Šta je CSS

- **CSS (*Cascading Style Sheets*)** je jezik koji se koristi za opis prezentacione semantike dokumenta pisanog u drugom opisnom jeziku (najčešće HTML).
- Jednostavnije rečeno, opisuje, tj. uređuje izgled i formatiranja bilo kog elementa na stranici. On nam omogućuje da, **kreirajući CSS opise stilizujemo elemente HTML** i drugih dokumenata.

Šta je CSS

- **CSS** skraćenica od **C**ascading **S**tyle **S**heets
- Stilovi nam dozvoljavaju kontrolu mnogih parametara dizajna koji ne mogu biti kontrolisani upotrebom HTML jezika.
- **Eksterni CSS fajl** nam omogućava da više HTML fajlova istovremeno stilizujemo

Šta je CSS

- CSS je nastao dosta kasnije nego što je to slučaj sa HTML-om. U početku, HTML stranice nisu bile vizuelno stilizovane. Potom smo koristili HTML tagove i atribute za stilizaciju, ali je sve to bilo previše komplikovano i nepraktično.
- Zato je osmišljen CSS jezik koji ostavlja HTML strukturi da prikazuje sadržaj, a CSS taj isti sadržaj stilizuje.
- U početku, mnogi autori su kombinovali HTML i CSS stilizaciju, ali danas, moramo, **u potpunosti poštovati pravilo razdvojenosti sadržaja (HTML) i stilizacije (CSS).**

Šta je CSS

- CSS nam omogućuje da kreiramo opise pomoću kojih ćemo stilizovati određene elemente.
- Na primer, CSS-om možemo definisati da pozadina strane bude plava i da sav tekst na njoj bude u fontu Arial i to u crvenoj boji. Takođe, možemo postaviti da svi paragrafi imaju određene margine i tako dalje.

CSS i HTML sintaksa

- **U CSS-u ne pišemo tagove niti postavljamo bilo kakav HTML sadržaj.** On služi isključivo za uređivanje i stilizaciju sadržaja u HTML i drugim dokumentima.
- CSS jezik ne koristi tagove i ima potpuno drugačiji način pisanja od HTML jezika
- CSS je odvojen od HTML jezika i njegovih tagova. Ukoliko imamo kreiranu HTML stranicu, nije potrebno da prepravljamo taj kôd kako bi uneli CSS kôd. Njega unosimo ili u posebnom fajlu (sa ekstenzijom *.css*) ili u head delu HTML dokumenta.

CSS primer

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

Primer kôda iznad predstavlja jedan jednostavan CSS opis (pravilo).

"Anatomija" CSS opisa

```
selektor {
    svojstvo: vrednost;
    svojstvo2: vrednost;
}
```

CSS opis

- CSS sintaksa se sastoji iz **CSS opisa (CSS Rule).**
- Sami CSS opisi su obavezno sastavljeni iz dva dela. Razlikujemo **selektor (selector)** i **deklaraciju (declaration)**.

- **Selektor** ukazuje na koji element se odnosi taj CSS opis.
- **Deklaracija** postavlja stilizaciju za element na koji se odnosi CSS opis.

CSS opis

- Deklaracije se takođe sastoje iz dva dela, a to su **svojstvo (property)** i **vrednost (value).**
- Kada kreiramo CSS opis, postavljamo **selektor**, zatim unosimo razmak i postavljamo vitičaste zagrade u kojima pišemo **deklaracije**, odnosno pišemo **svojstvo**, postavljamo dve tačke (:) i zatim **vrednost**.

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

CSS opis

- U istom opisu, možemo pisati više **deklaracija**, samo ih odvajamo znakom tačka-zarez (;).

```
p {
    font-family:Tahoma;
    font-size:12px;
}
```

CSS opis

- U ovom primeru, postavili smo selektor **p** koji označava da se taj opis odnosi na sve paragrafe u dokumentu (na <p> tagove u HTML-u).
- Postavili smo vitičaste zagrade i u njima pisali dve **deklaracije** koje se odnose na izbor fonta (font-family) i izbor veličine teksta (font-size). Primećujete da između **svojstva** i **vrednosti** u **deklaraciji** stoji uvek znak **:** , a da posle deklaracije uvek stoji znak **;** (koji označava kraj vrednosti)**.**

CSS opis

- Važno je pomenuti da novi redovi i razmaci u CSS jeziku ne igraju preterano značajnu ulogu, ali treba poštovati neku strukturu **radi preglednosti** kôda. Uobičajeno je da se pišu **selektor** i početna zagrada u jednom redu. Zatim, na sledećim redovima sve **deklaracije** (po jedna po redu) i na kraju zatvaramo zagradu u novom redu.
- CSS **selektori** razlikuju velika i mala slova tako da moramo voditi računa o tome.

CSS opis

- Dobro zapamtite ove termine koje smo pominjali jer ćemo se uvek vraćati na njih.
- Dakle, da ponovimo:

CSS jezik je sastoji iz opisa. Svaki opis je sastavljen iz selektora (selector) i deklaracija (declaration).

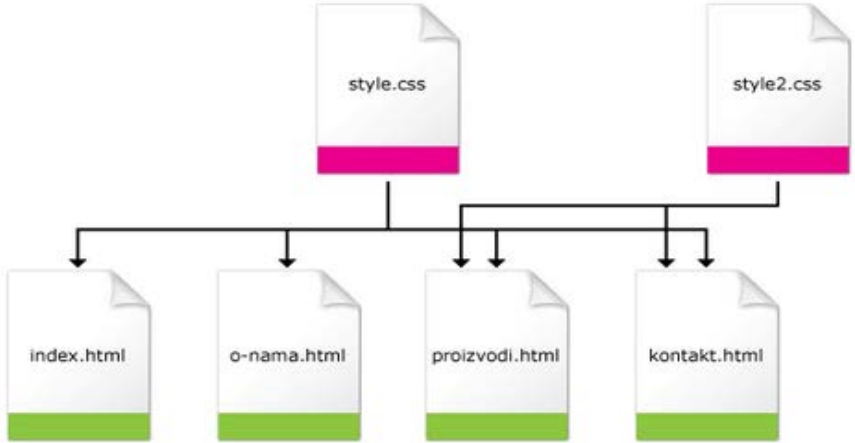
Deklaracija se sastoji iz svojstva (property) i vrednosti (value).

Način pisanja CSS opisa

- Pomenuli smo da je CSS odvojen od HTML koda, ali to znači da ga postavljamo...gde?
- Postoje tri varijante:
  - **Inline CSS** – opis se ubacuje u okviru samog taga u HTML dokumentu.
  - **Interni CSS** – pravila se upisuju u head sekciju same HTML strane i važe za stranu u koju su ugrađena.
  - **Eksterni CSS** – sva pravila se čuvaju u okviru *.css* fajla koji se vezuje za HTML strane i važi samo za te strane.

Eksterni CSS

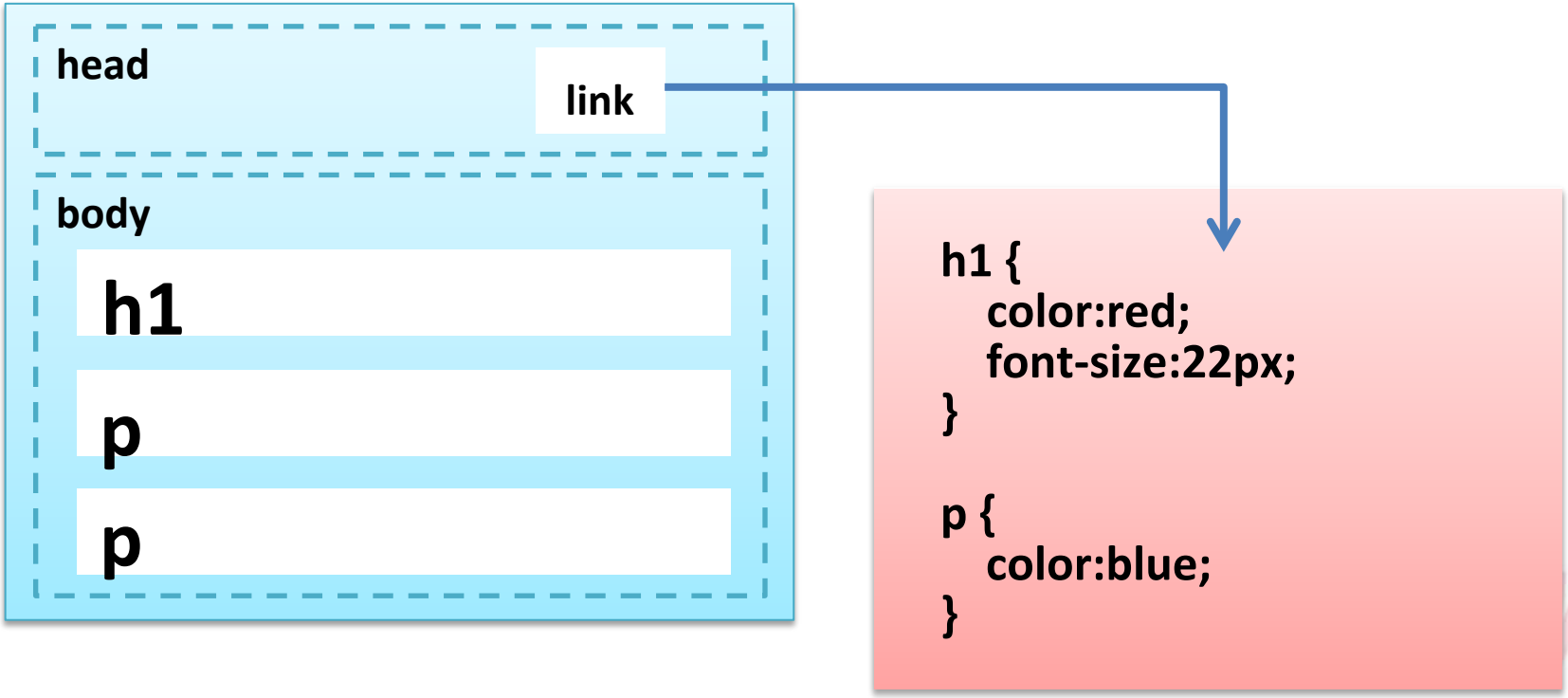
- Kao što mu i samo ime kaže, eksterni CSS se nalazi kompletno van HTML dokumenta u posebnom fajlu.



Eksterni CSS

- Najprostija varijanta je kada imamo jedan HTML fajl i pored njega jedan CSS fajl.
- Dovoljno je da u head delu HTML fajla unesemo određeni tag, koji će povezati eksterni CSS fajl. Sa druge strane, u samom CSS fajlu, pišemo samo opise.

Eksterni CSS



Eksterni CSS

Sam tag za povezivanje CSS opisa u head delu strane je **<link>.**
Ovo je samozatvarajući tag poput nekih drugih koje smo već upoznali (<img> npr), a sve vrednosti nosi u svojim atributima nad tagom. Ti atributi u slučaju <link> taga za povezivanje CSS fajlova su:

- **href** – definiše putanju do eksternog CSS fajla. Može biti relativna ili apsolutna. Preporuka je postavljati relativne putanje i voditi računa o vezama. **(obavezan)**
- **rel** – definiše odnos HTML fajla i linkovanog, eksternog fajla. Kada je u pitanju CSS postavljamo vrednost: ***stylesheet*** ***(obavezan)***
- **type** – definiše tip dokumenta ka kome se linkuje pošto se tag <link> može koristiti i za druge svrhe osim za CSS. Kada je u pitanju CSS postavljamo vrednost: ***text/css*** ***(u HTML5 opcioni i ne moramo ga pisati)***

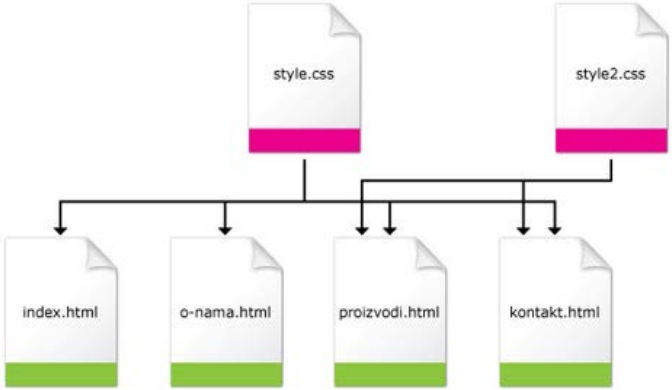
Evo primera za eksterno povezani CSS fajl, tačnije, početak HTML strane može izgledati ovako:

```
<!doctype html>
    <html>
    <head>
        <title>Eksterni CSS</title>
        <link href="css/stilizacija.css" rel="stylesheet">
        ......
    </head>
<body>
......
```

Ovaj tag se **uvek** postavlja u okviru **head dela** HTML fajla i praktično je uvek isti, samo se razlikuje putanja do fajla. Redosled atributa nije bitan.

Eksterni CSS

- U jednom HTML dokumentu može postojati više povezanih CSS fajlova. U tom slučaju postavićemo po jedan <link> tag za svaki eksterni css fajl.
- Jedan CSS fajl može skladištiti neograničen broj opisa, ali na autorima je da li će možda razdvajati grupe CSS opisa u različitim fajlovima radi lakše organizacije.
- Sa druge strane, jedan CSS fajl može biti pozvan, učitan, od strane više različitih HTML fajlova. Tako možemo stilizovati iste elemente na različitim stranama istim fajlom.



- Ne postoji ograničenje u broju opisa koje možemo definisati u okviru jednog .css fajla
- Na primer, definisanjem boje i veličine za tagove h1 i h2 unutar CSS fajla, menjamo izgled tih tagova na svim stranama na kojima je pozvan .css fajl

```
h1 {
 color: red;
 font-size: 2.2rem;
}

h2 {
 color: purple;
 font-size: 2rem;
}
```

Interni CSS

- Nasuprot eksternom, interni CSS se nalazi direktno u HTML dokumentu, tačnije u head delu.
- Umesto da postavljamo <link> tag kao malopre, možemo postaviti **<style>** tag i u okviru njega direktno upisati CSS opise.

Interni CSS

- Na ovaj način praktično smo ***ugradili*** ceo CSS fajl u strukturu HTML stranice.
- **Van <style> taga, nalazi se HTML strana i važe njena pravila**. Sa druge strane, **unutar njega se nalaze CSS opisi i važe pravila CSS jezika.**

Evo primera za interni CSS, tačnije, deo HTML dokumenta može izgledati ovako:

```
<html>
<head>
<title>Eksterni CSS</title>
<style type="text/css">
        p {
            font-family:Tahoma;
            font-size:12px;
        }
</style>
</head>
<body>
.....
```

- Postavljen je <style> tag sa atributom type koji određuje CSS tip stilizacije.
- Unutar <style> taga važe pravila CSS jezika.

Inline CSS

- Pored eksternog i internog CSS-a postoji i **inline** način pisanja. Inline se razlikuje po tome što je **selektor izostavljen** i sam CSS kod se postavlja direktno u HTML tagu, putem atributa.
- U ovom načinu pisanja nemamo ceo CSS opis, već samo deklaraciju (ili više njih) direktno na HTML tagu.

Inline CSS

- Kod inline CSS-a, postavljamo (na html tagu koji želimo da stilizujemo) atribut *style* u kome pod znacima navoda pišemo svojstva i vrednosti, kao i ranije.
- Zbog specifičnosti ove varijante, opis koji unesemo se odnosi samo na taj jedan element.
- **Zato, pisanje inline stilova je veoma nepraktično, a još teža je kasnija izmena.**

Inline CSS

Pogledajmo primer:

```
...
<p>Prvi paragraf</p>
<p style="font-family:Tahoma; font-size:12px;">Moj paragraf</p>
<p>Trećparagraf</p>
...
```

Ovako smo dobili stilizaciju nad samo jednim (drugim) paragrafom teksta. Ako bi želeli da svi budu stilizovani, morali bi i njima dodati atribut style i ova svojstva i vrednosti. Lakše je postaviti eksterni/ interni CSS i jednim opisom stilizovati paragrafe.

Tipovi CSS selektora

- Postoje razni tipovi selektora i pomoću njih možemo *gađati* skoro svaki element na stranici. Na nama je da odredimo koji ćemo način koristiti u zavisnosti od situacije.
- Zanimljivo je da, uglavnom, ne postoji samo jedan pravi način, već barem nekoliko njih.

Korišćena je reč „gađati“ kada govorimo o selektorima. Ovo ćete često čuti u žargonu. Iako bi pravilnije bilo reći nešto poput: „Ovaj opis stilizuje element koji je određen selektorom x“ češće se koristi „Ovaj opis gađa element x“.

neki tipovi selektora:

Tip selektora	Primer	Objašnjenje
Univerzalni	*	Gađa sve elemente na stranici
Opšti (tipski)	body, p, h2	Gađa tip elementa na stranici, na osnovu naziva taga
ID†	#primer	Gađa element sa označenom ID vrednošću.
CLASS‡	.primer	Gađa element sa označenom CLASS vrednošću.

† ID vrednost se u ID atributu HTML taga postavlja samo po imenu, dok se u CSS opisu piše sa prefiksom #

‡ CLASS vrednost se u CLASS atributu HTML taga postavlja samo po imenu, dok se u CSS opisu piše sa prefiksom . (tačka)

Opšti selektori

Ukoliko želimo da stilizujemo sve elemente određenog tipa (sve sa istim tagom), kao što su, na primer, svi p ili h1 tagovi, sve slike, body deo strane i tako dalje, možemo koristiti opšte selektore.
Imajte na umu da se ovakav CSS opis sa opštim tipom selektora, odnosi na sve tagove tog ciljanog tipa u dokumentu. Na primer, ako koristimo p, svi p tagovi će dobiti tu stilizaciju.

HTML:

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

Ovo je neki paragraf
Ovo je drugi paragraf

CSS:

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

ID selektor

Svaki HTML element može sadržati ID atribut. On se koristi da jednoznačno i unikatno odredi neki element. To znači da na određenoj HTML strani može postojati samo jedan element sa određenom ID vrednošću. Može se ponoviti u drugom dokumentu.

HTML:

```
<h1 id="naslov_glavni">Ovo je neki naslov kome je dodeljen ID</h1>
```

CSS:

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

Ovo je neki naslov kome je dodeljen ID

CLASS selektor

- CLASS atribut je vrlo sličan ID atributu, uz razliku da se može više puta pojaviti na jednoj HTML strani. Ponekad je korisno više elemenata označiti istom klasom (CLASS) kako bi ih odjednom sve stilizovali CSS-om. Drugim rečima, postavimo klasu nad više elemenata i jednim potezom u CSS-u ih sve stizujemo.
- Takođe, razlika u odnosu na ID je ta što jedan element može istovremeno sadržati više klasa. Sve ih pišemo pod znacima navoda u jednom atributu class, ali ih odvajamo razmakom.

HTML:

```
<p class=“moj_paragraf">Ovo je neki paragraf sa klasom</p>
```

CSS:

```
. moj_paragraf {
 color:blue;
}
```

Ovo je neki paragraf sa klasom

Imenovanje ID i CLASS vrednosti

- Samo slova i cifre, bez specijalnih znakova i bez "naših" slova.
- Razlikuju se velika i mala slova. Case-sensitive
- Ne smeju početi cifrom. Npr id vrednost *paragraf3* je ok, *3paragraf* nije.

Primer

Problem: *Postaviti **bold** nad svim paragrafima. Samo **trećem** paragrafu je potrebno promeniti boju fonta u zelenu.*

CSS:

```
p { font-weight:bold; }
.zeleni_text { color: green; }
```

HTML:

```
<p>Prvi paragraf</p>
<p>Drugi paragraf</p>
<p class="zeleni_text">Treći paragraf</p>
<p>Četvrti paragraf</p>
```

Prvi paragraf

Drugi paragraf

Treći paragraf

Četvrti paragraf

Definisali smo bold paragrafa selektorom p, a posebnom klasom smo definisali boju fonta za treći paragraf. Mogli smo pisati i ID u ovom slučaju.

Primer

Problem: *Postaviti **bold** nad svim paragrafima. Samo **prvom i trećem** paragrafu je potrebno promeniti boju teksta u zelenu.*

CSS:

```
p { font-weight:bold; }
.zeleni_text { color: green; }
```

HTML:

```
<p class="zeleni_text">Prvi paragraf</p>
<p>Drugi paragraf</p>
<p class="zeleni_text">Treći paragraf</p>
<p>Četvrti paragraf</p>
```

Prvi paragraf

Drugi paragraf

Treći paragraf

Četvrti paragraf

Definisali smo bold paragrafa selektorom p, a posebnom klasom smo definisali boju teksta za prvi i treći paragraf. Ovde je klasa bolji izbor nego ID jer jednu istu klasu možemo primeniti na više različitih elemenata. Upotrebom ID, morali bi koristiti dve zasebne ID vrednosti.

Složeni selektori

Osim osnovne upotrebe CLASS i ID selektora, možemo ih koristiti i okviru složenih selektora.
Ukoliko postavimo opšti selektor, a odmah zatim (bez razmaka) CLASS ili ID selektor, proveravamo da li ciljani element ispunjava oba ta uslova.

```
p.specijalno {
    font-weight:bold;
}
```

```
<p class="specijalno">Tekst</p>
```

```
h1#header {
    color:#ff0000;
}
```

```
<h1 id="header">Naslov teksta</h1>
```

Grupisanje selektora

Uz pomoć grupisanja selektora možemo za više elemenata definisati ista pravila. Nakon zareza se "ništa ne podrazumeva". To su i dalje dva selektora koja rade nezavisno, iako imaju ista svojstva i vrednosti.

```
h1, h2, h3, h4 {
    color: blue;
}
```

Sva četiri heading taga će biti ispisana plavom bojom

```
p.prvi, p.drugi {
    font-weight:bold;
}
```

Svi paragrafi sa klasom *prvi* i klasom *drugi* će biti prikazani bold

CSS BOX model

Block i inline elementi

- Svaki vidljivi element na stranici u startu (po defaultu) dobija neka pravila ponašanja - gde se nalazi, koje su mu dimenzije i tako dalje.
- Bilo da je u pitanju neki naslov, paragraf, slika, tabela ili nešto potpuno drugo, i bez ikakve stilizacije sa naše strane dobija pravila ponašanja i elementarnu stilizaciju od browsera koji to prikazuje.
- Najčešći tipovi prikaza su **block** i **inline**. Postoje i drugi o kojima će biti više reči kasnije.

Ne mešati inline elemente sa inline CSS. To nema veze jedno sa drugim!

Block i inline elementi

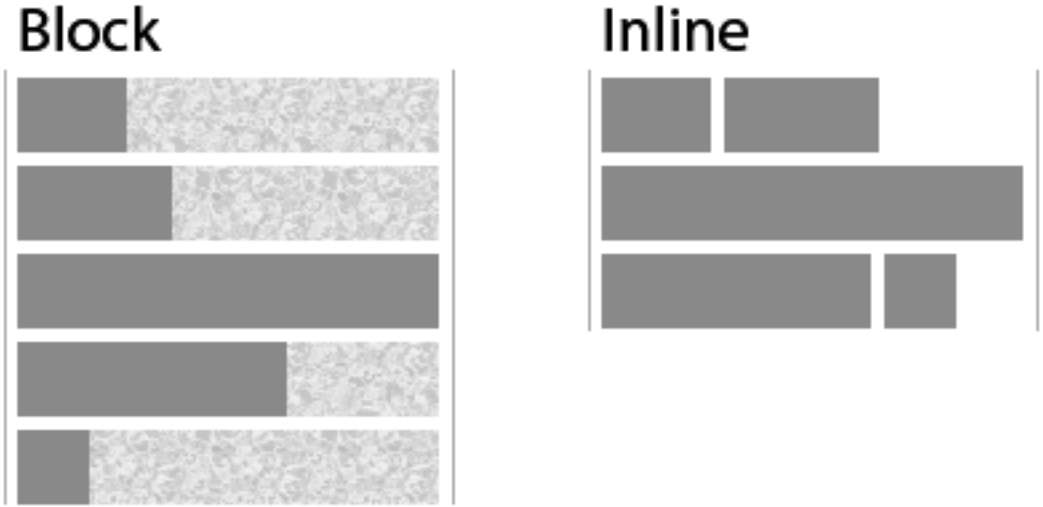
- Block elementi, ukoliko se ne definišu drugačije i nemaju postavljeno neko specijalno svojstvo pozicije (o tome kasnije), zauzimaju što je moguće više horizontalnog prostora i pomeraju sledeći element ispod, nakon sebe. Praktično *izguravaju* sve ispod i ostaju sami u tom redu.
- Block elementi su primarni alat za kreiranje rasporeda stranica.
- Inline elementi se baziraju na tekst formatiranju i tako se i postavljaju. Njihove dimenzije zavise samo od sadržaja unutar njih i nije moguće koristiti dimenzije i određene druge detalje na njima.

Block

- Maksimalna širina, automatska visina.
- Svaki element na novom redu.
- Možemo im menjati (zadavati) dimenzije i druge detalje CSS box modela.

Inline

- Automatska širina i visina, shodno sadržaju.
- Elementi u istom redu ukoliko prostor dozvoljava (inline).
- Ne možemo im menjati (zadavati) dimenzije i druge detalje CSS box modela bez dodatnih podešavanja.



Block elementi slažu jedan ispod drugog bez obzira iako je možda upisana manja širina. **Inline elementi se slažu jedan pored drugog** poput pojedinačnih karaktera u tekstu.

Block i inline elementi

- Po default vrednostima, većina elemenata koji se nalaze na stranici (unutar <body> dela) su ili block ili inline.
- Ipak, CSS-om možemo jednostavno promeniti neki inline element u block ili obrnuto. Na primer, stavke neuređene liste (block) možemo pretvoriti u inline elemente i dobićemo jedan red teksta sa stavkama koje stoje jedna za drugom. Takođe, možemo niz linkova u tekstu (inline) pretvoriti u block elemente i dobiti linkove vertikalno poređane.
- Prethodno navedeno znači da **svaki vidljivi element može biti ili block ili inline**, pitanje je samo da li je tako nešto potrebno u određenom kontekstu.

Block elementi

Po default vrednostima, u **block** elemente spadaju:

- Div elementi
- Paragrafi <p>
- Liste
- Stavke liste
- Naslovi <h1> – <h6>
- Osnovni HTML5 elementi <section>, <aside>, <nav>, <header> i <footer>
- Sam body tag <body>

i mnogi drugi.

Inline elementi

Po default vrednostima, u **inline** elemente spadaju:

- Span elementi
- Linkovi <a>
- Bold formatiranje ili <b>
- Italic formatiranje ili <i>
- Slike <img>
- Form oznake <label>

i mnogi drugi.

Ostali?

Elementi iz head dela stranice kao što su <script>, <meta>, <link>, kao i drugi iz body dela koji nisu vidljivi, **nisu ni inline ni block**, jer upravo su vidljivost i prikaz određeni ovim pravilima. Dodatno, možemo sakriti element iz prikaza ukidajući mu njegov tip prikaza (bilo block, inline..)

Grupisanje više elemenata u jedan block element

 …

Grupisanje više elemenata u jedan block element (DIV)

 element nam omogućuje da grupišemo više elemenata u jedan block element. Na primer, možemo kreirati *div* za header strane, i sve header elemente (logo, slogan, navigacija, pretraga) postaviti unutar njega.

** … **

Grupisanje više elemenata u jedan block element (DIV)

- Div tag, pošto je block tip elementa, počinje na novom redu.
- Po default vrednostima div-ovi nemaju nikakvu stilizaciju (transparentna je pozadina, border, margine i padding su 0, visina zavisi od sadržaja, širina je maksimalna).
- Ipak, možemo im dodati ID i/ili CLASS vrednosti i preko njih ih „gađati“ CSS opisima. Ta jednostavnost u osnovi, ali sa druge strane, velike mogućnosti stilizacije sa druge, čine da uz novije semantičke elemente **div elementi budu osnova građe svakog sajta.**
- Div elemenat (boks) može sadržati i više drugih elemenata, kreirajući hijerarhiju, što predstavlja ugnježdene elemente stranice.

Grupisanje više elemenata u jedan inline element

 …

Grupisanje više elemenata u jedan inline element (SPAN)

 element je inline ekvivalent elementa. Pomoću njega možemo grupisati više inline elemenata u jedan inline element. Koristi se najčešće za izdvajanje delova teksta, jer možemo, na primer, označiti deo teksta, dodati CLASS ili ID i zatim CSS opisom uticati na taj deo teksta (koji je okružen span-om).

** … **

Grupisanje više elemenata u jedan inline element (SPAN)

- Ono što je specifično za span je da **ne sme** sadržati block elemente, već samo druge inline elemente.
- U w3 specifikaciji za HTML se kaže:

  “Generally, block-level elements may contain inline elements and other block-level elements. Generally, inline elements may contain only data and other inline elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements”.

- Dakle, **block elementi mogu sadržati inline i block elemente, dok inline elementi mogu sadržati samo druge inline elemente.**

CSS Box model

- Kao što je već pomenuto, oko svakog block elementa na stranici, kreira se imaginarni okvir (box) koji možemo stilizovati.
- Ako bi razložili na korake prikaz nekog elementa, dobili bi ovakav redosled:
  1. Html postavlja sadržaj i kreira imaginarni okvir.
  2. CSS taj okvir stilizuje, menja po potrebi.
  3. Taj element se prikazuje korisniku.

CSS Box model (samo za block elemente!)

Box Model se sastoji od pet osnovnih svojstava elemenata:

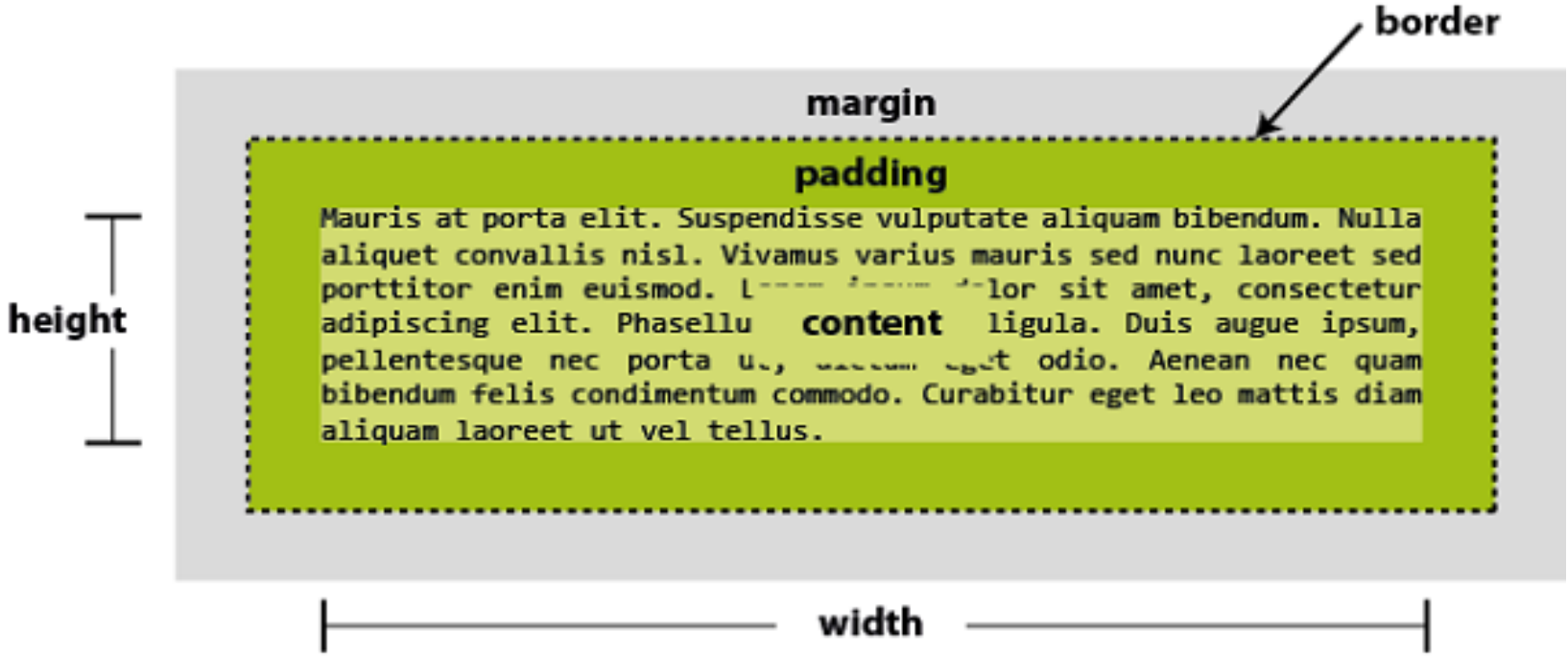
- **Width** (širina)
- **Height** (visina)
- **Margin** (margine ili spoljni razmak)
- **Padding** (padding ili unutrašnji razmak)
- **Border** (okvir, ivica)

- Iako ne spada u CSS Box model, često se pridodaje i **Background** svojstvo (pozadina).

Pogledajmo na primeru. Ovo je jedan stilizovan paragraf teksta:

Mauris at porta elit. Suspendisse vulputate aliquam bibendum. Nulla aliquet convallis nisl. Vivamus varius mauris sed nunc laoreet sed porttitor enim euismod. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at sodales ligula. Duis augue ipsum, pellentesque nec porta ut, dictum eget odio. Aenean nec quam bibendum felis condimentum commodo. Curabitur eget leo mattis diam aliquam laoreet ut vel tellus.

I on sadrži sve elemente CSS box modela, a koji su označeni na sledećoj slici:



Delovi Box modela

Margine su uvek transparentne i odvajaju element od drugih elemenata na stranici. Okviri **(Borders)** mogu biti različitih stilova i definišu granice elementa. **Padding** odvaja sadržaj elementa (Content) od njegovih okvira. Background podešavanja elementa se odnose na oblast unutar okvira zajedno sa **Padding i Content** delom (na slici Padding deo je prikazan u nešto tamnijoj zelenoj boji, mada je realno nevidljiv).

Delovi Box modela

- Margin, Border i Padding svojstva za svaku stranu elementa (Top, Right, Bottom, Left) mogu biti različite vrednosti.
- Po default vrednostima, ova svojstva su vrlo često 0, ali ne moraju biti.
- Margine mogu imati i negativnu vrednost, mada to nikako nije preporučeno.

CSS Box model „matematika“

- Važno je zapamtiti da Width i Height vrednosti određuju samo širinu, odnosno visinu sadržaja (content) elementa.
- **Realna visina i širina** koje element zauzima na strani je **content + padding + border**
- Na primer, ukoliko imamo div element, u kome je sadržaj 100 x 100 piksela, padding je 10px (sa svih strana), a border 2px (takođe sa svih strana), prostor koje takav elemenat zauzima je 124 x 124 piksela.
  Širina ukupno: **100px** za sadržaj +**10px padding x2** zbog levog i desnog paddinga + **2px za border x2** jer imamo levo i desno border. (100 + 10x2 + 2x2 = 124px).
  Isto važi i za visinu.

CSS Box model „matematika“

CSS kôd za element iz ovog primera bi mogao biti:

```
.mojDiv {
    width: 100px;
    height: 100px;
    padding: 10px;
    margin: 0;
    border: 2px solid #f90;
}
```

width i height

- Ova dva svojstva predstavljaju širinu, odnosno visinu sadržaja elementa. Po osnovnim vrednostima, širina i visina su taman tolike da uokvire sadržaj (inline), odnosno širina maksimalna, a visina spram sadržaja (block).
- Vrednost može biti u **pikselima, procentima, em vrednostima i td**. Ukoliko koristimo procente, veličina je određena veličinom prozora browsera, odnosno veličinom roditeljskog elementa (ako postoji). Kod em vrednosti, veličina zavisi od veličine teksta (fonta) unutar njega.

```
.mojDiv {
    width:40%;
    height:80px;
}
```

Padding

Ovo svojstvo definiše unutrašnji razmak (padding). Ukoliko ne postavimo vrednost, podrazumeva se 0. Ako svojstvo ne sadrži sufiks, onda se odnosi na sve četiri strane:

```
padding:10px;
```

Ukoliko želimo različite padding vrednosti gore, desno, levo i dole možemo pisati na primer:

```
padding-top:10px;
padding-right:15px;
padding-bottom:20px;
padding-left:25px;
```

Postoji i skraćeni način pisanja (shorthand) pomoću kojeg u jednom redu, koristeći samo padding svojstvo (bez sufiksa) možemo odrediti sve četiri vrednosti. Kod shorthand padding svojstva možemo pisati:

```
padding:10px 15px 20px 25px;
```

Ovako napisana deklaracija će imati isti rezultat kao malopređašnje četiri. Važno je samo naglasiti da se vrednosti **uvek** pišu u pravcu kazaljke na satu počevši od gornje. U našem primeru, 10px je vrednost gornjeg, 15px desnog, 20px donjeg, a 25px levog paddinga.

Margin

Ovo svojstvo definiše spoljašnji razmak (margin). Sva pravila koja važe za padding, važe i za margin. Podrazumevana vrednost je 0. Možemo pisati opšte svojstvo (margin), ili koristiti iste sufikse za strane kao kod paddinga. Ili pak možemo pisati shorthand (skraćeno) svojstvo.

```
...
margin:100px;
/* sve iste */
...
```

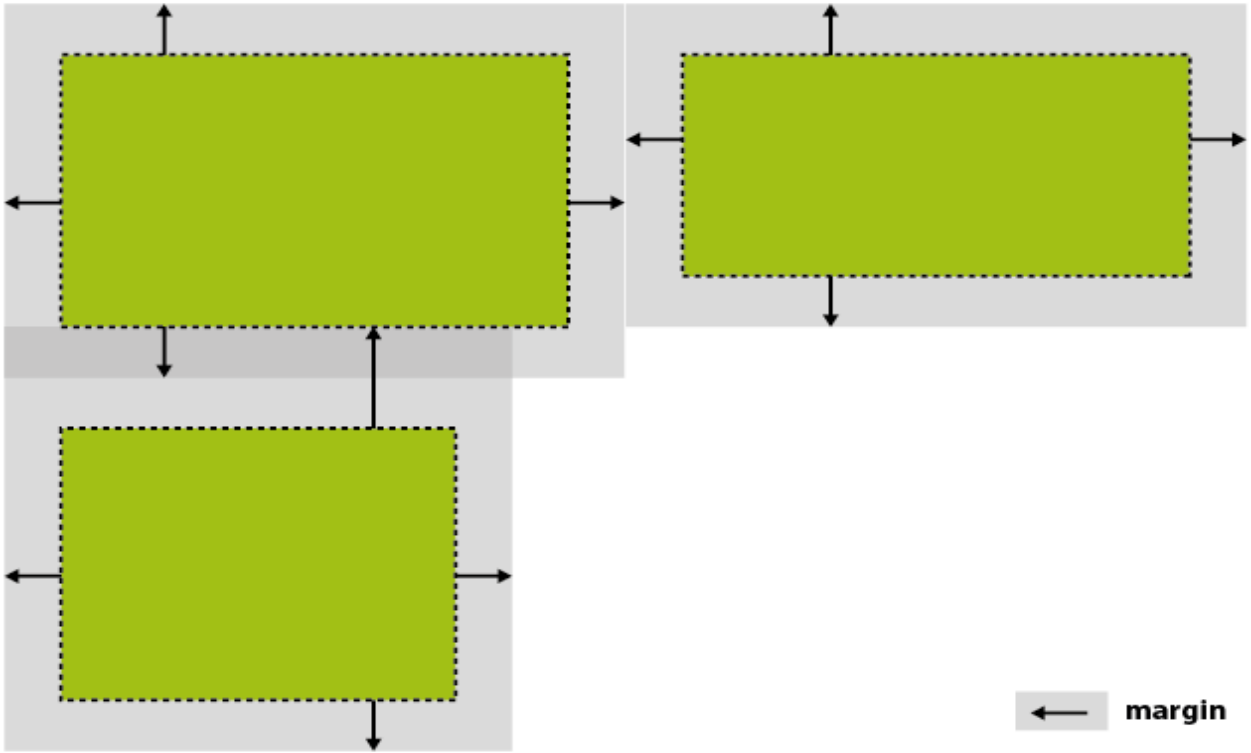
```
...
margin-top:100px;
margin-right:30px;
margin-bottom:200px;
margin-left:25px;
...
```

```
...
margin:100px 30px 200px 25px;
...
```

Margine i collapse

Kod margina je bitno znati da one mogu **kolapsirati (collapse),** odnosno, kada se nađu dva elementa jedan ispod drugog, ukoliko postoje margine između njih (donja margina na gornjem i gornja margina na donjem elementu), ukupan prostor **neće** biti zbir njihovih margina, već će se uzeti vrednost veće od te dve. Nasuprot tome, ukoliko su block elementi drugačije postavljeni margine ne kolapsiraju, već se sabiraju.

Margine i collapse



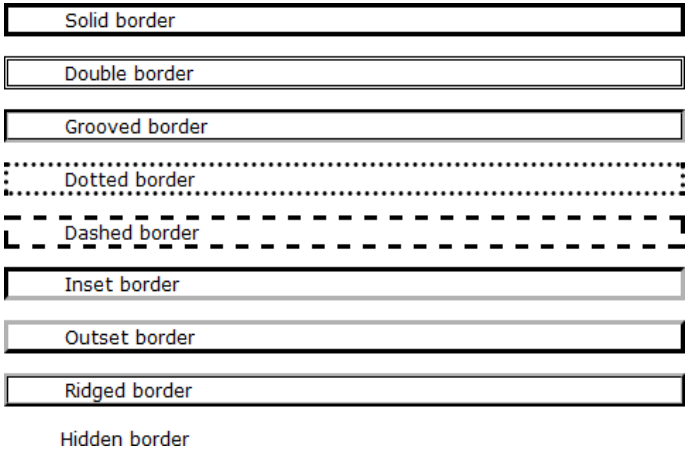
Border-width

Ovo svojstvo koristimo kako bi definisali debljinu ivice oko elementa. Kao vrednost možemo koristiti piksele (preporučeno) ili jednu od tri vrednosti: *thin, medium, thick (izbegavati).* Procenti i ostale jedinice u ovom slučaju nisu dozvoljene. Poput margine i paddinga, i kod border svojstva vrednost se odnosi na sve četiri strane. Različite možemo pisati skraćeno (shorthand) ili odvojeno, na primer:

```
border-top-width: 6px;
border-right-width: medium;
border-bottom-width: 3px;
border-left-width: thin;
```

Border-style

Ovim svojstvom određujemo tip ivice, okvira (border) oko elementa. Na slici ispod vidimo moguće varijante. Uglavnom se koristi *solid* (obična linija). Vrednost je sam naziv tipa (solid, double, grooved i tako dalje).



Border-style

- Ukoliko su sve ivice elementa iste, onda pišemo, na primer

  border-style:solid;

- Takođe, možemo individualno odrediti tip ivice koristeći:

  border-top-style: solid;
  border-right-style: groove;
  border-bottom-style: dotted;
  border-left-style: outset;

Border-color

Kao što samo ime kaže, ovo svojstvo definiše boju ivice. Oznaku boje (vrednost) možemo pisati na različite načine, pomoću hex, rgb, hsl ili imena boje.

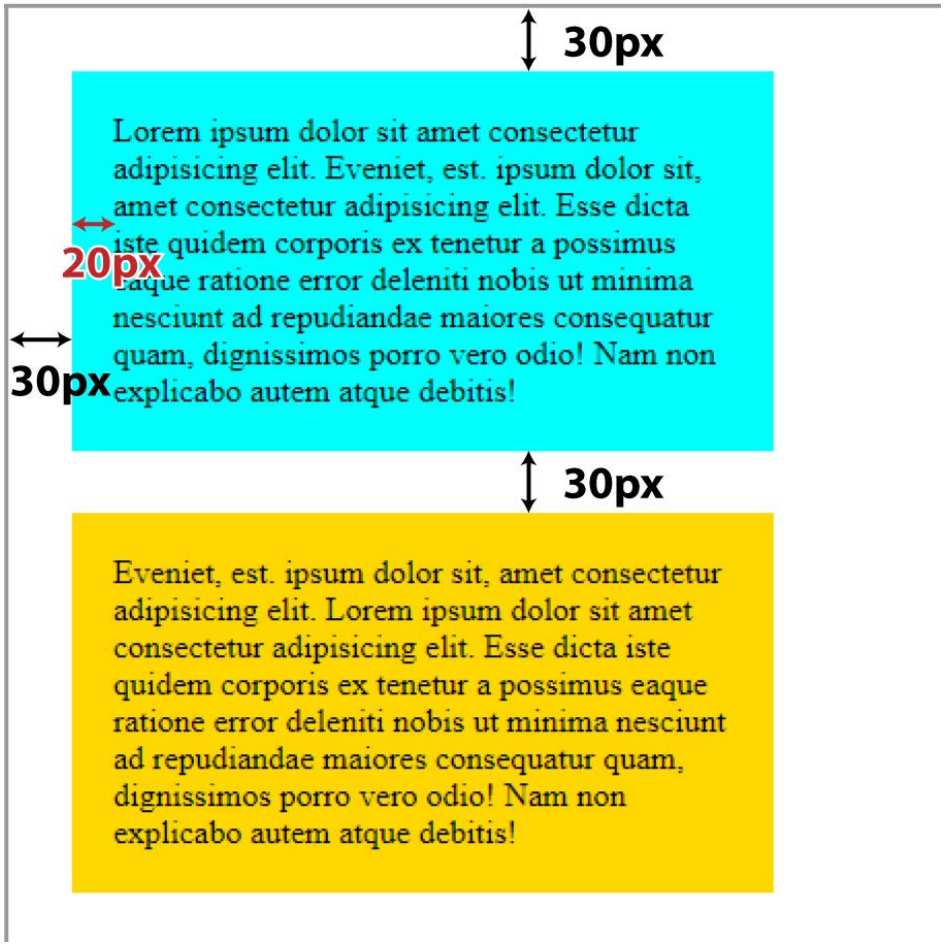
border-color: #FF9900;

Border

- Možemo iskombinovati i skratiti sve border varijante.
- Na primer, možemo pisati:

border: 1px solid #FF9900;

vežba



Display

Block elemente možemo pretvarati u inline i obrnuto.

Tip elementa može biti određen korišćenjem svojstva **Display** kroz CSS. Ako na primer, svojstvu Display nekog Inline elementa dodelimo vrednost Block, tada će taj element biti tretiran kao i svaki drugi Blok element.

```
span {

   display:block;
}
```

Ovim primerom smo span elemente pretvorili u block.

Display svojstvo ima i drugih upotreba o čemu više kasnije.

Visibility

Visibility svojstvo nam omogućuje da sakrijemo određeni element, ali da pritom ostane rezervisan prostor za njega. Može imati jednu od dve vrednosti, *hidden* (sakriva element) ili *visible* (prikazuje element – ovo je default vrednost i ne moramo je pisati).

```
span {
   visibility:hidden;
}
```

Razlika između *visibility:visible;* i *display:none;* je u tome što u prvom slučaju element nestaje, ali ostaje rezervisan prostor za njega, i ostali elementi se ne premeštaju, dok u drugom slučaju, element nestaje sa strane, ali i prostor rezervisan za njega, te se i ostali elementi razmeštaju po strani.

Overflow

Ovim svojstvom definišemo šta će se desiti ukoliko sadržaj prelazi okvire koji su mu zadati. Na primer, možemo definisati širinu i visinu nekog div-a na 200x300px, a pritom uneti celu stranu teksta. Pošto su definisane dimenzije manje od potrebnih, deo teksta će biti van okvira. Koristeći overflow svojstvo možemo definisati da se sadržaj koji ide van elementa ne prikazuje pomoću *overflow:hidden;* ili da skroluje pomoću *overflow:scroll;* Još jedna moguća vrednost je *overflow:visible;* koja je primenjena po defaultu.

```
div.mojTekst {
  overflow:scroll;
}
```

Postoje i zasebna overflow-x i overflow-y svojstva za X odnosno Y osu.

Box-sizing svojstvo

CSS3 je doneo svojstvo **box-sizing**. Ukoliko njega postavimo nad određenim elementom, možemo definisati da li će se padding i border dodavati na unete dimenzije (što je default stanje) ili će se oduzimati od unetih dimenzija.

Za box-sizing svojstvo, dostupne su vrednosti:

Content-box – klasično, default stanje CSS Box modela. Na unetu širinu CSS-om dodaju se padding i border ukoliko postoje, i tako dobijamo vidljivu širinu.

Border-box – ukoliko postavimo ovu vrednost, element na koji je primenjeno, ponašaće se po drugačijem CSS Box modelu. Unete dimenzije će biti iste kao vidljive dimenzije. Eventualni padding i / ili border će biti unutar unete širine.

Napomena:
Postojala je i vrednost: **Padding-box** – ali je izbačena iz specifikacije i više se ne koristi.