

Tok dokumenta

Do sada smo se upoznali sa osnovama CSS jezika i osobinama elemenata. U ovoj jedinici proširujemo svoje znanje i obrađujemo tok dokumenta, odnosno na koji način se elementi slažu i kreiraju strukturu. Objasnićemo razlike između normalnog toka (engl. *natural stack*) i ostalih tipova. Videćemo šta znači float vrednost i kada se primenjuje.

Pozicioniranje

Kod pozicioniranja elemenata možemo razlikovati sledeće tipove:

- static (default)
- relative
- absolute
- fixed.

Static i relative pozicioniranje čine normalni tok dokumenta.

Static pozicioniranje

Static je po default vrednostima osnovno pozicioniranje svakog elementa na stranici. U literaturi se često i ne spominje (jer je osnovno) ili se označava kao default, normal i sl. Ono što je bitno je da postoji i da je svakako primenjeno na elemente.

Static je primenjen na sve elemente koji nemaju specificiranu poziciju putem *Absolute* ili *Fixed* i koji nisu *Floated* (upoznaćemo float kasnije).

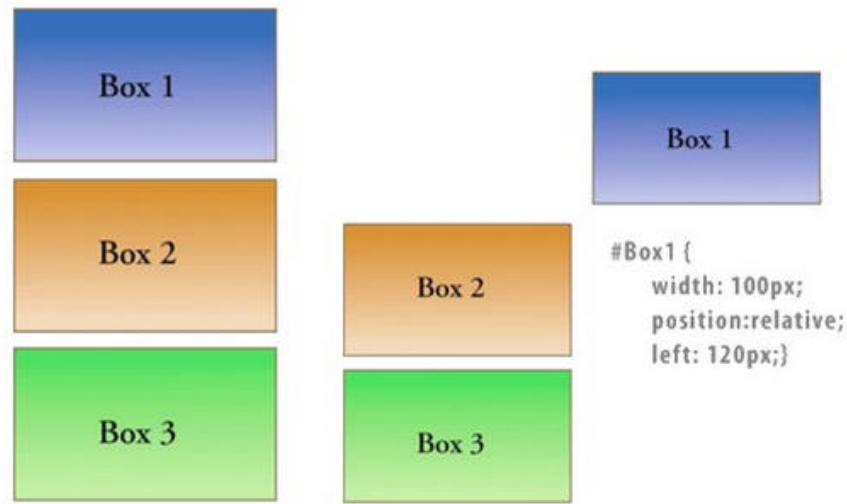
Već smo saznali, u priči o elementima, osnovna pravila normalnog toka, ali ćemo ponoviti. Block elementi su pozicionirani jedan ispod drugog po vertikali, dok su Inline elementi pozicionirani po horizontali, sleva nadesno. Vertikalne margine kolapsiraju (preklapaju se) u normalnom toku dokumenta. To znači da se umesto sabiranja vrednosti margina gornjeg i donjeg elementa računa samo margina koja je veća. Horizontalne margine nikad ne kolapsiraju.

Relativno pozicioniranje

Kada je elementu dodeljeno svojstvo *position: relative*, tada je element inicijalno pozicioniran prema pravilima normalnog toka, odnosno potpuno isto, ali uz jednu razliku u odnosu na static, samo zato se vodi kao poseban način pozicioniranja.

Element možemo pomeriti iz njegovog osnovnog položaja, dok će prostor koji on zauzima na osnovnoj poziciji ostati prazan. Pomeramo ga prema **Offset** svojstvima, koje mogu biti *Left*, *Right*, *Top* i *Bottom*. Vrednost svakog ovog svojstva predstavlja razdaljinu za koju bi

odgovarajuća ivica elementa trebalo da bude pomerena u odnosu na poziciju elementa u static varijanti.



Relative pozicioniranje

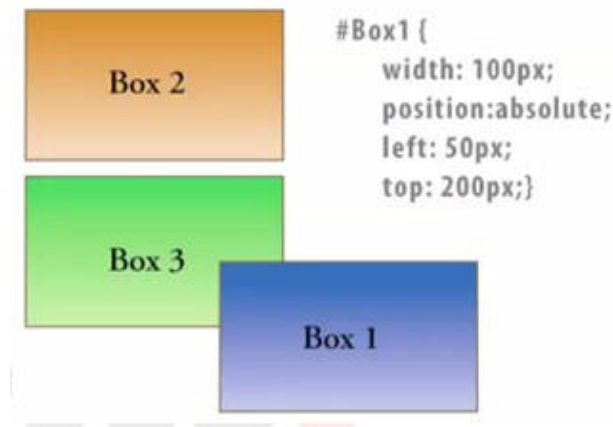
Boks 1 koristi relativno pozicioniranje da bi se pomorio udesno za 120px u odnosu na njegovu startnu poziciju. Boks 2 i Boks 3 ostaju u istom mestu, nepromenjene pozicije, (ponašaju se kao da Boks 1 nije ni promenio poziciju, tj. kao da je ostao na istom mestu) jer relativno pozicioniranje ne pomera elemente iz normalnog toka.

Obratite pažnju da pomeranje u desno označavamo svojstvom *left*, jer udaljavamo element za njegovu vrednost od leve ivice osnovne pozicije. Analogno tome, *top* svojstvo pomera element nadole.

Apsolutno pozicioniranje

Elementi sa svojstvom *position: absolute;* se uklanjaju, tj. pomeraju iz normalnog toka i ne utiču na ostale elemente. Pozicija elemenata se određuje na osnovu Offset vrednosti koje mogu biti Left, Right, Top i Bottom. Ove vrednosti se ponašaju skoro isto kao i kod relativnog pozicioniranja, sem što se kod apsolutnog pozicioniranja element pomera u odnosu na roditeljski element, koji takođe mora biti pozicioniran, tj. kome je dodeljeno svojstvo position. Najlakše je dodati *position: relative* roditeljskom elementu ako njega ne želimo da pomeramo.

Apsolutno pozicioniranje nam dozvoljava da definišemo gde bi element trebalo da bude pozicioniran apsolutno u odnosu na stranicu ili u odnosu na roditeljski element.



Absolute pozicioniranje

Boks 1 je apsolutno pozicioniran, Left i Top svojstva ga pomeraju na desno i dole, a Boks 2 i Boks 3 se ponašaju kao da Boks 1 ni ne postoji i zauzimaju njegovu poziciju u normalnom toku. Zbog svoje nove pozicije Boks 1 se preklapa sa Boksom 3 i bilo koja promena u sadržaju Boks 3 ne utiče na poziciju ostala dva elementa.

Česta zamka u koju početnici upadnu je upravo korišćenje apsolutnog pozicioniranja za sve elemente. Iako deluje primamljivo, može (i svakako hoće) stvoriti ogromne probleme u kasnijem radu.

Ovaj tip pozicioniranja treba koristiti samo ponekad kada je neophodno i veoma oprezno.

Fiksno pozicioniranje

Fiksno pozicioniranje je specijalan način apsolutnog pozicioniranja. Kada se skroluje stranica, fiksiran element se ne pomera kao i svi ostali elementi. Ovaj atribut se ne ponaša isto u svim čitačima, a IE6, kao i ranije verzije Internet Explorera ga uopšte ne podržavaju.

```
#mojElement {  
  position: fixed;  
  left: 10%;  
}
```

Element je pomeren na desno za 10% od ukupne širine prozora čitača i fiksiran je u tom mestu tako da, kada skrolujemo kroz stranicu, mojElement ostaje na istom mestu bez obzira na sve ostalo.

Upotreba block elemenata za layout

Da bismo kreirali osnovnu strukturu stranice, koristimo block elemente (uglavnom DIV-ove). Pomenuli smo da su se ranije koristile i nevidljive tabele za tu namenu, ali to danas nikako nije preporučljivo.

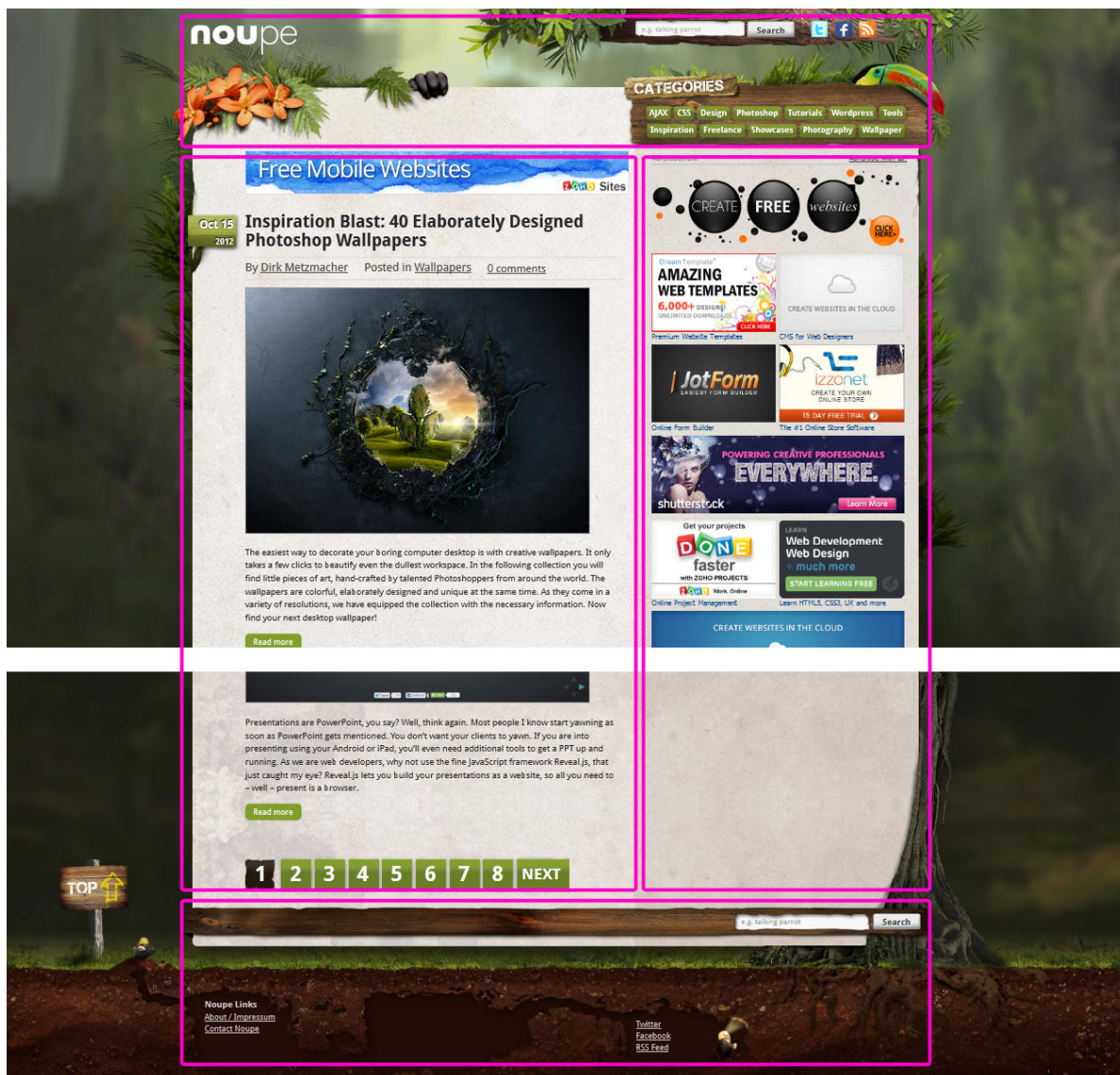
Dakle, kreirajući DIV elemente i raspoređujući ih u celine, postavljamo osnovnu strukturu (layout) naše stranice, odnosno sajta.

Da bismo to malo plastičnije prikazali, pogledajte sledeći primer:



Čest raspored na sajtovima

Na slici nije prikazana stranica sajta, već jedan čest layout kakav se može naći danas na sajtovima. Pogledajmo popularan web dizajn blog – www.noupe.com i njegov osnovni layout. Ljubičaste linije su dodate radi lakšeg prepoznavanja regiona, ne postoje na sajtu.



www.noupe.com

Na datom primeru vidimo da je vrlo sličan našem konceptu na prethodnoj slici, iako se zbog dizajna to možda ne primećuje na prvi pogled. U svakom slučaju, iako korisnik možda toga nije svestan, osnovni regioni sajta se kreiraju tako što se postave div elementi (ili ekvivalenti u HTML5 jeziku).

Postavljanje layouta

Postoje četiri pravila, tačnije, aspekta na koje bi trebalo paziti oko postavljanja block elemenata za layout:

1. **Dimenzije:** prostor koji element zauzima i na koji utiče.
2. **Float:** uticaj na ponašanje blocka. Izdvajanje iz normalnog toka.
3. **Clear:** ponovno uspostavljanje normalnog toka.

4. **Ugnježdavanje:** element je ili unutar drugog elementa ili u istom nivou hijerarhije sa njim, ne postoji ništa između.

Važno: sve ovo se odnosi na normalni tok (static i relative pozicioniranje). Absolute i fixed su po konceptu drugačiji i ovi pojmovi se ne odnose na njih.

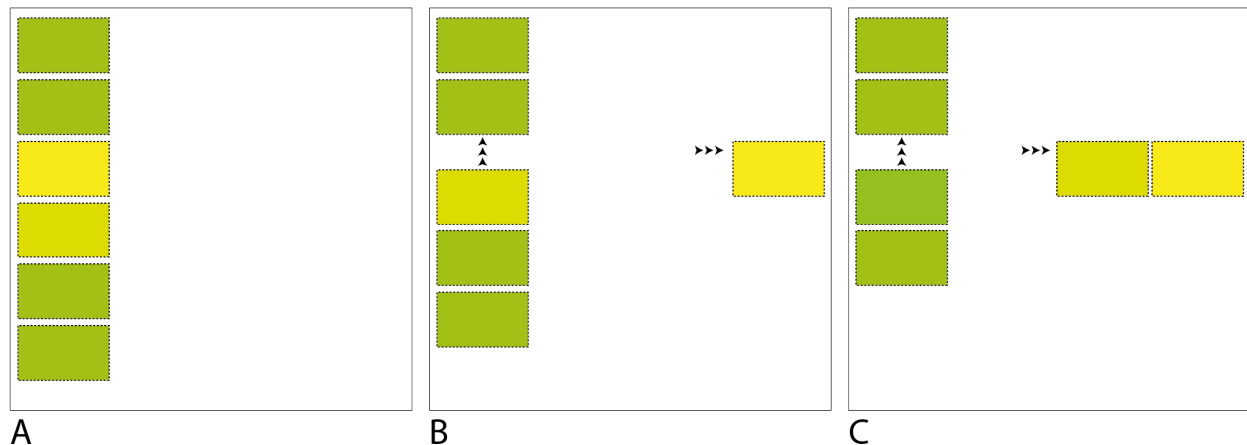
1. Ukupna širina

Ukupna širina zavisi od više osobina elementa, kako je i objašnjeno u prethodnoj jedinici. Na ukupne dimenzije utiču sama širina/visina, ali i padding i border. Pored toga, margine elementa su vizuelno van samog elementa, ali takođe utiču na prostor koji zauzima i na koji utiče.

Ukoliko pogrešno saberemo dimenzije, često dolazi do grešaka. Na primer, ukoliko postavimo div od 1000px širine, a unutar njega želimo da postavimo dva druga diva, jedan pored drugog, možemo im dati širine po 500px samo ako su njihove margin, padding i border vrednosti 0. Ukoliko dođe do promene (povećanja) tih dimenzija, moramo smanjiti širine kako bi zbir opet ostao 1000px (ili manje od 1000px).

2. Float

Floating se ostvaruje preko svojstva *float* sa vrednostima *left* (levo) i *right* (desno). Kada dodelimo ovu vrednost nekom elementu, on izlazi iz normalnog toka koji smo ranije pomenuli i ponaša se drugačije. Vertikalno je pozicioniran na isti način (kao kod normal toka – static ili relative), dok je po horizontali pozicioniran skroz levo ili desno, koliko je moguće unutar svog bloka u kome je sadržan. Za razliku od bokseva u normalnom toku, vertikalne margine floated elemenata nikada ne kolapsiraju sa marginama elemenata iznad ili ispod.



Prikaz ponašanja float elemenata

Na ilustraciji iznad, na delu koji je označen kao **A** vidimo više block elemenata koji su u normalnom toku (nije im postavljen float). Pod oznakom **B** vidimo jedan element kome je postavljena vrednost `float:right`; njegova pozicija se odmah premešta na desnu stranu u visini u kojoj je originalno i bio postavljen. Svi elementi koji slede se više ne obaziru na float element i zauzimaju njegovo mesto. Ukoliko postavimo float i na sledeći element (prikaz **C**)

i on će se pomeriti skroz desno do ranije floatovanog elementa, ali se neće preklopiti. Ostali elementi se opet premeštaju ne obazirući se na ta dva.

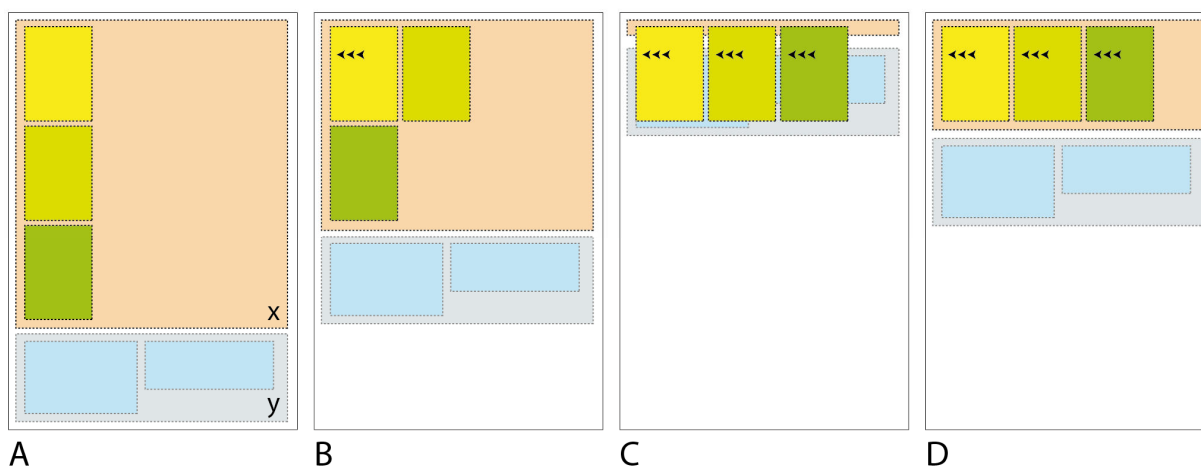
Float se postavlja veoma jednostavno, dodavanjem *float* svojstva i vrednosti *left* ili *right* u željenom CSS opisu.

```
#sidebar {  
    float:right;  
    width:200px;  
}
```

Preporuka je da float vrednost uvek prati i širina elementa.

3. Clear

Jedna (moglo bi se reći nezgodna) osobina float svojstva je to što roditeljski element koji sadrži float element ne prepoznaje njegovu visinu (Height). Dakle, visina float elementa ne utiče na ponašanje (visinu) roditeljskog elementa. Roditeljski element se trudi uvek da obuhvati sve svoje elemente, osim one koji su float. Zašto je to bitno? Bitno je jer ako roditeljski element sadrži samo floatovane elemente, neće imati visinu, odnosno biće jednaka nuli (ukoliko nije ručno data preko Height svojstva). Na stranici kod koje većina elemenata stoji u normalnom toku to može dovesti do haosa. Pogledajmo sledeću ilustraciju:



U početku (**A**) postavljena su dva div elementa, X i Y, kojima nije eksplicitno data visina, tj. zavise od unutrašnjih elemenata. X sadrži tri block elementa.

Ukoliko postavimo float jednom od tri elementa X (**B**), druga dva će se rasporediti pored njega i samo njihova visina (elementa koji nisu float) utiče na div X (vidimo da mu se smanjuje visina) i element Y se pomera na gore.

Ukoliko su sva tri elementa unutar X elementa *float:left* (prikaz **C**), sam element X će kompletno izgubiti visinu. Njegovi elementi se i dalje vide, ali se sledeći element Y pomorio skroz do X. Elementi se preklapaju i nastaje haos.

Rešenje je jednostavno, a to je dodavanje svojstva **clear** na elementu koji sledi. U ovom slučaju, to je element Y. Pomenuto clear svojstvo može imati vrednosti *left*, *right* ili *both*. Njima označavamo sa koje strane elementa se ne smeju nalaziti float elementi. U našem primeru, ako sva tri elementa unutar X imaju *float:left*; onda *clear:left* ili *clear:both*; nad y elementom rešavaju stvar.

```
#footer {  
    clear:both;  
}
```

Možda se pitate zašto bismo koristili clear ili overflow. Zašto jednostavno ne postavimo visinu (Height) okružujućim elementima? Ovo bi moglo biti rešenje, ali time ograničavamo mnoge aspekte svoje stranice. Najčešće se ovi elementi koriste za osnovne elemente strukture, a to znači da će različite stranice istog sajta (sa ovim istim elementima) imati različite sadržaje. Postoji staro pravilo među web dizajnerima koje glasi: *ukoliko možete da duplirate sav sadržaj na stranici (dva puta duži tekst, dve slike umesto jedne i sl.) a da se layout ne raspadne, onda ste dobro postavili layout*.

Još malo o float svojstvu

Iako zamišljen sa namerom da sređuje float, clear svojstvo nije uvek najsrećnije rešenje. Uglavnom se koristilo i koristi tako što se doda što jednostavniji block element direktno posle floatovanih elemenata (npr
 tag), i kome se visina postavi na 1px, visibility skloni i tako dalje (jer nije potrebno da ga uopšte primetimo na strani), ali mu se i doda *clear:both*. To je možda najelegantnije rešenje, ali zahteva dodatak tog ekstra taga, koji nema drugu svrhu i sadržaj.

Još jedno, možda bolje rešenje, a koje se pojavilo manje više slučajno, jeste upotreba *overflow* svojstva o kojem smo pričali ranije. Naime, ukoliko element koji okružuje float elemente sadrži *overflow:hidden*; uzeće u obzir njihove visine, postaviće se pravilno oko njih, kao da nemaju float i tako nema potrebe za clear svojstvom. Ovom solucijom izbegavamo dodatne tagove, a i elementi koji slede ne moraju imati dodatne CSS redove. U našem primeru od malopre, mogli smo dodati *overflow:hidden*; na element X i tako bi rešili problem.

4. Ugnježdavanje

Prilikom planiranja treba jasno odvojiti i isplanirati elemente, odnosno definisati koji elementi će biti na istom nivou, jedan posle drugog, a koji će se nalaziti jedan u drugom. Podsećam, svi elementi se nalaze u korenom elemenu (body), ali onda možemo ostale smeštati unutar njih.

Primer layouta

Pošto smo naučili šta je float i kako možemo postaviti elemente, pogledajmo jedan opšti primer, recimo da želimo postaviti dve kolone sa centralni deo strane, main div (za sadržaj) i sidebar div (za sporedni sadržaj). U HTML-u bismo jednostavno dodali sledeće:


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Float</title>
</head>
<body>

    <div id="wrapper">

        <div id="main">
        </div>

        <div id="sidebar">
        </div>

    </div>

</body>
</html>
```

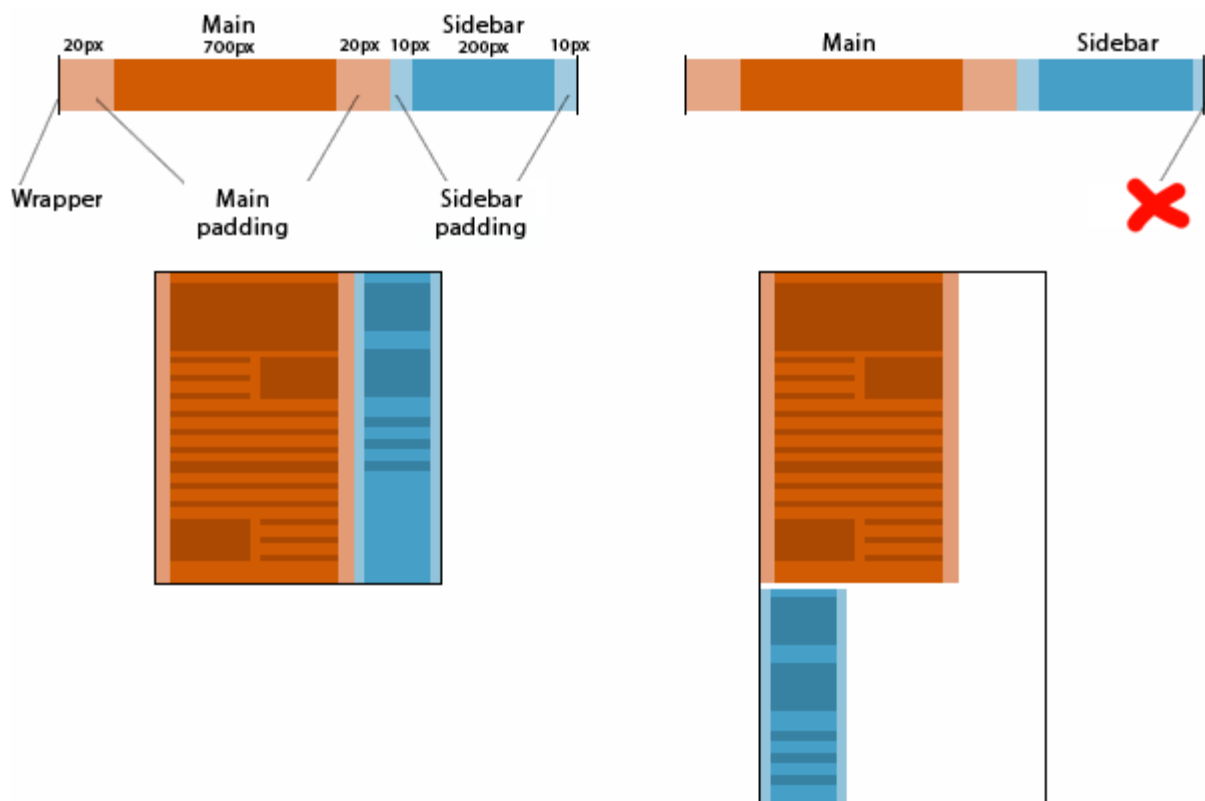
Dok bismo u CSS-u postavili sledeće:

```
#wrapper {
    width:960px;
}

#main {
    float:left;
    width:700px;
    padding:0 20px 0 20px;
}

#sidebar {
    float:right;
    width:200px;
    padding:0 10px 0 10px;
}
```

I ukoliko unesemo sadržaj u main i sidebar elemente, dobijamo nešto poput prikaza na sledećoj ilustraciji (levi deo slike).



Float glavnih elemenata

Ukoliko pažljivije pogledamo levi deo ilustracije, i naš kôd, primetićemo da je ukupni zbir širina tačno 960px, što odgovara širini okružujućeg elementa.

700px (main) + 2x 20px (main padding) + 200px (sidebar) + 2x 10px (sidebar padding) = 960px

Ukoliko bismo makar za 1 piksel povećali neku od ovih vrednosti, dobili bismo nešto poput desnog dela ilustracije. Pošto nema dovoljno prostora (širine), sidebar bi prešao u sledeći red, odnosno ispod main dela.

Naravno, ovo su samo dva elementa. Možemo sad dodati header div pre main dela (unutar wrappera) i footer posle sidebar dela (pre zatvaranja wrapper diva). Sa tom razlikom da njima nije potreban float i možemo ih pustiti da se prostiru celom širinom (po defaultu za block elemente).

Z index

Ukoliko su elementi postavljeni pomoću relative, absolute ili fixed pozicioniranja, i ukoliko se preklapaju, onaj element koji je kasnije u kôdu dokumenta pojaviće se vizuelno iznad elementa koji je ranije u kôdu i time ga sakrivajući u manjoj ili većoj meri.

Ipak, možemo uticati na to preklapanje upotrebom z-index vrednosti. Ovo je slično kao opcija *send to back*, *bring forward* i slične koje možemo naći u Wordu, Photoshopu i sl.

Kao vrednost unosimo brojčanu vrednost po svom nahođenju i element sa većim brojem će biti vidljiv. Vrednosti mogu biti negativne.

```
.elementIspred {  
    z-index:100;  
}  
  
.elementIza {  
    z-index:-20;  
}
```

Dakle, samo vrednosti nisu bitne, već njihov međusobni odnos.

Pitanje u jedinici

Da li absolute pozicioniran element utiče na druge elemene na stranici?

- **Ne**
- Da
- Da, ali samo na one koji su takođe absolute
- Da, ali samo na one koji nisu absolute

Najvažnije iz lekcije

- Kod pozicioniranja elemenata možemo razlikovati sledeće tipove: static (default), relative, absolute, fixed.
- Osnovni layout postavljamo pomoću block elemenata.
- Postavljanjem float vrednosti nad elementom izdvajamo ga iz normalnog toka.
- Float elementi moraju imati širinu
- Float elemene moramo srediti (postaviti *clear* svojstvo).