

**Visoka škola za informacijske tehnologije**

---

**Oblikovanje web stranica**  
skripta

Jurica Đurić

---

Zagreb, 2013

## Sadržaj

1	Povijest web dizajna.....	5
1.1	Prvi dizajni.....	6
1.2	Programi za izradu web stranica .....	8
2	Strategija dizajna.....	9
3	Dizajn sučelja.....	10
3.1	Pravila dizajna.....	11
3.1.1	Jednostavan dizajn .....	11
3.1.2	Sadržaj stranice .....	12
3.1.3	Čitljivost teksta .....	12
3.1.4	Jednostavnost korištenja – navigacija.....	15
3.1.5	Podrška za sve preglednike .....	16
3.1.6	Ažuriranost stranice .....	18
3.1.7	Ključne riječi .....	18
3.1.8	Pružanje informacija korisniku .....	18
3.1.9	Razlog za vraćanje na stranicu .....	19
3.1.10	Testiranje.....	19
4	Oblikovanje stranica.....	20
4.1	Dizajn pomoću frame-ova .....	20
4.2	Dizajn pomoću tablica .....	20
5	Multimedija.....	21
5.1	Formati za slike .....	21
5.1.1	Razlike rasterske i vektorske grafike .....	21
6	CSS.....	23
6.1	Povezivanje HTML-a i CSS-a.....	23
6.1.1	Linijski način pisanja stilova.....	23
6.1.2	Pisanje CSS-a unutar oznake <style>.....	23
6.1.3	Pisanje stilova u vanjskoj datoteci .....	24
6.2	Pravila CSS-a .....	26

6.3	Grupiranje selektora .....	27
6.4	Nasljeđivanje svojstva .....	27
6.6	Svojstva za uređivanje teksta .....	29
6.6.1	Boja teksta .....	29
6.6.2	Veličina teksta .....	29
6.6.3	Debljina slova .....	29
6.6.4	Oblik slova .....	30
6.7	Svojstva za uređivanje elemenata .....	30
6.7.1	Pozadinska boja elemenata.....	30
6.7.2	Obrub elemenata .....	30
6.7.3	Margina .....	31
6.7.4	Padding .....	31
6.7.5	Širina i visina elemenata.....	32
6.7.6	Svojstvo overflow .....	32
6.7.7	Postavljanje pozadinske grafike elementa .....	35
6.8	Model kutije (Box model) .....	36
6.9	Klasa i ID selektori.....	37
6.9.1	Klasa.....	37
6.9.2	ID atribut .....	38
6.10	Elementi <div> i <span>.....	39
6.11	Pseudoklase i pseudoelementi .....	40
6.12	Pseudoelementi .....	41
6.13	Selektori atributa .....	42
6.14	Specificity .....	43
6.14.1	Important .....	44
6.15	Razmještaj elemenata .....	45
6.15.1	Izrada stranice koristeći jedan stupac .....	47
6.15.2	Izrada stranice koristeći dva stupca .....	50
6.15.3	Izrada stranice koristeći tri stupca .....	54
6.16	Pozicioniranje elemenata .....	57
6.16.1	Static.....	57
6.16.2	Absolute .....	57

6.16.3	Relative .....	57
6.16.4	Fixed .....	58
6.16.5	z-indeks.....	58
6.16.6	Display .....	61

## 1 Povijest web dizajna

Samo 20-tak godina nakon što je izmišljen, World Wide Web ili www ili web kako se još naziva promijenio je naše živote u svakom pogledu. Promijenio je način rada, način učenja, način življena. Prvobitno web nije bio zamišljen za ono što ga mi danas koristimo tako prije daljnog učenja treba se upoznati sa prošlosti web-a. Web treba razlikovati od interneta. Web je samo jedan od servisa koji se koristi na internetu slično kao i e-mail ili FTP ili bilo koji drugi servis.



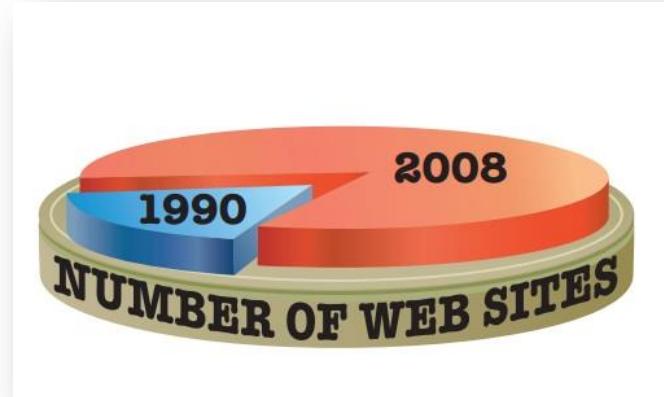
1990. godine u <sup>1</sup>CERN-u engleski inženjer i znanstvenik Tim Berners-Lee i belgijski znanstvenik Robert Cailliau rade na projektu koji je kasnije nazvan World Wide Web ili www ili web te predlažu da se koristi hipertekst za povezivanje i dohvaćanje informacija. 1994. Tim Berners-Lee napušta CERN i osniva World Wide Web Consortium (W3C), međunarodno tijelo koje nadgleda razvoj standarda za Web kako bi stranice izrađene na takav način bile dostupnije korisnicima Interneta. HTML, CSS su neki od standarada koje nadzire W3C. Uz te standarde W3C nadzire i promovira nove grafičke, video i zvučne formate koji se koriste na web-u.



---

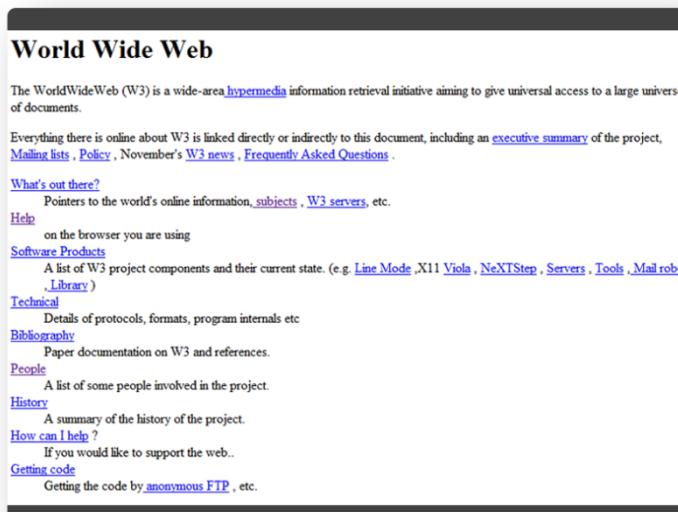
<sup>1</sup> Conseil Européen pour la Recherche Nucléaire

Broj web stranica je vrlo teško točno definirati no većina procjena govori o nekoliko bilijuna. 2008. godine Google je objavio kako je indeksirao jedna trilijun jedinstvenih web adresa. Kako je prošlo tek 20 godina od osnivanja web-a razvoj je stvarno vrlo velik.



## 1.1 Prvi dizajni

Prve web stranice nisu koristile prebogati grafički dio, razloga je bilo nekoliko. Jedan od razloga je bio taj što prve web stranice su se koristile samo za prikaz tekstualnih informacija te samim time potreban za multimedijskim sadržajem nije postojala.



Slika 1. Prva web stranica

Prva web stranica je bila stranica koja je sadržavala samo tekst i nekoliko poveznica na druge stranice. Kasnije većina drugih stranica je izgledala vrlo slično sa dodatkom jednostavne grafike. Daljnjim razvojem postojala je sve veća potreba za uvođenjem raznih multimedijskim sadržajem. Kako u to vrijeme osim nerazvijenosti web-a nije bila razvijena niti infrastruktura



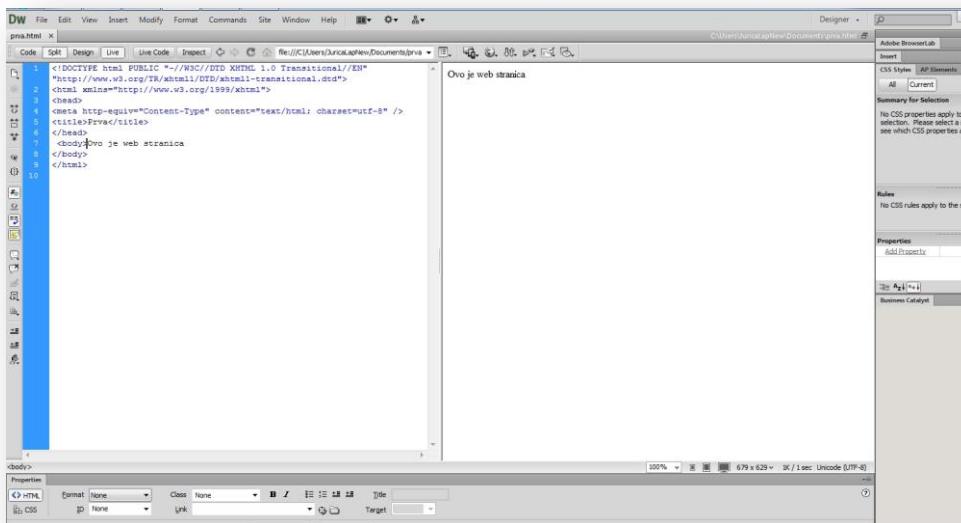
Slika 2. Web stranica Amazon-a



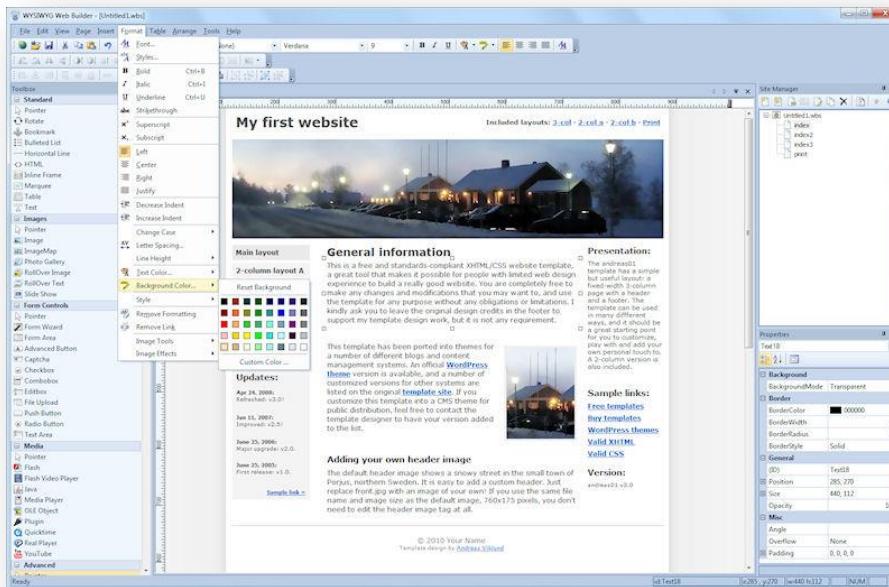
Slika 3. Stranica Google-a iz 1997. godine

## 1.2 Programi za izradu web stranica

Neki od programa za izradu web stranica su Adobe Dreamweaver kojeg je razvila tvrtka Macromedia, no 2005. postaje dio Adobe Creative Suite paketa. DreamWeaver pruža i prikaz dizajna i prikaz kôda te mješoviti prikaz. Pomoću njega moguće je vidjeti kako će stranica izgledati u pregledniku prilikom izrade bez potrebe otvaranja preglednika.



Također postoji niz raznih programa koji omogućavaju izradu i dizajn web stranica pomoću gotovih predložaka. Jedan od takvih programa je **WYSIWYG** (eng. **What You See Is What You Get**) Web Builder koji omogućava vrlo jednostavno dizajniranje web stranica i ljudima koji nemaju prevelikog znanja u području izrade web stranica. Naravno takve programe mogu koristiti i oni koji posjeduju napredna znanja izrade web stranica.



Programi koji omogućavaju brzu izradu web stranica ne znače da su najbolji izbor prilikom

izrade web stranica. Većina njih automatski ubacuje kôdove za neke dijelove stranica koje kasnije mogu usporiti očitavanje stranice ili donijeti probleme prilikom održavanja stranica. Iz tog razloga najbolje je ovisno o potrebama izrade web stranica donijeti pravilnu odluku s kojim programom kreirati web stranicu. Ponekad je puno bolje i kvalitetnije napraviti web stranicu koristeći program tipa Notepad-a iako je u startu potrebno više vremena no na kraju se pokaže da je brže jer sav kôd koji se napiše zna se gdje je i koju svrhu na web stranici ima.

## 2 Strategija dizajna

Web stranica (eng. Web page) je HTML dokument koji omogućava prezentaciju teksta i poveznica, a dostupan je preko svojeg URL adrese. Na web stranici uz tekst moguće je prikazivati i razne multimedejske sadržaje, slike, zvukove, video i drugi sadržaji.

Web sjedište (eng. Web site) je mjesto (direktorij) na računalu (serveru) na kojem se nalazi skup web stranica koje su povezane međusobno preko poveznica te čine jednu cjelinu.

Prije izrade samih web stranica prvo je potrebno isplanirati web sjedište kako kasnije kada se krene u izradu ne bi došlo do nepotrebnih problema. Planiranje se izvodi kroz 4 koraka koja su prikazana sljedećom slikom.



Prilikom izrade web sjedišta prvi korak je definiranje ciljeva u kojem se određuju ciljevi izrade web sjedišta te namjena istog. U drugom koraku razmatra se ciljana publika koja će primarno koristiti web sjedište. Ovisno o ciljanoj publici zasniva se daljnji dizajn stranica jer nije isti dizajn ako želi privući mlađa publiku ili starija, isto tako nije isto ako se radi o web sjedištu neke tvrtke ili ako se radi o web sjedištu koji služi za razmjenu primjerice fotografija. U ovom koraku također je vrlo bitno saznati i tehničke detalje vezano za način spajanja na web sjedište kako bi se znalo u kolikoj mjeri je moguće implementirati multimedijalne sadržaje. Ako ciljana publiku koristi vrlo spore veze za spajanje na internet bilo kakva multimedija na web stranici bi učinila stranicu vrlo teškom za učitavanje i pregled. Kako su danas brzine spajanja na Internet i po nekoliko desetaka puta brže u odnosu na vrijeme kada su nastale prve web stranice multimedijalni sadržaji danas spada u nezaobilazni dio svake web stranice. Internet preglednici su također jedan od mogućih problema te se prilikom ovog koraka mora paziti koji će preglednik koristiti ciljana publiku kako se ne bi koristile tehnologije koje ti preglednici ne podržavaju. Sljedeći korak u razvoju web sjedišta je prikupljanje informacija u kojem se prikupljaju svi potrebni podaci kako bi se bez ikakvih problema u potpunosti kreirati web sjedište. U tom koraku potrebno je prikupiti sve tekstove, sve slike i ostale multimedijalne sadržaje kako bi se u sljedećem koraku moglo pristupiti što prije. Sljedeći korak u razvoju je kreiranje idejnog rješenja web sjedišta. U tom koraku se stvaraju mogući izgledi web sjedišta koji se prezentiraju naručiteljima te ovisno o njihovim zahtjevima ili se prepravlja ili s kreće u finalnu izradu cijelog web sjedišta. Naravno u ovom koraku je vrlo bitno držati se pravila grafike i tipografije kako bi konačno web sjedište bilo funkcionalno i kako bi služilo svojoj svrsi. Također u ovom koraku je bitna i optimizacija konačnih web stranica kako zbog bržeg učitavanja tako i zbog veće mogućnosti izlistavanja prilikom pretraživanja u tražilicama.

### 3 Dizajn sučelja

Do 2007. godine web dizajneri su mogli biti sigurni da se korisnici koji pregledavaju web stranice koriste računalo sa velikim ekranom. Tada su sve stranice trebale biti široke oko 960 piksela što je bio neki standard. Jedini problemi koje su mučili web dizajnere su bili prikazi u različitim internet pretraživačima. Iako se i prije 2007. godine stranicama moglo pristupiti preko mobilnog telefona većina korisnika je za pregledavanje web stranica koristilo računalo. Pojavom pametnih uređaja (iPhone-a i Android operacijskog sustava) sve više korisnika koristi i svoj mobilni uređaj za pregledavanje web stranica. Samim time javlja se novi veliki problem koji web dizajneri moraju riješiti, različite rezolucije ekrana kako samih mobilnih uređaja tako sada velika razlika između ekrana na mobilnim uređajima i ekrana koja koriste stolna računala. Uz mobilne uređaje u zadnje vrijeme na tržištu ima sve više i tablet-a koji također imaju različite rezolucije ekrana.



U počecima izrade web stranica, web dizajn se odnosio samo na statičke HTML (Hypertext Markup Language) stranice koje su koristile veze i grafiku. Danas web stranicu je gotovo nemoguće zamisliti bez dinamičke web stranice koja u sebi ima dijelove koristeći programske jezike kao na primjer .NET, JSP(JavaServer Pages) ili PHP.

### 3.1 Pravila dizajna

Prilikom dizajna trebalo bi paziti na nekoliko pravila koja bi se trebala poštivati zbog uspješnosti web stranice.

1. Jednostavan dizajn
2. Sadržaj stranice
3. Čitljivost teksta
4. Jednostavnost korištenja – navigacija
5. Podrška za sve preglednike
6. Ažuriranost stranice
7. Ključne riječi
8. Pružanje informacija korisniku
9. Razlog za vraćanje na stranicu
10. Testiranje

#### 3.1.1 Jednostavan dizajn

Jednostavan dizajn je jedan prvo pravilo koje bi svaki web dizajner trebao znati. Korisnici ne vole komplikirane i kompleksne web dizajne, posjetitelj dođe na stranicu pogledati ono što ga zanima i što je stranica jednostavnija to je veća vjerojatnost da će se vratiti na istu stranicu. Primjer je recimo Google pretraživač koji ima vrlo jednostavan dizajn koji gotovo svi posjetitelji prihvataju. Posjetitelj koji dođe na Google traži informaciju i želi ju što prije dobiti.



### 3.1.2 Sadržaj stranice

Istraživanjem je dokazano kako ljudi čitaju web stranice u obliku slova F. Što znači da posjetitelji najviše gledaju gornji lijevi dio prema desnoj strani i odozgore po lijevoj strani prema dolje. Samim time dizajnerima je to naputak gdje trebaju staviti najvažnije informacije koje žele kako bi korisnici što prije došli do informacija koje ga zanimaju.



### 3.1.3 Čitljivost teksta

#### 3.1.3.1 Fontovi

Veličinu i odabir fonta treba prilagoditi namjeni web stranice. Postoje dvije veće podijele font-ova (stilova pisma) a to su Serif i Sans-serif. U serif skupinu fontova spadaju oni koji imaju malene ukrasne poteze, zaobljenja na rubovima slova. Zbog te karakteristike najčešće se upotrebljavaju za naslove i tamo gdje treba naglasiti velike dijelove teksta jer kod manjih veličina serif fontovi smanjuju čitljivost zbog ukrasa koji oduzimaju bijeli prostor između slova. To vrijedi samo za web dok za tisk na papiru to nije pravilo. Rezolucija ima najviše

utjecaja na čitljivost teksta pa je tako rezolucija kod web-a 72dpi-a dok se za tisk koristi najmanje 300dpi-a. Kod nižih rezolucija slova postaju manje čitljiva jer pikseli dolaze do izražaja. Primjer serif fonta su: Times, Times New Roman, Courier, Courier New, Georgia, Garamond.



Slika 4. Serif font

Sans-serif fontovi nemaju rubne ukrase na slovima te samim time se povećava čitljivost jer povećava bjelinu između slova. Sans-serifni fontovi su najidealniji za ispisivanje sadržaja stranica te su zbog toga najčešće korišteni na web-u. Primjer sans-serif fonta su: Arial, Tahoma, Verdana, Trebuchet.



Slika 5. Sans serif font

Većina fontova dolazi zajedno sa operacijskim sustavom no može se dogoditi da web dizajner na stranicu stavi neki font koji je sam kreirao ili skinuo sa interneta. Na to se treba obratiti pažnja jer se može dogoditi da krajnji korisnik nema instaliran taj font te će se prikazivati zamjenski font što može narušiti cijeli dizajn stranice. Prilikom izrade web stranica trebaju se koristiti takozvani web safe fontovi koji su zapravo sistemski fontovi koji su sastavni dio svakog operacijskog sustava.

Uz vrste fontova postoje i grupe fontova od kojih su najpoznatije:

- Serif (serifni fontovi)
- Sans-serif (sans serifni fontovi)
- Cursive (ukošeni fontovi)
- Fantasy (dekorativni fontovi)
- Monospace (fontovi jednake širine)

Primjer grupe fontova: font-family: Arial, 'Trebuchet MS', Helvetica;

U navedenom primjeru ako se dogodi da korisnik koji pregledava stranicu nema instaliran prvi font, kao zamjenski koristiti će se drugi a ako njega nema onda će se koristiti treći. Naravno uvijek bi se trebalo paziti na to da obavezno jedan bude sistemski kako bi se na kraju stranica mogla bez problema pregledavati. Ukoliko se dogodi da ne postoji niti jedan od navedenih fontova kao zadnji font može se staviti cijela grupa fontova koja će se tada koristiti kao zamjenski font.

Primjer grupe fontova: font-family: Arial, Helvetica,sans-serif;

Veličinu fonta je bitno prilagoditi namjeni stranice. Naslovi, navigacija te bitne informacije potrebno je naglasiti sa većim fontom dok sami sadržaj stranice može biti manjeg fonta.

Većina današnjih preglednika omogućava povećanje prikaza tako da ako korisnik ne može sa lakoćom čitati može povećati prikaz cijele stranice.

### 3.1.3.2 Boje

Boje su vrlo bitne u web dizajnu te pravilnim odabirom boja web stranica može privući ili odbiti posjetitelja. Većina tvrtki koji ulažu u svoj imidž vode brigu oko pravilnog odabira boja te na kraju boje su njihov simbol prepoznavanja. Na primjer Milka (ljubičasta), Coca-Cola (crvena), McDonald's (žuta- crvena), Facebook (plava-bijela).

Postoje dva sustava boja a to su RGB (**R**ed, **G**reen, **B**lue) i CMYK (**C**yan, **M**agenta, **Y**ellow, **B**lack). RGB sustav boja koriste ekran, fotoaparati, kamere dok CMYK sustav boja koriste pisači. Sustav boja koji se koristi na webu je RGB. Miješanjem te tri boje moguće je dobiti bilo koju nijansu boje od potpuno crne do potpuno bijele. Svaki dio boje ima raspon od 0 do 255 gdje 0 predstavlja minimalni udio te boje dok 255 predstavlja maksimalni udio te boje.

Unutar kôda boja se može definirati na tri načina:

- Ime boje,
- RGB vrijednost boje, te
- Heksadecimalni kôd boje.

Ime boje podrazumijeva korištenje naziva boje koja se želi koristiti recimo „blue“ je primjerice za plavu boju. Na ovaj način moguće je vrlo jednostavno koristiti boje bez potrebe znanja rgb ili heksadecimalne vrijednosti boje. Drugi način korištenja boje je upisivanjem pojedine vrijednosti boja koje želimo, na primjer za potpuno crnu boju treba napisati rgb(0,0,0), za potpuno bijelu (255,255,255) dok primjerice za plavu boju treba napisati rgb(0,0,255). Treći način korištenja boja je pomoću heksadecimalnog kôda koji zapravo predstavlja heksadecimalni prikaz rgb broja. Heksadecimalni kôd sastoji se od znaka # i 6 znakova (brojki i/ili slova). Tih 6 znakova su heksadecimalne vrijednosti – kombinacija osnovnih boja (crvene, zelene i plave). Prva dva heksadecimalna znaka se odnose na crvenu boju, srednja dva na zelenu dok zadnja dva na plavu boju. Heksadecimalno označavanje kreće se od #00 (0) do #FF (255) i predstavljaju kao i kod RGB označavanja udio određene osnovne boje u konačnoj boji.

Kod web dizajna najčešće se koristi heksadecimalno označavanje boja.

Naravno nije potrebno pamtiti heksadecimalni kôd određene boje jer svi programi koji omogućavaju rad sa bojama imaju ugrađene palete boja ili takozvane Color Mixer-e koji nude mogućost odabira određene boje klikom miša te se nakon toga prikazuje i rgb i heksadecimalna vrijednost odabrane boje.

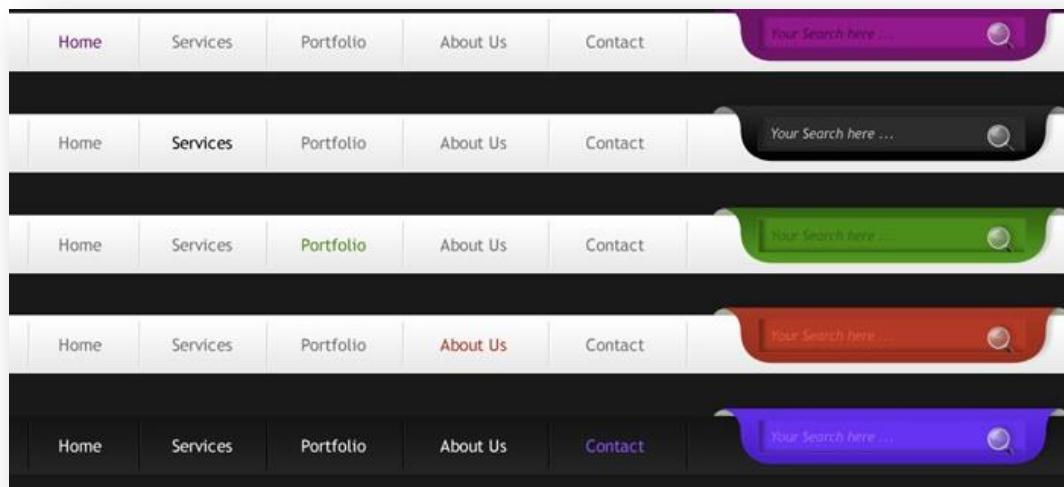
### 3.1.4 Jednostavnost korištenja – navigacija

Svaka web stranica ima poveznice na druge stranice unutar istog web sjedišta ili vanjske poveznice. Prilikom dizajna vrlo je bitno osmisliti pravilnu navigaciju koja će posjetitelju omogućiti jednostavno korištenje i navigaciju između nekoliko stranica. Kod dužih stranica potrebno je postaviti poveznicu koja vodi na vrh stranice kako bi posjetitelj vrlo brzo i jednostavno se vratio na početak stranice. Postoje dvije vrste navigacija a to su:

- Horizontalna, te
- Vertikalna

#### 3.1.4.1 Horizontalna navigacija

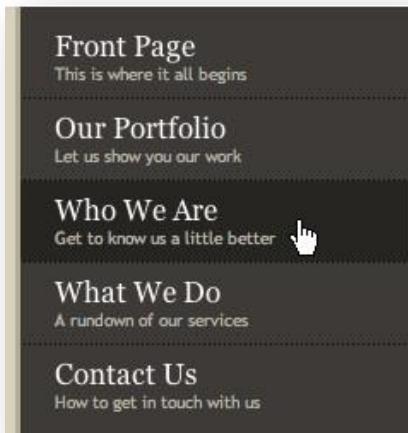
Najčešće se nalazi na vrhu stranice i sadrže glavnu navigaciju. Unutar glavne navigacije može se nalaziti i pomoćna tj. sekundarna koja ima podelemente svakog glavnog elementa. Na primjer ako se radi o stranici neke web trgovine koja prodaje računala i pripadajuću tehniku, jedan podelement u grupi računala može biti stolno računalo a drugi podelement može biti prijenosno računalo. Nakon toga se može sve dalje granati koliko je potrebno i koliko se želi.



Slika 6. Primjer horizontalne navigacije

#### 3.1.4.2 Vertikalna navigacija

Vertikalna navigacija je najstarija navigacija koja se koristi na web stranicama. Može biti pozicionirana na lijevoj ili desnoj strani stranice. Isto kao i kod horizontalne navigacije mogu se koristiti izbornici u nekoliko razina ovisno o namjeni stranice.



Slika 7. Primjer vertikalne navigacije

Uz ove dvije vrste navigacija postoje i navigacije koje su prilagođene namjeni koje ne spadaju u niti jednu vrstu navigacije. Primjerice jedno vrijeme stranica od Nike-a je imala navigaciju koja je izgledala kao daljinski upravljač. Takva vrsta navigacije se najčešće koristi tvrtki koje prodaju razne proizvoda kako bi privukli što više posjetitelja koji će na kraju kupiti proizvod koji se prikazuje na stranici.



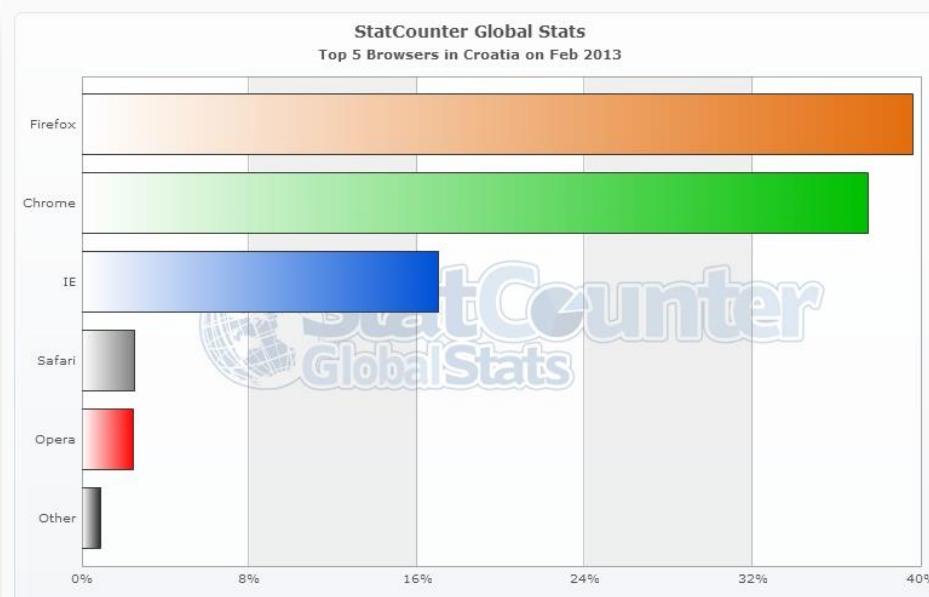
Slika 8. Primjer navigacije koja izgleda kao daljinski upravljač

Koja god navigacija se odabere treba biti konzistentna tokom izrade cijelog web sjedišta, odnosno kroz svaku web stranicu treba biti pozicionirana na istom mjestu i imati isti dizajn kako bi posjetitelj znao gdje se nalazi i lakše se snalaziti na stranici.

### 3.1.5 Podrška za sve preglednike

Početkom razvoja interneta nije bilo previše preglednika koji su se koristili za pregledavanje web stranica. Razvojem tehnologija pojavilo se sve više preglednika pomoću kojih su krajnji

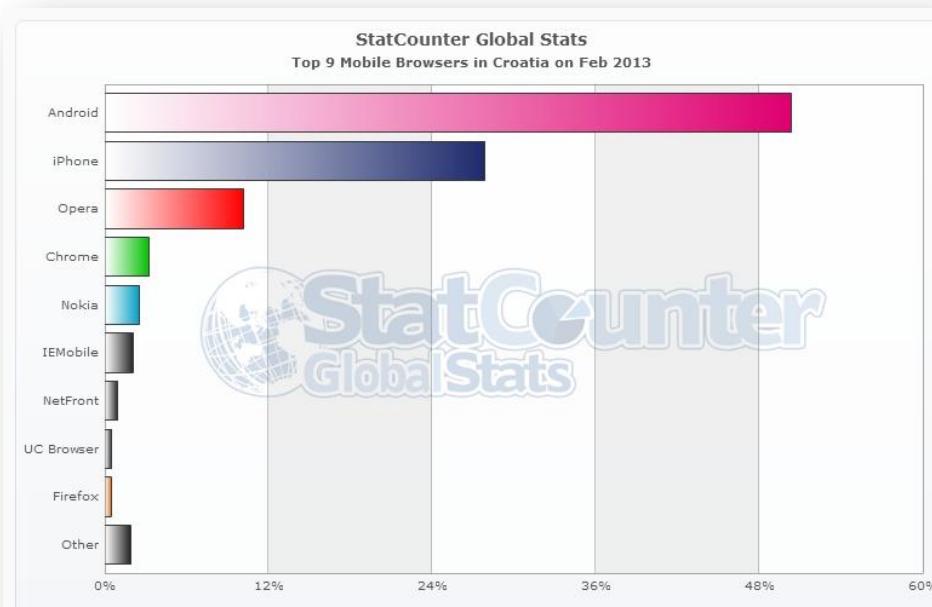
korisnici mogli pregledavati web stranice. Najpopularniji internet preglednici koji se danas koriste su Internet Explorer, Mozilla Firefox, Opera, Google Chrome, Safari i drugi. Internet Explorer dolazi kao dio Windows-a te ga mnogi korisnici koriste kao jedini internet preglednik. Mozilla Firefox i Google Chrome su trenutno najpopularniji internet preglednici.



Slika 9. Korištenje internet preglednika

### 3.1.5.1 Mobilni web – mobile web

Pregledavanje web stranica u zadnje vrijeme je sve popularnije i preko mobilnih uređaja koji koriste razne vrste internet preglednika. Općenito pojam mobilnog weba ne predstavlja podvrstu weba nego znači da se istim web stranicama pristupa preko mobilnih uređaja. Naravno zbog različitih rezolucija ekrana potrebno je prilagoditi stranicu mobilnom uređaju no to je dalje ista web stranica koju je moguće pregledavati i preko bilo kojeg drugog internet preglednika. Ponajprije ono na što se mora paziti prilikom izrade stranica koje će se pregledavati i preko mobilnih uređaja je navigacija koja je zbog manjih ekrana mobilnih uređaja otežana u odnosi na velike ekrane koje koriste računala.



Slika 10. Korištenje internet preglednika na mobilnim uređajima

### 3.1.6 Ažuriranost stranice

Stranicu bi redovito trebalo osvježavati sa novim informacijama. Ako je riječ o stranici koja prodaje neki proizvod potrebno je uvijek na stranici imati aktualne cijene, opise proizvoda koji su u trenutnoj ponudi. Isto tako potrebno je osvježavati i kontakt informacije u slučaju promjene adrese, promjeni broja telefona ili bilo kakvih drugih bitnih informacija koje su bitne krajnjem posjetitelju stranice. Ako se radi o stranici koja izvještava o vijestima i događajima potrebno je imati najnovije vijesti koje su aktualne. Isto tako vrlo je bitno imati arhivu starih vijesti kako bi se u bilo kojem trenutku moglo doći do vijesti i događaja koji su se dogodili u prošlosti. Uvijek je dobro na stranicu staviti i tražilicu koja omogućava pretraživanje informacija unutar stranice ili cijelog web sjedišta ili u nekim slučajevima čak direktno pretraživanje interneta vez otvaranja tražilice.

### 3.1.7 Ključne riječi

Ključne riječi su vrlo važne najviše zbog internetskih tražilica koje upravo pomoću toga pronalaze web stranice. Što je više ključnih riječi korišteno unutar samo web stranice to je veća šansa da će se pojaviti prije u tražilici. Naravno potrebno je odabrat ključne riječi koje imaju doticaja sa sadržajem teksta koji se nalazi na stranici ili ključne riječi koje imaju veze sa proizvodima i uslugama koje se nalaze na stranici.

### 3.1.8 Pružanje informacija korisniku

Cilj web stranice je brzo, efikasno i točno posjetitelju pružiti informaciju zbog koje je i došao. Svaki posjetitelj ne želi trošiti puno vremena tražeći informaciju na stranici i samim time dizajn web stranice mora biti takav da posjetitelj odmah uoči ono što ga zanima.

Sve bitne informacije trebaju biti istaknute na vidljivim mjestima te se treba izbjegavati predugačka ili preširoka stranica na kojoj posjetitelj mora koristiti ili vertikalne ili horizontalne klizače tzv. scroll. Naravno nije uvijek moguće napraviti stranicu gdje se ne moraju koristiti klizači no ako je baš neophodno onda bi bilo bolje kada bi to bili samo vertikalni klizači.

### 3.1.9 Razlog za vraćanje na stranicu

Razlog za vraćanje posjetitelja na stranicu može biti mini blog koji se vodi sa novi aktualnostima, forum preko kojeg posjetitelji komuniciraju međusobno ili sa vlasnikom web stranice. Jedan od mogućih načina privlačenja posjetitelja je mogućnost pretplate na novosti takozvani newsletter. To je vrsta pretplate u kojoj je posjetitelj koji želi primati novosti vezano za novosti na stranici, novosti o proizvodima ili nove usluge ostavlja svoju e-mail adresu te se nakon toga automatski šalju novosti na koje se posjetitelj pretplatio.

### 3.1.10 Testiranje

Svaku stranicu na cijelom web sjedištu treba provjeriti kako se ne bi dogodilo da bilo koja poveznica sa drugom stranicom nije dobra tj. da ne postoje takozvane „broken links“. Uz provjeru valjanosti svih poveznica potrebno je stranice testirati koristeći W3C validatore. W3C validatori mogu testirati HTML i CSS kôd te na taj način dizajneru dati potpuni izvještaj da li je sve napravljeno po standardima koje bi trebalo poštivati.



Slika 11. W3C validator

A screenshot of the W3C Validator interface showing validation results. It includes sections for "Notes and Potential Issues" and "Validation Output". The "Notes and Potential Issues" section contains a note about experimental features and a warning about character encoding. The "Validation Output" section shows 24 errors, with one specific error highlighted: "Line 9, Column 2182: The bgcolor attribute on the body element is obsolete. Use CSS instead." The code snippet for this error is shown as "&lt;body dir="ltr" bgcolor="#ffff"&gt;".

Slika 12. W3C validator - pregled grešaka

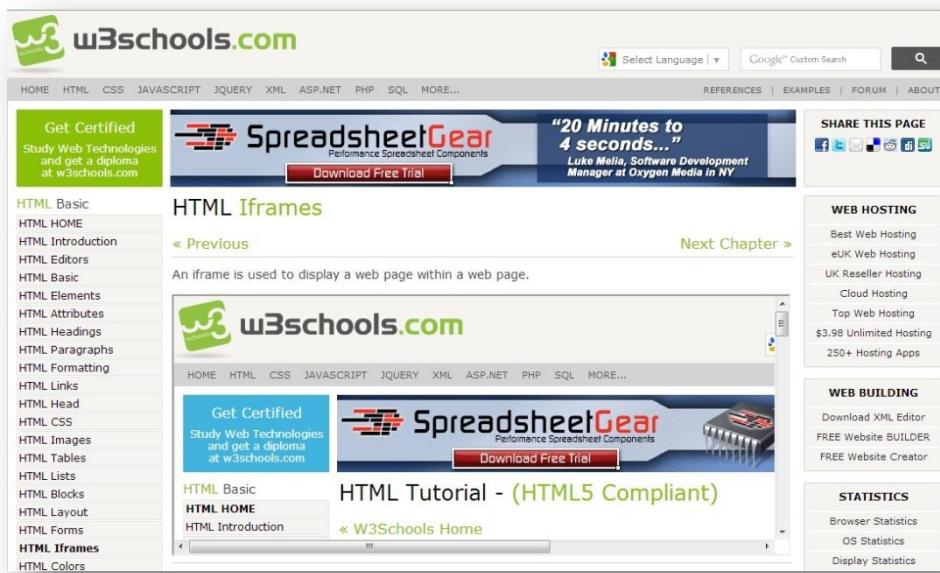
Također svakako je potrebno provjeriti stranice na različitim rezolucijama kako na većim ekranima tako i na mobilnim ekranima te na različitim preglednicima kako se ne bi dogodilo da posjetitelj koji koristi stariji preglednik ne može pregledavati stranicu.

## 4 Oblikovanje stranica

Prilikom oblikovanja web stranica vrlo je bitno znati postaviti pravilni dizajn te način na koji će se taj dizajn i kreirati. Također vrlo je bitno znati postaviti navigaciju unutar stranice i veze prema drugim web stranicama.

### 4.1 Dizajn pomoću frame-ova

Stranice koje koriste okvire su se koristile kada su nastale web stranice i oni se više ne koriste previše. Naprednija verzija okvira su i-frame-ovi oni se mogu koristiti kada se želi ograničiti duljina stranice no i dalje se želi imati relativno dugačka „podstranica“ unutar stranice. To se recimo može koristiti kada se želi unutar stranice staviti nekakva tablica koja može biti vrlo dugačka te kako se povećanjem tablice ne bi povećala cijela stranica.



### 4.2 Dizajn pomoću tablica

Još jedan od starijih načina dizajniranja web stranica koji koristi tablice za pozicioniranje elemenata. Ovaj način oblikovanja stranica može se i danas koristiti kada se radi recimo jednostavna galerija slika u kojoj se žele staviti slike jednakih veličina.



## 5 Multimedija

Od prve pojave web stranica multimedija čini osnovni dio svake web stranice, grafika, zvuk i video su danas nezaobilazni dio svake web stranice. Upravo zato vrlo je bitno znati osnove grafike kako bi se mogla napraviti što kvalitetnija web stranica koja se brzo učitava. Kod izrade web stranica vrlo je bitna kvalitetna kompresija multimedije kako bi se što brže stranice mogle učitavati no naravno bitno je i da kompresija ne bude prevelika kako se ne bi previše izgubila kvaliteta.

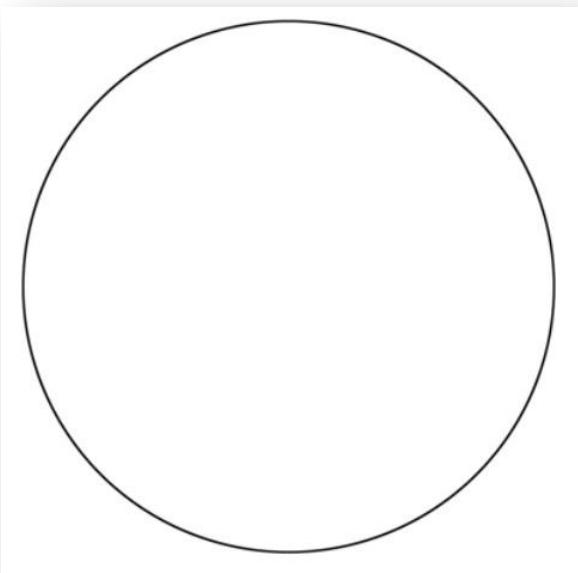
### 5.1 Formati za slike

Postoji nekoliko formata za slike koje svakodnevno koristimo no neki od njih nisu primjereni za web stranice zbog svoje veličine i neoptimiziranosti. Primjer takvog formata je BMP koji je format za nekompresiranu slikovnu datoteku. Neki od primjera kompresiranih slikovnih datoteka su JPG, PNG i GIF.

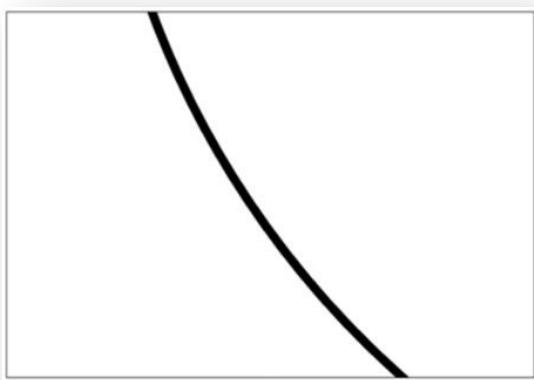
#### 5.1.1 Razlike rasterske i vektorske grafike

Većina poznatih formata za slike kao što su JPG, PNG, GIF su rasterski tipovi grafike i oni se najčešće koriste za web stranice no kako je to rasterska grafika imaju veliku manu.

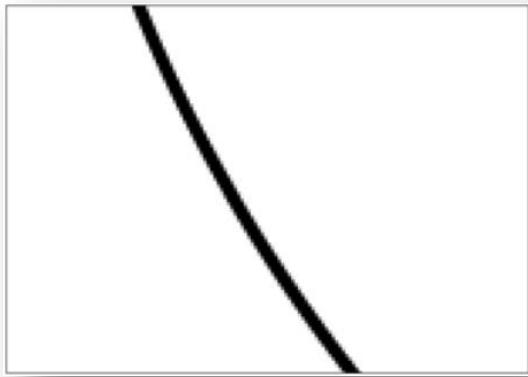
Povećanjem slike gubi se njihova kvaliteta te pri velikom povećanju izgledaju kockasto. Kod vektorske grafike to nije slučaj i one su uvijek iste kvalitete bez obzira na njihovu veličinu recimo SVG format. Sljedeći primjer će to najbolje prikazati, ako se napravi krug u vektorskem i rasterskom formatu i nakon što se poveća samo dio kruga dobiti će se različiti rezultati.



Krug koji je napravljen u vektorskog grafici iako je povećan nije izgubio na kvaliteti prikaza, što znači da bez obzira povećavanje ili smanjivanje kvaliteta ostaje ista što je vrlo bitno kod web stranica gdje se koriste ekran različitih rezolucija.



Krug koji je napravljen u rasterskog grafici kada je povećan kvaliteti prikaza se promijenila, tj. rubovi su postali zamućeni i kada bi se povećalo još više vidjelo bi se kako je crna boja na rubovima postala siva. Iz ovoga se vidi kako u rasterskoj tehnologiji promjena rezolucije mijenja kvalitetu prikaza no u većini slučajeva gubitak kvalitete nije toliko jako vidljiv jer se slike optimiziraju na određenu rezoluciju na kojoj se toliko ne vidi gubitak kvalitete.



## 6 CSS

CSS (*Cascading Style Sheets*) je tehnologija koja služi za dizajniranje HTML dokumenta. CSS je uveden zbog otežanog dizajniranja grafičkih elemenata unutar HTML-a te zbog odvajanja strukture sadržaja i prezentacije.

### 6.1 Povezivanje HTML-a i CSS-a

CSS stilove je moguće povezati sam HTML-om na tri načina:

- Linijski (atribut style unutar svake HTML oznake),
- Unutar `<style>` HTML oznake u `<head>` oznaci HTML-a te
- U vanjskoj datoteci koju povezujemo s HTML dokumentom.

Najčešće upotrebljavani način je odvajanje stilova u zasebnu datoteku jer se na taj način omogućava pristup svim stilovima u više HTML datotekama.

#### 6.1.1 Linijski način pisanja stilova

Linijski način naziva se pisanje CSS-a unutar HTML oznake. Svakoj HTML oznaci se dodaje atribut **style** pomoću kojeg se definiraju pojedine CSS oznake.

Na primjer:

```
<p style="color:blue; font-size:20px">Ovdje će se prikazati tekst plave boje veličine 20 px</p>
```

U ovom primjeru koristi se **style** atribut HTML oznake `<p>` (oznaka paragrafa) unutar kojeg će se prikazati tekst plave boje i veličine slova 20px.

#### 6.1.2 Pisanje CSS-a unutar oznake `<style>`

Pisanjem CSS stilova unutar `<head>` HTML oznake omogućava lakši pregled svih CSS oznaka unutar jedne HTML datoteke jer se sve oznake stavljuju na jedno mjesto unutar `<style>` HTML oznake.

Na primjer:

```
<html>
  <head>
    <title>Naslov</title>
    <style type="text/css">
      h2 {font-size: 12px;}
      p {font-size: 10px;
          color: #222222;}
    </style>
  </head>
  <body>
    <h2>Ovo je tekst h2</h2>
    <p>Tekst paragrafa</p>
  </body>
</html>
```

Primjer korištenja CSS oznaka unutar `<style>` HTML oznake koja omogućava korištenje nekoliko stilova unutar iste `<style>` oznake. Za razliku od linijskog načina pisanja na ovaj način moguće je vrlo lako upravljati izgledom stranice koristeći samo jednu `<style>` oznaku.

### 6.1.3 Pisanje stilova u vanjskoj datoteci

Ovaj način omogućava pisanje stilova koji se mogu koristiti u nekoliko HTML datoteka. Na taj način moguće je napraviti isti dizajn za sve stranice koje se žele koristiti. Također na taj način vrlo je lako promijeniti izgled cijele stranice ili nekoliko stranica promjenom samo jedne CSS oznake koja se nalazi u vanjskoj CSS datoteci. CSS datoteka je obična tekstualna datoteka koja ima nastavak `.css` te ju je moguće uređivati u bilo kojem tekstu editoru koji se koristi na računalu.

```
h1 {  
    font-size:24px;  
    color:yellow;  
}  
  
h2 {  
    font-size:18px;  
    color:#ff0010;  
}  
  
p {  
    font-size:10px;  
    color:#00ee00;  
}
```

```
<html>  
    <head>  
        <title>1. primjer </title>  
        <link href="stil.css" media="all" type="text/css" rel="stylesheet" />  
    </head>  
    <body>  
        <h1>Ovo je naslov h1</h1>  
        <p>Tekst paragrafa koji koristi stilove iz css datoteke</p>  
        <p>Drugi tekst paragrafa koji koristi stilove iz css datoteke</p>  
    </body>  
</html>
```

```
<html>  
    <head>  
        <title>2. primjer </title>  
        <link href="stil.css" media="all" type="text/css" rel="stylesheet" />  
    </head>  
    <body>  
        <h1>Ovo je naslov h1</h1>  
        <h2>Ovo je podnaslov</h2>  
        <p>Tekst paragrafa koji koristi stilove iz css datoteke</p>  
    </body>  
</html>
```

Izgled prvog primjera HTML dokumenta koji je uređen koristeći CSS iz vanjske datoteke.



Izgled drugog primjera HTML dokumenta koji je uređen koristeći CSS iz vanjske datoteke.



## 6.2 Pravila CSS-a

Svako pravilo unutar CSS-a sastoji se od:

- selektora,
- svojstva te
- vrijednosti

Selektor služi za određivanje HTML elementa na koji će se stil primjenjivati (body, p,h1, itd.)

Na primjer ako se želi primijeniti stil na HTML oznaku <p> CSS oznaka će biti **p**.

Svojstvo određuje koje se svojstvo određenog selektora želi urediti. Svojstva su **color** za boju, **font-size** za veličinu slova, **border** za obrub.

Vrijednost predstavlja vrijednost određenog svojstva kao što za svojstvo **color** vrijednost može biti **yellow** ili za svojstvo **font-size** može biti vrijednost **18px**.

Deklaracija stila sastoji se od svojstva i vrijednosti na primjer:

```
color: yellow;
```

Svojstvo i vrijednost odvajaju se dvotočkom, dok se deklaracije odvajaju točkom-zarez. Sve deklaracije koje pripadaju nekom selektoru grupiraju se vitičastim zagradama.

Na primjer ako se želi oblikovati `<p>` HTML element kojem se želi postaviti boja teksta u plavu i postaviti veličinu slova na 20px to se može napraviti ovako:

```
h1 {  
    color: blue;  
    font-size: 20px;  
}
```

### 6.3 Grupiranje selektora

Ponekad postoji potreba da se više različitih HTML elemenata urede sa istim stilom. Kako se ne bi ponavljao kôd za isti stil nekoliko puta, postoji način da se svi potrebni elementi grupiraju te se nakon toga uredi ta grupa kao jedan stil. Selektori se grupiraju tako da se između njih stavi zarez.

```
h1, h2 {  
    color: #222222;  
    font-size: 14px;  
}
```



Na ovaj način je moguće grupirati neograničen broj selektora što smanjiti veličinu css datoteke u kojoj se nalaze stilovi te na taj način ubrzati učitavanje stranice.

### 6.4 Nasljeđivanje svojstva

Grupiranje selektora podrazumijeva korištenje nekoliko identičnih vrijednosti svojstava za sve grupirane selektore. Ponekad se može dogoditi da određeni selektori imaju vrijednosti nekih svojstava te se tada koristi nasljeđivanje svojstava. Na primjer:

```
h1, h2 {  
    color: #222222;  
}  
h2 {  
    font-size: 14px;  
}
```

Selektor h2 u primjeru se pojavljuje na dva mesta. Prvi puta je grupiran sa selektorem h1 i definirano im je zajedničko svojstvo boja. Drugi puta selektor h2 se pojavljuje kako bi se definiralo svojstvo veličine slova pri čemu boja teksta ostaje ista. Ovim primjerom se vidi kako je selektor h2 naslijedio boju teksta iz prethodne deklaracije. U slučaju da se neka svojstva preklapaju veći prioritet ima svojstvo koje je zadnje napisano.

```
h1, h2 {  
    color: #222222;  
    font-size: 24px;  
}  
h2 {  
    font-size: 14px;  
}
```



Grupiranje selektora zajedno sa nasljeđivanjem svojstava su vrlo bitna za optimiziranje veličine css datoteke. Grupiranje se obično radi nakon što se deklariraju sva svojstva koja su potrebna za dizajn stranica.

## 6.6 Svojstva za uređivanje teksta

### 6.6.1 Boja teksta

Za određivanje boje teksta unutar HTML elementa može se koristiti svojstvo **color**.

Vrijednost svojstva može se postaviti na neki nekoliko načina:

- Imenom boje (blue, red, yellow),
- RGB vrijednošću boje (rgb(255,200,200)) te
- Heksadecimalnim brojem (#00ff00).

Najčešći način postavljanja boje je korištenjem heksadecimalnim brojem.

```
color: blue;  
color:rgb(0,0,255);  
color: #0000ff;  
color: #00ff00;
```

### 6.6.2 Veličina teksta

Za promjenu veličine teksta koristi se svojstvo **font-size**. Svojstvo se može postaviti na nekoliko načina od kojih je najpopularniji unošenje absolutne vrijednosti u pikselima (px).

```
font-size: 12px;  
font-size: x-small;  
font-size: 200%;  
font-size: 2em;
```

W3C preporuča korištenje ili % ili em za veličinu slova jer se na taj način smanjuje mogućnost problema kod različitih rezolucija.

### 6.6.3 Debljina slova

Svojstvo za debljinu slova je **font-weight** a vrijednosti svojstva mogu biti sljedeće:

- normal,
- bold,
- bolder te,
- lighter.

Umjesto tih vrijednosti moguće je napisati i brojčanu vrijednost (100, 200, 300, 400, 500, 600, 700) s tim da 400 predstavlja normalnu debljinu slova dok 700 predstavlja bold debljinu slova.

```
font-weight: normal;  
font-weight: 400;
```

#### 6.6.4 Oblik slova

Oblik slova je također dio koji se može uređivati unutar svojstava fonta. U oblike slova spadaju razne grupe fontova ili **font-family**. Postoje dvije osnovne verzije oblika slova a to su serif i sans-serif. Slova sa serifima koriste ukrase na rubovima slova te nisu toliko čitljiva kao što su slova bez serifa.

```
font-weight: arial, verdana, sans-serif;  
font-family: courier, serif;
```

### 6.7 Svojstva za uređivanje elemenata

#### 6.7.1 Pozadinska boja elemenata

Pozadinske boje elemenata se postavljaju na vrlo sličan način kao što se postavlja boja slova. Boja se može postaviti koristeći nekoliko načina:

- imenom boje,
- RGB vrijednosti boje te,
- Heksadecimalnim brojem boje.

```
background-color: blue;  
background-color:rgb(0,0,255);  
background-color: #0000ff;  
background-color: #00f;
```

Kao što je vidljivo iz prethodnih primjera imena svojstava u CSS-u su riječi engleskog jezika. Kada se naziv svojstva sastoji od dvije riječi međusobno se spajaju koristeći znak '-'.

#### 6.7.2 Obrub elemenata

Svakom HTML elementu se može postaviti obrub svojstvom **border**. Vrijednost svojstva sastoji se od nekoliko dijelova:

- debljine obruba (najčešće u pikselima),
- vrsta obruba (puna linija, točkasta linija, itd.) te
- boja obruba.

```
border: 2px solid #252525;
```

Ovo je naslov h1

Ovo je podnaslov h2

Kako je svaki HTML element pravokutnog oblika moguće je postaviti obrub na samo jednu stranicu pravokutnika (lijevu, desnu, gornju ili donju).

```
border-left: 2px solid #252525;  
border-right: 2px solid #252525;
```

Ovo je naslov h1  
Ovo je podnaslov h2

### 6.7.3 Margina

Razmak između dva HTML elementa naziva se margina (**margin**). Kao i kod obruba margini se mogu postaviti na svakoj stranici posebno. Također moguće je staviti marginu za sve stranice samo sa jednom vrijednosti.

```
margin: 15px;
```

Ovo je naslov h1  
Ovo je podnaslov h2

Pojedinačne margini je moguće postaviti kao i obrube koristeći sljedeće:

- margin-top (gornja margina),
- margin-left (lijeva margina),
- margin-right (desna margina) i
- margin-bottom (donja margina).

Sve margini je moguće postaviti unutar jedne linije koristeći jednu oznaku margin. Prvi broj se odnosi na gornju marginu, drugi na desnu, treći na donju te zadnji na lijevu marginu.

```
margin: 15px 10px 10px 15px;
```

### 6.7.4 Padding

Padding je svojstvo koje određuje udaljenost sadržaja HTML elemenata od obruba. Način na koji se deklarira je isti kao i kod margina.

```
padding: 15px;  
padding-left: 10px;  
padding: 15px 10px 15px 10px;
```

Padding je svojstvo koje se često koristi kako bi se tekst odvojio od obruba i na taj način olakšao čitljivosti teksta.

```
Padding
```

### 6.7.5 Širina i visina elemenata

Prepostavljena vrijednost širine većine HTML elemenata je 100% širine <body> elementa, točnije širina prozora preglednika. Za podešavanje širine koristi se svojstvo **width**, dok se za visinu koristi svojstvo **height**. Prepostavljena visina HTML elemenata je **auto**, odnosno povećava se proporcionalno količini sadržaja.

```
width: 400px;  
width: 50%  
  
height: 200px;  
height:10%;
```

## Naslov h1

Paragraf visine 100px i širine 250px

Paragraf visine 100px i širine 250px

### 6.7.6 Svojstvo overflow

Svojstvo **overflow** definira na koji način će se ponašati element ako se unutar njega nalazi više teksta nešto što je visinom zadano. Ukoliko je postavljena visina elementa te sadržaji

koji se nalazi unutar tog elementa prelazi zadanu visinu koristeći ***overflow*** svojstvo ***auto***, automatski će se pojaviti klizna traka (scrollbar) koji će omogućiti pregled cijelog sadržaja tog elementa. Ako visina nije postavljena elementu unutar kojeg se nalazi sadržaj će se automatski povećati kako bi se mogao vidjeti cijeli sadržaj. Predefinirana vrijednost ***overflow*** svojstva je ***visible***.

#### **6.7.6.1 Overflow: visible**

```
height: 100px;  
overflow: visible;
```

### **Overflow svojstvo: visible**

Etiam et urna eu ligula fermentum pulvinar. Quisque quis justo lorem. Vestibulum hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam quis nisi sit amet nibh malesuada sagittis. Pellentesque non mauris ut arcu suscipit hendrerit imperdiet porta quam. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla facilisi. Pellentesque nec elit dolor, nec mollis urna. Quisque volutpat, mi sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Maecenas ut nisi non metus venenatis pharetra. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum aliquet, neque vel tempor vulputate, tortor ligula auctor justo, in scelerisque dolor ante vel tortor.

Kao što je vidljivo ako se postavi vrijednost na ***visible*** to znači da će se sadržaj prikazivati i izvan zadanih veličina.

### 6.7.6.2 *Overflow:auto*

```
height: 100px;  
overflow: auto;
```

# Naslov h1

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Cras mollis,  
elit rhoncus pretium adipiscing, turpis  
neque pretium erat, et aliquam nibh  
augue sed lorem. Lorem ipsum dolor  
sit amet, consectetur adipiscing elit.

```
height: auto;  
overflow: auto;
```

# Naslov h1

*Etiam* non possumus. *Quod* si quis dicit: *Non* possumus, *quoniam* non possumus. *Quod* si quis dicit: *Possumus*, *quoniam* possumus. *Quod* si quis dicit: *Possumus*, *quoniam* non possumus. *Quod* si quis dicit: *Possumus*, *quoniam* possumus. *Quod* si quis dicit: *Possumus*, *quoniam* non possumus. *Quod* si quis dicit: *Possumus*, *quoniam* possumus.

### **6.7.6.3 Overflow: hidden**

**Hidden** vrijednošću svojstva **overflow** sadržaj koji prelazi izvan granica elementa neće se prikazati tj. neće biti vidljiv sadržaj koji je izvan dimenzija elementa.

```
height: 100px;  
overflow: hidden;
```

### **Overflow svojstvo: hidden**

**Etiam** et **urna** eu ligula fermentum pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam quis nisl sit amet nibh

#### **6.7.6.4 Overflow:scroll**

Koristeći ***scroll*** vrijednost svojstva ***overflow*** pojavljuju se i vertikale i horizontalne pomicne trake (scroll) i kada je potrebno i kada nije. To znači da bez obzira na veličinu sadržaja i količinu teksta koji se nalazi unutar elementa trake će biti vidljive.

```
height: 100px;  
overflow: scroll;
```

## Overflow svojstvo: hidden

```
 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam  
 et urna eu ligula fermentum pulvinar. Quisque quis justo lorem.  
 Suspendisse porttitor lobortis erat ut posuere. Vestibulum  
 hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed  
 < >
```

### 6.7.7 Postavljanje pozadinske grafike elementa

Za postavljanje pozadinske grafike elementu koristi se svojstvo ***background-image***. Kao vrijednost svojstva potrebno je postaviti relativnu ili apsolutnu putanju do željene grafike. Postoji i opcija ponavljanja grafike, postavljanje pozicije te postavljanje veličine. Kod postavljanja pozadine ovo svojstvo se postavlja na `<body>` element.

```
background-image: url(slika1.jpg);  
background: url(pozadina.jpg) repeat-x right;
```

#### Naslov h1

```
 Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit. Cras mollis, elit rhoncus  
 pretium adipiscing, turpis neque pretium  
 erat, et aliquam nibh augue sed lorem.  
 Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit. Nulla pharetra interdum  
 nisi id adipiscing. Duis commodo feugiat  
 nisi quis ullamcorper. Nunc vel lorem sed  
 est porta lacinia. Sed vel nisl vitae lectus  
 posuere semper. Suspendisse laoreet,  
 turpis interdum cursus vestibulum, arcu  
 augue congue sapien, non mattis ligula  
 quam non nibh.
```

VsiteVsiteVsiteVsiteVsiteVsiteVsiteVsiteVsite

```
background: url(pozadina.jpg) no-repeat right;
```

## Naslov h1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras mollis, elit rhoncus pretium adipiscing, turpis neque pretium erat, et aliquam nibh augue sed lorem. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla pharetra interdum nisi id adipiscing. Duis commodo feugiat nisi quis ullamcorper. Nunc vel lorem sed est porta lacinia. Sed vel nisl vitae lectus posuere semper. Suspendisse laoreet, turpis interdum cursus vestibulum, arcu augue congue sapien, non mattis ligula quam non nibh.

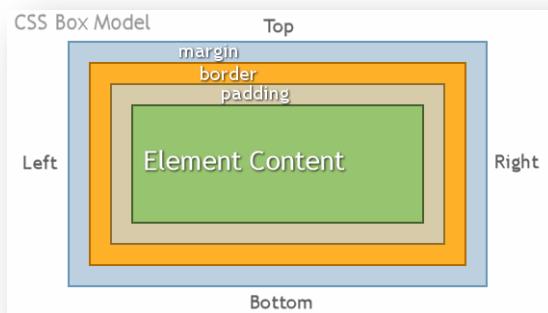


## 6.8 Model kutije (Box model)

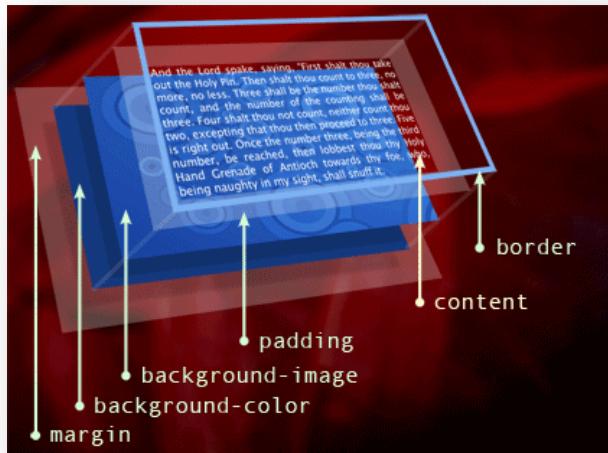
Svi HTML elementi mogu se smatrati "kutijama". CSS "box model" (model kutije) je u biti okvir koji uokvirava HTML elemenata, a sastoji se od:

- sadržaja - sadržaj HTML elementa (text, slika, drugi elementi),
- margina - prostor između ruba HTML elementa i ostalih elemenata,
- bordera - obrub HTML elementa te
- paddinga - prostor između sadržaja i obruba).

CSS "box model" (model kutije) omogućava postavljanje obruba oko elemenata, vanjske razmake između različitih elemenata i razmake između obruba elemenata i sadržaja elemenata. Razumijevanje box modela ima uvelike veze sa layout-om stranica



Uz sve navedeno u modelu kutije nalazi se i pozadinska boja i pozadinska slika.



## 6.9 Klasa i ID selektori

U radu sa HTML-om dokumenata često nije dovoljno upotrebljavati samo selektori koji su uključeni u HTML. U tim slučajevima potrebno je definirati vlastite stilove za jedan HTML element. Klase su ekstenzije selektora koje omogućavaju da se za jedan selektor koristi više stilova.

### 6.9.1 Klasa

Klase nekom selektoru se pridjeljuje tako što se iza imena selektora stavi točka i naziv klase

```
p.pocetak {color:#252525;}
```

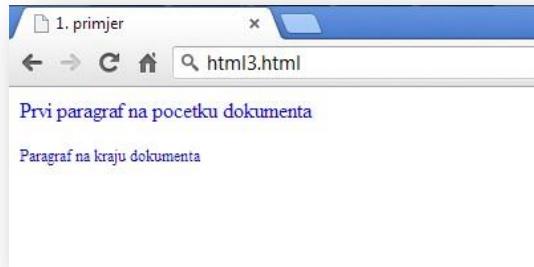
Nakon kreiranja klase unutar CSS-a potrebno je tu klasu i pozvati unutar HTML dokumenta.

```
<p class="pocetak">Prvi paragraf</p>
```

Na ovaj način je moguće definirati poseban stil za svaki HTML element unutar dokumenta.

```
p {color: #252525; font-size: 14px;}  
p.pocetak {color: #0000ff; font-size: 16px;}  
p.kraj {color: #0000dd; font-size: 12px;}
```

```
<body>  
    <p class="pocetak">Prvi paragraf na pocetku dokumenta</p>  
    <p class="kraj">Paragraf na kraju dokumenta</p>  
</body>
```



Klase nužno se ne moraju pisati kao ekstenzija nekog selektora već mogu biti samostalne.

```
.tekst {  
    color:#252525;  
    font-size: 16px;  
    padding-left: 50px;  
}
```

```
<body>  
    <p class="pocetak">Prvi paragraf na pocetku dokumenta</p>  
    <p class="tekst">Tekst koji se nalazi unutar HTML dokumenta. </p>  
    <p class="kraj">Paragraf na kraju dokumenta</p>  
</body>
```



### 6.9.2 ID atribut

Uz klase vrlo često se koriste i ID atributi koji su jedinstveni identifikatori elemenata. U CSS-u se pišu tako što se ispred njih stavi znak **#**. Jedan ID atribut može se koristiti samo za jedan element za razliku od klase koja se može koristiti na više elemenata.

```
#prviparagraf {background-color:#999999; color:yellow;}
```

```
<p id="prviparagraf">Prvi paragraf</p>
```

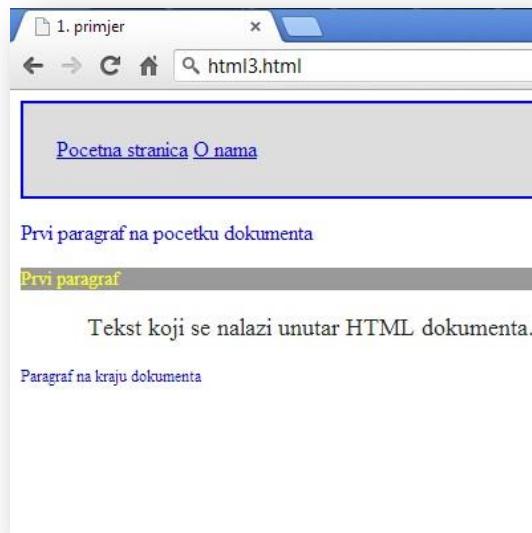


## 6.10 Elementi <div> i <span>

Prilikom uređivanja izgleda HTML dokumenata potrebno je grupirati više elemenata u logičke cjeline kao što su na primjer zaglavje gdje se nalazi logo i navigacija. Takvo grupiranje moguće je napraviti koristeći **<div>** element. Koristeći taj element moguće je postaviti ista svojstva za niz elemenata koji se nalaze unutar pojedine cjeline.

```
<div id="zaglavlje">
    <a href="index.html">Pocetna stranica</a>
    <a href="o_nama.html">O nama</a>
</div>
```

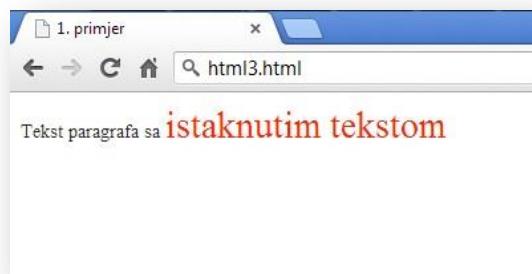
```
#zaglavlje {
    background-color: #dddddd;
    padding: 25px;
    border: 2px solid #0000fe;
}
```



Element **<span>** se također može koristiti za odvajanje logičkih cjelina ali za razliku od div elementa koji stvara blokove, span je linijski element.

```
<p>Tekst paragrafa sa <span class="istaknuto"> istaknutim tekstrom </span></p>
```

```
.istaknuto {  
    color: #ff2500;  
    font-size: 28px;  
}
```



## 6.11 Pseudoklase i pseudoelementi

Korištenje pseudoklasa je uvedeno verzijom CSS-a 2.1. Time je omogućena veća fleksibilnost pri uređivanju elemenata. Pomoću njih je moguće promijeniti izgled prve linije unutar teksta ili postavljanja prvog slova velikog unutar odlomka (drop cap).

Pseudoklasama moguće je selektirati element ili stanje elemenata koji se nalaze u html datoteci. Tako je recimo moguće urediti izgled veze koji se može mijenjati ovisno na položaj

pokazivača miša. Neke od pseudoklasa koje se mogu koristiti u vezama su:

- a:link {color:#FF0000;}
- a:visited {color:#00FF00;}
- a:hover {color:#FF00FF;}
- a:active {color:#0000FF;}

Koristeći pseudoklase moguće je u potpunosti promijeniti izgled veza ovisno o njihovim stanjima.

- **:link** – se koristi za prikazivanje veza koje još nisu posjećene tj. korisnik nije nikad koristio tu vezu.
- **:visited** – se koristi za prikazivanje veza koje je korisnik posjetio.
- **:hover** – se koristi za prikazivanje veza u trenutku kada se pokazivačem miša prolazi preko veze.
- **:active** – se koristi za prikazivanje veza koje su aktivne veze.

Postoji još nekoliko pseudoklasa od kojih je zanimljiva ***lang()*** koja se može koristiti ako se želi napraviti više jezična stranica.

```
:lang(en) {color: #880088;}  
:lang(hr) {color: #550055;}  
<body lang="hr">
```

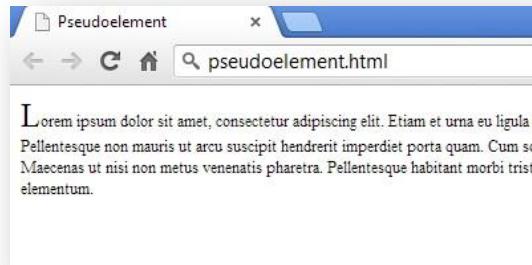
## 6.12 Pseudoelementi

Pseudoelementi su dijelovi html dokumenta koji nisu određeni elementom ali su njegovi dijelovi. Zbog toga pseudoelementi moraju stajati uz selektor elementa. Neki od pseudoelemenata su **:first-line** i **:first-letter**. Pseudoelement **:first-line** selektira prvi red teksta unutar nekog elementa, dok će pseudoelement **:first-letter** selektirati će prvo slovo unutar teksta elementa.

```
body {  
    font-size: 12px;  
}  
p:first-line {  
    font-size: 16px;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et urna eu ligula fermentum pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam quis nisi sit amet nibh malesuada sagittis. Nulla facilisi. Pharetra id odio. Maecenas ut nisi non metus venenatis pharetra. Pharetra id odio. Vestibulum aliquet, neque vel tempor vulputate, tortor ligula auctor justo, in scelerisque volutpat, mi sed sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Maecenas ut nisi non metus venenatis pharetra. Pharetra id odio. Vestibulum aliquet, neque vel tempor vulputate, tortor ligula auctor justo, in scelerisque volutpat, mi sed sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Morbi ut enim justo, a semper nibh. Cras eget orci eu massa adipiscing elementum.

```
body {  
    font-size: 12px;  
}  
p:first-letter {  
    font-size: 26px;  
}
```



### 6.13 Selektori atributa

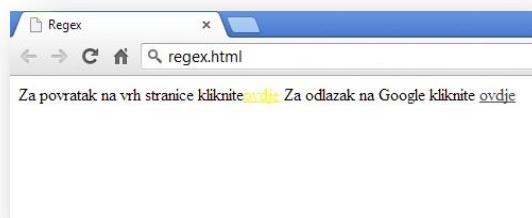
Od verzije CSS2 omogućeno je selektiranje nekog elementa s obzirom na njegove atribute. S time je omogućeno uređivanje istih elementa sa različitim atributima. Selektori atributa označavaju se uglatim zagradama [ i ] unutar kojih ide ime i vrijednost atributa.

```
a [title="Naslov"] {color: yellow;}
```

U navedenom primjeru selektirani su svi elementi koji imaju atribut title riječ naslov, boja tih elemenata će biti žuta.

Koristeći se sintaksom regularnih izraza moguće je selektirati i posebne slučajevе vrijednosti atributa (na primjer atributi koji sadrže ili počinju nekim tekstrom)

```
a {color: yellow;}  
a[href^="http:"] {color: #00ffff;} /*boja vanjskih veza*/
```



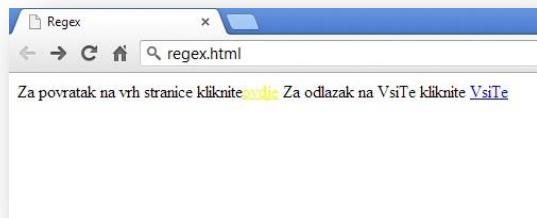
Oznaka ^ (kapica) ispred znaka jednakosti označava tekst na početku vrijednosti atributa

mora odgovarati tekstu u navodnicima selektora.

[atribut~=vrijednost] odgovara elementima koji u vrijednosti atributa imaju više vrijednosti od kojih je jedna vrijednost jednaka „vrijednost“.

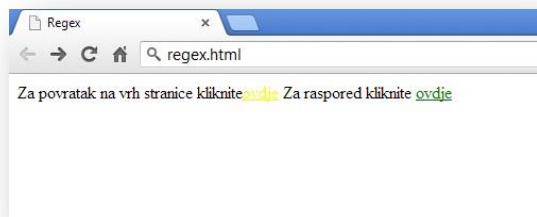
```
a[rel~="tehnologije"] {color: blue;}
```

Ovaj primjer će selektirati sve veze koje u svom rel atributu imaju riječ „tehnologije“.



Ukoliko se želi selektirati element čija vrijednost završava sa određenom vrijednošću koristi se znak \$.

```
a[href$=".pdf"] {color: green;}
```



Ovim primjerom selektira se veza koja vodi na pdf datoteku.

## 6.14 Specificity

Specificity je mehanizam unutar CSS-a koji služi za lakše rješavanje mogućih problema u slučaju nasljeđivanja. Koristeći specificity moguće je odrediti način nasljeđivanja svojstava.

Kao što je prikazano na slici najmanju vrijednost imaju elementi, nakon toga idu klase, pseudoklase i atributi, nakon toga ID, te najveću vrijednost imaju inline atributi unutar <style> oznake. Vrijednosti se računaju kao kod pretvorbe binarnih brojeva u dekadske, skroz desno su brojevi manje vrijednosti dok su sa lijeve strane brojevi veće vrijednosti. Na temelju izračuna konačne vrijednosti postavljaju se CSS pravila.

```
p { color: #fff; /*0,0,0,1*/
.intro { color: #98c7d4; /*0,0,1,0*/
#header { color: #444245; /*0,1,0,0*/
<h1 style="color:#000;">Naslov</h1> /*1,0,0,0*/
```

```
#content p {
    color: blue;
}
p {
    color: lightblue;
}
#content .intro {
    color: yellow;
}
```

```
<div id="content">
    Sadržaj
    <p class="intro">Tekst unutar klase</p>
    <p>Tekst unutar p elementa</p>
</div>
```

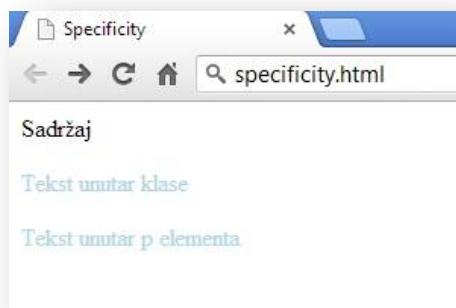


### 6.14.1 Important

Svojstvo ***Important*** može promijeniti redoslijed nasljeđivanja unutar CSS pravila. Ponekad je potrebno koristiti to svojstvo kako bi bili sigurni da će se neko svojstvo sigurno postaviti no nije ga preporučljivo koristiti uvijek. Kada se koristi mora se posebno paziti da li se zaista želi postaviti ili se problem koji se želi izbjegći korištenjem svojstva important može drugačije riješiti.

```
#content p {
    color: blue;
}
p {
    color: lightblue !important;
}
#content .intro {
    color: yellow;
}
```

```
<div id="content">
    Sadržaj
    <p class="intro">Tekst unutar klase</p>
    <p>Tekst unutar p elementa</p>
</div>
```



## 6.15 Razmještaj elemenata

Prije pojave CSS-a postojala su dva načina razmještaja elemenata. Koristeći framesetove te koristeći tablice. Oba načina su imali velike probleme prilikom pregledavanja stranica na ekranima različitih rezolucija. Uz to niti jedna od ta dva načina nije bio u potpunosti prihvatljiv jer su koristili previše internetskog prostora što je uzrokovalo prevelike promete koji su u to vrijeme bili vrlo ograničeni zbog brzina veza sa internetom. Pozicioniranje elemenata može se podijelit u nekoliko grupa. Jedna podjela je prema širini cjelokupne stranice:

- stranice apsolutne širine (širina je točno zadana nekom mjernom jedinicom npr. pikselima) te
- stranice relativne širine (širina ovisi o prozoru preglednika ili ekrana računala).

Druga podjela je prema broju stupaca u kojima se nalazi sadržaj stranice pa tako postoje stranice koje imaju:

- jedan stupac,
- dva stupca,
- tri stupca te
- više stupaca.

Kod izrade web stranica vrlo je bitno pozicionirati sadržaj u sredinu stranice odnosno ekrana. Postoji nekoliko načina za to no jedan od jednostavnijih je postaviti isti razmak od lijevog i desnog ruba stranice do sadržaja stranice. Za pozicioniranje stranica se koriste **<div>** HTML elementi koji uvelike olakšavaju uređivanje cijelih web stranica.

```
<h1>Centriranje stranice</h1>
<div id="wrapper">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et urna eu ligula fermentum
    pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum
    hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam
    quis nisl sit amet nibh malesuada sagittis. Pellentesque non mauris ut arcu suscipit hendrerit
    imperdiet porta quam. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur
    ridiculus mus. Nulla facilisi. Pellentesque nec elit dolor, nec mollis urna. Quisque volutpat, mi
    sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Maecenas
    ut nisi non metus venenatis pharetra. Pellentesque habitant morbi tristique senectus et netus et
    malesuada fames ac turpis egestas. Vestibulum aliquet, neque vel tempor vulputate, tortor ligula
    auctor justo, in scelerisque dolor ante vel tortor.
</div>
```

```
#wrapper {
    background-color: #eeeeee;
}
```

### Centriranje stranice

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et urna eu ligula fermentum pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam quis nisl sit amet nibh malesuada sagittis. Pellentesque non mauris ut arcu suscipit hendrerit imperdiet porta quam. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla facilisi. Pellentesque nec elit dolor, nec mollis urna. Quisque volutpat, mi sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Maecenas ut nisi non metus venenatis pharetra. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum aliquet, neque vel tempor vulputate, tortor ligula auctor justo, in scelerisque dolor ante vel tortor.

Ovim primjerom je vidljivo da je stranica već centrirana, no razlog tomu je div element koji kao i bilo koji drugi element uvijek zauzima 100% širine ako se ne navede proizvoljna širina elementa.

```
#wrapper {  
    background-color: #eeeeee;  
    width: 400px;  
}
```

Ukoliko se postavi širina div elementa na neku vrijednost stranica više nije centrirana te je prema automatskim postavkama pomaknuta prema lijevom rubu ekrana.

## Centriranje stranice

Etiam et una uia fermentum pulvris. Quisque quis justo lorem. Suspenside portori loboris erat ut posse. Vestibulum hender portu orci ut mattis. Fucus eras, vehicula sed pharidra, et tristis a et. Aliquam quis nisi sit at nimb. maledusa sagittis. Pharetra non maurus ut arcu suscipit hender imperdict porta quam. Cum socius natque penitus et magis dicit partiment montes, nescire ridiculus mus. Nulla facili. Pellenquet nec elit dolor, nec mollis ura. Quisque volupat, nisi sodales fermentum blandit, leo quam convallis parus, sed serlecurus arcu idio. Macenesas ut nisi non metus venenatis pharetra. Pharetales habent morbi tristique senectus et netus et maledissa famae ut tigrids egates. Vestibulum aliquat, neque vel tempor vulputate, tortor ligula auctor pasto, in sclerisque dolor ante vel tortor.

Ako se sadržaj želi postaviti u sredinu stranice potrebno je podesiti margine. Ukoliko se margine postane na vrijednost **auto** time se postiže automatsko centriranje sadržaja.

```
#wrapper {  
    background-color: #eeeeee;  
    width: 400px;  
    margin: auto;  
}
```

## Centriranje stranice

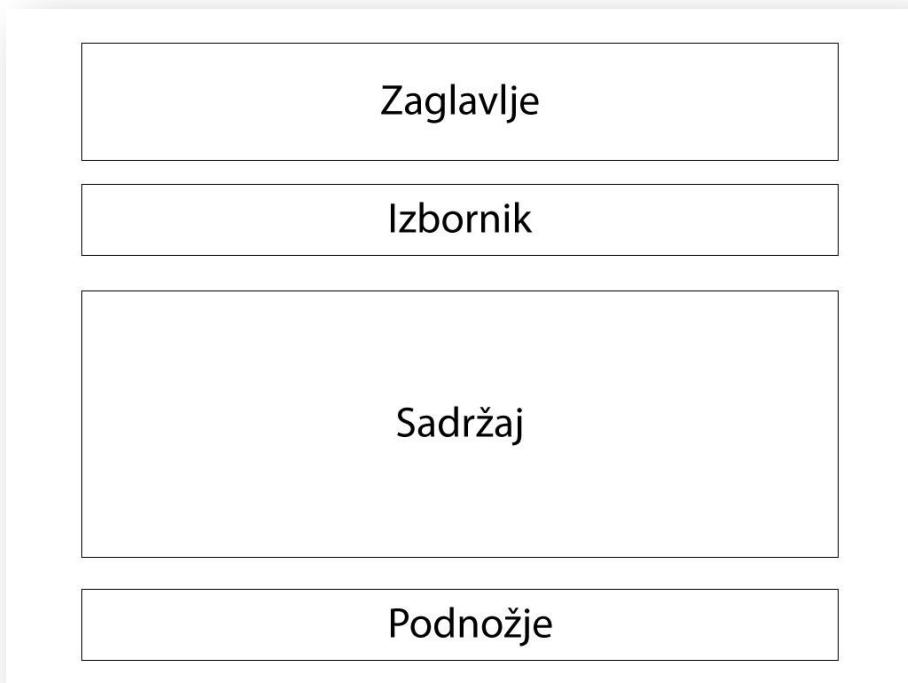
Etiam et una eu ligula fermentum pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum hendrerit porta orci ac mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam qui nisl sit amet nibh malesuada sagittis. Pellentesque non mauris ut arcu suscipit hendrerit imperdiet porta quam. Cum sociis natoque penatibus et magis dis parturient montes, nascetur ridiculus mus. Nulla facilis! Pellentesque nec elit dolor, nec mollis urna. Quisque volutpat, mi sodales fermentum blandit, leo quam convallis purus, sed scelerisque arcu lacus id odio. Maecenas ut nisi non metus veneant pharetra. Pellentesque habitab morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum aliquet, neque vel tempore vulputate, tortor ligula auctor justo, in scelerisque dolor ante vel tempor.

U primjeru za absolutnu širinu elementa postavljena je na 400px no u praksi je puno veće i obično iznosi 964 (za ekrane širine 1024px) no sve češće se koriste i veće širine jer ekran imaju veću rezoluciju.

### 6.15.1 Izrada stranice koristeći jedan stupac

Prije početka izrade dizajna potrebno je skicirati dizajn stranice te na taj način olakšati

kasniju izradu i pozicioniranje elemenata. Skiciranje nije nužno uvijek potrebno no preporučljivo je jer može znatno olakšati kasniji rad.



Na skici se vidi da je stranica podijeljena na 4 logička dijela:

- zaglavje,
- izbornik,
- sadržaj te
- podnožje.

U zaglavju se najčešće stavlja naslov stranice, logotip te ponekad izbornik koji se može nalaziti i kao poseban element ispod zaglavlja. Izbornik se sastoji od poveznica na druge stranice unutar istog web site-a ili vanjske poveznice. U sadržaju se nalazi sadržaj stranice i sve važne informacije koje se mijenjaju ovisno o stranici koja je odabrana. U podnožju se nalaze podaci o autoru i autorskim pravima stranice te obično o godini izrade stranice.

Logičke cjeline na skici potrebno je pretvoriti u HTML div elemente koji se mogu nazvati:

- wrapper (container unutar kojeg će biti postavljena cijela stranica),
- header (prostor za zaglavje),
- navigation (prostor za izbornik),
- content (prostor za sadržaj) i
- footer (prostor za podnožje).

```
<div id="wrapper">
    <div id="header"></div>
    <div id="navigation"></div>
    <div id="content"></div>
    <div id="footer"></div>
</div>
```

Nakon kreiranja HTML strukture potrebno je napraviti i CSS kôd. Wrapper ima fiksnu dimenziju od 964px jer je stranica predviđena za ekran veličine 1024px. Svi elementi su odmaknuti od ruba 5px dok su međusobno elementi odmaknuti isto 5px.

```
#wrapper {
    background-color: #eeeeee;
    width: 964px;
    margin: auto;
    padding: 5px;
    border: 1px solid black;
}
#header, #navigation, #content, #footer {
    background-color: #dddddd;
    padding: 5px;
    border: 1px solid black;
}
```

```

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Centriranje</title>
    </head>
    <body>
        <div id="wrapper">
            <div id="header">
                <h1>Izrada web stranice pomoću jednog stupca</h1>
            </div>
            <div>
                <a href="#">Početna stranica</a>
                <a href="#">O nama</a>
            </div>
            <div id="content">
                <h2>Naslov sadržaja</h2>
                Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et urna eu ligula fermentum pulvinar. Quisque quis justo lorem. Suspendisse porttitor lobortis erat ut posuere. Vestibulum hendrerit porta orci ut mattis. Fusce lacus eros, vehicula sed pharetra id, tristique a elit. Aliquam quis nisl sit amet nibh malesuada sagittis. Pellentesque non mauris ut arcu suscipit hendrerit imperdiet porta quam. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla facilisi.
            </div>
            <div id="footer">
                <p>VsiTe, 2013</p>
            </div>
        </div>
    </body>

```

Izrada web stranice pomoću jednog stupca

Početna stranica O nama

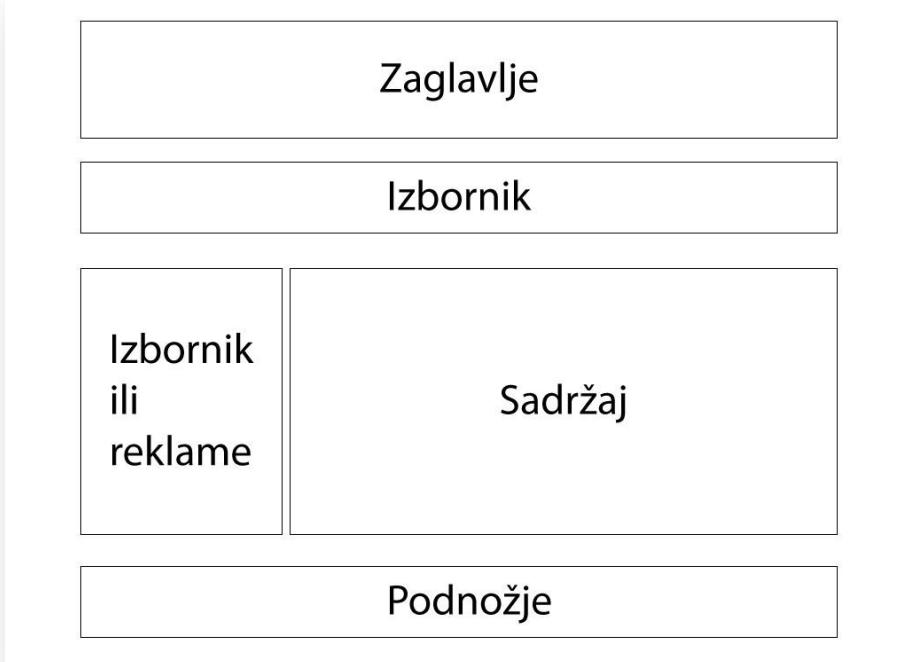
Naslov sadržaja

Placeholder text about a one-column page design.

VsiTe, 2013

### 6.15.2 Izrada stranice koristeći dva stupca

Dizajn sa dva stupca je puno češće korišteni u odnosu na dizajn sa jednim stupcem. Koristeći dizajn sa dva stupca unutar dodatnog stupca koji se može nalaziti ili sa lijeve ili sa desne strane sadržaja obično se nalazi ili vertikalni izbornik ili oglasi tj. reklame.



Izgled HTML kôda može izgledati ovako:

```

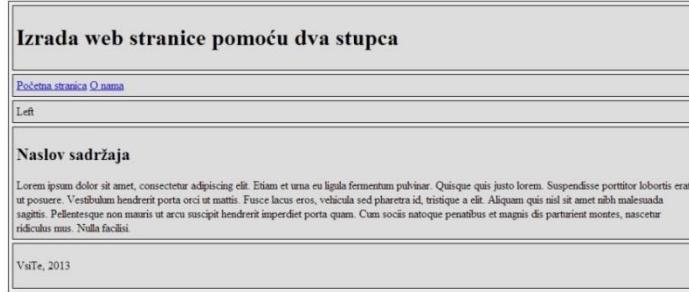
<div id="wrapper">
    <div id="header"></div>
    <div id="navigation"></div>
    <div id="left"></div>
    <div id="content"></div>
    <div id="footer"></div>

```

```

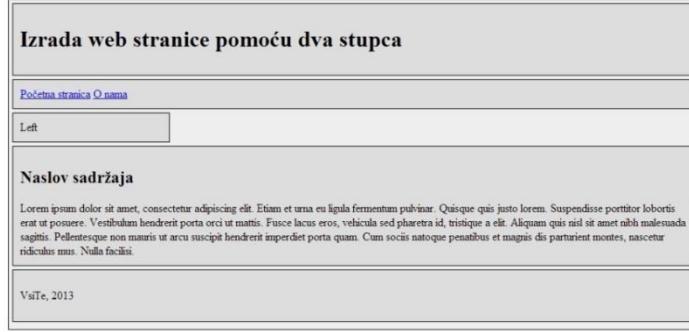
#wrapper {
    background-color: #eeeeee;
    width: 964px;
    margin: auto;
    padding: 5px;
    border: 1px solid black;
}
#header, #navigation, #content, #footer, #left {
    background-color: #dddddd;
    padding: 5px;
    border: 1px solid black;
}

```



Kao što je vidljivo na slici ukoliko se ne postavi širina lijevog stupca automatski se postavlja na 100% roditelja, tj. postavlja se punu širinu ekrana ukoliko ne postoji roditelj. Ukoliko se postavi fiksna širina lijevog stupca izgled stranice se mijenja.

```
#wrapper {  
    background-color: #eeeeee;  
    width: 964px;  
    margin: auto;  
    padding: 5px;  
    border: 1px solid black;  
}  
  
#header, #navigation, #content, #footer, #left {  
    background-color: #dddddd;  
    padding: 5px;  
    border: 1px solid black;  
}  
  
#left {  
    width: 200px;  
}
```

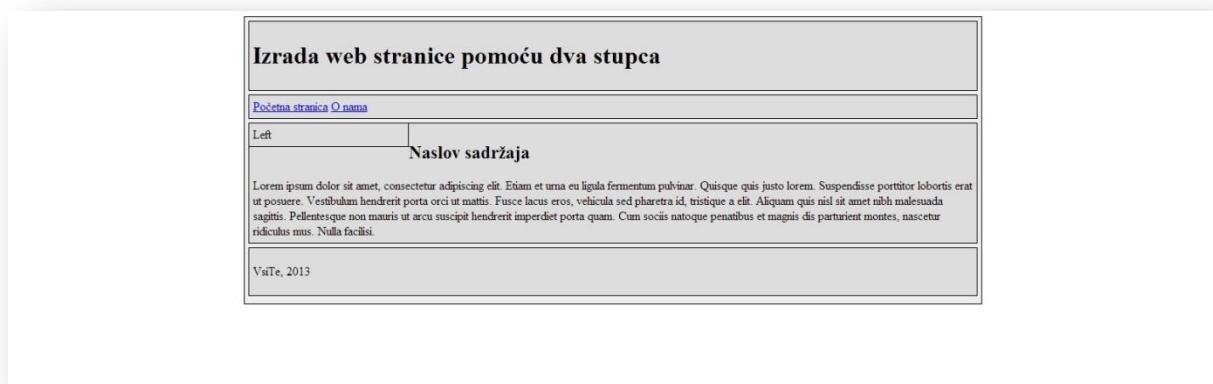


Kao što je vidljivo na slici iako je postavljena fiksna širina lijevog stupca i dalje stupac sa sadržajem se nalazi ispod njega.

### 6.15.2.1 Svojstvo float i clear

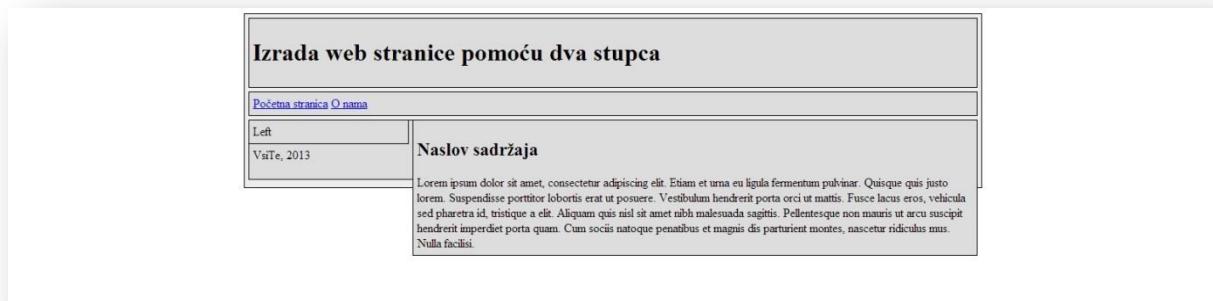
Svojstvo **float** omogućuje kako bi se ciljani element postavio ili uz lijevi ili uz desni rub ekrana. Ukoliko na više elemenata postavimo svojstvo **float** to neće imati utjecaja na konačni izgled. Ako se na lijevi stupac postavi svojstvo **float** na vrijednost **left** rezultat će biti kao na slici.

```
#left {  
    width: 200px;  
    float: left;  
}
```



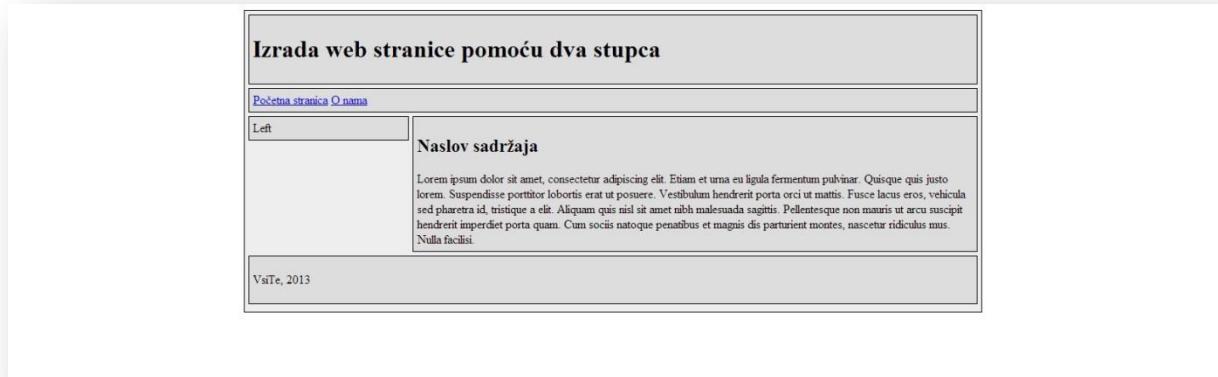
Kako bi se postavio sadržaj s desne strane potrebno je postaviti i širinu elementa unutar kojeg se nalazi sadržaj. Širina elementa se može izračunati koristeći sljedeću formulu:  $964 - 200 - 4 - 20 = 740$  od čega je 200 piksela širina lijevog stupca 4 je obrub po 2 piksela sa svake strane i 25 je padding, 15 piksela lijeve i 10 piksela sa desne strane.

```
#content {  
    width: 735;  
    float: right;  
}
```



Korištenjem fiksnih širina oba stupca postiže se postavljanje obaju stupaca kako je planirano no zadnji dio odnosno footer se nalazi ispod sadržajnog stupca što se želi izbjegići. Kako bi se to izbjeglo potrebno je koristiti svojstvo ***clear*** koje ima tri vrijednosti ***left***, ***right*** i ***both***. Postavljanjem svojstva ***clear*** i vrijednosti ***both*** na footer postiže se konačni izgled stranice.

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



### 6.15.3 Izrada stranice koristeći tri stupca

Na internetu najčešće susrećemo stranice koje koriste dizajn sa tri stupca. Kao i kod dizajna sa dva stupca, dodatni stupci sadrže dodatne informacije kao što su reklame ili kalendar ili drugo. Uobičajeno je da se dodatni stupac nalazi ili s lijeve ili s desne strane sadržajnog stupca.



Izgled HTML kôd može izgledati ovako:

```
<div id="wrapper">
<div id="header"></div>
<div id="navigation"></div>
<div id="left"></div>
<div id="content"></div>
<div id="right"></div>
<div id="footer"></div>
```

Kao i u slučaju sa dizajnom koji koristi dva stupca pozicioniranje elemenata se izvodi koristeći **float** i **clear** svojstvima. Lijevi i sadržajni stupac koriste svojstvo **float** sa vrijednostima **left** dok desni stupac koristi **right** vrijednost svojstva **float**.

```

#wrapper {
    background-color: #eeeeee;
    width: 964px;
    margin: auto;
    padding: 5px;
    border: 1px solid black;
}

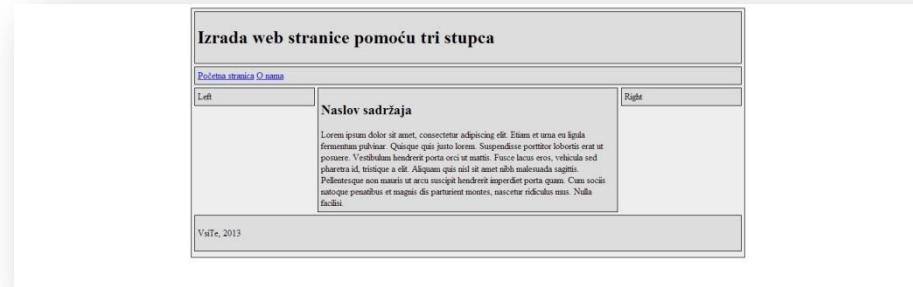
#header, #navigation, #content, #footer, #left, #right {
    background-color: #dddddd;
    padding: 5px;
    margin-bottom: 5px;
    border: 1px solid black;
}

#left {
    width: 200px;
    margin-right: 5px;
    float: left;
}

#content {
    width: 517px;
    float: left;
}

#right {
    width: 200px;
    margin-left: 5px;
    float: right;
}

```



## 6.16 Pozicioniranje elemenata

Koristeći svojstvo **position** moguće je pozicionirati pojedini element na stranicu ovisno o njegovoј vrijednosti. Uz ovo svojstvo također se koriste svojstva **left**, **right**, **top** i **bottom** pomoću kojih se pože odrediti pozicija nekog elementa u odnosu na navedene strane.

Svojstvo **position** može imati sljedeće vrijednosti:

- static,
- absolute,
- relative te
- fixed.

### 6.16.1 Static

Vrijednost **static** je predefinirana vrijednost svojstva **position**, na taj način se elementi dodaju jedan za drugim na stranicu. Svojstva **left**, **right**, **top** i **bottom** nemaju utjecaja prilikom korištenja ove vrijednosti.

### 6.16.2 Absolute

Korištenjem vrijednosti **absolute** elementi se pozicioniraju apsolutno u odnosu na druge elemente tj. na prvi element koji ima poziciju različito od **static** ukoliko takav element ne postoji onda se element pozicionira u odnosu na <html> oznaku.

```
h2 {  
    position: absolute;  
    left: 10px;  
    top: 40px;  
}
```



### 6.16.3 Relative

Korištenjem svojstva **relative** elementi se pozicioniraju relativno na normalnu poziciju.

```
h2.pos_left {  
    position:relative;  
    left:-20px;  
}
```



#### 6.16.4 Fixed

Kod korištenja svojstva **fixed** potrebno je znati se element postavlja na fiksnu poziciju te će taj element uvijek biti na točno toj poziciji. Prilikom pomicanja kliznih traka (scrollanja) element uvijek ostaje na istoj poziciji.

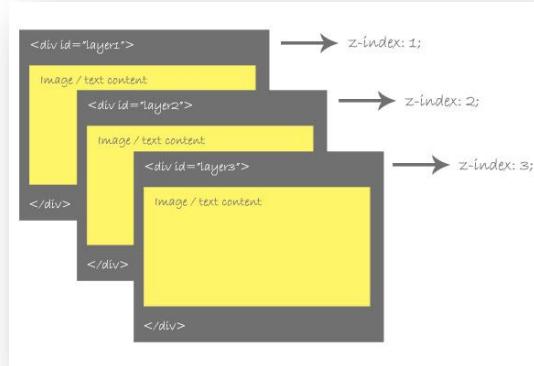
```
h2.pos_fixed {  
    position:fixed;  
    left: 20px;  
    top: 50px;  
}
```



#### 6.16.5 z-index

Uobičajeno HTML stranice su dvodimenzionalne, no ponekad je potrebno elemente složiti i u tri dimenzije, svojstvo **z-index** to omogućava. **Z-index** se može primjenjivati samo kada se primjenjuje svojstvo **position** sa vrijednostima **absolute**, **relative** i **fixed**. Vrijednosti **z-indexa** prikazuju koliko je element „blizu“ korisniku tj. koliko je udaljen.

Naime vrijednosti mogu biti ili negativne što znači da je element u pozadini drugih elemenata ili mogu biti pozitivne vrijednosti što znači da je element iznad svih. Predefinirana vrijednost je **auto**; Ukoliko se koristi nekoliko elemenata koji imaju vrijednost **auto** elementi će se slagati jedan na drugi.



```
#prvi {
    position:fixed;
    left: 20px;
    top: 55px;
    z-index: auto;
    background-color: lightblue;
}

#drugi {
    position:fixed;
    left: 30px;
    top: 70px;
    z-index: auto;
    background-color: red;
}

#treci {
    position:fixed;
    left: 40px;
    top: 85px;
    z-index: auto;
    background-color: lightgreen;
}
```



```
#prvi {  
    ...  
    z-index: 0;  
    ...  
}  
#drugi {  
    ...  
    z-index: 2;  
    ...  
}  
#treci {  
    ...  
    z-index: 1;  
    ...  
}
```



```
#prvi {  
    ...  
    z-index: 3;  
    ...  
}  
#drugi {  
    ...  
    z-index: 0;  
    ...  
}  
#treci {  
    ...  
    z-index: 2;  
    ...  
}
```



## 6.16.6 Display

Svojstvo **display** određuje na koji će se način HTML element prikazivati. Najčešće vrijednosti ovog svojstva su:

- none,
- block te
- inline.

### 6.16.6.1 None

Koristeći **none** vrijednost element se ne prikazuje na stranici.

```
#prvi {  
    display: none;  
    border: 1px solid black;  
}
```

```
<h1>Display: none</h1>
<div id="prvi">Prvi element </div>
<div id="drugi">Drugi element </div>
<div id="treci">Treći element </div>
```

### Display: none

Drugi element
Treći element

Kao što je vidljivo na slici prvi element koji postoji unutar HTML dokumenta se ne prikazuje na stranici jer mu se postavljeno svojstvo **display** na vrijednost **none**.

#### 6.16.6.2 Block

Koristeći **block** vrijednost element se prikazuje kako je inače i predviđeno tj. širina se postavlja na širinu elementa koji je iznad njega. Elementi div, p, ul, ol, li i od h1 do h6 imaju predefinirane vrijednosti **block**. Ponašaju je kao da je postavljen prijelom retka prije i poslije elementa.

```
#prvi {
    display: block;
    border: 1px solid black;
}
```

### Display: block

Prvi element
Drugi element
Treći element

#### 6.16.6.3 Inline

Koristeći **inline** vrijednost element zauzima mesta samo onoliko koliko mu je potrebno tj. koliko je velik sadržaj elementa. Obično se nalaze unutar nekog „block“ elementa. Elementi span, a, img, em, strong imaju predefinirane vrijednosti **inline**. Ne rade prijelom retka prije i poslije elementa.

```
#prvi {  
    display: block;  
    border: 1px solid black;  
}  
.element {  
    display: inline;  
    border: 1px solid black;  
}  
p {  
    border: 1px solid black;  
}
```

```
<h1>Display: inline</h1>  
<div id="prvi">Prvi element  
    <div class="element">Drugi element unutar prvog elementa </div>  
    <p>Treći element</p>  
</div>
```

## Display: inline

Prvi element	Drugi element unutar prvog elementa
	Treći element