

CSS model

Ranije u kursu smo pomenuli da svaki element na HTML stranici možemo predstaviti sa zamišljenim okvirom (kutijom) oko njega i da postoje inline i block elementi. U ovoj lekciji ćemo se detaljno pozabaviti tim pojmovima i na koji način možemo upravljati pomenutim okvirima oko elemenata kroz CSS box model.

VAŽNO: nemojte mešati pojmove **inline elemenata** i **inline CSS stilizacije**. Inline elementi su tip elementa na HTML stranicama, dok je inline CSS stilizacija, način postavljanja CSS stilizacije na bilo kom elementu (više o tom tipu stilizacije u prethodnoj lekciji).

Osnove rasporeda (layout)

Ukoliko radimo u Adobe Illustratoru, možemo ređati okvire za tekst kako god želimo pomerajući ih po dokumentu. U Photoshopu možemo crtati po slojevima (layerima) gde želimo. U MS Wordu sliku možemo premeštati sa mesta na mesto prevlačenjem. Iako bismo to možda i očekivali i ovde, u HTML-u i CSS-u, istina je da je raspoređivanje elemenata sasvim drugačije. Ovde je to igra guranja i ređanja. Možemo napraviti analogiju sa zidanjem zida pomoću cigli, moramo imati prethodni red (ili barem jednu ciglu), kao podršku za sledeće. Razlika je samo to što se kod nas cigle slažu odozgo na dole, ali je princip isti.

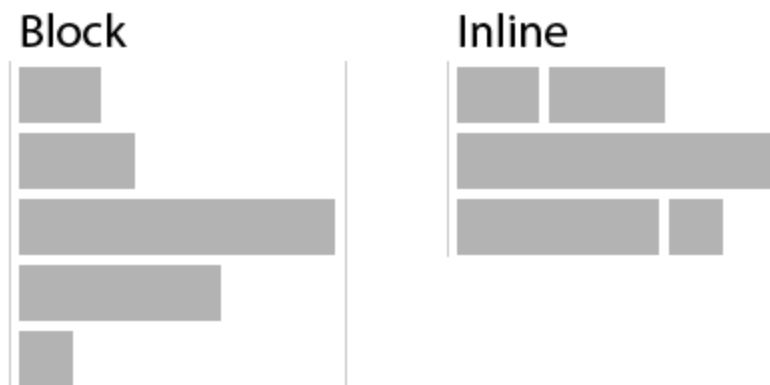
Ukoliko ne shvatimo i usvojimo znanja vezana za raspoređivanje elemenata i na koji način funkcioniše HTML/CSS kombinacija jezika, imaćemo velikih problema kasnije oko pretvaranja „idealnog“ dizajna koji smo nacrtali/dobili u Photoshopu (Fireworksu, Illustratoru...), u funkcionalni HTML dokument.

Block elementi se slažu jedan na drugi, inline jedan pored drugog

Svi vidljivi elementi na strani, počev od slika i paragrafa do linkova, su smešteni u nevidljive okvire (u engleskom se koristi termin *box* – doslovno *kutija*).

Ti okviri se pojavljuju u dve varijante: block i inline. Razlika između block i inline elemenata leži u njihovom ponašanju na strani.

- **Block** elementi su pravougaonici koji teže da popune sav horizontalan prostor.
- **Inline** elementi su pravougaonog oblika i uklapaju se uz druge elemente.



Block i inline elementi

Ono što je prikazano na ilustraciji iznad, a što čini suštinu razlika je to da se **Block elementi slažu jedan ispod drugog** bez obzira na njihovu širinu. **Inline elementi se slažu jedan pored drugog** poput pojedinačnih karaktera u tekstu.

Block elementi, ukoliko se ne definišu drugačije, zauzimaju što je više moguće horizontalnog prostora i pomeraju prethodni i sledeći element iznad, odnosno ispod sebe. Praktično *izguravaju* sve pored sebe iznad ili 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. Ukoliko definišemo širinu inline elementa na npr. 200px, neće se ništa desiti, njegova širina će i dalje zavisiti od širine sadržaja. Nasuprot tome, dodavanje još teksta će povećati širinu.

Po default vrednostima, svaki element koji se nalazi na stranici (unutar <body> dela) je 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.

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

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

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

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

Elementi (tagovi) iz head dela stranice kao što su <script>, <meta>, <link> i tako dalje nisu ni inline ni block, jer nisu vidljivi na stranici i ne spadaju pod ova pravila.

Grupisanje elemenata

Grupisanje više elemenata u jedan block element (DIV)

<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.

Div tag, pošto je block tip elementa, počinje na novom redu. Osim toga, ne utiče na izgled strane. Po default vrednostima, div-ovi nemaju nikakvu stilizaciju (transparenta je pozadina, border, margine i padding su 0, visina zavisi od sadržaja i tako dalje). 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 **div elementi** budu **osnova građe svakog sajta**.

Div elemenat (boks) može sadržati i više drugih elemenata, kreirajući hijerarhiju, što predstavlja ugnježdene elemente stranice. Prozor čitača u ovoj hijerarhiji predstavlja Root (koreni) element.

Grupisanje više elemenata u jedan inline element (SPAN)

 element je inline ekvivalent <div> 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 spanom).

Ono što je specifično za span jeste da **ne sme** sadržati block elemente, već samo druge inline elemente. U w3c 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.¹

¹ Izvor: <http://www.w3.org/TR/html401/struct/global.html#h-7.5.3>

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

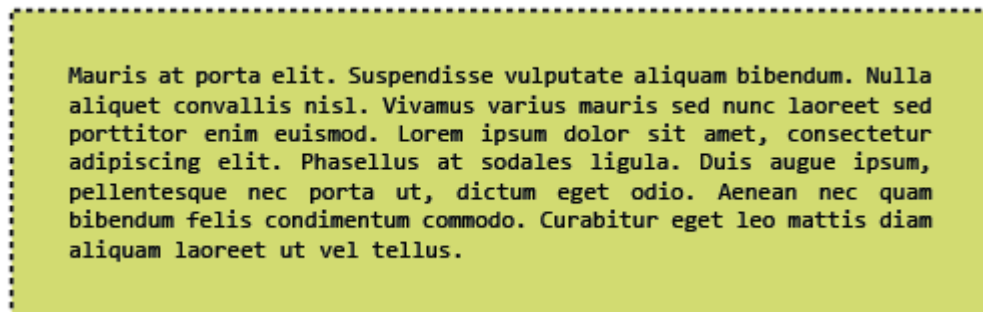
CSS Box model

Kao što je već pomenuto, oko svakog vidljivog elementa na stranici kreira se imaginarni okvir (box) koji možemo stilizovati. Ako bismo razložili na korake prikaz nekog elementa, dobili bismo ovakav redosled: HTML postavlja sadržaj i kreira imaginarni okvir. CSS taj okvir stilizuje, menja po potrebi, i na kraju se taj element prikazuje korisniku.

Box Model se sastoji od pet osnovnih svojstava elemenata:

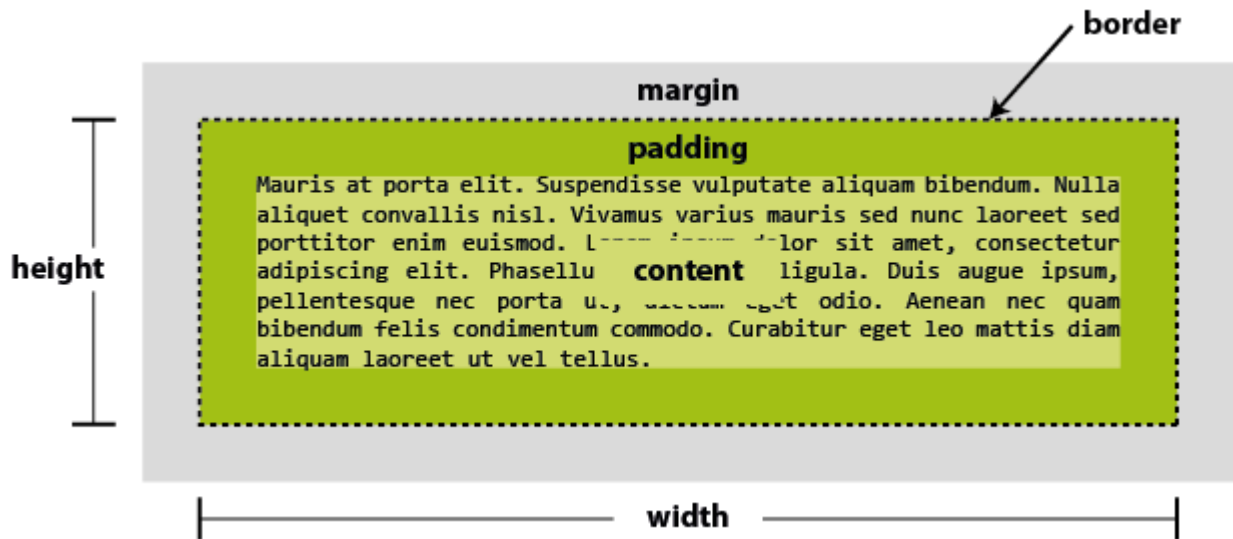
- **Width** (širina)
- **Height** (visina)
- **Margin** (margine ili spoljni razmak)
- **Padding** (padding ili unutrašnji razmak)
- **Border** (okvir, ivica)
- Iako ne spada u CSS Box model, često se pridodaje i **Background** svojstvo (pozadina).

Pogledajmo na primeru. Ovo je jedan stilizovan paragraf teksta:



Stilizovan paragraf teksta na stranici

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



Isti paragraf teksta sa prikazanim CSS box model elementima

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).

Margins, Borders i Padding svojstva za svaku stranu elementa (Top, Right, Bottom, Left) mogu biti različite vrednosti. Po default vrednostima, ova svojstva su u većini slučajeva 0. Margine mogu imati i negativnu vrednost, mada to nikako nije preporučeno.

Važno je zapamtiti da Width i Height vrednosti određuju samo širinu, odnosno visinu sadržaja (content) elementa. **Realna visina i širina** koje elemenat zauzima na strani je **Content + Padding + Border**.

Na primer, ukoliko imamo div element, u kome je sadržaj 100 x 100 piskela, padding je 10px (sa svih strana), a border 2px (takođe sa svih strana), mesto koje takav elemenat zauzima je 124 x 124 piksela. To je zato što je širina ukupno: **100px** za sadržaj + **10px za padding x2** zbog levog i desnog paddinga + **2px za border x2** jer imamo levu i desnu. ($100 + 10 \times 2 + 2 \times 2 = 124$ px). Isto je i za visinu.

CSS kôd za element iz ovog primera bi mogao biti:

```
div {
  width:100px;
  height:100px;

  padding:10px;
  margin:0;

  border-width:2px;
  border-style:solid;
  border-color:#f90;
}
```

U nastavku jedinice ćemo videti detalje ovih svojstava.

CSS svojstva u box modelu

Width i Height

Ova dva svojstva predstavljaju širinu, odnosno visinu sadržaja elementa. Po osnovnim vrednostima, širina i visina su taman tolike da uokvire sadržaj. Izuzetak je širina kod block elemenata koja se prostire maksimalno.

Vrednost može biti u **pikselima, procentima ili em vrednostima**. Pikseli su najpopularniji način i u većini slučajeva se oni koriste. Ukoliko koristimo procenete, 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:400px;  
    height:80%;  
}
```

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 malopredašnje 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

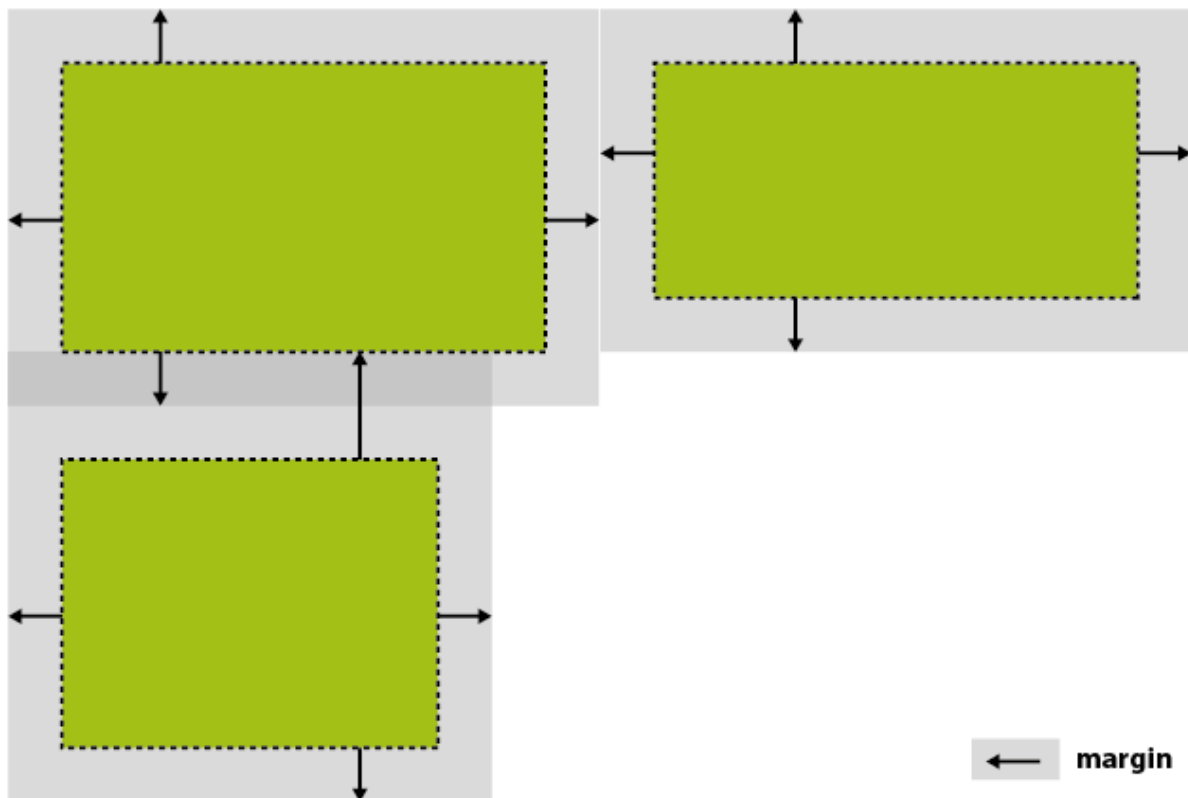
Ovo svojstvo definiše spoljašnji razmak (margin). Sva pravila koja važe za padding, važe i za margin. Podrazumevana vrednost je 0. Možemo pisati opšte svojstvo (margin), ili koristiti iste sufikse za strane kao kod paddinga. Ili pak možemo pisati shorthand (skraćeno) svojstvo.

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

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

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

Kod margina je bitno znati da one mogu **kolapsirati (collapse)**, odnosno, ukoliko 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 elementi jedan pored drugog, margine ne kolapsiraju, već se sabiraju.



Ilustracija margina

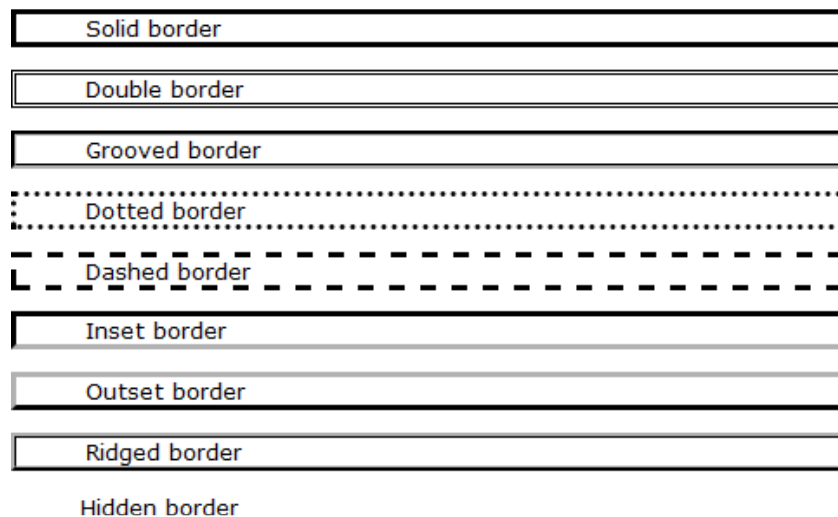
Border-width

Ovo svojstvo koristimo kako bismo definisali debljinu ivice oko elementa. Kao vrednost možemo koristiti piksele ili jednu od tri vrednosti: *thin*, *medium*, *thick*. 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).



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 sistema ili imena boje (o bojama na webu u posebnoj lekciji).

```
border-color: #FF9900;
```

Druga važnija svojstva

Display

Ranije u lekciji smo pomenuli da 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.

Pored vrednosti *block* i *inline* koje koristimo za promenu tipa, display svojstvo može imati i vrednost *none*. U tom slučaju, ako je primenjeno *display:none* browser će se ponašati kao da element ne postoji, mada će biti u kôdu. Takođe i bilo koji ugnježdeni elementi unutar tog elementa neće biti prikazani iako im je dodeljena neka druga Display vrednost.

Na primer, span tag je inline elemenat. Ukoliko dodelimo display:block vrednost, ponašaće se, praktično, kao div. Ukoliko sa druge strane postavimo display:none, pomenuti span neće biti vidljiv. Ipak, morate zapamtiti da pomoću ove opcije ne uklanjamo kompletno elemenat već samo sakrivamo njegov prikaz, ali korisnik može uvek očitati sirov HTML kod koji sadrži taj deo. Ne bi trebalo sakrivati tako neke osetljive podatke.

Visibility

Visibility svojstvo nam omogućuje da sakrijemo određni element, ali da pri tom 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. Podsećam, koristeći obe tehnike, elementi su samo vizuelno sakriveni od korisnika, i dalje su dostupni u kôdu, te ne treba sakrivati osetljive podatke na ovaj način.

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 pri tom uneti celu stranu teksta. Pošto su definisane dimenzije manje od potrebnih, ceo tekst neće biti vidljiv (deo koji izlazi van div-a). Ipak, koristeći overflow svojstvo možemo definisati da se sadržaj unutar elementa skroluje koristeći *overflow:scroll;* Druga moguća vrednost je *overflow:hidden;* koja će „odseći“ višak koji prelazi van okvira. Po default vrednostima, primenjen je *overflow:visible;*

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

Ranije se ovo svojstvo više koristilo za unos dužih tekstova u ograničene okvire. Sada, sa razvojem svih aspekata weba, postalo je izlišno, čak i nepraktično. Ipak, jedna sporedna osobina ovog svojstva postala je korisna na modernim sajtovima, a o njoj ćemo pričati nešto kasnije, u jedinici o raspoređivanju elemenata.

Min-width, max-width

Kao što i sami nazivi ovih svojstava govore, radi se o minimalnoj, odnosno o maksimalnoj širini elementa. Umesto da koristimo fiksiranu dimenziju, možemo definisati minimalnu i/ili maksimalnu dimenziju.

```
#wrapper {  
    min-width:960px;  
    max-width:1280px;  
}
```

Nije obavezno korišćenje oba svojstva na istom elementu. Ne treba kombinovati sa fiksnim *width*.

Min-height, max-height

Poput prethodno pomenutih svojstava, min-height i max-height se istovetno ponašaju, samo sa tom razlikom da se odnose na visinu elementa.

```
p {  
    min-height:10px;  
    max-height:26px;  
}
```

Pitanje u jedinici

Realna, vidljiva širina elementa na strani je?

- **width + padding + border**
- width + margin + border
- width + padding
- width + border

Najvažnije iz lekcije

- Postoje Block i Inline elementi.
- Inline CSS stilizacija i Inline elementi nisu isti pojam.
- Elementi na strani zavise od drugih elemenata.
- Block elementi su pravougaonici koji teže da popune sav horizontalan prostor.
- Inline elementi su pravougaonog oblika i uklapaju se uz druge elemente.
- Elemente možemo grupisati u celine (DIV kao block ili SPAN kao inline)
- CSS box model elementa se sastoji od širine (Width), visine (Height), margine (Margin), paddinga (Padding), ivice (Border), a često se uključuje i pozadina (Background).