



Osnove razvoja web i mobilnih aplikacija

Laboratorijske vježbe

LV 2 Uvod u CSS

**Fakultet elektrotehnike računarstva i
informacijskih tehnologija Osijek**

Kneza Trpimira 2b

www.ferit.unios.hr

Sadržaj:

1.	CSS - Struktura i sintaksa CSS-a	5
	Selektori i deklaracije	5
	Čitljivi CSS	6
	Prazni prostor	6
	Komentari	6
	Skraćena svojstva	6
	Kako radi CSS	7
	DOM	7
	Primjena CSS-a u DOM	8
2.	Kako primijeniti CSS na HTML	9
	Stil unutar linije (engl. <i>Inline style</i>)	9
	Unutarnja lista stilova (engl. <i>Internal stylesheet</i>)	9
	Vanjska primjena liste stilova (engl. <i>External stylesheet</i>)	10
3.	Selektori	11
	Jednostavni selektori – element selektori	11
	Jednostavni selektori – <i>class</i> selektori	12
	Jednostavni selektori – <i>id</i> selektori	13
	Univerzalni selektor	13
	Pseudo selektori	14
	Selektori atributa	16
	Kombinatori i grupni selektori	18
4.	CSS vrijednosti i jedinice	21
	Numeričke vrijednosti	21
	Bez jedinične vrijednosti	22
	Postotak	22
	Boje	23
	Ključne riječi boja	24
	Heksadekadske vrijednosti boja	24
	RGB	24
	Prozirnost	24
5.	Model kutije	25
	Svojstva kutije	25
	Širina i visina	25
	Ispunjenje	25
	Rub	26

Margine	26
Preljev.....	26
Pozadinski isječak	27
Obrub.....	27
Tipovi CSS kutija	28
Static.....	29
Relative	30
Absolute	30
Fixed	30
Sticky	30
6. Stiliziranje teksta	32
Boje.....	32
Obitelji fontova.....	32
Sigurni fontovi za web	32
Po zadanom fontovi	33
Lista fontova	33
Veličina fonta.....	33
Poravnanje teksta.....	34
Razmak između slova i riječi.....	35
Skraćena naredba	35
7. Liste	36
Stiliziranje hiperveza	36
Stanje veze.....	36
Stiliziranje veze	38
Stiliziranje liste koja je navigacija	39
8. Manipuliranje pozadinom u CSS-u	41
Boja pozadine	41
Slika pozadine.....	42
Ponavljanje pozadine	42
Pozicija pozadine	42
Pozadinska vezanost.....	43
Veličina pozadine.....	44

1. CSS - Struktura i sintaksa CSS-a

CSS (Cascading Style Sheets), odnosno kaskadni stilovi, jednostavan su mehanizam za dodavanje stilova: fontova, boja razmaka između paragrafa, uređivanje tablica. Svojom dolaskom CSS je izazvao pravu revoluciju na internetu i to zahvaljujući nizu prednosti koje ima pred tabličnim layoutom (korištenje tablica za formiranje stranice). Korištenjem CSS-a postalo je moguće odvojiti prezentaciju podataka i dizajn od same strukture podataka. HTML kôd postaje pregledniji i manji što znači da ga je puno lakše kontrolirati, a također je moguće jednostavnim primjenom parametara promijeniti izgled stranice.

Stilski obrasci sačinjeni su od stilskih pravila. Svako pravilo ima dva dijela:

- Selektor: Određuje element na koji se stilsko pravilo odnosi
- Deklaracija: Određuje kako izgleda sadržaj opisan CSS-om

Koristimo se sa setom interpunkcijskih i posebnih znakova kako bi definirali stilsko pravilo. Sintaksa za stilska pravila uvijek slijedi sljedeći uzorak:

selektor {deklaracija;}

Deklaracija se dijeli na dvije stavke:

1. Properties (aspekti kako da računalo prikaže tekst i grafiku).
2. Values (podaci koji određuju kao želimo da tekst i slike izgledaju na našoj stranici).

Properties se od vrijednosti (value) u deklaraciji odvaja dvotočkom, a svaka deklaracija završava s točkom-zarezom:

selector {property: value;}

CSS specifikacija¹ sadrži listu svih opcija s kojima možemo raditi u svojim stilskim pravilima, te vrijednostima koje pravila mogu zauzeti. Stilski obrasci nadjačavaju unutrašnja pravila prikaza u internet pregledniku. Vaše odredbe za formatiranje utječu na krajnji prikaz stranice u korisnikovom prozoru.

Korisnici mogu promijeniti svoje postavke tako da njihov Internet preglednik ne prikazuje vaš CSS. Iako većina korisnika koristi CSS, trebali bi isprobati da li vaša stranica izgleda prihvatljivo bez uključenog

Selektori i deklaracije

Za jedan selektor možemo napraviti nekoliko stilskih pravila, svaki s jednom deklaracijom:

```
h1 {color: teal;}  
h1 {font-family: Arial;}  
h1 {font-size: 36px;}
```

¹ Pogledajte specifikaciju prema W3C (<http://www.w3.org/TR/CSS2>)

No, tako velika kolekcija stilskih pravila postaje teška za upotrebu. CSS vam dozvoljava da kombinirate nekoliko deklaracija u jedno stilsko pravilo koje utječe na karakteristike prikaza pojedinog selektora:

```
h1 {  
  
color: teal;  
  
font-family: Arial;  
  
font-size: 36px;  
  
}
```

Sve deklaracije za h1 selektor su unutar istih vitičastih zagrada ({}) i odvojene su točkazarezom (;). Možemo staviti koliko god hoćemo deklaracija u stilsko pravilo, samo trebamo svaku deklaraciju završiti s točka-zarezom.

Čitljivi CSS

CSS sintaksa nije teška, čak kada se neka pravila krivo napišu, ona će se jednostavno ignorirati. Unatoč tome, postoje dobre prakse kako bi CSS kôd bio jednostavniji za korištenje i održavanje.

Prazni prostor

Prazni prostor označava razmake, tabulator i nove linije. Prazni prostor se dodaje u izvorni kôd kako bi CSS bio čitljiviji.

Kao i kod HTML, preglednik nastoji ignorirati sav prazni prostor unutar CSS kôda jer većina praznog prostora je tu kako bi se povećala čitljivost kôda. Ipak treba pripaziti na razdvajanje vrijednosti koja se dodaju određenom svojstvu, s barem jednim razmakom.

Komentari

Kao i s HTML, savjetuje se korištenje komentara unutar CSS-a u svrhu dokumentiranja kôda. Također, komentari su korisni za komentiranje određenog dijela kôda u svrhu testiranja i uklanjanja grešaka.

```
/*  
  
*/
```

Skraćena svojstva

Neka svojstva, kao što su font, background, padding, border, margin, dozvoljavaju postavljanje više vrijednosti unutar iste linije, čime se štedi vrijeme i sam kôd postaje uredniji.

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

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

Kako radi CSS

Kada preglednik prikazuje dokument, mora prvo kombinirati sadržaj dokumenta s informacijama stila. Dokument se obrađuje u dva koraka:

1. Preglednik pretvara HTML i CSS u DOM (engl. *Document Object Model*). DOM predstavlja dokument u memoriji računala. Kombinira sadržaj dokumenta sa stilovima.
2. Preglednik prikazuje sadržaj DOM-a

DOM

Ima strukturu stabla, gdje svaki element, atribut, dio teksta postaju DOM čvorovi. Čvorovi su definirani po svojim odnosima s drugim DOM čvorovima. Neki elementi su roditelji čvorova koji su djeca, a djeca čvorovi imaju podčvorove. Razumijevanje DOM-a pomaže prilikom dizajniranja i održavanja CSS, jer je DOM mjesto gdje se susreću CSS i sadržaj dokumenta.

Primjer:

```
<p>  
  Koristit ćemo:  
  <span>Kaskadne</span>  
  <span>Liste</span>  
  <span>Stilova</span>  
</p>
```

U DOM-u čvor koji odgovara <p> elementu predstavlja roditelj čvor. Njegova djeca su tekstovi i elementi koji predstavljaju čvorove. elementi su čvorovi koji su ujedno i roditelji s tekстом koji je čvor i koji predstavlja njihovo dijete.

```
p
|-Koristit ćemo:
|-SPAN
|   |-Kaskadne
|-SPAN
|   |-Liste
|-SPAN
|       |-Stilova
```

Ovo je način na koji preglednik interpretira HTML dijelove – prevodi prethodno prikazano DOM stablo i prikazuje rezultat unutar preglednika.

Koristit ćemo: Kaskadne Liste Stilova

Primjena CSS-a u DOM

Koristeći prethodni primjer, na isječak kôda primijenit će se određeni CSS stil.

```
span {
  border: 1px solid black;
  background-color: lime;
}
```

Primjenom navedenog stila, preglednik će analizirati HTML i kreirati iz tog HTML-a DOM, a nakon toga će preglednik analizirati CSS. Kako je jedino pravilo unutar CSS span selektor, primijenit će se pravilo na svakom od tri navedena span elementa.

2. Kako primijeniti CSS na HTML

Postoje tri različita načina kako primijeniti CSS na HTML dokument.

Stil unutar linije (engl. *Inline style*)

Stil unutar linije predstavlja CSS deklaraciju koja ima utjecaj samo na jedan element, a nalazi se unutar style atributa.

```
<p>
  Koristit ćemo:
  <span style="border: 1px solid black; background-color:
lime;">Kaskadne</span>
  <span>Liste</span>
  <span>Stilova</span>
</p>
```

Savjetuje se izbjegavanje ovog načina primjene stila, osim ako postoji iznimno opravdani razlog. Ako se primjenjuje ovakav način primjene stilova samo na individualnim elementima, održavanje postaje iznimno teško, jer se iste informacije moraju mijenjati na nebrojnim mjestima unutar istog dokumenta. Ovo također miješa prezentacijske informacije CSS-a s strukturnim informacijama HTML-a, što čini CSS teže čitljivim i težim za razumijevanje. Jedini opravdani razlog korištenja stilova na ovakav način, bio bi prilikom rada u razvojnom okruženju koji ima velike restrikcije, npr. rad s CMS-om (engl. *Content Management System*) koji dozvoljava izmjene samo unutar <body> elementa.

Unutarnja lista stilova (engl. *Internal stylesheet*)

Unutarnja lista stilova primjenjuje se unutar HTML dokumenta tako da se primjeni <style> element unutar <head> elementa HTML dokumenta.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Primjena CSS stilova</title>
  <style>
    span {
      border: 1px solid black;
      background-color: lime;
    }
  </style>
</head>
```

Ovakva primjena stilova može biti praktična u određenim situacijama, npr. rad s CMS-a koji dozvoljavaju direktne izmjene nad CSS-om. U ovom slučaju CSS bi se morao ponavljati na svakoj stranici, kao i primjena ažuriranja.

Vanjska primjena liste stilova (engl. *External stylesheet*)

Vanjska primjena stilova predstavlja primjenu CSS koji se nalazi u odvojenoj datoteci s ekstenzijom .css, a poveznica na tu datoteku se radi iz HTML dokumenta pomoću elementa <link> koji se postavlja u <head> element HTML dokumenta.

HTML dokument

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Primjena CSS stilova</title>
  <link rel="stylesheet" href="style.css">
</head>
```

CSS dokument

```
span {
  border: 1px solid black;
  background-color: lime;
}
```

Ova metoda je najbolja jer se s jednom listom stilova može stilizirati više HTML dokumenata i ažuriranjem jedne liste stilova bi ažuriralo stilove u svim HTML dokumentima. Razdvajanjem različitih dijelova kôda čini jednostavnijim održavanje kôda.

3. Selektori

U CSS-u, selektori se koriste za ciljanje određenog HTML elementa na kojemu se želi primijeniti određeni stil. Postoji široka varijacija CSS selektora, čime se omogućava velika preciznost prilikom odabira elemenata za stiliziranje.

Selektori se mogu podijeliti u sljedeće kategorije:

- Jednostavni selektori: odabiru jedan ili više elemenata ovisno o tipu elementa ili o elementu koji u svojoj otvorenoj oznaci imaju ugrađen atribut *class* ili *id*.
- Selektor atributa: odabire jedan ili više elemenata ovisno o njihovim atributima ili vrijednosti atributa.
- Pseudo klase: odabire jedan ili više elemenata koji se nalaze u određenom stanju, npr. pokazivačem miša se prelazi preko elementa, checkbox koji je prazan ili odabran, dijete element koji je prvi element svoga roditeljskog elementa unutar DOM stabla.
- Pseudo elementi: odabire jedan ili više dijelova sadržaja koji se nalaze na određenoj poziciji u odnosu na element, npr. prva riječ svakog paragrafa.
- Kombinatori: to nisu selektori, već način kombiniranja dva ili više selektora za precizni odabir, npr. odabir samo paragrafa koji su djeca unutar roditeljskog elementa <div>, odabir paragrafa koji dolazi odmah nakon naslova.
- Grupni selektori: ovo nisu razdvojeni selektori, već ideja je da se stavi više selektora odvojeni zarezom za isto CSS pravilo, kako bi se primijenilo jedno pravilo na sve elemente koji su odabrani navedenim selektorima.

Jednostavni selektori – element selektori

Ovaj selektor dohvaća HTML element na temelju imena elementa. Ovo je najjednostavniji način odabira svih elementa istog tipa.

```
<p>
  Koristit ćemo:
  <spanKaskadne</span>
  <span>Liste</span>
  <span>Stilova</span>
</p>
```

```
p {
  border: 1px solid black;
  background-color: yellow;
}

span {
  border: 1px solid black;
  background-color: lime;
}
```

Jednostavni selektori – *class* selektori

Class selektor sastoji se od znaka (.) iza koje slijedi ime klase. Ime klase je bilo koja vrijednost, bez razmaka, koja se nalazi unutar HTML *class* atributa. Atribut *class* se može staviti u bilo koji element unutar *<body>* elementa HTML dokumenta, a ime (vrijednost) koja se predaje *class* atributu treba bit navedena od strane programera. Također treba obratiti pozornost kako više elemenata unutar HTML dokumenta može imati isto ime za *class* atribut, ali također kako i jedan element može imati više imena za *class* atribut razdvojeni razmakom.

```
<ul>
  <li class="jedan drugi-naziv">Jedan</li>
  <li class="dva treci-naziv">Dva</li>
  <li class="tri drugi-naziv">Tri</li>
  <li class="cetiri treci-naziv">Četiri</li>
</ul>
```

```
.jedan {
  text-transform: uppercase;
}
.tri {
  text-transform: lowercase;
}
.drugi-naziv {
  text-decoration: underline;
}
.treci-naziv {
  color: cyan;
}
```

-
- JEDAN
 - Dva
 - tri
 - Četiri

Jednostavni selektori – *id* selektori

ID selektor se sastoji od znaka (#) iza kojega odmah slijedi ID ime danog elementa iz HTML-a. Svaki HTML element mora imati jedinstveno ID ime, postavljeno s *id* atributom. Ime atributa treba biti navedeno od strane programera. Ovo je najefikasniji način kako odabrati jedan element. Također, treba obratiti pozornost kako ID treba biti jedinstven unutar dokumenta. Ako se unutar HTML dokumenta pojavljuje više elemenata s istim ID imenom, može doći do nepredvidljivo ponašanje.

```
<p id="pristojno">"Dobro jutro."</p>
<p id="ruzno">"Odlazi!"</p>
```

```
#pristojno {
    font-family: cursive;
}

#ruzno {
    font-family: monospace;
    text-transform: uppercase;
}
```

Univerzalni selektor

Univerzalni selektor (*) dozvoljava odabir svih elemenata na stranici. Rijetko se koristi za primjenu stila za svaki element na stranici, često se koristi u kombinaciji s drugim selektorima.

```
<div>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit.<br>
    Ullam aperiam deserunt delectus, <strong>nemo odio
nobis</strong>.<br>
    Est deserunt architecto <em>odit</em>.<br>
    Natus consequuntur laborum excepturi libero.<br>
    Perspiciatis, <strong>repellat</strong>? Ex, recusandae
reprehenderit.<br>
    Numquam.
  </p>
</div>
```

```
* {  
  border: 1px solid black;  
  background: rgba(255, 0, 0, 0.25);  
}
```

Pseudo selektori

Pseudo selektori ne odabiru elemente, već određene dijelove elemenata ili elemente samo u određenom kontekstu.

Pseudo klase

CSS ključna riječ pseudo-class dodaje se na kraju selektora nakon znaka (:) koji se koristi za određivanje elementa kojega se želi stilizirati samo kada se nalazi u određenom stanju. Npr. ako se želi stilizirati element veze samo kada se dođe iznad veze s pokazivačem miša ili prvi element dijete unutar roditeljskog elementa u DOM stablu... . Neki selektori pseudo klase:

- a:link,
- a:visited,
- a:hover,
- a:active,
- p:nth-of-type(1), :nth-of-type(even), :nth-of-type(odd) (značenje nth-of-type(n) selektora odgovara svakom elementu koji je n-to dijete roditelja, određenog tipa)
- th:nth-child(1), :nth-child(even), :nth-child(odd),
- li:last-child;

```
<ul>  
  <li><a href="#">United Kingdom</a></li>  
  <li><a href="#">Germany</a></li>  
  <li><a href="#">Finland</a></li>  
  <li><a href="#">Russia</a></li>  
  <li><a href="#">Spain</a></li>  
  <li><a href="#">Poland</a></li>  
</ul>
```

```

a {
    text-decoration: none;
    color: black;
}

a:hover {
    text-decoration: underline;
    color: red;
}

li:nth-of-type(even) {
    background-color: #00ff00;
}

li:nth-of-type(odd) {
    background-color: #0000ff;
}

```

Pseudo element

Pseudo element je vrlo sličan pseudo klasi. To su ključne riječi koje se navode nakon znakova (::) i dodaju se na kraju selektora kako bi se odabrao određeni element. Neki elektori pseudo elementa:

- p::first-letter,
- p::first-line.

```

<p>
Lorem ipsum, dolor sit amet consectetur
adipisicing elit. <br>Sit sunt perspiciatis
dolorum perferendis veniam id, ad a voluptatum
vitae, qui nostrum debitis ratione aspernatur
voluptate! Itaque ut cum aut officiis.
</p>

```

```

p::first-line {
    font-weight: bold;
}

p::first-letter {
    font-size: 3em;
    font-weight: bold;
}

```

Selektori atributa

Selektor atributa su posebna vrsta selektora kojima se odabire element na temelju njegova atributa i vrijednosti atributa. Opća sintaksa se sastoji od uglatih zagrada ([]), unutar kojih se nalazi ime atributa iza kojeg može slijediti uvjet kojim se provjerava vrijednost atributa.

Selektor atributa prisutnosti i vrijednosti

Ovi selektori atributa usklađuju se s točnom vrijednosti atributa:

- [attr]: ovim selektorom odabiru se svi elementi s atributom (attr), nevezano koju vrijednost ima.
- [attr=value]: ovim selektorom odabiru se svi elementi s atributom (attr), ali samo ako mu je vrijednost (value).
- [attr~=value]: ovim selektorom odabiru se svi elementi s atributom (attr), ali samo ako je (value) vrijednost (attr), odnosno, jedna od riječi liste razdvojene razmakom.

```
<a href="http://www.ferit.unios.hr" title="Na izvoru znanja elektrotehnike, računarstva i informacijskih tehnologija">FERIT</a>
<a href="http://www.wikipedia.org" target="_blank">Wikipedia</a>
<a href="http://www.bug.hr" target="_self">Bug</a>
```

Primjer 1: [attr]

```
[target]{
    background-color: lime;
}
```

Primjer 2: [attr=value]

```
[target=_blank]{
    background-color: aqua;
}
```

Primjer 3: [attr~=value]

```
[title~=izvoru]{
    border: 5px solid yellow;
}
```


Atribut selektor podstring vrijednosti

Ovi selektori atributa su poznati kao "selektori slični regularnim izrazima" jer omogućavaju fleksibilan odabir kao regularni izrazi:

- `[attr^=value]`: ovaj selektor će odabrati sve elemente s atributom (`attr`) kojemu vrijednost započinje s (`value`).
- `[attr$=value]`: ovaj selektor će odabrati sve elemente s atributom (`attr`) kojemu vrijednost završava s (`value`).
- `[attr*=value]`: ovaj selektor će odabrati sve elemente s atributom (`attr`) kojemu vrijednost sadrži podstring (`value`). Podstring predstavlja dio stringa, npr. *cat* koji je dio *caterpillar*.
- `[attr|=value]`: ovaj selektor će odabrati sve elemente s atributom (`attr`) kojemu je vrijednost točno (`value`) ili započinje s (`value-`). Crtica nije slučajno navedena, već se može koristiti prilikom dohvaćanje elementa koji ima atribut (`lang="en"`, `lang="en-US"`).

Primjer 1: `[attr^=value]`

```
<h1 class="top-header">Dobro došli</h1>
<p class="top-text">Vrijedne kolege</p>
<p class="topcontent-text">Danas učimo CSS</p>
```

```
[class^="top"] {
    background: yellow;
}
```

Primjer 2: `[attr$=value]`

```
[class$="text"] {
    background: lime;
}
```

Primjer 3: [attr*=value]

```
[class*="ead"] {  
    background: aqua;  
}
```

Primjer 4: [attr|=value]

```
[class|=top] {  
    background: gold;  
}
```

Kombinatori i grupni selektori

Korištenjem jedno selektora po elementu je korisno, ali može biti neefikasno u pojedinim situacijama. CSS selektori postaju korisniji kada se počnu kombinirati za dobivanje preciznijeg odabira elemenata. CSS ima način odabiranja elemenata na temelju srodnosti elemenata. Ta srodstva se izražavaju kombinatorom:

- Grupa selektora (A, B): bilo koji element koji odgovara A i/ili B, grupni selektor ne predstavlja kombinator.
- Kombinator potomka (A B): bilo koji element koji odgovara B, tj. koji je potomak elementa A.
- Kombinator djeteta (A > B): bilo koji element koji odgovara B i koji je direktni element djeteta elementa A.
- Kombinator srodnih susjeda (A + B): bilo koji element koji odgovara B i koji je sljedeći srodni susjed elementa koji odgovara A. A i B moraju imati istog roditelja, B se odabire, ako slijedi odmah nakon A.
- Opći srodni kombinator (A ~ B): bilo koji element koji odgovara B i koji slijede nakon srodnog elementa koji odgovara A. A i B moraju imati istog roditelja i odabiru se svi B elementi nakon elementa A.

Primjer 1: A, B:

```
<div>
  <p>Paragraf 1 unutar diva.</p>
  <p>Paragraf 2 unutar diva.</p>
  <span>
    <p>Paragraf 3 unutar diva.</p>
  </span>
</div>

<p>Paragraf 4 koji nije unutar diva.</p>
<p>Paragraf 5 koji nije unutar diva.</p>
```

```
div, p {
  background-color: yellow;
}
```

Primjer 2: A B

```
div p {
  background-color: yellow;
}
```

Primjer 3: A > B

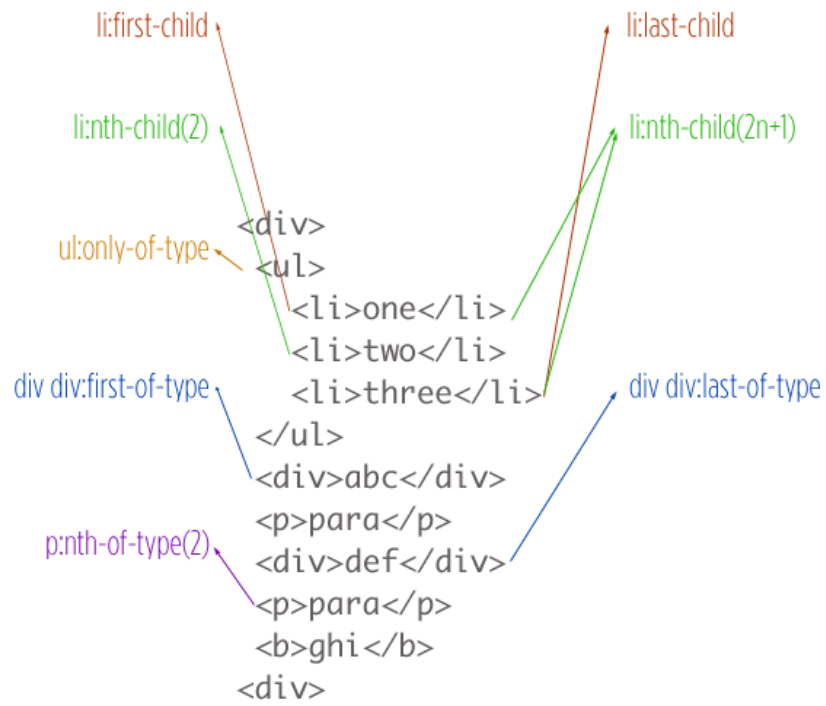
```
div > p {
  background-color: orange;
}
```

Primjer 4: A + B

```
div + p {
  background-color: orange;
}
```

Primjer 5: A ~ B

```
div ~ p {  
  background-color: orange;  
}
```



4. CSS vrijednosti i jedinice

Postoje mnogi vrijednosni tipova u CSS-u, neki se češće koriste nego drugi, spomenuti će se sljedeći:

- Numeričke vrijednosti: vrijednosti za duljinu, npr. širina elementa, debljina rubova, veličina fonta, cjelobrojne vrijednosti bez jedinice za definiranje relativne duljine linije... .
- Postoci: mogu se koristiti za definiranje veličine ili širine – relativno prema širini ili visini roditeljskog kontejnera
- Boje: za definiranje pozadinske boje, boje teksta... .

Numeričke vrijednosti

Primjer korištenja jedinice za duljinu i veličinu

```
<p>Ovo je paragraf</p>
<p>Ovo je paragraf</p>
<p>Ovo je paragraf</p>
```

```
p {
  border: 2px solid black;
  background-color: cyan;
}

p:nth-child(1) {
  width: 150px;
  font-size: 18px;
}

p:nth-child(2) {
  width: 250px;
  font-size: 24px;
}

p:nth-child(3) {
  width: 350px;
  font-size: 30px;
}
```

Pixel (px) odnosi se na apsolutnu jedinicu jer će uvijek biti iste veličine nevezano o bilo kojoj srodnoj postavci.

Postoje i relativne jedinice, koje su relativne prema trenutnom elementu kao što je *font-size* ili viewport veličina:

- `em`, `1em`: odgovara *font-size* svojstvu s vrijednosti roditeljskog elementa (širina velikog slova M). Po zadanom, osnovna veličina fonta, *font-size* koji je zadan web stranici od strane web preglednika prije CSS stiliziranja iznosi 16px, što znači da je izračunata veličina za 1em iznosi 16px za element. Potrebno je obratiti pozornost na sljedeće, veličina fonta za element se nasljeđuje od njegovih roditelja, stoga ako je zadane drugačija veličina fonta za roditeljski element, *px* ekvivalent *em* jedinice može postati kompliciran. *em* jedinice su najčešće korištene relativne jedinice na web-u.
- `rem`: (root *em*) radi na isti način kao i *em*, samo što će uvijek biti jednaka veličini zadanog osnovnog *font-size* svojstva.

Viewport – je korisnički vidljivi prostor web stranice, on varira o veličini uređaja i bit će manji na mobitelima naspram ekranima računala.

Bez jedinične vrijednosti

Ponekad se može susresti numeričke vrijednosti bez jedinice i ne predstavljaju uvijek pogrešku u sintaksi, u pojedinim slučajevima je dopuštena. Ako se želi ukloniti margina i punjenje, mogu se koristiti bez jediničnu numeričku vrijednost nula.

Drugi primjer je visina linije (svojstvo *line-height*), kojim se postavlja visina svake linije teksta unutar elementa. Mogu se koristiti jedinične vrijednosti za postavljanje visine linije, ali jednostavnije je korištenje numeričke bez jedinične vrijednosti, koje se ponašaju kao faktor umnoška.

```
p {
    line-height: 1.5;
}
```

Objašnjenje: *font-size* je 16px, *line-height* je 1.5 i visina linije će iznositi 24px.

Postotak

Za većinu svojstava kod kojih se koristi numerička vrijednost može se koristiti i vrijednost izražena u postocima. Npr. kreiranje kutija čija će se širina uvijek mijenjati za određeni postotak u odnosu na širinu njihova roditeljskog kontejnera. Usporedbe radi, kutije koje imaju fiksno postavljenu širinu (u *px* ili *em*) uvijek će ostati iste duljine, čak ako se promijeni širina njihovog roditeljskog kontejnera.

```
<div>
  <div class="kutije">Fiksna širina rasporeda u pikselima</div>
  <div class="kutije">Nefiksirani raspored u postotku</div>
</div>
```

Zanimljivo je za spomenuti kako veličina fonta (*font-size*) je postavljena na 200% i ovakva promjena veličine je relativna prema roditeljskoj veličini fonta, kao što je slučaj s *em* jedinicom. Po zadanom, veličina fonta za stranicu je 16px, te je veličina fonta za vrijednost od 200% iznosi 32px. Primjena

```
div .kutije {  
  font-size: 200%;  
  color: white;  
  height: 150px;  
  border: 2px solid black;  
}  
  
.kutije:nth-child(1) {  
  background-color: red;  
  width: 650px;  
}  
  
.kutije:nth-child(2) {  
  background-color: blue;  
  width: 75%;  
}
```

postotka za veličinu fonta radi vrlo slično kao korištenje *em* jedinice, stoga za iznos 200% ekvivalentna vrijednost u *em* jedinici iznosi 2em.

U primjeru s paragrafima koji se nalaze unutar *div* elementa, prikazana su dva različita rasporeda kutija. Jedan se zove tekući raspored (engl. *Liquid layout*), a drugi se zove fiksna širina rasporeda (engl. *Fixed width layout*) i oba imaju svoju primjenu:

- Tekući raspored se može koristiti kako bi standardni dokument uvijek odgovarao ekranu i kako bi izgledao prihvatljivo na širokoj ponudi mobilnih ekrana različite veličine.
- Fiksna širina rasporeda može se koristiti kako bi npr. karta zadržala istu veličinu. Viewport preglednika bi se mogao pomicati po karti, pregledavajući samo određeni dio karte u bilo kojem trenutku. Količina vidljivih informacija u jednom trenutku ovisi o veličini viewpota.

Boje

Postoji više načina kako odabrati boju u CSS-u, neki su načini implementirani u novije vrijeme. Ista vrijednost boje može se koristiti bilo gdje u CSS-u, bilo da se odabire boja teksta, pozadine ili nečeg drugoga.

Standardni prikaz boja na modernim računalima iznosi 24bit-a, što omogućuje prikaz oko 16.7 milijuna različitih boja, kombinacijom različitih crvenih, zelenih i plavih kanala s 256 različitih vrijednosti po kanalu ($256 \times 256 \times 256 = 16.777.216$).

Ključne riječi boja

Najjednostavniji i najstariji tipovi boja u CSS su ključne riječi boja. To su stringovi koji predstavljaju određenu vrijednost boje, npr. crvena - red, zelena - green, plava - blue... .

Ovo je jednostavan način pamćenja boja, no ograničeno je na oko 165 različitih ključnih riječi boja za primjenu u moderni preglednicima.

Heksadekadske vrijednosti boja

Vrlo česti način prikaza boje je heksadekadska boja ili hex kôd. Svaka hex vrijednost se sastoji od znaka ljestve (#) iza kojeg slijedi šest heksadekaskih brojeva, gdje svaki može primiti vrijednost od 0-f(15). Svaki par vrijednosti predstavlja jedan kanal – crveni, zeleni i plavi i dozvoljava odabiranje bilo koje od 256 mogućih vrijednosti ($16 \times 16 = 256$). Npr. crvena boja - #ff0000, zelena boja - #00ff00, plava boja - #0000ff.

Ove vrijednosti su malo kompliciranije, ali su puno više svestranije za odabir boja i može se odabrati bilo koja boja.

RGB

RGB vrijednost je funkcija – rgb(), kojoj se predaju tri argumenta koji predstavljaju crveni, zeleni i plavi kanal za boje, slično kao hex kôd. Svaki kanal u RGB je predstavljen dekadskim brojem od 0 – 255. Npr. crvena boja – rgb(255, 0, 0), zelena boja – rgb(0, 255, 0), plava boja – rgb(0, 0, 255).

RGB sustav boja je dobro podržan kao i hex vrijednosti i RGB vrijednosti su nešto više intuitivnije i lakše za razumjeti naspram hex vrijednosti.

Prozirnost

Moguće je ne samo postaviti vrijednost boje, već i prozirnost na boju. Za to se koristi CSS funkcija rgba() koja prima četvrti argument za prozirnost unutar intervala [0.0 – 1.0]. Taj se kanal još naziva alfa kanal (engl. *Alpha channel*), gdje nula predstavlja potpunu prozirnost, a jedan predstavlja potpunu neprozirnost. Npr. crvena boja s 50% prozirnosti – rgba(255, 0, 0, 0.5).

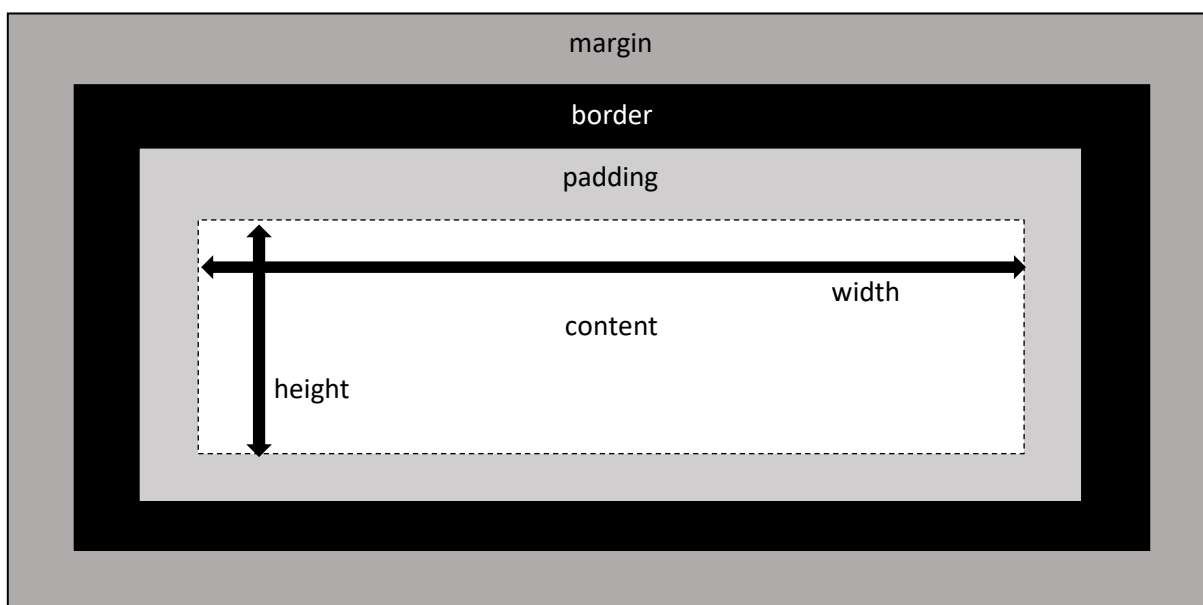
Prozirne boje su jako korisne za primjenu bogatih vizualnih efekata, za miješanje pozadine, polu prozirne UI elemente... .

5. Model kutije

CSS model kutije predstavlja temelj rasporeda elemenata na stranici. Svaki element je predstavljen kutijom u obliku pravokutnika zajedno sa sadržajem kutije, ispunjenjem, rubom i marginom koji se koji se nadovezuju jedan na drugi kao slojevi luka. Kada preglednik treba prikazati sadržaj stranice, primjenjuje stilove na sadržaj svake kutije, provjerava kolika je veličina, ispunjenja, rubova i margina, te gdje su pozicionirane pojedine kutije u odnosu na druge.

Svojstva kutije

Svaki element unutar dokumenta je strukturiran kao kutija u obliku pravokutnika. Veličina ispunjenja, ruba i margine mogu se podesiti pomoću određenih CSS svojstava.



Širina i visina

Svojstva *width* i *height* postavljaju širinu i visinu kutije sa sadržajem. Kutija sa sadržajem je prostor unutar kojeg se prikazuje sadržaj te kutije. Taj sadržaj uključuje tekst unutar te kutije kao i druge kutije koje predstavljaju ugniježdenu djecu elemente.

Postoje i druga svojstva koja omogućavaju suptilniji način upravljanja veličinom kutije sa sadržajem, kao npr. postavljanjem ograničenja na veličinu naspram apsolutne veličine. To se omogućava svojstvima kao što su: *min-width*, *max-width*, *min-height*, *max-height*.

Ispunjenje

Svojstvo *padding* odnosi se na unutarnju marginu CSS kutije koja se nalazi između vanjskog ruba kutije sa sadržajem i unutarnje strane ruba. Veličina ovog sloja se može postaviti na sve četiri strane od jednom sa svojstvom *padding* ili sa svojstvima koji se odnose na svaku stranu zasebno kao što su: *padding-top*, *padding-right*, *padding-bottom*, *padding-left*.

Rub

Rub CSS kutije nalazi se između vanjskog ruba ispunjenja i unutarnjeg ruba margine. Po zadanom, veličina ruba je postavljena na nulu, što ga čini nevidljivim, ali pomoću CSS-a moguće mu je promijeniti debljinu, stil, boju. Svojstvo *border* omogućava postavljanje svih prethodno navedenih svojstava od jednom, npr. *border: 1px solid black*. Ovakva naredba se može podijeliti na više svojstava kojima se omogućava preciznije stiliziranje:

- Postavljanje debljine, stila i boje jedne strane ruba: *border-top*, *border-right*, *border-bottom*, *border-left*.
- Postavljanje individualno debljine, stila i boje: *border-width*, *border-style*, *border-color*.
- Može se postaviti individualno jedan od tri svojstva na jednoj strani: *border-top-width*, *border-top-style*, *border-top-color*... .
- Zaokruženi rubovi mogu se postaviti pomoću svojstva *border-radius*.

Rubovi ignoriraju širinu postavljenu u postotcima.

Margine

Margine okružuju CSS kutiju i odguruje ostale CSS kutije unutar rasporeda. Pomoću svojstva *margin* mogu se postaviti sve četiri margine ili individualno pomoću: *margin-top*, *margin-right*, *margin-bottom*, *margin-left*.

Margine imaju posebno ponašanje koje se naziva urušavanje margine (engl. *Margin collapsing*). Kada se dodiruju dvije kutije, udaljenost između njih je vrijednost najveće margine, a ne njihova suma.

Margina se ne broji kao dio područja elementa, već kao područje izvan elementa.

Ako se margini postavi vrijednost *auto*, tada se margina centrira u sredini svojeg roditeljskog elementa (kontejnera). Elementu je potrebno definirati određenu širinu, a preostali prostor će se podjednako raspodijeliti s lijeve i desne strane.

Preljev

Kada se postavi veličina kutije (širina i visina) pomoću apsolutnih vrijednosti, moguće je kako sadržaj neće stati unutar definirane veličine i u tome slučaju sadržaj se može preliti izvan kutije. Kako bi se kontrolira takav događaj potrebno je koristiti svojstvo *overflow* koje može primiti nekoliko vrijednosti:

- *auto*: ako postoji previše sadržaja, sadržaj koji se preljeva je sakriven i prikazane su trake za pomicanje kako bi korisnik mogao vidjeti sav sadržaj.
- *hidden*: ako postoji previše sadržaja, sadržaj koji se preljeva je sakriven.
- *visible*: ako postoji previše sadržaja, sadržaj koji se preljeva se prikazuje izvan kutije (ovo ponašanje je postavljeno po zadanom).

Pozadinski isječak

Pozadina kutije je sačinjena od boje i slike, postavljene jedna na drugu pomoću svojstva *background-color* i *background-image*. Primijenjeni su na kutiju i nacrtani ispod kutije. Po zadanom, pozadine se protežu do vanjskog dijela ruba kutije. U većini slučajeva ovo je uredu, ali ako se ima pozadina koja se ponavlja i za koju se želi da se proteže do ruba sadržaja, potrebno je primijeniti određeno svojstvo, kao što je *background-clip* svojstvo kutije.

```
<div class="default"></div>
<div class="padding-box"></div>
<div class="content-box"></div>
```

```
div {
  width: 60px;
  height: 60px;
  border: 20px solid rgba(0, 0, 0, 0.5);
  padding: 20px;
  margin: 20px 0;
  background-size: 130px;
  background-position: center;
  background-image: url("../images/ferit-logo-
223x300.png");
  background-color: gold;
}

.default {
  background-clip: border-box;
}

.padding-box {
  background-clip: padding-box;
}

.content-box {
  background-clip: content-box;
}
```

Obrub

Obrub kutije izgleda kao rub, ali nije dio modela kutije. Ponaša se kao rub, ali je nacrtan na vrhu kutije bez da mijenja veličinu kutije, preciznije, obrub je nacrtan izvan ruba kutije, a unutar područja margine.

Potrebno je pripaziti na svojstvo *outline*, koje se koristi u nekim slučajevima pristupne pomoći kako bi se naglasili aktivni dijelovi web stranice, kao što su hiperveze kada ih korisnik odabere.

Tipovi CSS kutija

Sve što se dosada spomenulo primjenjuje se na kutije koje predstavljaju blok elemente. Ipak CSS ima i druge tipove kutija koje se ponašaju drugačije. Tip kutije koji se primjenjuje na neki element definira se svojstvom *display*. Svojstvom *display* je najvažnije svojstvo u CSS za kontrolu rasporeda. Postoji više vrijednosti koje svojstvo *display* može primiti, no spomenut će se tri najčešća:

- block kutija: definirana je kao kutija koja se slaže na drugu kutiju, sadržaj prije i poslije kutije se nalazi u odvojenim linijama, te se može primijeniti svojstvo širine i visine.
- inline kutija: je suprotno od block kutije, ona ploviti s tekстом dokumenta, npr. pojaviti će se u istoj liniji s okružujućim tekstom i s ostalim linijskim elementima, a sadržaj će joj se slomiti kako plovi tekst, kao tekst paragrafa. Postavke svojstava za širinu i visinu nemaju utjecaj na inline kutiju, a postavljanje ispunjenja, margine i ruba ažurirat će poziciju okružujućeg teksta, ali neće utjecati na poziciju okružujućih blok kutija.
- inline-block kutija: je nešto između block i inline kutije. Plovi s okružujućim tekstom i ostalim linijskim elementima bez stvaranja prekida linije prije i poslije kao block kutija, ali joj je moguće promijeniti veličinu koristeći svojstva širine i visine i zadržava svoj blok integritet kao blok kutija. Neće biti slomljene preko paragrafa kao inline kutija

Po zadanom blok elementi imaju primijenjeno svojstvo *display: block;* , a linijski elementi imaju promijenjeno svojstvo *display: inline;* .

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="inline">Mauris tempus turpis id ante mollis
dignissim.</span>
  Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
</p>

<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="block">Mauris tempus turpis id ante mollis
dignissim.</span>
  Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
</p>

<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="inline-block">Mauris tempus turpis id ante mollis
dignissim.</span>
  Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
</p>
```

```
p {
  padding: 1em;
  border: 1px solid black;
}

span {
  border: 1px solid green;
  background-color: yellow;
}

.inline {
  display: inline;
}

.block {
  display: block;
}

.inline-block {
  display: inline-block;
}
```

No, osim gore navedenih opcija, postoje i drugi načini za pozicioniranje kutija. Navedeni su svi na slici niže.

```
.el {
  position: static;
  position: relative;
  position: absolute;
  position: fixed;
  position: sticky;
  position: inherit;
}
```

Static

Ovo je zadana vrijednost za svaki pojedinačni element stranice. Različiti elementi nemaju različite zadane vrijednosti za pozicioniranje, svi počinju kao statični. Jedini razlog zbog kojeg biste ikada

postavili element *position: static*; je nasilno ukloniti neko pozicioniranje koje je primijenjeno na element izvan vaše kontrole. Ovo je prilično rijetko jer pozicioniranje nije kaskadno.

Relative

Ova vrsta pozicioniranja je vjerojatno najviše zbunjujuća i zloupotrijebljena. Ono što zapravo znači je "u odnosu na sebe". Ako postavite *position: relative*; na elementu, ali bez drugih atributa pozicioniranja (gore, lijevo, dolje ili desno), to uopće neće utjecati na njegovo pozicioniranje, bit će točno ono što bi se dogodilo da ste ga ostavili kao statičan; No, ako mu dodijelite neki drugi atribut pozicioniranja, recimo, *top: 10px*;, on će pomaknuti svoju poziciju 10 piksela prema dolje u odnosu na to gdje bi inače bio.

Postoje dvije druge stvari koje se događaju kada postavite element na poziciju koja je relativna:

- Jedan je da uvodi mogućnost korištenja z-indexa na tom elementu, što nije moguće sa statički pozicioniranim elementima. Čak i ako ne postavite vrijednost z-indexa, ovaj će se element sada pojaviti na vrhu bilo kojeg drugog statički pozicioniranog elementa.
- Druga stvar koja se događa je da ograničava opseg apsolutno pozicioniranih podređenih elemenata. Svaki element koji je dijete relativno pozicioniranog elementa može biti apsolutno pozicioniran unutar tog bloka. Ovo donosi neke snažne prilike o kojima ovdje govorim.

Absolute

Ovo je vrlo moćna vrsta pozicioniranja koja vam omogućuje da doslovno postavite bilo koji element stranice točno tamo gdje želite. Za postavljanje lokacije koristite attribute pozicioniranja gore, lijevo, dolje i desno. Zapamtite da će te vrijednosti biti relativne u odnosu na sljedeći nadređeni element s relativnim (ili apsolutnim) pozicioniranjem. Ako ne postoji takav nadređeni element, prema zadanim će se postavkama vratiti sve do samog elementa <html>, što znači da će biti postavljen u odnosu na samu stranicu.

Kompromis (i najvažnija stvar koju treba zapamtiti) o apsolutnom pozicioniranju je da su ti elementi uklonjeni iz toka elemenata na stranici. Na element s ovom vrstom pozicioniranja ne utječu drugi elementi i on ne utječe na druge elemente. Ovo je ozbiljna stvar koju treba uzeti u obzir svaki put kada koristite apsolutno pozicioniranje. Njegova prekomjerna ili nepravilna uporaba može ograničiti fleksibilnost vaše stranice.

Fixed

Element s fiksnim položajem postavljen je u odnosu na okvir za prikaz ili sam prozor preglednika. Okvir za prikaz se ne mijenja kada se prozor pomiče, tako da će fiksni pozicionirani element ostati tamo gdje je kada se stranica pomiče.

Ovo se može koristiti za nešto poput navigacijske trake za koju želite da ostane vidljiva cijelo vrijeme bez obzira na položaj pomicanja stranica. Problem s fiksnim pozicioniranjem je taj što može uzrokovati situacije u kojima fiksni element preklapa sadržaj tako da je nedostupan.

Sticky

Ljepljivo pozicioniranje je stvarno jedinstveno! Ljepljivi element samo će stajati tamo kao statični element, ali dok se pomičete pokraj njega, ako njegov nadređeni element ima mjesta (obično: dodatna visina), ljepljivi element će se ponašati kao da je fiksni sve dok nadređeni element ne bude

ostao bez dodatnog prostora. Zvuči čudno u takvim riječima, ali lako je vidjeti što se događa u demonstraciji: <https://css-tricks.com/position-sticky-2/>

6. Stiliziranje teksta

CSS svojstva za stiliziranje teksta mogu se podijeliti u dvije skupine:

- Stiliziranje fonta: svojstva koja utječu na font koji se primjenjuje na tekst, koliko je velik, gdje je podebljan, ukošen...
- Stiliziranje tekstualnog rasporeda: svojstva koja utječu na razmake i druga svojstva rasporeda teksta, kojima se utječe na prostor između linija teksta i poziciju teksta unutar kutije sadržaja.

Boje

Svojstvo *color* postavlja boju sadržaja koji je u prvom planu odabranog elementa.

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

Obitelji fontova

Za primjenu drugačijeg fonta za tekst, potrebno je koristiti svojstvo *font-family* koji omogućava određivanje fonta ili liste fontova koje će preglednik primijeniti na odabranom elementu. Preglednik će primijeniti font samo ako navedeni font dostupan na računalu s kojega se pristupa web stranici. Ako preglednik ne uspije primijeniti navedeni font unutar CSS-a, preglednik će primijeniti svoj font koji ima po zadanom.

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

Sigurni fontovi za web

Postoji određeni broj fontova koji su općenito dostupni na svim platformama, to su tzv. sigurni fontovi za web.

Problem je pronaći font koji je dostupan na računalima s kojima se pristupa web stranicama. Sigurni fontovi za web su poznati po dostupnosti na skoro svim operacijskim sustavima. Lista sigurnih fontova za web će se mijenjati kako se budu mijenjali operacijski sustavi, ali za sada zahvaljujući Microsoftu i inicijativi Osnovni fontovi za web (engl. *Core fonts for Web*) koja je bila aktivna kasnih 90 i ranih 2000 izdvojeni su fontovi koji danas spadaju u sigurne fontove za web.

Neki od fontova:

- Arial, Helvetica; generički tip: sans-serif,
- Courier New; Courier: generički tip: monospace,
- Georgia; generički tip: serif,

- Times New Roman, Times; generički tip: serif
- Trebuchet MS; generički tip: sans-serif
- Verdana; generički tip: sans-serif

Po zadanom fontovi

CSS definira pet generičkih imena za fontove: *serif*, *sans-serif*, *monospace*, *cursive*, *fantasy*. Ovo su vrlo općeniti fontovi i njihov izgled ovisi o svakom pregledniku i može varirati na svakom operacijskom sustavu na kojemu se pokreću preglednici. Njihov odabir predstavlja najgori mogući scenarij, gdje će preglednik pokušati primijeniti font koji barem izgleda prikladno.

Lista fontova

Kako se ne može garantirati dostupnost određenog fonta koji se želi primijeniti na web stranici, može se pružiti lista fontova i time omogućiti pregledniku izbor između više ponuđenih fontova. Za to je potrebno primijeniti svojstvo *font-family* i vrijednosti koje predstavljaju fontove razmaknute s znakom (,). U takvom slučaju, preglednik započinje od početka liste i provjera da li može primijeniti font na računalu s kojim se pristupa web stranici, ako ne može, pomiče se po listi i nastavlja dalje provjeravati preostale fontove.

Dobra je praksa pružiti na kraju liste generičko ime fonta koje se može primijeniti, ako nijedan font ne odgovara listi. Ako se ime fonta sastoji od dvije riječi i više, takva imena je potrebno navesti unutar navodnika.

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

Veličina fonta

Kada se pričalo o jedinicama veličine, spomenulo se postavljanje veličine fonta pomoću svojstva *font-size*:

- px (pixel): broj piksela kojim se govori kolika treba biti visina teksta. To je apsolutna vrijednost koja daje isti rezultat veličine fonta u svim situacijama.
- ems: 1ems je jednak veličini fonta koji je postavljen u roditeljskom elementu u kojemu se nalazi trenutni element, odgovara širini velikog slova M u roditeljskom elementu. Može postati izazov radeći s elementima koji imaju različite veličine fonta i koji su dosta ugniježđeni. Ali ispravnom primjenom, moguće je postaviti veličinu cijeloj stranici pomoću *em* jedinice.
- rems: funkcioniraju kao *ems*, samo što je 1rem ekvivalentan veličini fonta korijenskog elementa unutar HTML dokumenta (<html>), a ne roditeljskog elementa. Ovom jedinicom je puno lakše postavljati veličinu fonta, ali *rems* nisu podržani unutar Internet Explorera 8 i niže.
- Postoci: vrlo slično kao i em jedinica.

CSS pruže četiri česta svojstva za izmjenu teksta:

- **font-style:** koristi se za uključivanje italic teksta:
 - normal: postavlja tekst u normalni font (isključuje italic).
 - italic: postavlja italic tekst, ako font ima tu opciju, a ako ne postoji italic font, simulirat će se ukošenim tekstom.
 - oblique: postavlja tekst u simuliranu inačicu italic fonta, ukošavanjem normalnog tekst.
- **font-weight:** postavlja zadebljani tekst. Ovo svojstvo ima mnoge vrijednosti u slučaju da su dostupne različite varijante fonta kao što su: -light, -normal, -bold, -extrabold, -black... , najčešće se koriste normal i bold.
 - normal, bold: po zadanom izgled fonta i bold font.
 - 100 – 900: numerička vrijednost zadebljanog teksta koja omogućava precizniju kontrolu naspram prethodno navedenih ključnih riječi,
- **text-transform:** omogućava transformaciju fonta:
 - none: onemogućava bilo kakvu transformaciju.
 - uppercase: transformira sav tekst u velika slova.
 - lowercase: transformira sav tekst u mala slova
 - capitalize: transformira sve riječi da im prvo slovo bude veliko slovo.
 - full-width: transformira sve simbole u zapis fiksne širine.
- **text-decoration:** omogućava i onemogućava dekoraciju teksta nad fontovima:
 - none: onemogućava bilo koju dekoraciju teksta koja je bila aktivna.
 - underline: podcrtava tekst,
 - overline: daje tekstu gornju crtu,
 - line-through: postavlja crtu preko teksta.

Svojstvo *text-decoration* je skraćeno svojstvo što znači da može primiti više vrijednosti od jednom.

```
p {  
    text-decoration: underline overline;  
}
```

Poravnanje teksta

Svojstvo *text-align* se koristi za kontrolu kako će tekst biti poravnat unutar svoje kutije. Naredbe koje se koriste za poravnavanje su isti kao i one koje se koriste u naprednim obrađivačima dokumenta:

- left: lijevo poravnanje teksta,
- right: desno poravnanje teksta,
- center: centriranje teksta,
- justify: raširuje tekst uz varijaciju razmaka kako bi sve linije teksta bile iste duljine. Potrebno je ovu vrijednost svojstva koristiti oprezno jer može dati ružan rezultat.

Visina teksta

Svojstvo *line-height* postavlja visinu svake linije teksta. Ovo svojstvo može primiti različite jedinice, a ako primi numeričke vrijednosti, tada se ta vrijednost ponaša kao množitelj čime se smatra najboljom opcijom, *font-size* se množi, kako bi se dobilo *line-height*.

Razmak između slova i riječi

Svojstvo *letter-spacing* i *word-spacing* omogućavaju postavljanje razmaka između slova i riječi unutar teksta. Ovo se rijetko koristi, ali se može naći primjena za dobivanje određenog izgleda teksta.

Skraćena naredba

Mnoga svojstva fonta se mogu postaviti pomoću skraćena svojstva *font* i moraju biti zapisanu u slijedećem redoslijedu: *font-style*, *font-variant*, *font-weight*, *font-strech*, *font-size*, *line-height* i *font-family*.

Od svih navedenih svojstava samo *font-size* i *font-family* su obavezni kada se navodi skraćeno svojstvo *font*, dok se između *font-size* i *line-height* treba navesti kosa crta (/).

7. Liste

Liste se ponašaju kao i običan tekst, ali postoje određena CSS svojstva koja vrijede samo za liste. Ova se svojstva primjenjuju samo na numeriranim (ol) i na nenumeriranim (ul), kao što su:

- `List-style-type`: postavlja natuknice liste, npr. kvadratić ili kružić za nenumerirane liste ili arapske i rimske brojeve za numerirane liste.
- `List-style-position`: postavlja prikaz natuknice unutar stavke liste ili izvana prije početka svake stavke.

Svojstvo `list-style-type` za nenumeriranu listu može primiti vrijednosti: `none`, `circle`, `disc`, `square`.

Svojstvo `list-style-type` za numeriranu listu može primiti vrijednosti: `none`, `decimal`, `decimal-leading-zero`, `lower-alpha`, `upper-alpha`, `lower-roman`, `upper-roman`...

Svojstvo `list-style-position`:

```
<ol>
  <li>Primjer</li>
  <li>Primjer</li>
  <li>Primjer</li>
  <li>Primjer</li>
</ol>
```

```
ol {
  list-style-type: upper-roman;
  list-style-position: outside;
}
```

```
ol {
  list-style-type: upper-roman;
  list-style-position: inside;
}
```

Liste imaju i skraćeno svojstvo `list-style` kojemu se od vrijednosti može pridružiti `list-style-type` i `list-style-position`, nevezano kojim redoslijedom.

Stiliziranje hiperveza

Prilikom stiliziranja veza, važno je znati primjenu pseudo klase kako bi se ispravno stiliziralo stanje veze i kako stilizirati vezu za primjenu kod čestih varijacija sučelja kao što su navigacijski izbornik i kartice.

Stanje veze

Prvi koncept koji je potrebno razumjeti je stanje veze. Veza može imati različita stanja koja se mogu stilizirati korištenjem pseudo klase:

- Neposjećena veza: po zadanom postavljeno stanje veze, kada se ne nalazi niti u jednom drugom stanju. Ovo stanje se može eksplicitno postaviti korištenjem :link pseudo klase.
- Posjećena veza: stanje veze nakon što se posjetila web stranica pomoću te veze, stilizira se pomoću :visited pseudo klase.
- Lebdeće stanje: stanje veze kada se pokazivačem miša dođe iznad veze, stilizira se :hover pseudo klasom.
- Fokusirano stanje: postiže se kada se pomoću tab tipke na tipkovnici označi veza na stranici, stilizira se pomoću :focus pseudo klase.
- Aktivna veza: veza koja se aktivira klikom miša, stilizira se pomoću :active pseudo klase.

Po zadanom stanje veze

Kada se analizira po zadanom stanje veze, mogu se zamijetiti uključeni stilovi kao:

- Veze su podcrtane,
- Neposjećene veze su plave boje,
- Posjećene veze su ljubičaste boje,
- Prelaskom pokazivača miša, pokazivač mijenja ikonu,
- Fokusirane veze imaju pravokutni obrub oko teksta,
- Aktivne veze su crvene boje.

Prilikom stiliziranja važno je ne odmaknuti se predaleko od očekivano ponašanja veze, te je potrebno barem:

- Podcrtati vezu, ako se ne podcrtavaju, tada ih je potrebno barem na neki način sjenčati,
- Učiniti ih da reagiraju kada se prelazi pokazivačem miša iznad veze, kada su fokusirane i kada su aktivne.

Sa sljedećim CSS svojstvima moguće je isključiti ili izmijeniti stilove koji su postavljeni po zadanom:

- color: za boju teksta,
- cursor: stil pokazivača miša, savjetuje se ne isključivati,
- outline: obrub teksta, obrub je sličan rubu, jedino je razlika što rubovi zauzimaju mjesto u kutiji, a obrub ne zauzima, već samo sjedi na vrhu pozadine

Stiliziranje veze

Dobra je praksa prvo navesti CSS pravila koja će se primijeniti na veze.

```
a {}  
  
a:link {}  
  
a:visited {}  
  
a:focus {}  
  
a:hover {}  
  
a:active {}
```

Važno je poštovati redoslijed CSS pravila, jer se nadovezuju jedan na drugi, ako se navede drugačiji redoslijed pravila, konačni rezultat neće dobro izgledati.

```
a {
    outline: none;
    text-decoration: none;
    padding: 2px 1px 0;
}

a:link {
    color: #265301;
}

a:visited {
    color: #437A16;
}

a:focus {
    border-bottom: 1px solid;
    background: #BAE498;
}

a:hover {
    border-bottom: 1px solid;
    background: #CDFEAA;
}

a:active {
    background: #265301;
    color: #CDFEAA;
}
```

Stiliziranje liste koja je navigacija

Veze se često stiliziraju kako bi se ponašale i izgledale kao tipke u određenim situacijama, npr. glavna navigacija se opiše pomoću liste koja sadrži veze i to se može jednostavno stilizirati kako bi izgledalo kao skup kontrolnih tipki koje omogućavaju korisniku navigaciju kroz web stranicu.

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">Pizza</a></li>
  <li><a href="#">Burger</a></li>
  <li><a href="#">Sadwich</a></li>
</ul>
```

```
body, html {  
    margin: 0;  
    font-family: sans-serif;  
}  
  
ul {  
    padding: 0;  
    width: 100%;  
}  
  
li {  
    display: inline;  
}  
  
a {  
    outline: none;  
    text-decoration: none;  
    display: inline-block;  
    width: 19.5%;  
    margin-right: 0.625%;  
    text-align: center;  
    line-height: 3;  
    color: black;  
}  
  
li:last-child a {  
    margin-right: 0;  
}  
  
a:link, a:visited, a:focus {  
    background: yellow;  
}  
  
a:hover {  
    background: orange;  
}  
  
a:active {  
    background: red;  
    color: white;  
}
```

Ukoliko želite kreirati izbornik koji će imati više levela pomoću HTML i CSS-a, može pogledati ovaj link: <https://css-tricks.com/solved-with-css-dropdown-menus/>

8. Manipuliranje pozadinom u CSS-u

Pozadina elementa je područje koje se nalazi ispod sadržaja elementa, ispunjenja i ruba. Pozadina se ne nalazi ispod margine, jer se margina ne broji kao dio elementa. Postoji nekoliko svojstava pomoću kojih se mogu mijenjati vrijednosti pozadine, kao što su:

- `background-color`: postavlja boju pozadine elementa,
- `background-image`: postavlja određenu sliku pozadine elementa,
- `background-position`: definira poziciju pozadine unutar elementa,
- `background-repeat`: definira da li će se pozadina iza elementa ponavljati,
- `background-attachment`: definira ponašanje pozadine elementa kada se sadržaj elementa pomiče prilikom pregledavanja, da li će se pozadina pomicati zajedno sa sadržajem ili će biti fiksna,
- `background`: kratko svojstvo kojim se može uključiti u jednoj liniji svih pet prethodnih svojstava,
- `background-size`: omogućava dinamičku promjenu veličine pozadine

Boja pozadine

Pomoću svojstva *background-color* moguće je promijeniti boju pozadine i često se koristi zbog:

- Po zadanom, boja pozadine za većinu elemenata nije bijela boja već prozirna. Ako se roditeljskom elementu postavi određena boja pozadine, i ako se za dijete element želi staviti bijela boja pozadine, to se mora učiniti eksplicitno za svaki element.
- Uvijek je potrebno postaviti boju pozadine, npr. u slučaju da se ne uspije prikazati slika pozadine.

```
<p>Uzbudljiva kutija!</p>
```

```
p {  
  font-family: sans-serif;  
  padding: 20px;  
  background-color: yellow;  
}
```

Slika pozadine

Svojstvo *background-image* definira sliku pozadine koja će se primijeniti za određeni element. Kako bi se uključila slika za pozadinu elementa, potrebno je koristiti CSS funkciju *url()* kojoj se predaje apsolutna ili relativna putanja na resurs slike.

Slika se pozadini elementa pridružuje korištenjem CSS svojstva i služe samo za dekoraciju, a ne kao sadržaj.

```
p {  
    font-family: sans-serif;  
    padding: 20px;  
    background-color: yellow;  
    background-image: url("https://mdn.mozillademos.org/files/13026/fire-ball-icon.png");  
}
```

Po zadanom, trenutno se pridružena slika ponavlja horizontalno i vertikalno u pozadini elementa, to se može promijeniti korištenjem svojstva *background-repeat*.

Ponavljanje pozadine

Svojstvom *background-repeat* moguće je definirati kako se slika pozadine ponaša. Po zadanom, slika će se ponavljati, vrijednost je postavljena na *repeat*. Druge često primjenjivane vrijednosti su:

- no-repeat: slika se neće ponavljati, samo će se jedanput pojaviti,
- repeat-x: slika će se ponavljati horizontalno preko cijele pozadine,
- repeat-y: slika će se ponavljati vertikalno, do dna pozadine,
- repeat: slika se ponavlja horizontalno i vertikalno.

```
p {  
    font-family: sans-serif;  
    padding: 20px;  
    background-color: yellow;  
    background-image: url("https://mdn.mozillademos.org/files/13026/fire-ball-icon.png");  
    background-repeat: no-repeat;  
}
```

S vrijednosti *no-repeat*, slika se pojavila samo jedanput, ali je njezina pozicija nezgodno prikazana, i potrebno ju je izmijeniti.

Pozicija pozadine

Svojstvo *background-position* omogućava pozicioniranje slike pozadine. Općenito, svojstvo će primiti dvije vrijednosti odvojene razmakom koje predstavljaju horizontalnu (x) i vertikalnu (y) koordinatu

slike. Gornji lijevi kut slike predstavlja koordinate (0, 0), a pozadinu možemo zamisliti kao graf, gdje x koordinate ide s lijeve ka desnoj strani, a y koordinata ide od vrha prema dnu. Svojstvo može primiti različite vrijednosti od kojih su najčešće:

- Absolutna vrijednost u pikselima: `background-position: 200px 25px;`.
- Relativna vrijednost u rem-a: `background-position: 20rem 2.5rem;`.
- Postotak: `background-position: 90% 25%;`.
- Ključne riječi: `background-position: right center`, i može primiti vrijednosti kao; `left`, `center`, `right`, `top`, `center`, `bottom`.

Moguće je miješati vrijednosti koje se navode svojstvu *background-position*. Ako se navede samo jedna vrijednost, ta će vrijednost bit predstavljena kao horizontalna vrijednost, a vertikalna vrijednost će biti po zadanom `center`.

```
p {  
  font-family: sans-serif;  
  padding: 20px;  
  background-color: yellow;  
  background-image: url("https://mdn.mozillademos.org/files/13026/fire-ball-icon.png");  
  background-repeat: no-repeat;  
  background-position: 99% center;  
}
```

Pozadinska vezanost

Svojstvo *background-attachment* omogućava definiranje ponašanje slike pozadine elementa kada se sadržaj može pomicati prilikom pregledavanja. Vrijednosti koje svojstvo može primiti su:

- `scroll`: omogućava pozadini elementa pomicanje kada se pomiče stranica prilikom pregledavanja. Ako se pomiče sadržaj elementa, pozadina se ne pomiče. Pozadina je fiksirana na istoj poziciji stranice, tako da se pomiče kako se pomiče stranica.
- `fixed`: omogućava fiksnu pozadinu elementa kako se pozadina ne bi pomicala prilikom pregledavanja sadržaja.
- `local`: dodana je naknadno jer se vrijednost *scroll* ponaša konfuzno i ne čini uvijek ono za što je namijenjena. Vrijednost *local* fiksira pozadinu elementu za kojeg je postavljena, tako kada se pomiče sadržaj elementa, pomiče se i pozadina zajedno s tim sadržajem.

Svojstvo *background-attachment* ima efekt samo ako postoji sadržaj koji se pomiče

Veličina pozadine

Svojstvom *background-size* moguće je dinamički primijeniti dimenziju slike pozadine kako bi bolje odgovarala dizajnu. Ovo svojstvo prima dvije vrijednosti, jednu za horizontalnu dimenziju i jednu za vertikalnu dimenziju, te se mogu koristiti sve jedinice za veličinu.

Moguće je deklarirati sve vrijednosti svojstva u jednoj liniji kôda koristeći svojstvo *background*. Bilo koje svojstvo koje se ne navede unutar jedne linije bit će postavljeno na zadanu vrijednost.

```
p {  
    font-family: sans-serif;  
    padding: 20px;  
    background-image: url("https://mdn.mozillademos.org/files/13026/fire-  
ball-icon.png");  
    padding: 15px 15px;  
    background: yellow no-repeat 99% center;  
}
```

Zadaci

1. Osiguran su sljedeći resursi, HTML dokument i slike, dok je CSS datoteku potrebno kreirati i povezati s HTML dokumentom. Organizirati projektnu strukturu kako se savjetovalo na prošlim LV-i. Primijeniti resurse koji dolaze kao prilog uz ovaj zadatak.

Potrebno je stilizirati numeriranu listu nogometnih klubova, gdje svaki klub predstavlja vezu.

- Numeriranoj listi postaviti rub veličine 2px, fiksnu širinu 250px i ispunjenje od 5px.
- Postaviti svakoj stavki unutar liste ispunjenje od 3px, samo lijevo ispunjenje postaviti na 34px. Postaviti donju marginu na 5px osim zadnje stavke, a poziciju numeriranja postaviti unutar liste.
- Potrebno je dodijeliti zastave lijevoj strani svake stavke unutar liste, koristiti svojstvo *background*, navodeći listu vrijednosti: putanju do slike, 5px margina, centrirana slika, bez ponavljanja, s bojom pozadine. Boje su: Njemačka – crvena s 75% prozirnosti, Španjolska – plava s 50% prozirnosti i Velika Britanija – zelena s 25% prozirnosti.
- Onemogućiti stil vezi i postavljajući crnu boju teksta. Tek kada se pokazivačem miša dođe iznad stavke liste, prikazati stavku kao vezu, postavljajući podcrtanu dekoraciju, s velikim slovima i crvenom bojom teksta.

