozadina elementa je žute boje, a granica debljine 5 piksela.

- Sva gore navedena svojstva možemo pisati skraćeno sa za svaku pojedinu stranu granice:

```
<html>
<head>
<style type="text/css">
div.granica{ background-color:yellow;
              border-top:5px red ridge;
              border-right:10px blue hidden;
              border-bottom:15px green dashed;
              border-left:20px black solid;
              width:500px }
</style>
</head>
<body>
<div class="granica">Element koji ima postavljene granice.<br>
Postavili smo na skraćeni način svojstva sviju strana granice.
</div>
</body>
</html>
```

- Dobijemo rezultat:



Dimenzije CSS kutije

- Da bismo definirali dimenzije CSS kutije ili nekog sadržaja koristimo svojstva širine i visine, koja mogu biti postavljena za bilo koji blok element:

width - određuje širinu CSS kutije.

height - određuje visinu CSS kutije.

- Pogledajmo primjer:

```
<html>
<head>
<style type="text/css">
img.slika{ height:200px;
           width:300px; }
</style>
</head>
<body>
<p>Slika kojoj nismo mijenjali dimenzije.</p>

<hr>
<p>Slika kojoj smo mijenjali dimenzije.</p>

<hr>
</body>
</html>
```

- U web pregledniku kao rezultat dobijemo:

Slika kojoj nismo mijenjali dimenzije.



Slika kojoj smo mijenjali dimenzije.



Svojstva popisa (listi)

- Dolaskom CSS-a listama se značajno povećala uporaba, jer je CSS unio nove mogućnosti formatiranja i omogućio da se popisi od običnog elementa za nizanje teksta pretvore u vrlo koristan alat. Pomoću CSS možemo obične popise pretvoriti u izvrsne izbornike. Popisi se sastoje od ``, odnosno `` oznake, koje ustvari čine blok element, te od jedne ili više `` oznaka koje označavaju elemente popisa. Svaku se od ovih oznaka možemo zasebno formatirati.

list-style-type- određuje tip grafičke oznake koja će se nalaziti ispred pojedinog elementa popisa, a može imati sljedeće vrijednosti: circle, square, disc, decimal, lower-alpha, upper-alpha, lower-roman, upper-roman i none.

list-style-image- svojstvo koje nam omogućava da umjesto grafičkih oznaka ispred elemenata liste dodamo sličicu.

list-style-position- svojstvo koje definira uvlačenje elemenata liste a poprima vrijednosti inside i outside.

list-style- skraćeni način pisanja svih svojstava.


```
<html>
<head>
<style type="text/css">
ul{ list-style-type:square;
    list-style-position:inside; }
</style>
</head>
<body>
Ovo je primjer liste:
<ul>
<li>Prvi element liste.</li>
<li>Drugi element liste.</li>
<li>Treći element liste.</li>
<li>Četvrti element liste.</li>
</ul>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je primjer liste:

- Prvi element liste.
- Drugi element liste.
- Treći element liste.
- Četvrti element liste.

- Primjer kada umjesto grafičkih oznaka koristimo sliku:

```
<html>
<head>
<style type="text/css">
ul { list-style-image:url(slika.gif); }
</style>
</head>
<body>
```

Ovo je primjer liste sa slikom umjesto grafičke oznake:

```
<ul>
<li>Prvi element liste.</li>
<li>Drugi element liste.</li>
<li>Treći element liste.</li>
<li>Četvrti element liste.</li>
</ul>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je primjer liste sa slikom umjesto grafičke oznake:

- ▶ Prvi element liste.
- ▶ Drugi element liste.
- ▶ Treći element liste.
- ▶ Četvrti element liste.

Svojstva pozicioniranja elemenata na stranici

- Pozicioniranje elemenata na stranici može biti apsolutno, relativno ili fiksno. Pri korištenju sva tri načina pozicioniranja možemo koristiti sljedeća svojstva:
 - left** - udaljenost elementa od lijeve referentne točke, tj. udaljenost od nekog drugog elementa ili prozora preglednika.
 - right** - udaljenost elementa od desne referentne točke, tj. udaljenost od nekog drugog elementa ili prozora preglednika.
 - top** - udaljenost elementa od gornje referentne točke, tj. udaljenost od nekog drugog elementa ili prozora preglednika.
 - bottom** - udaljenost elementa od donje referentne točke, tj. udaljenost od nekog drugog elementa ili prozora preglednika.
 - z-index** - ukoliko se dva elementa na stranici preklapaju vrijednost z-indexa određuje koji će element biti iznad, a koji ispod. Element sa većom vrijednosti z-indexa biti će prikazan iznad elementa sa manjim z-indexom.
 - position** - određuje da li je pozicioniranje apsolutno, relativno ili fiksno. Svojstvo prima sljedeće vrijednosti: absolute, relative i fixed.
 - overflow** - određuje što se događa ako sadržaj prelazi granice elementa. visible, hidden, scroll i auto.
- Svojstva **left**, **right**, **top** i **bottom** mogu imati dužine izraženu u pikselima, postocima ili vrijednost auto kad sam preglednik određuje udaljenost.

- Primjer apsolutnog pozicioniranja:

```
<html>
<head>
<style type="text/css">
h2{ position:absolute;
    left:80px;
    top:120px; }
</style>
</head>
<body>
<h2>Ovo je naslov sa apsolutnim pozicioniranjem</h2>
<p>Pomoću apsolutnog pozicioniranja element možemo
smjestiti bilo gdje na stranici. <br>Naslov je smješten 80
piksela od lijeve strane prozora preglednika i 100 piksela od
<br>gornje strane prozora preglednika.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Pomoću apsolutnog pozicioniranja element možemo smjestiti bilo gdje na stranici. Naslov je smješten 80 piksela od lijeve strane prozora preglednika i 100 piksela od gornje strane prozora preglednika.

Ovo je naslov sa apsolutnim pozicioniranjem

- Ako nekom svojstvu damo negativnu vrijednost dužine npr. `left:-20px`, to znači da se element prikaže 20 piksela lijevo od mjesta na kojem bi se inače prikazao da mu nismo zadali vrijednost dužine.
- Pogledajmo primjer relativnog pozicioniranja:

```
<html>
<head>
<style type="text/css">
h2.lijevo{ position:relative;
           left:-20px; }
h2.desno{ position:relative;
          left:20px; }
</style>
</head>
<body>
<h2>Ovo je naslov bez pozicioniranja.</h2>
<h2 class="lijevo">Ovo je naslov pomaknut 20 piksela lijevo od prvotnog
položaja.</h2>
<h2 class="desno"> Ovo je naslov pomaknut 20 piksela desno od
prvotnog
položaja.</h2>
<p>Pomoću apsolutnog pozicioniranja element pomičemo u odnosu na
njegov originalni položaj.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovo je naslov bez pozicioniranja.

Ovo je naslov pomaknut 20 piksela lijevo od prvotnog položaja.

Ovo je naslov pomaknut 20 piksela desno od prvotnog položaja.

Pomoću apsolutnog pozicioniranja element pomičemo u odnosu na njegov originalni položaj.

- Primjer fiksnog pozicioniranja:

```
<html>
<head>
<style type="text/css">
h2.lijavo{ position:fixed;
    left:20px;
    top:30px; }
h2.desno{ position:fixed;
    right:145px;
    top:65px; }
p{ position:fixed;
    top:400px;
    font-size:16pt; }
</style>
</head>
<body>
<h2 class="lijavo">Ovo je naslov pomaknut 20 piksela od lijev strane i 30 od
gornje strane.</h2>
<h2 class="desno"> Ovo je naslov pomaknut 145 piksela od desne strane i 65
od gornje strane.</h2>
<p>Pomoću fiksnog pozicioniranja element smještamo u odnosu prozor web
preglednika. <br> To jest određujemo koliko je element udaljen od ruba
prozora preglednika.</p>
</body>
</html>
```


- Kao rezultat dobijemo:

Ovo je naslov pomaknut 20 piksela od lijev strane i 30 od gornje strane.

Ovo je naslov pomaknut 145 piksela od desne strane i 65 od gornje strane.

Pomoću fiksnog pozicioniranja element smještamo u odnosu prozor web preglednika.
To jest određujemo koliko je element udaljen od ruba prozora preglednika.

Ostala svojstva

- Od niza svojstava CSS-a, ovdje smo napravili izbor još nekolicine atraktivnih i češće korištenih:

cursor - definira vrstu pokazivača kada se on nalazi iznad elementa. Može imati sljedeće vrijednosti: auto, crosshair, default, help, move, pointer, text, wait itd.

visibility - definira vidljivost elementa, a može imati vrijednosti visible (vidljiv) i hidden (nevidljiv). Često puta elemente koristimo kao spremnike podataka kada koristimo JavaScript i ne želimo da su vidljivi na stranici pa ih onda ovim svojstvom sakrijemo.

display - svojstvo koje omogućava promjenu predefiniranog načina prikaza različitih elemenata. Može imati vrijednosti block, inline, list-item i *none*. Koristi se kad želimo blok element prikazati kao linijski i obrnuto. Primjere ćemo vidjeti u kasnijim poglavljima.

Vrste CSS elemenata

- Postoje tri vrste CSS elemenata:

Blok elementi (eng. *block level elements*)

Linijski elementi (eng. *inline elements*)

Popisi (eng. *lists*).

Blok elementi

- Blok elementi imaju predefinirano ponašanje da se u dokumentu automatski postavljaju jedan ispod drugog tj. slijedeći element započinje u novoj liniji. Tipičan primjer blok elemenata su `<p>` i `<h1>`. Ponašanje blok elemenata prikazano je idućim primjerom:

```
<html>
<head>
<style type="text/css">
p { border:2px solid red;
    width:450px;
    height:150px;
    font-size:16pt;
    padding:5px; }
</style>
</head>
<body>
```

`<p>`Ovo je prvi blok koji nam služi kao primjer ponašanja blok elemenata na HTML stranici. Blok elementi se ponašaju tako da se uvijek smještaju jedan ispod drugoga ako to mi sami ne promijenimo oblikovanjem pomoću CSS-a. `</p>`

`<p>`Drugi odlomak nam je potreban da bismo dokazali da će se on smjestiti ispod prvog odlomka, a ne odmah do njega iako je bilo mjesta da se smjesti u istu liniju kao prvi odlomak. `</p>`

```
</body></html>
```

- Kao rezultat dobijemo:

Ovo je prvi blok koji nam služi kao primjer ponašanja blok elemenata na HTML stranici. Blok elementi se ponašaju tako da se uvijek smještaju jedan ispod drugoga ako to mi sami ne promijenimo oblikovanjem pomoću CSS-a.

Drugi odlomak nam je potreban da bismo dokazali da će se on smjestiti ispod prvog odlomka, a ne odmah do njega iako je bilo mjesta da se smjesti u istu liniju kao prvi odlomak.

Linijski elementi

- Za razliku od blok elemenata, linijski elementi predefinirano se postavljaju u nizu, tj. jedan element za drugim. Tipični primjeri blok elemenata su `<a>` i ``. Ponašanje linijskih elemenata prikazano je idućim primjerom:

```
<html>
<head>
<style type="text/css">
p { border:2px solid red;
    width:500px;
    height:200px;
    font-size:16pt;
    padding:5px; }
</style>
</head>
<body>
<p>Ovaj odlomak je blok element koji u sebi sadrži linijske elemente kao što su
slika  i još jedan linijski element kao što je link
<a href="http://www.etfos.hr">ETFOS</a>. Iz ovih primjera vidimo da se
linijski elementi postavljaju jedan do drugog u istom retku i ne
prelaze u idući redak.</p>
</body>
</html>
```

- Kao rezultat dobijemo:

Ovaj odlomak je blok element koji u sebi sadrži linijske



elemente kao što su slika Sveučilište Josipa Jurja Strossmayera u Osijeku i još jedan linijski element kao što je link [ETFOS](#) Iz ovih primjera vidimo da se linijski elementi postavljaju jedan do drugog u istom retku i ne prelaze u idući redak.

Popisi

- Popisi (liste) su blok elementi, tj. imaju svojstva blok elemenata, ali uz to imaju i grafičku ili brojčanu oznaku ispred teksta. Ponašanje popisa prikazano je na idućoj slici:

Ovo je primjer liste:

- Prvi element liste.
- Drugi element liste.
- Treći element liste.
- Četvrti element liste.

Kontejneri SPAN i DIV

- **Span**
- Element `` je linijski kontejner i ponaša se isto kao svi linijski elementi. Smisao kontejnera je strukturirano obuhvaćanje i formatiranje raznih elemenata. Uloga kontejnera da unutar sebe sadržava druge elemente. Tako formatiranjem samog kontejnera formatiramo i elemente koji se nalaze unutar njega. Za razliku od oznaka koje su predefinirane i unaprijed definirane (`<p>`, `<h1>`, `<a>`), kontejneri nisu unaprijed definirani i dizajnerima je dopušteno da sami odrede svrhu kontejnera, tj. kako će biti formatirani i koji će se sadržaj nalaziti unutar njih.

- Tipičan primjer `` kontejnera je formatiranje nekolicine riječi u tekstu:

```
<html>
<head>
<style type="text/css">
p { font-size:12pt; }
```

```
.plava { font-size:16pt;
        color:blue;
        border:2px solid red; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Odlomak sa tekstom veličine 12pt, <span class="plava">unutar
kojeg se nalazi linijski kontejner sa plavim tekstom veličine 16pt.
```

```
</span>
```

```
</p>
```

```
</body>
```

```
</html>
```

Odlomak sa tekstom veličine 12pt, unutar kojeg se nalazi linijski kontejner sa plavim tekstom veličine 16pt.

- **Oznaka div**
- Element `<div>` je blok kontejner i ponaša se kao blok elementi. Najčešće služi za obuhvaćanje više blok ili linijskih elemenata unutar jedne strukture koja kao i kod spana dijeli zajedničko formatiranje. Primjer upotrebe ovog elementa su prikazi CSS i HTML koda.

Primjer izrade izbornika pomoću CSS-a

- Sada ćemo pokazati jedan vrlo jednostavan primjer CSS izbornika koji ima različite efekte, kao što su mijenjanje boja, obruba i tako dalje. Izbornik je sastavljen od nekoliko hiperveza koji se mogu nalaziti na stranici. Sastavni dio svakog izbornika je podizbornik, ovdje prikazan samo za jednu nadređenu hipervezu.
- Prvi što moramo napraviti pri izradi izbornika je popisa hiperveza koje ćemo imati u izborniku. Popisi se zbog svojih CSS svojstava redovito koriste za kreiranje izbornika.

- Popis glavnog izbornika sastoji se od četiri linka:

 Fakultet

 Znanost

 Struka

 Studiji

- Kao rezultat dobijemo:

- Fakultet
- Znanost
- Struka
- Studiji

- Nadalje svaki izbornik ima svoje linkove podizbornika dodajemo kao još jedan popis:

```
<ul>
  <li><a href="http://www.etfos.hr/?fakultet"> Fakultet </a></li>
  <ul>
    <li><a href="http://www.etfos.hr/?opci_podaci"> Opći podaci</a></li>
    <li><a href="http://www.etfos.hr/?sluzbe">Službe</a></li>
  </ul>
  <li><a href="http://www.etfos.hr/?znanost"> Znanost</a></li>
  <ul>
    <li><a href="http://www.etfos.hr/?zavodi"> Zavodi</a></li>
    <li><a href="http://www.etfos.hr/?projekti_mzos">Projekti</a></li>
  </ul>
  <li><a href="http://www.etfos.hr/?struka"> Struka</a></li>
  <li><a href="http://www.etfos.hr/?studiji"> Studiji</a></li>
  <ul>
    <li><a href="http://www.etfos.hr/?sveucilisni_studij">Sveučilišni </a></li>
    <li><a href="http://www.etfos.hr/?strucni_studij">Stručni</a></li>
  </ul>
</ul>
```

- Kao rezultat dobijemo:

- Fakultet
 - Opći podaci
 - Službe
- Znanost
 - Zavodi
 - Projekti
- Struka
- Studiji
 - Sveučilišni
 - Stručni

- Sada kad smo napravili popis linkova slijedi njegovo formatiranje. Ovim smo i završili sve izmjene HTML koda, i sve preostalo formatiranje radimo pomoću CSS-a.

- Za formatiranje popisa hiperveza koristimo jednostavne selektore (ul i li) i kontekstnim selektorima za hiperveze izbornika (ul, li i a) i kontekstne selektore za hiperveze podizbornika (ul, ul, li i a). Prvi kontekstni selektor definira samo one hiperveze koji su unutar prvog popisa `` `` elementu , a drugi samo one linkove unutra dvostrukog popisa `` `` ``.
- Pored samog razmještanja elemenata na stranici, izbor boja i dodavanje ukrasa isključivo pripada poslu web dizajnera. Ovdje se može vidjeti snaga CSS-a, jer web dizajner, bez obzira na napisan kôd, mijenjanjem CSS-a u potpunosti može promijeniti izgled popisa. Hiperveze izbornika smo podebljali i obojili plavom bojom, a podizbornika crvenom bojom, uklonili smo grafičke oznake ispred listi i podcrtanu liniju ispod linkova.

- Slijedi kôd:

```
ul, li { list-style:none;
}
ul li a, ul li a:visited { text-decoration:none;
    margin: 5px 5px 5px 10px;
    padding:5px;
    width:150px;
    display:block;
    color:blue;
    font-weight:bold;
}
ul ul li a, ul ul li a:visited { text-decoration:none;
    margin:5px 5px 5px 20px;
    padding:5px;
    width:150px;
    display:block;
    color:red;
    font-weight:bold;
}
```

- Kao rezultat dobijemo:

Fakultet

Opći podaci

Službe

Znanost

Zavodi

Projekti

Struka

Studiji

Sveučilišni

Stručni

- Već u ovome trenutku imamo gotove linkove gotovi i funkcionalni, ali im nedostaju dinamički efekti, jer su korisnici u današnje vrijeme naviknuti da im pri pomicanju pokazivača miša nekakav efekt označi iznad kojeg linka se trenutno nalazi pokazivač. Takvi efekti, osim uljepšavanja, imaju i ulogu lakše orijentacije i snalaženja među linkovima. Dinamičke efekte postićemo dodavanjem pseudoklasa u CSS stilove. Upotrijebit ćemo pseudoklasu **hover**, a mijenjat ćemo izgled linkova tako da im se pojavi obrub kada mišem pređemo iznad njih. Slijedi primjer:

```
ul, li { list-style:none; }
ul li a, ul li a:visited {
    text-decoration:none;
    margin: 5px 5px 5px 10px;
    padding:5px;
    width:150px;
    display:block;
    color:blue;
    font-weight:bold; }
ul li a:hover {
    text-decoration:none;
    margin: 5px 5px 5px 10px;
    padding:5px;
    width:150px;
    display:block;
    color:blue;
    font-weight:bold;
    border:2px solid blue; }
```

```
ul ul li a, ul ul li a:visited {
    text-decoration:none;
    margin:5px 5px 5px 20px;
    padding:5px;
    width:150px;
    display:block;
    color:red;
    font-weight:bold; }
ul ul li a:hover {
    text-decoration:none;
    margin:5px 5px 5px 20px;
    padding:5px;
    width:150px;
    display:block;
    color:red;
    font-weight:bold;
    border:2px solid red; }
```

- Kao rezultat dobijemo:

Fakultet
Opći podaci
Službe
Znanost
Zavodi
Projekti
Struka
Studiji
Sveučilišni
Stručni

- Dodavanjem klase **hover** dobili smo konačan prikaz u kojem se linkovi mijenjaju kako mišem prelazimo preko njih.

Primjer izrade horizontalnih izbornika

- U prethodnom poglavlju smo prikazali izradu CSS izbornika, a sada ćemo prikazati jednostavan način izrade navigacijskih elemenata koji bi kao jedinstvena navigacijska cjelina trebali podsjećati na kartice (eng. tab).
- Kartice su već duže vrijeme jedno od češćih rješenja internetske navigacije i možemo ih napraviti isključivo pomoću CSS-a. Valja posebno napomenuti da je ovakav pristup izabran ne samo zbog jednostavnosti već i radi potpuno jednakog prikaza u svim web preglednicima koji podržavaju CSS, što se često ne postiže kod primjene složenijih rješenja.

- Valja posebno napomenuti da je ovakav pristup izabran ne samo zbog jednostavnosti već i radi potpuno jednakog prikaza u svim web preglednicima koji podržavaju CSS, što se često ne postiže kod primjene složenijih rješenja.
- Osnovu našeg rješenja kao i u prošlom primjeru čini popis linkova:

 Fakultet

 Znanost

 Struka

 Studiji

- Kao rezultat dobijemo:

- Fakultet

- Znanost

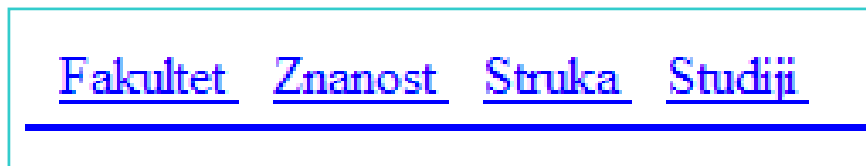
- Struka

- Studiji

- Trebamo napraviti da se linkovi prikazuju jedan do drugog, dodati pozicioniranje (smještanje), boje znakova, crtu koja odvaja kartice od sadržaja i tako dalje. Smještanje se uspostavlja marginama i odmacima, a kao crta širine 5 piksela upotrebljava se donja granica nesortiranog popisa. Slijedi primjer:

```
ul li { display:inline;
        padding:5px;}
ul {   margin: 50px 0px 0px 0px;
        padding:5px;
        width:300px;
        display:block;
        color:blue;
        border-bottom:2px solid blue;}
```

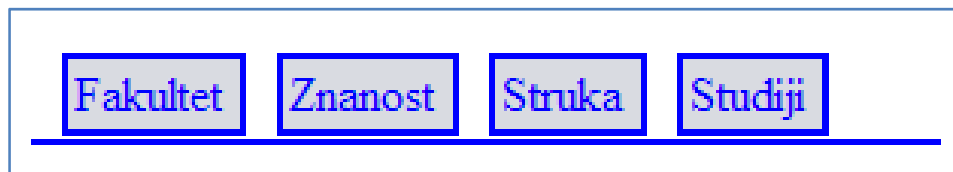
- Kao rezultat dobijemo:



- Da bismo odredili izgled za svaki pojedini karticu uvodimo novi selektor i formatiramo kartice:

```
ul li { display:inline;
padding:5px; }
ul { margin: 50px 0px 0px 0px;
padding:5px;
width:300px;
display:block;
color:blue;
border-bottom:2px solid blue; }
ul li a { background:#d9dbe1;
padding:2px;
margin:0px;
border:2px solid blue;
text-decoration:none; }
```

- Kao rezultat dobijemo:



- Nakon što smo dobili konačni izgled kartica potrebno nam je uraditi dinamičke efekte kako bi kartice mijenjale izgled kada mišem pređemo preko njih. Naime, osim same atraktivnosti kojoj se obično teži, promjena izgleda ima i funkcionalan značaj jer korisniku olakšava izbor mjesta na koje se može kliknuti. Primjenu dinamičkih elemenata na hipervezama, kao i u prethodnom primjeru, postićemo upotrebom pseudoklasa **active** i **hover** - kako bismo promijenili izgled linka bilo klikom, bilo prelaskom miša preko hiperveze. Promotrimo slijedeći primjer:

```
ul li {
    display:inline;
    padding:5px;}
ul {
    margin: 50px 0px 0px 0px;
    padding:5px;
    width:300px;
    display:block;
    color:blue;
    border-bottom:2px solid blue;}
ul li a{
    background:#d9dbe1;
    padding:2px;
    margin:0px;
    border:2px solid blue;
    text-decoration:none; }
ul li a:hover{
    background:#d9dbe1;
    border:2px solid red;
    color:red;}
ul li a:active{
    background:#d9dbe1;
    border:2px solid grey;
    padding:grey;}
```

Fakultet

Znanost

Struka

Studiji