

Želja da se na jednom mjestu omogući definiranje ili izmjena oblikovanja elemenata web-stranice, dovela je do stvaranja CSS-a (Cascading Style Sheets).

Prošlo je gotovo desetljeće od prve objave našeg tutoria [HTMLupute.pdf](#). U njemu su date osnove html-a s uputama za izradu jednostavnih web-stranica. "Jednostavne" pri tome znači da nije uključen rad i obrada baza podataka, što je materija za profesionalce. Ako niste do sada upoznali html i te osnove, za preporuku je da prije konzumiranja ovih uputa proučite barem prvih sedam-osam stranica tog našeg tutorijala. U protivnom, neki pojmovi i oznake u ovom tekstu mogli bi Vam biti nerazumljivi. Sve što je u tom tutorialu objašnjeno, važi i danas, a većinu toga morate znati ako se namjeravate upuštati u izradu ili samo doradu-prilagodbu Vaše web-stranice, koju je možda netko drugi izradio.

U tih desetak godina, došlo je međutim do promjena u web-dizajnu. Najveća je promjena u odvajanju vizualnog oblikovanja od sadržaja. Dok se prije oblikovanje obavljalo u samom html-dokumentu, danas se sadržaj kreira gotovo bez oblikovanja, a taj se sadržaj onda može različito oblikovati ili mijenjati uz pomoć css-datoteke, gotovo bez intervencije u html dokumentu. Time je postignuta i izvjesna optimalizacija, tj. smanjenje "težine" web-sjedišta, a stranica koja "teži" manje kilobajta, brže se učitava.

Druga je promjena u tome, što se danas web-stranice rijetko rade "ispočetka", tj. od praznog papira. Sve složenije stranice imaju zaglavlje, naslov, navigaciju (menu-e), sadržaj i podnožje, tj. logičke cjeline koje ćemo zvati segmentima stranice. Sadržaj se sastoji od teksta, statičnih ili animiranih slika, te poveznica na interne ili vanjske sadržaje (dokumente, pod-stranice ili druge stranice, te multimediju). Danas su svima na raspolaganju tisuće "gotovih" layout-a koji se mogu s Interneta preuzeti skupa sa svojim css-datotekama, pa korisniku ostaje samo da umjesto pokaznog, unese svoj sadržaj. Stranica FREE CSS ( <http://www.free-css.com/free-css-templates> ) nudi besplatno oko 500 uzoraka web-dizajna. Izmjenama u css-datoteci, ponuđeni web-dizajn možemo prilagoditi svojim željama, a ako treba i značajnije preoblikovati, ako poznamo html i CSS.

U ovom tutorialu obrađujemo upravo CSS, pa nećemo ponavljati ono što smo obradili u osnovama [HTMLupute.pdf](#) koje su još uvijek dostupne sa uvodne stranice našeg sajta <http://inovatori.hr> pod podnaslovom SOFTVER I RAČUNALO, ili sa ovog linka: [HTML šalabakter-za jednostavne web-stranice \(PDF\)](#)

Ipak, podsjetimo se da u HTML dokumentu, tj. izvornom kodu (source code) web-stranice - imaju slijedeću strukturu (pri tome se svaka kodna definicija uklapa između početnog < > i završnog </> taga unutar kojih se upisuju konkretne HTML oznake, ili tzv. selektori):

**<!DOCTYPE HTML 5>** nalaže korištenje HTML 5 standarda (važno za ispravan prikaz hrvatskih znakova pri primjeni kodne stranice "ISO-8859-2", koju valja koristiti ako kod pišemo u Notepad-u. Ako koristimo bolji editor, bolje je umjesto ISO-885-2 u <meta> tagu staviti "UTF-8" koji se najviše i koristi. UTF-8 onda treba postaviti i u tekst-editoru.

```
</html>
  <head>
    <meta charset="ISO-8859-2" />      propisuje hrvatsku znakovnu tablicu
    <title> naslov</title>           (naslov će se prikazati u naslovnoj traci web-stranice)
  </head>
  <body>
    <p>ovdje se upisuje tekst odlomka</p>
    ..... <-- ostali sadržaji tijela web-stranice
  </body>
</html>
```

(konkretno <p> tag opisuje karakteristike odlomka, dok druge vrste sadržaja – tabele, slike i dr. imaju druge HTML oznake - vidi [HTMLupute.pdf](#) )

Unutar tijela stranice, u složenijima (a to su uvodne stranice koje se prve učitavaju kad pozovemo neko web-sjedište) smješteni su nabrojani segmenti.

U navedenom tutorialu spomenuti su i načini oblikovanja sadržaja stranice, tzv. stilovi (style). Već je tamo spomenuto da se razlikuju tri načina stilske oblikovanja u web-dizajnu:

**Ugrađeni stilovi** djeluju na tijelo (body), odnosno sadržaj čitavog dokumenta u koji su ugrađeni. Ugrađuju se na vrhu HTML dokumenta između <style> i </style> smještenih unutar zaglavlja dokumenta <head> i </head>. Ugrađeni stil djeluje samo na one dokumente (stranice unutar web-sjedišta) u koje je ugrađen, tj. u kojima ga želimo primijeniti.

**Linijski stilovi** dograđuju se unutar ciljanog taga smještenog u tijelu (tj. između <body> i </body>) html-dokumenta dodavanjem style="stilske definicije" na izvornu html-oznaku (npr na <table kao u primjeru ispod ili na <p ako se oblikuje odlomak i sl.). Djeluju lokalno samo na tag u koji su ugrađeni, definirajući u pravilu manji broj stilskih obilježja (boju pisma ili podloge, vrstu centriranja, širinu retka i dr.) na primjer: **<table style="background:aqua; text-align:center; color:blue">** U ovom je primjeru definirana svijetlo-plava pozadina teksta u tabeli, plavi tekst i središnje centriranje.

Primijetimo da se stilske definicije stavljaju među navodnike, karakteristika od argumenta odvaja se dvotočkom, a u slučaju definiranja više karakteristika, one se međusobno odvajaju znakom ;

Korištenjem ova dva načina definiranja dizajna stranica, definicije stilova razmještaju se po web-sjedištu na raznim mjestima, pa ako želimo izmijeniti izgled stranica, moramo korigirati te definicije na svim mjestima i svim stranicama gdje su primijenjene, odnosno ugrađene.

**Vezani stilovi** Da bi se izbjeglo takvo definiranje (i/ili izmjena) oblikovanja na mnogo mjesta razasutih po HTML kodu web-sjedišta (koje u pravilu sadrži više web-stranica odnosno html- dokumenata), stvoren je treći oblik definiranja tzv. "vezanog stila". Kod njega se željene stilske karakteristike ugrađuju u posebnu datoteku s nastavkom **.css**. Karakteristike iz css datoteke Internet-preglednici (browseri) primjenjuju na one HTML dokumente, s kojima smo ih povezali na način, da u zglavlje tih dokumenata (tj. unutar <head> i </head> tagova) ugradimo tag:

**<link rel="stylesheet" href="imeStila.css" type="text/css"> .**

Pri tome je "imeStila" proizvoljni naziv datoteke. U konkretnom primjeru riječ je o najčešćim - tekstualnim stilovima (type="text/css"), međutim postoje i drugi tipovi stilskih definicija.

Iako se ovaj tag također treba ugraditi u sve HTML stranice u kojima stilove iz .css faila želimo primijeniti, primjena se ipak pojednostavljuje, jer ne treba u svim stranicama ugrađivati ili mijenjati kompletne stilske definicije kao kod ugrađenih stilova, nego samo navedeni tag. Eventualne izmjene ili definicije iz .css datoteke bit će primijenjene u svim HTML dokumentma u koje je masno ispisani tag ugrađen.

Može se naravno kreirati više .css datoteka različitih imena, pa se odgovarajući tagovi (po uzoru na navedeni) ugrađuju u zaglavlja onih dokumenata u kojima pojedine stilove želimo primijeniti.

Nakon usvajanja osnovnih stilskih definicija (prije svega tekstualnog oblikovanja, razmaka odlomaka, centriranja i dr.) koji su obuhvaćeni u najranijem kompletu takvih definicija (**CSS1**), usvojene su i složenije definicije oblikovanja (**CSS2**), primjerice za definiranje razmještaja sadržaja – tekstova, okvira, slika, tabela po dokumentu i dr. Konačno, za napredno oblikovanje sadržaja koje uključuje primjerice zaobljenje uglova slika ili tekstualnih okvira, sjenčane pozadine pojedinih okvira s postepenim prijelazom nijanse u prozirno (gradijenti) i sl. razvijen je i **CSS3**, koji međutim u ovom sažetku primjerenom početnicima nije uključen.

## struktura jednostavne html-datoteke

kompletan **HTML-dokument** s vezanim stilom ima slijedeću strukturu:

```

<!DOCTYPE html 5           ← uvodni tag html-dokumenta (web-stranice) u starijim kodovima
                             može izgledati i drugačije, osobito npr. ako stranica sadrži okvire ili dr.
                             (detaljnije vidjeti u HTMLupute.pdf str. 1)
<html>
  <head>
    <meta charset="ISO-8859-2" />      ← Kodna stranica za hrvatsko pismo (bolja je "UTF-8" )
    <link rel="stylesheet" href="mojStil.css" type="text/css"> ugrađuje stilove iz css datoteke.
    Parametar href defnira (lokaciju i) ime css datoteke koja sadrži stilove. Ako je css datoteka smještena u
    istom folderu sa HTML-dokumentom, u href je dovoljno navesti ime bez lokacije (kao u ovom primjeru).
    <b><tittle style="font-size:16px">Naslov</title></b> ← naslov se prikazuje u naslovnoj traci browsera
    U ovom primjeru postavljen je masni ispis <b> i veličina naslova na 16 piksela linijskim stilom
  </head>
  <body>
    <p>Ovdje se upisuje sadržaj odlomka<br/>
    ..... <br/>
    ..... <br/>
    kraj odlomka (u ovom slučaju i sadržaja) </p>
  </body>
</html>

```

unutar tagova <body> smješteno je "tijelo", tj. sadržaj HTML dokumenta (web-stranice) na koga će se primijeniti **mojStil.css** . <br/> prebacuje tekst u novi red

Ovako definiran HTML-dokument u bilo kom jednostavnom teksteditoru sprema se kao tekstualna datoteka s proizvoljnim imenom i nastavkom `.html` (na primjer: **mojaStranica.html**) a pregledava se u pregledniku (dvoklikom na ime dokumenta u windows-exploreru, ili lokaciju s diska s imenom.html upisati u adresnu traku preglednika)

**Sadržaj css datoteke:** CSS datoteka je također obična tekstualna datoteka s nastavkom `.css` Može se kreirati i uređivati u Notepad-u, iako je bolje koristiti prikladnije editore namijenjene baš za pisanje HTML kodova.

**tittle**{font-size:20px; ← definira veličinu znakova u naslovu  
background:#E1EBC8; ← boja pozadine naslova (način definiranja boje u hekadecimalnom obliku)  
color: rgb(51, 51, 255)} ← boja teksta naslova (način definiranja boje udjelima crvene, zelene i plave)  
**p** {font-style:italic; color:red} ← stil (nagnuto) i boja teksta u odlomcima (opisni način definiranja boja engleskim nazivom)

Ove oblikovne definicije (u ovom slučaju naslova i odlomka) spremaju se kao tekstualna datoteka, (najbolje u isti folder sa web-dokumentima na koje se odnose) s nastavkom `.css`, npr: **mojStil.css**

Iz primjera vidimo, da definicija css-datoteke za element koga želimo oblikovati sadrži:

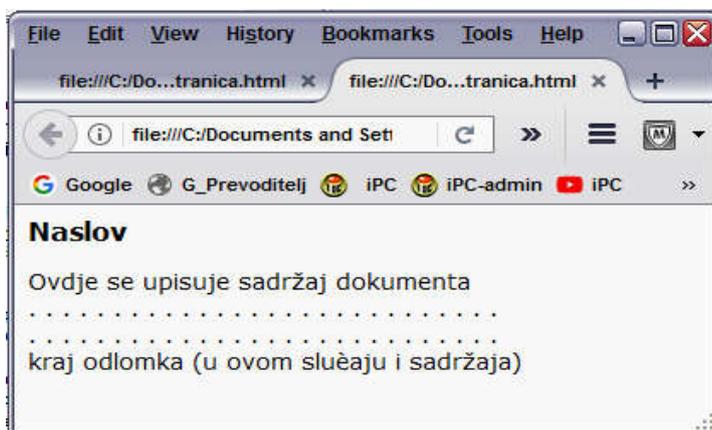
**selektor { karakteristika1:atribut; karakteristika2:atribut; itd }**

Ako za isti element oblikujemo više karakteristika, parovi se međusobno odvajaju znakom `;` Pri tome definicije možemo nizati u jednom retku (kao kod **p** u primjeru) ili u više redaka (kao kod **tittle**). Sve definicije za jedan element odnosno selektor navode se odvojene znakom `;` unutar jedne vitičaste zagrade, međutim dopušteno je za isti selektor postaviti više deklaracija.

Ako je za isti selektor ista karakteristika višestruko definirana, za cijeli segment dokumenta važit će posljednja deklaracija (ako npr. preglednik ignorira neki css nalog, možda je isti element drugačije definiran nekim kasnijim nalogom). Prazni redovi i razmaci veći od jednog znaka u kodu se ignoriraju Sve oznake elemenata koje su u html-kodu korištene kao selektori, u css-datoteci mogu prihvaćati naloge za oblikovanje tih elemenata. **Za selektore mogu se koristiti proizvoljne riječi ili kombinacije engleskih znakova**, no mnogi su općeprihvaćeni kao standardni. Tako je primjerice uobičajeno da se `h` koristi kao oznaka naslova (header), a `h1`, `h2`, `h3` itd. kao oznake podnaslva raznih nivoa. Njihove su karakteristike predefinirane, ali ih se css nalogom može korigirati.

Ako u css-u neka deklaracija treba oblikovati više elemenata, njihove oznake (selektore) treba navesti odvojene zarezom. Primjerice: **h,h1,h2,h3 {font-weight:bold}** je nalog za masni ispis svih naslova i podnaslova.

Gore smo naveli jednostavan primjer html- dokumenta **mojaStranica.html** i pripadajuće css-datoteke **mojStil.css**.

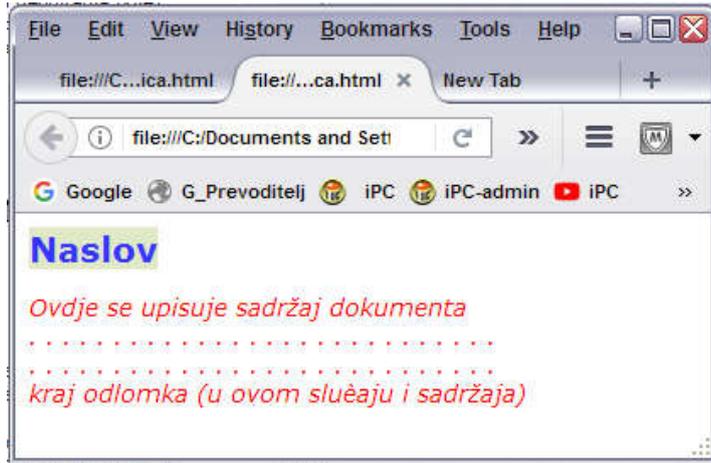
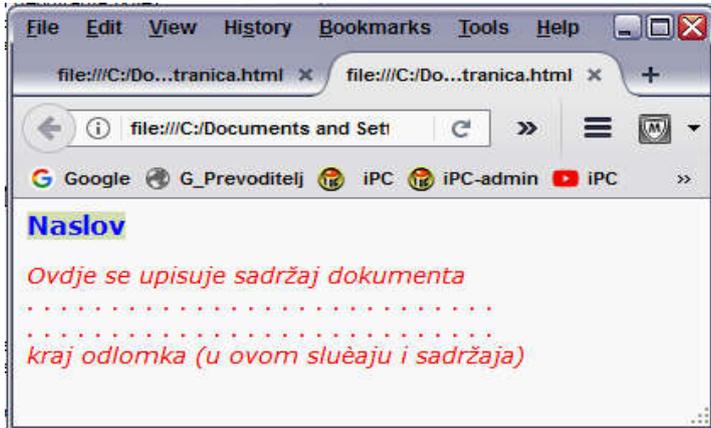


Ovako izgleda mojaStranica iz gornjeg primjera **bez css datoteke**.

Naslov je podebljan i postavljen na veličinu 16 piksela linijskim stilom u tagu `<tittle>` koji se nalazi u zaglavlju HTML dokumenta **mojaStranica.html**

Sav ostali tekst je neformatiran, tj. nismo ga propisivali u html-kodu pa ima pretpostavljene (default-ne) karakteristike.

Izgled iste mojaStranice kad se u istu mapu s njom doda i datoteka **mojStil.css** vidi se na slijedećoj stranici:



Naslov je dobio obojenu pozadinu, font je plavi, a odlomak u tijelu teksta je obojen crveno i skošen (`font-style:Italic`). Vidi se međutim, da veličina naslova nije povećana kako se zahtijeva u css failu. Znači da lokalna definicija iz linijskog stila (koja je veličinu fonta postavila na 16 px) ima prednost pred vezanim stilom iz css-faila.

Ako se definicija veličine naslova ukloni iz linijskog stila dokumenta mojaStranica, ispoljit će se djelovanje deklaracije za veličinu naslova iz datoteke mojStil.css. Slijedi da se ne smiju definirati linijskim stilovima oni elementi oblikovanja, koje želimo regulirati vezanim stilom u css-u.

Css datoteka iz primjera bi naravno djelovala na neki drugi dokument, u kome ista karakteristika koja je uređena u css datoteci, nije lokalno prejudicirana linijskim stilom.

Na opisani način se uz pomoć css datoteke mogu propisati i druge karakteristike dizajna stranice, od boje ili podvlake hiperveza (linkova) do razmještaja sadržaja (tekstualnih odlomaka, okvira, tabela, menu-a, slika i dr), bez intervencije u samom html-dokumentu. Priložen je popis elemenata dizajna koji se mogu koristiti za oblikovanje stranica u css-u.

## OSNOVNO OBLIKOVANJE

Nalozu u css-datoteci imaju oblik: **selektor {svojstvo: vrijednost}**, primjerice **p {font-size: 16px}** Ako je karakteristika (svojstvo) sastavljena iz više riječi, one se odvajaju znakom – kao **font-size**.

Kao **selektor** može se u načelu koristiti bilo koji znak ili riječ ili kombinacija znakova engleskog pisma, ali se brojni selektori smatraju standardnima, među kojima i engleski nazivi sekcija stranice kao i oblikovni tagovi iz html-dokumenata. Evo popisa uobičajenih selektora:

Sekcije: **#container, #tittle, #header, #nav, #content, #footer** – detaljnije na strani 13.

|                  |                   |                   |                                  |
|------------------|-------------------|-------------------|----------------------------------|
| <b>p</b>         | odlomak           | <b>tablice</b>    | <b>poveznice</b>                 |
| <b>body</b>      | tijelo stranice   | <b>table</b>      | <b>a</b> poveznica               |
| <b>h</b>         | naslov u tekstu   | <b>th</b>         | <b>a:link</b> neposjećena veza   |
| <b>h1, h2,..</b> | podnaslovi        | <b>tr</b>         | <b>a:hover</b> veza pod kursorom |
| <b>ol</b>        | brojčana lista    | <b>td</b>         | <b>a:visited</b> posjećena veza  |
| <b>ul</b>        | lista sa simbolom | <b>table.nav</b>  | <b>a:active</b> aktivna veza     |
| <b>hr</b>        | linija            | <b>table.rest</b> |                                  |

Brojna **svojstva** (karakteristike) koje se mogu propisati nalozima css-datoteka data su u tabeli na kraju tutoriala, a u nastavku ćemo se pozabaviti detaljnijim opisom najčešćih oblikovnih svojstava.

Osnovne oblikovne definicije uključuju svojstva za oblikovanje:

- **veliçine** (pisma, visine, širine ili debljine elementa: retka, tablice, ćelije, linije, obruba i dr.)
- **boje** (fonta, pozadine pisma ili ćelije u tabeli ili cijele stranice-dokumenta, okvira, i dr)

Dok se karakteristike veličine i boje u definiciji svojstava opisuju jednakim vrijednostima kod različitih elemenata, slijedeće karakteristike koriste vrijednosti specifične za:

- **font** ← **oblici pisma** - koriste se najčešći fontovi koje poznamo iz WORD-a
- **border** ← kompletan ili lijevi, desni, gornji ili donji **obrub** tablice, ćelije, okvira ili slike
- **background** ← boja, grafika-slika, fiksna ili klizeća **pozadina** stranice, okvira, ćelije ili teksta
- **margin** ← komplet ili lijeva, gornja, desna ili donja **margin** tabele, ćelije, okvira ili slike
- **style** ← **oblici pisma** (`font-style`), liste (`list-style`), obruba (`border-style`)

Ovi elementi se kombiniraju sa svojstvom koje želimo definirati, na primjer:

|                            |   |
|----------------------------|---|
| <i>font-size</i>           | ← veličina pisma  |
| <i>font-style</i>          | ← izgled pisma (tj. vrsta fonta, npr: arial ili sans serif) |
| <i>font-weight</i>         | ← "debljina" pisma  |
| <i>border-color</i>        | ← boja obruba   |
| <i>background-image</i>    | ← pozadinska grafika  |
| <i>border-bottom-style</i> | ← izgled donjeg obruba (puna, crtkana, točkana crta i sl)   |

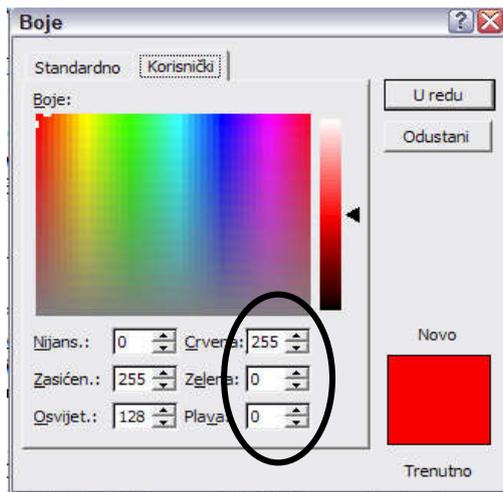
Opišimo najprije detaljnije karakteristike veličine i boje

**veličina** uz svojstva specifična za pismo (*font-size* i *font-weight*) uključuje slijedeća svojstva:

- *height* ← visina
  - *min-height* ← minimalna visina
  - *max height* ← maksimalna visina
  - *min-width* ← minimalna širina
  - *max width* ← maksimalna širina
- Veličina elemenata uobičajeno se izražava u pikselima (px), no može se izraziti i u postotku od širine ili visine dokumenta (npr. može se zadati širina kolone, tablice, slike ili linije i sl. od x% od ukupne širine web-stranice). Kod pisma se može zadati veličina u postotku prema normalnoj veličini pisma koja je default-no (tj. ako se ne propiše drugačije - pretpostavljeno) 11px.

**boja (color)** Riječ **color** se dodaje na element čiju boju definiramo, npr: *border-color*. Može se odnositi na font, podlogu, pozadinu, ćeliju idr. Izražava se na tri načina:

- opisno, tj. **white black gray blue navy red silver green yellow brown**  
engl.imenom bijelo crno sivo plavo morn.pl. crveno srebrno zeleno žuto smeđe
- RGB oznakom, npr: **rgb(255,255,255)** za **crno**, **rgb(0,0,0)** za **bijelo** ili **rgb(255,0,0)** za **crveno**
- heksadecimalno, npr: **#000000** **#ffffff** **#ff0000**



Često ćete trebati i prozirnu boju (engleski: **transparent**)

RGB oznaka definira udio crvene, zelene i plave boje u željenoj nijansi.

RGB ili heksadecimalnom oznakom boje, mogu se daleko preciznije definirati željene nijanse nego korištenjem imena boje.

Oznake se mogu dobiti primjerice izborom boje iz korisničke palete u nekom foto-editoru, a vrijedosti RGB oznaka su vidljive i u korisničkoj paleti za izbor boje u WORD-u (slika lijevo).

Dok se veličina i boja kombinira sa svojstvima različitih elemenata, u nastavku se navode oblikovne karakteristike specifične za pojedine vrste elemenata html-dokumenata.

## PISMO

- font** prihvaća više svojstava pisma u jednoj definiciji
- font-family** vrsta (oblik) pisma – pretpostavljen je Times New Roman. Mogu se zadati i drugi najčešće korišteni fontovi koje koristimo u WORD-u obavezno uz grupe:
- sans serif** grupa fontova koji su čitljiviji od serif (to su **arial, verdana, helvetica** i sl.)
- serif** ostali fontovi (npr. **courier, Times New Roman** i drugi)  
Preporuča se zadati više sličnih fontova obavezno s grupom, odvojenih zarezom, za slučaj da posjetitelj web-stranice nema instaliran baš taj koga smo predvidjeli. Primjer: **font-family: verdana, arial, sans serif**
- font-size** veličina pisma u pikselima (px) ili u postotku u odnosu na normalnih 11px. No može se zadati i opisno: **normal, x-large, xx-large, xxx-large**
- font-style** dopunsko oblikovanje, npr: **italic** (nagnuto pismo). **none** nalaže odsustvo dopunskog oblikovanja.
- font-color** boja teksta (jednaki učinak ima i sam **color**)
- font-weight** debljina teksta (100, 200, . . . . 400, . . . . .700) ili opisno:  
**light normal bold** (ili u html-u tag **<b> </b>**)
- Prihvaćaju se i debljine **lighter** i **bolder**
- text-align** poravnanje: **left** (lijevo), **right** (desno), **center** (centrirano)

|                        |   |
|------------------------|---|
| <b>text-decoration</b> | dopunsko oblikovanje: <b>underline</b> , <b>overline</b> , <b>linethrough</b> , <b>blink</b> i <b>none</b><br>podvučeno, pod crtom, prekríženo, bljeskanje, ništa |
| <b>text-indent</b>     | uvučen početak teksta   |
| <b>line-height</b>     | visina retka  |
| <b>letter-spacing</b>  | razmak između znakova u riječi  |
| <b>word-spacing</b>    | razmak između riječi  |
| <b>text-shadow</b>     | tekst sa sjenom   |

**LISTE** se mogu također smatrati dijelom oblikovanja teksta

|                         |   |
|-------------------------|---|
| <b>list-style</b>       | prihvaća više svojstava u jednoj deklaraciji                                      |
| <b>list-style-type</b>  | definira tip oznake ispred retka (markera) – puni ili prazan kružić, rombić i dr. |
| <b>list-style-image</b> | postavlja grafiku u funkciju markera  |

**OBRUBI I PADDING** oko odlomka, ovira, tabele, ćelije, slike i dr. *Padding* je razmak-praznina između obruba i sadržaja.

Deklaracija **border** se piše ovako: **border: debljina stil boja** (vidi primjer na slici).

Ako boja nije zadana, pretpostavlja se crna. Ako nedostaje stil (izgled) linije, preglednik bi mogao ignorirati cijeli zahjev, tj. izostaviti obrub. No razni preglednici mogu različito interpretirati odstupanja, pa je bolje držati se standardnih struktura i pravila pisanja.

|                     |  |
|---------------------|--|
| <b>border</b>       | regulira više svojstava <b>obruba</b> u jednoj deklaraciji                     |
| <b>border-color</b> | boja obruba  |
| <b>border-width</b> | širina obruba u pikselima ili postocima širine stranice (pretpostavljeno 100%) |
| <b>border-style</b> | oblik linija. Na raspolaganju su:  |
| • <b>solid</b>      | puna linija  |
| • <b>dashed</b>     | crtkana  |
| • <b>dotted</b>     | točkana  |
| • <b>double</b>     | dvostruka  |

Gornje karakteristike se odnose na čitav obrub, međutim moguće je posebno propisati svaku stranicu obruba, tako da umjesto "border" iz gornjih karakteristika koristimo dijelove obruba:

- **border-left** lijevi obrub
- **border-right** desni obrub
- **border-top** gornji obrub
- **border-bottom** donji obrub

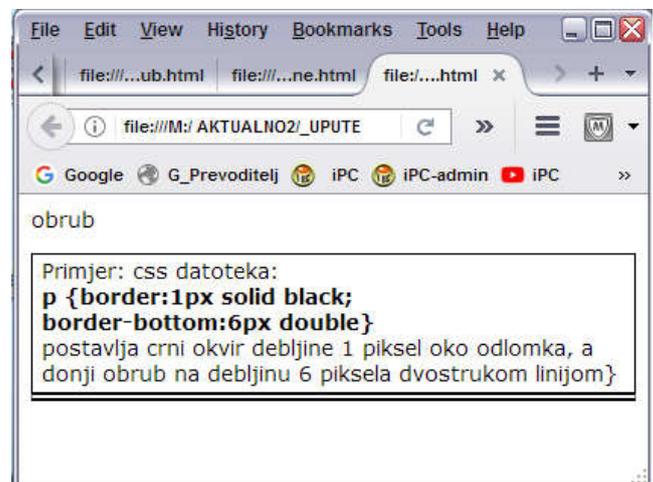


Gornja slika prikazuje stranicu **obrub.html** sa uokvirenim odlomkom uz pomoć datoteke **obrub.css** koja sadrži masno tiskane deklaracije, kao primjer primjene opisanih svojstava.

Da sadržaj ne bi bio "prilijepljen" uz obrub (kao na gornjoj slici), koristi se nalog: **padding:4px** koji postavlja jednaki razmak (u ovoj naredbi 4px) od obruba sadržaja na sve četiri strane obruba.

Moguće je propisati različite razmace definicijom: **padding: Tpx Rpx Bpx Lpx** (Paziti na redosljed) Pri tome je **T** širina gornjeg, **R** desnog, **B** donjeg, a **L** lijevog razmaka sadržaja od obruba u pikselima. Primjerice **padding:2px 1px 4px 6px** dodan u deklaracije za oblikovanje odlomka u datoteci **obrub.css** rezultira razmacima od obruba prema desnoj slici. Tako oblikovan odlomak bolje izgleda nego na prethodnoj slici.

Umjesto jedne definicije za sve četiri strane, mogu se koristiti i karakteristike za svaku stranu posebno: **padding-top**, **padding-right**, **padding-bottom**, **padding-left**



*Padding* je primjenjiv za sve vrste sadržaja (tekstne okvire, slike, uokvirene odlomke sa border i dr)

## MARGINE

Dok padding definira razmak između sadržaja i okvira uokvirenih sadržaja, margine su praznine između (uokvirenog ili neuokvirenog) sadržaja i rubova web-stranice. Mogu se primijeniti na sve sadržaje web-stranice (tekstne okvire, uokvirene ili neuokvirene odlomke, naslove, liste, slike i dr. Kao i kod paddinga, **margin** postavlja jednake margine na sva četiri ruba web-dokumenta – npr:

**margin:4px**

Također je moguće definirati svaku marginu posebno, uz pomoć svojstava:

**margin-top**  
**margin-right**  
**margin-bottom** i  
**margin-left**

Umjesto toga, može se koristiti struktura **margin:TpX RpX BpX LpX** (poštivati redosljed T-R-B-L) I ovdje je **T** vrijednost gornje, **R** desne, **B** donje i **L** lijeve margine u pikselima.

Margine se mogu primijeniti na sve uokvirene ili neuokvirene sadržaje, pa i na odlomke sa ili bez obruba, koje (ako su bez obruba) odvajaju za vrijednost donje margine zadanu u nalogu **margin**. Evo i primjera uporabe karakteristike margin za oblikovanje tekstualnog sadržaja:

Datoteka **margin.css** sadrži samo masno ispisan nalog iz primjera. Vidi se da se definicija za margine u primjeru odnosi samo na odlomke (ali ne i na naslov).

Ako bismo htjeli da iste margine važe i za naslov, deklaracija bi trebala glasniti:

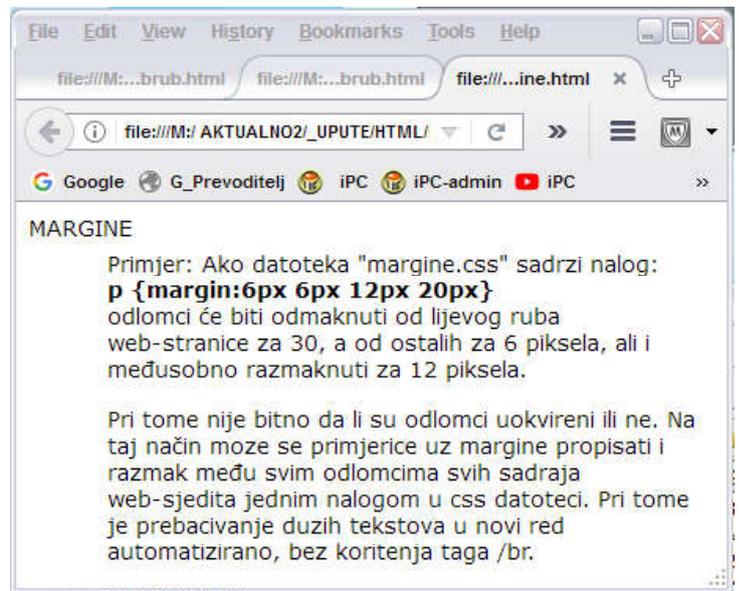
**tittle, p {margin:6px 6px 12px 20px}**

Umjesto toga, u css datoteku možemo dodati npr. deklaraciju za pomak početka naslova na 30% širine web-stranice:

**tittle {margin-left:30%}**

Ako je atribut u postocima širine stranice, pri širenju ili suženju okvira stranice mišem, premještat će se položaj naslova.

Ne zaboravimo, da se u slučaju višestrukog definiranja istog elementa (npr. lijeve margine) istom naredbom, na sav sadržaj uvijek primijenjuje posljednji zahtjev.



**POZADINSKA GRAFIKA** je grafički sadržaj koji se može postaviti ispod primarnih sadržaja.

Deklaracija za postavu pozadinske grafike (npr. ove desno) ima oblik:

**body {background-image:url(grafika.jpg)}** gdje je grafika.jpg ime slike.

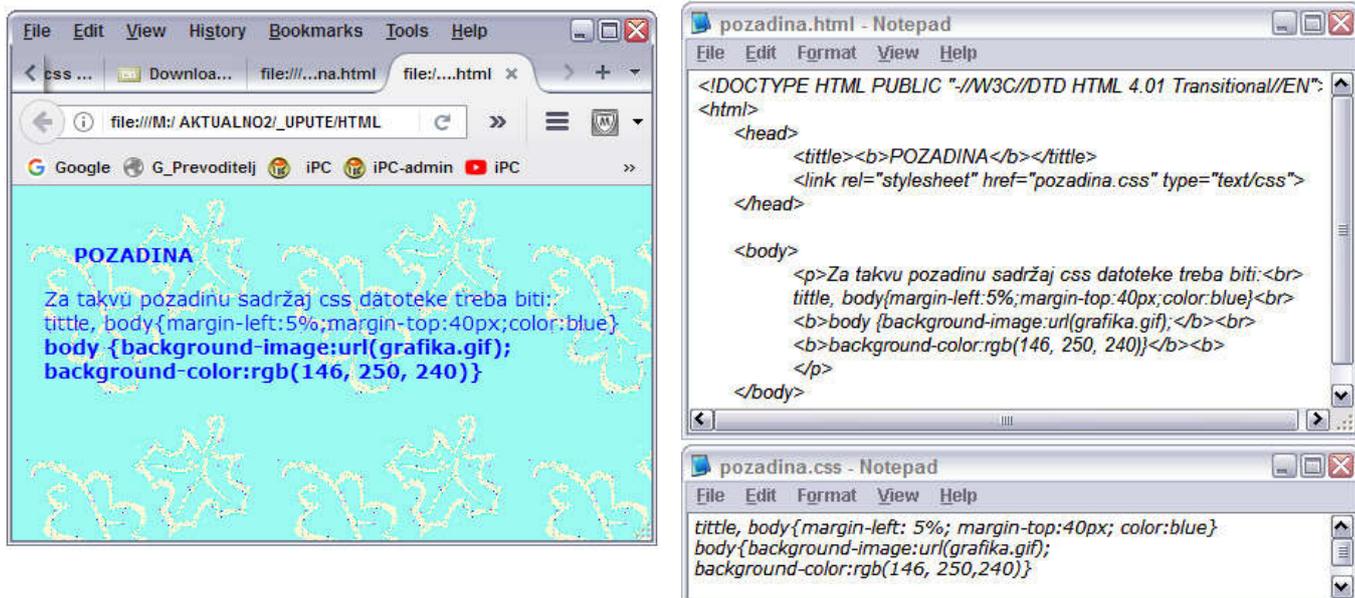
Takva deklaracija napunit će cijelu web-stranicu sa ponavljajućom slikom. Ako želimo da se na stranicu postavi slika bez ponavljanja, deklaracija glasi:

**body {background-image:url(grafika.jpg) no-repeat top}**

"Top" na kraju deklaracije gornji rub slike pozicionira na vrh web-stranice, središnje centrirano, no moguća su druga pozicioniranja: left, right, center (stavlja sredinu slike na sredinu prostora) ili bottom (poravnava donji rub slike s donjim rubom prostora), ili kombinacije (bottom right i sl.)

U deklaraciji trebaju doslovno stajati slova **url** ispred zagrade u kojoj je ime slike. Sliku je najbolje smjestiti u isti folder sa html-dokumentima. Time se izbjegava potreba navođenja puta do slike. Primjer takve web-stranice je na slici lijevo, a izvorni kod tog html-dokumenta (**pozadina.html**) i pripadajuća css datoteka (**pozadina.css**) su desno od nje, pisani u Notepad-u.



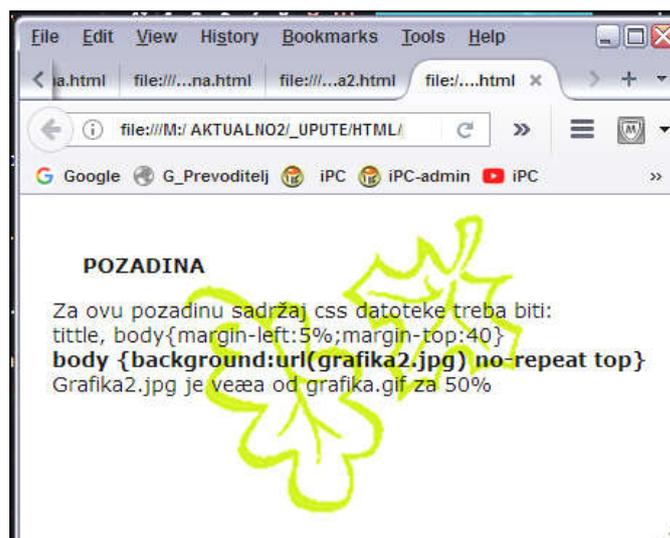


Vidi se da izvorni html-kod sadrži samo naslov, redak za vezu sa css-datotekom i tekstualni odlomak bez oblikovanja sadržaja (između <p> i </p> je samo tekstualna uputa a ne kod).

Oblikovanje stranice je u cijelosti sadržano u css-datoteci. Prvi redak css-datoteke sadrži oblikovanje teksta (margine i boju pisma), a masno ispisani redovi definiraju pozadinu. Slika se ponavlja kroz čitav prostor stranice. Zadnji redak postavlja pozadinsku boju cijele stranice (body), međutim da bi se ta boja vidjela, slika grafika.gif treba biti prozirna, što jpg format ne podržava, pa je zato korišten gif format slike s prozirnom pozadinom.

Desno je pozadina s nešto povećanom slikom koja se ne ponavlja, bez obojene podloge. U prvom retku css-datoteke izostavljeno je **color:blue**, pa je tekst sada pretpostavljene, crne boje. Također, vidi se da je gornja margina zadana sa **margin-top:40** (bez **px**), iz čega zaključujemo da se pikseli pretpostavljaju, jer je preglednik uredno izvršio nalog.

Događa se i mušićavo ponašanje preglednika, pa tako npr. u drugoj varijanti **pozadina2.css** pozadinska grafika se odbila pojaviti po nalogu **background-image:url(grafika2.jpg)** što je uredan nalog, ali se uredno prikazuje po nalogu **background:url(grafika2.jpg)** bez "-image". Zato, ako preglednik ignorira deklaraciju iz css datoteke, valja malo eksperimentirati, a po potrebi i pogledati kako su problem riješili drugi. U tu svrhu će dobro poslužiti uzorci gotovih css-datoteka. No **najprije pažljivo provjerite da li ste ispravno utipkali engleske riječi i strukturu naloga u deklaracijama**. Najmanja greška proizvest će ignoriranje naloga ili sasvim neočekivano razmještanje elemenata. Ispušteno slovo ili zamjena ; i : u nalogima najčešće su greške (nemojte pisati kod s tipkovnicom koja ispušta slova).



## POSTAVA KLIZAĆA

Ako u "prozoru" na ekranu posjetitelja nije vidljiv čitav HTML-dokument (web-stranica), preglednik će automatski postaviti klizač na desni ili donji rub prozora (ili oba ako je potrebno) i tako omogućiti "skrolanje" i pregledvanje čitavog sadržaja.

Slično tome, moguće je da u okviru definiranih dimenzija na stranici neće stati čitav sadržaj okvira. U takvom slučaju preglednik će okvir rastegnuti koliko treba, da u njega stane čitav sadržaj, možda i van vidljivog dijela stranice, pa on dobiva (najčešće bočni) klizač. Želimo li zadržati deklarirane dimenzije okvira, nalog **overflow:auto** postaviti će klizač uz njegov rub, čime čitav sadržaj postaje dostupan. Nalog možemo dodati u deklaraciju s nalogima za dimenzioniranje okvira.

**OBLIKOVANJE POVEZNICA** (hyperlink-ova) omogućuje lakše razlikovanje poveznica od običnog teksta bojom i/ili podvlakom, kao i razlikovanje veza koje smo već posjetili od neposjećanih.

- **a:link**
- **a:visited**
- **a:hover**

**a:hover** mijenja izgled veze kad mišem prelazimo preko nje.

Primjer: **a:link,a:visited, a:hover {text-decoration:underline}** ← sve veze će biti podvučene  
**a:link{color:blue}** ← neposjećena veza će biti plavo pisana tj. ovako: [veza](#)  
**a:visited{color:aqua}** ← posjećena veza bit će svijetloplava [veza](#)  
**a:hover{color:yellow; background-color:blue}** [veza](#)  
 Po zadnjem nalogu, za prelaska mišem preko veze, pismo će biti žuto na plavoj podlozi.

## NAPREDNO OBLIKOVANJE CSS-om

**ID I KLASE** U prethodnim primjerima vidjeli smo kako možemo oblikovati npr. odlomke na web-stranici. Međutim takvo oblikovanje će se odnositi na sve odlomke. Klase i ID omogućuju različito oblikovanje istovrsnih elemenata web-stranice. Pri tome ID-atribut uvijek identificira isključivo jedan element, dok se klasa može primijeniti na više različitih elemenata. Ona se dodaje poput ekstenzije na željeni selektor, odvojena točkom. I klasa i ID su proizvoljne riječi, jedinstvene u jednom projektu (na jednom web-sjedištu). Pri tome, klasa se umeće u html sa: `<selektor.class="imeKlase">` a u css-datoteci u ulogu selektora igrat će: **selektor.imeKlase**

ID u html-u počinje sa: `<selektor id="imeGrupe">`, a u css-datoteci selektor će biti: **#imeGrupe**.

Evo primjera za **ID**:

Ako u css-u zadamo : **#uputa** {font-family:Times New Roman; font-style:italic; font-color:brown}

**#uputa** će kao selektor sve upute na web-sjedištu pisati kosim Times New Roman-om smeđe boje. Da bi preglednik znao koji su odlomci na stranici upute, one se u html-dokumentu moraju početi sa:

`<p id="uputa">ovdje slijedi tekst upute, koja završava s uobičajenim završnim tagom</p>`

U pregledniku bi ta uputa izgledala ovako: *ovdje slijedi tekst upute, koja završava s uobičajenim završnim tagom*

Evo primjera i za **KLASU (class)**:

Uzmimo da prijevod u nekom tekstu želimo pisati podebljano zelenom bojom. U css-datoteci definirat ćemo za to potrebnu klasu **.prijevod** na slijedeći način:

**p.prijevod {color:green; text-weight:bold}** Klasu **.prijevod** dodali smo na selektor **p**, no ona se u css-failu može definirati i samostalno (**.prijevod** mjesto **p.prijevod**)  
 Da bi preglednik znao koji je odlomak u tekstu prijevod, u html-u ga treba početi sa:

`<p class="prijevod">This text is the translation.</p>`

Rezultat će upregledniku izgledati ovako: **This text is the translation.**

No, istu klasu možemo koristiti i za drugi selektor, npr. za prevedenu tabelu. U html-dokumentu definiranje prevedene tabele počet ćemo sa:

`<table class="prijevod" iza čega slijede definicije elemenata tabele>` Tabela će se ispisati zeleno.

**SPAN** se može smatrati specifičnom uporabom klase za oblikovanje samo dijela neke logičke cjeline (npr. dijela retka ili odlomka u tekstu): Način primjene najbolje je opet razjasniti na primjeru. U source-kodu html-dokumenta definiramo:

`<p>Na benzinskoj postaji <span class="vazno">strogo je zabranjeno pušenje</span>`

Karakteristike važnog dijela teksta opisat ćemo u slijedećoj css-datoteci:

```
.vazno {font-weight:bold;
font-size:125%;
color:red}
```

Odlomak će u pregledniku izgledati ovako:

Na benzinskoj postaji **strogo je zabranjeno pušenje**

**PSEUDOKLASE** omogućuju oblikovanje onih elemenata ili stanja elemenata web.stranice koji nisu izravno prisutni u html-kodu. Počinju dvotočkom i dodaju se na selektor, ali se mogu navoditi i bez selektora (npr. **:link** ) Tipičnu i čestu uporabu predstavljaju pseudoklase:

**a:link** definira izgled nekorištene poveznice  
**a:hover** izgled poveznice u trenutku kada kursorom prelazimo preko nje  
**a:visited** izgled posjećene poveznice  
**a:active** izgled poveznice dok se određite prikazuje

Ovih pseudoklasa nema u html-dokumentu, već je u njemu samo selektor a, međutim korisnik ih može uvesti u css-datoteku ako je potrebno.

**DIV** Obrnuto od **span**, koji mijenja dio nekog elementa stranice, tag **<div>** omogućuje propisivanje istih svojstava oblikovanja za grupu elemenata web-dokumenta koji čine neku logičku cjelinu, ovaj puta koristeći identifikaciju ID.

Recimo da na uvodnoj stranici web-sjedišta želimo jednako oblikovati horizontalnu i lijevu bočnu menu-traku, i to tako da su bijeli linkovi bez podvlake smješteni na smeđoj dugmadi, pri čemu se veličina dugmeta nešto povećava kad je miš iznad njega, tako da slični na uzdignuto dugme spremno za kliknuti. Izvorni kod takve stranice (uvodna stranica često nosi ime "**index.html**") i css-datoteka **menu.css** dati su na lijevim, a izgled stranice u pregledniku na desnoj slici.

Spomenimo da će poveznice na nepostojeći sadržaj prikazati stranicu **File not found** pa je poželjno kreirati, i takve poveznice usmjeriti, na stranicu **STRANICA U IZRADI** da se izbjegne utisak o greški.

```

Index.html - Notepad
File Edit Format View Help
<html>
<head>
<title>HOME</title>

<link rel="stylesheet" href="menu.css" type="text/css">
</head>

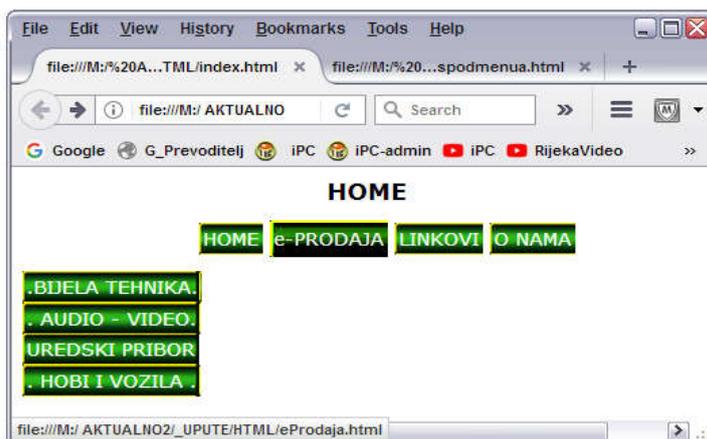
<body>

<div id="Menu"><p>
<a href="index.html">HOME</a>
<a href="eProdaja.html">e-PRODAJA</a>
<a href="linkovi.html">LINKOVI</a>
<a href="onama.html">O NAMA</a></p>

<p.v>
<a href="bTehnika.html">BIJELA TEHNIKA.</a></br>
<a href="avTehnika">. AUDIO - VIDEO.</a></br>
<a href="uredski.html">UREDSKI PRIBOR</a></br>
<a href="hobiVoz.html">. HOBI I VOZILA.</a></p.v></div>

</body>
</html>

```



```

menu.css - Notepad
File Edit Format View Help
title {margin-left:45%;font-size:150%; font-weight:bold}

p {margin-left:130px;width:420px;height:30px}
p.v {width:196px;height:300px;background-color:brown}

#Menu
a:hover {padding:6px 3px 6px 3px;background-image:url(dugmeG.jpg)}
a {line-height:21px; color:white; padding:4px 3px 4px 3px;
background-image:url(dugmeD.jpg);text-decoration:none}

```

U izvornom kodu **index.html** iza naslova imamo liniju `<link rel="stylesheet" href= . . . itd` kojom se dokument povezuje sa css datotekom. Unutar `<body>` tagova imamo dva odlomka, i to `<p>` u kome je smješten horizontalni izbornik i `<p.v>` kojim se definira lijevi vertikalni menu.

Tag `<div id="Menu">` obuhvaća, oba odlomka, jer je završni tag `</div>` iza drugog odlomka pa će i oblikovanja deklarirana pod selektorom `#Menu` u css-datoteci važiti za oba odlomka. Upravo je to namjena taga `<div>`, da koristi istu deklaraciju za oblikovanje elemenata na više mjesta, čime se smanjuje broj naredbi u kodu, a time i ubrzava učitavanje naše web-stranice.

U datoteci **Menu.css** iza podebljanog i na 150% povećanog naslova horizontalni menu je pomaknut u desno nalogom za lijevu marginu od 130px u definiciji prostora za odlomak `<p>`, da bi početak menua bio poravnat s desnim rubom vertikalnog menua. Širina odlomka `p` postavljena je na širinu u koju će stati sva dugmeta horizontalnog menua, tj na 420 piksela. Umjesto toga, širinu tog izbornika mogli smo postaviti i na punu širinu stranice (100%).

U drugom odlomku `<p.v>`, širina prostora je "rezervirana" nalogom `width:196` na širinu dugmadi (196 piksela). Ne zaboravimo da se link, pa ni dugme menu-a možda neće ni pojaviti na ekranu, ako za njega nema dovoljno širine u prostoru koji smo osigurali u drugom i trećem retku css-datoteke,

ili će npr. u horizontalnom menu-u pola dugmadi preskočiti u novi red, pa treba dobro izračunati prostor. Usput, *p.v* je proizvoljna oznaka, međutim sasvim uredno funkcionira kao bilo koji selektor.

Za pozadinu linkova korištene su dvije slike dugmeta, i to slika **dugmeD.jpg** za neaktivno dugme, a **dugmeG.jpg** za dugme nad kojim je kursor (na gornjoj slici, kursor je nad dugmetom e-PRODAJA). Ako je slika prevelika, na mjestu dugmeta prikazat će se samo dio slike, a ako je premala, u prostoru dugmeta će se javiti ponavljajuće slike manjih dugmeta. Zato veličinu slike dugmadi u foto-editoru treba prilagoditi visini linije i širini definiranoj najdužim natpisom, + paddingi (u primjeru 196 piksela). Takova podešavanja se lijepo i precizno mogu izvesti u besplatnom fotopregledniku **Irfanview** (opcija *resize/resample*).

Povećanje dugmeta pod kursorom postignuto je povećanjem gornjeg i donjeg paddinga. Dugme se nije širilo u horizontalnom smjeru, jer bi to u horizontalnom menu-u pomicalo i susjedna dugmeta, a u vertikalnom stvaralo stršeci kraj dugmeta u koloni, što ne bi lijepo djelovalo.

Veličina dugmeta u pregledniku ne zavisi od veličine slike, nego je dužina definirana dužinom ispisa na njemu s dodatkom bočnih paddinga, pa zato na vertikalnom menu-u natpise treba ujednačiti tako da svi imaju jednako dugačak natpis (koristeći na primjer točke i prazna mjesta kao na slici).

Vertikalni raspored dugmeta u lijevom menu-u zahvaljujemo tagu `</br>` u html-kodu, koji je svaki link vertikalnog menua prebacio u novi red. Izvorno je visina dugmeta bila premala, što je povećano povećanjem visine retka (nalog `line-height:21px`). Time je visina dugmeta podešena tako da se gotovo dotiču kad kursor miša nije iznad njih.

Sa svega pet redaka naloga u css datoteci, tako je oblikovana funkcionalna i sasvim pristojna navigacija, no dotjerivanje zahtijeva dosta eksperimentiranja dok se pogode prave vrijednosti margina, paddinga i veličine prostora rezerviranih dimenzijama odlomaka **p** i **p.v**.

Početcnici mogu iskusiti dosta frustracija, dok ne "uhvate konce" i nadmudre eventualne mušice preglednika. Tako je primjerice jedan od trikova: Ako se neki nalog neće aktivirati, dobro ga je premjestiti na početak linije iza `{`, ili u prvi redak, tj. odmah iza početnog taga, pa će to možda natjerati preglednike da shvate "što je umjetnik želio reći".

Ako zapnemo s nekom deklaracijom koju preglednik nikako da shvati, dobro je pogledati u slične uzorke kako su to riješili drugi. No, može se i problematični element ubaciti u nekom objektnom programu za kreiranje web-stranica, pa onda u css-datoteci pogledati kako je program to riješio. No, prije toga napravimo kopiju css-datoteke, jer bi je objektni program mogao temeljito preraditi.

Svakako za dotjerivanje detalja, moramo imati otvorene html i css datoteke u tekst-editoru, a html datoteku i u pregledniku. Nakon svake korekcije u tekst-editoru fail treba spremiti i osvježiti preglednik

**PSEUDOELEMENTI** dodani iza selektora (najčešće iza pseudoklasa) počinju znakom `-`, a oblikuju dio nekog elementa u bloku (npr. odlomka). Najčešće se koriste:

**p:first-line** koji oblikuje prvi redak u bloku (ovdje u odlomku *p*). Pseudo-element je **-line**.  
**p:first-letter** prvo slovo u bloku (u ovom primjeru u odlomku)

U css-datoteci, iza pseudoelementa slijede uobičajene deklaracije oblikovanja unutar vitičastih zagrada – Evo primjera:

```
body {background-color:rgb(170, 200,254)}
p:first-letter{color:yellow;letter-spacing:-4px; font-size:50px}
p:first-line{color:yellow;font-size:20px; font-weight:bold}
p{font-style:verdana; font-style:italic;font-size:12px}
```

ako ovako oblikujemo css-datoteku  
Odlomak će u pregledniku izgledati ovako:

Nalogom **p:first-letter** oblikovano je prvo slovo. **Letter-spacing: -4px** smanjio je razmak iza slova *W*, jer je izvorno djelovao prevelik. Naredba **p:first-line** oblikovala je prvi redak. Plavu boju osigurala je prva deklaracija kao podlogu čitave stranice.



Usput, okanite se pisanja crnim slovima na tamnoplavoj pozadini i slične "estetike" kakve se nažalost nerijetko viđaju na web-stranicama, jer nitko nije toliki mazohista da se maltretira s odgonetavanjem vašeg nečitkog teksta.

**SELEKCIJA PO ATRIBUTU** još je jedna od naprednih mogućnosti, koja odabire elementa na koje će *css*-datoteka utjecati, navođenjem nekog atributa sadržanog u elementima dokumenata.

Općenito, u *css* datoteci, izraz **[atribut=vrijednost]** u **uglatoj zagradi** dodan iza selektora primijenit će deklaracije iz *css*-datoteke na sve elemente u kojima navedeni atribut ima navedenu vrijednost. Ponovo je ovo najbolje pojasniti na primjeru:

Želimo da sve interne poveznice našeg *html* dokumenta budu plavo, a one iz vanjskih izvora crveno ispisane. Takav zahtjev ispunit će slijedeće deklaracije u *css*-datoteci:

|  |   |
|--|---|
| <code>a{color:blue;}</code>  | nalaže plavu boju poveznica   |
| <code>a[href="http:"]{color:red}</code>  | nalaže da veze koje sadrže <code>http:</code> (tj. vanjske) budu crvene<br>Može se ići još detaljnije, pa tako drugi redak možemo pisati:   |
| <code>a[href^="http:"]{color:red}</code>   | Pri tome znak <code>^</code> bira samo one elemente, u kojima je <code>http:</code><br><b>na početku</b> lokacije, a to su opet vanjski izvori.   |
| <code>x[xxx~="yyy"]</code><br>između ostalih atributa i<br><code>x[xxx\$="zzz"]</code> | Za razliku od toga, ako je ispred jednakosti stavljen znak: <code>~</code><br><i>css</i> će se primijeniti i ako je "yyy" jedan od više atributa.<br><br>ako je ispred jednakosti <code>\$</code> <i>css</i> se primijenjuje na elemente čiji<br>atribut <b>završava sa ZZZ</b> . |

`a[href$=".pdf"]{color:green}` Ovaj nalog će na primjer, zeleno obojiti poveznice na PDF, a  
`a[href$=".xls"]{color:brown}` smeđe poveznice na XLS dokumente.

**CSS3** nudi još neke novije mogućnosti koje nisu od tolikog značaja, pa se s njima nećemo baviti.

## WEB-STRANICE S JEDNIM STUPCEM

Počtnici ponekad ostaju frustrirani pri pokušajima oblikovanja svojih uradaka, jer ne uspijevaju vladati pozicioniranjem elemenata web-stranice, već imaju dojam da elementi skakuću svagdje po stranici bez nekog suvislog pravila. Ranije se razmještanje elemenata sadržaja po prostoru web-stranica rješavalo oblikovanjem tablice sa ili bez bordera u koju su onda smješteni pojedini sadržaji (tekstovi, tabele, slike i dr.) ili ugradnjom okvira (*frame*) u stranicu (kao što je objašnjeno i u našem prvom tutorialu iz 2008.g.), no, danas se to rješava *css*-om, bez uplitanja oblikovanja u strukturu *html*-koda, kojim se danas definira samo sadržaj praktički bez oblikovanja..

To s jedne strane smanjuje ukupnu memorijsku težinu kodova koje treba učitati pri pozivanju web-stranice, a time se ubrzava učitavanje, a s druge strane imamo potpunu slobodu (pre)oblikovanja dizajna stranice bez diranja u sadržaj. Te su prednosti omogućile i današnju koncepciju izrade web-stranica na temelju gotovih uzoraka dizajna (*template*), u kojima korisnik treba samo unijeti svoje sadržaje (tekstove, slike, liste menu-naredbe i dr.). Na takvom principu u biti funkcionira i *WORDPRESS*, ali na Internetu se nude tisuće *layout*-a za besplatni download skupa sa pripadajućim *css*-datotekama. Izrada funkcionalne "konfekcijske" web-stranice time se značajno pojednostavljuje i bez specijaliziranih programa za izradu web-stranica, korištenjem *Notepad*-a ili baš web-dizajnu namijenjenog tekst-editora. Ako korisnik želi korigirati ponuđene detalje oblikovanja, razmješta i dr. web-dizajna, mora međutim poznavati *html* i *css*, i razumije se, mora imati puni pristup administriranju svog web-sjedišta na serveru odabranog providera (*CARNET* primjerice to ne dopušta).

Oblikovanje *css*-om iskorištava mogućnosti i svojstva taga `<div>` i identifikacije **ID**. Na strani 9 i 10 objašnjeno je, da `<div>` omogućuje oblikovanje grupe elemenata koj čine neku logičku cjelinu. Kako je i cijela web-stranica zapravo logička cjelina koju čine njezini segmenti: **zaglavlje, izbornik, sadržaj i podnožje**, to se i na nju može primijeniti uporaba `<div>` -a. Svi nabrojani elementi smješteni su na prostoru monitora, odnosno u pregledniku na prostoru prozora preglednika. Za taj prostor je u engleskom uvriježen pojam **container** koji analogno pravom kontejneru predstavlja kutiju u kojoj su smješteni svi nabrojani segmenti, pa i glavni segment **sadržaj**. Pri tome je "sadržaj" (engl: **content**) ustvari hrpa različitih elemenata (teksta, slika, listi i dr.).

**Pretpostavljeni parametri i centriranje stranice** Preglednici poštuju slijedeća pravila:

- širina stranice (bijelo područje na donjoj slici) je pretpostavljeno 100%, tj. četrdesetak px uža od rezolucije monitora za koju je stranica prvenstveno kreirana. Ta se širina upisuje kao širina kontejnera. Na primjer za monitor 1600x900, **#container{width:1560px}** daje širinu stranice

od 100%. Na manjim monitorima preglednici će automatski ugraditi klizače za dohvaćanje čitavog sadržaja, a na većima će stranica biti umanjena na nativnoj rezoluciji.

- Razlika od četrdesetak px koristi se za: pretpostavljenu marginu do ruba containera (dakle i monitora) i padding kontejnera (2 x desetak px). Eventualni klizači smanjuju korisni prostor za širinu klizača.
- Kod širine sadržaja od 100%, stranica (tj. container) će automatski biti centriran, jer su sve četiri margine i paddingi pretpostavljeno jednaki.
- Ako je veličina sadržaja manja od 100%, preglednik će ga pozicionirati lijevo gore. Za centriranje stranice manje od 100% treba stoga koristiti nalog: **#container{margin:auto}**.

Grafički prikazano, struktura web-stranice dakle općenito izgleda ovako:

Crtkani pravokutnik predstavlja granice monitora, (odnosno prozora preglednika)-dakle **container**, plavo je padding container-a, a žuti razmaci među sekcijama stranice definirani su gornjom marginom tih sekcija.

**Zaglavlje** u pregledniku uz naslov web-stranice sadrži "zaglavnu sliku", eventualno s par specifičnih linkova.

**Izbornik** (menu) je prostor rezerviran za poveznice na druge (pod)stranice ili dokumente korisnika.

**Content** sadrži ili prikazuje glavne sadržaje stranice.

**Podnožje** sadrži specifične korisnikove informacije poput datuma objave, autorskih prava i slično, tj. podatke koji nisu neposredno u funkciji sadržaja. Visina svih sekcija automatski se prilagođava količini upisa.



Pretočena na jezik html-a i css-a gornja grafička interpretacija (uključivo boje) će glasti:

u **html dokumentu** koristeći `<div>` i **ID**:

```
<html><head><title>NASLOV</title>      ← početni tag stranice i zaglavlja s naslovom
<link rel="stylesheet" href="prazno.css" type="text/css"> </head>      ← veza na css datoteku "prazno.css"
<div id="container">
  <div id="header">ZAGLAVLJE</div>      ← početni tag za definicije web-stranice,tj."container"-a
  <div id="nav">navigacija</div>      ← definirano je zaglavlje kao sastavnica web-stranice
  <div id="content">
    <p><span class="naslov">sadržaj</span> (content)</br>      ← definiran je prostor za horizontalni menu (navigaciju)
    (uključuje odlomke,slike,liste i dr.)</br> .</p></div>      ← definira prostor glavnih sadržaja web-stranice
    <div id="footer"></div>      ← u prvom retku koristi klasu ".naslov" za naglašavanje naslova, slijedi ostatak odlomka.
  </div>      ← definira prostor podnožja
</div>      ← završni tag kontejnera
```

Preglednici će na monitoru uz pomoć `<div>`-a kreirane segmente nizati redosljedom iz html-faila.

U kodu su tanko pisani atributi i tekst sadržaja iz konkretnog primjera. Identifikatori sekcija (container, header, nav, content i footer) bi se mogli pisati i hrvatski (bez hrvatskih znakova) jer kao identifikator mogu poslužiti proizvoljne riječi, ali u gotovim layout-ma će se vjerojatno najčešće koristiti navedena imena sekcija, pa se je bolje na njih naviknuti. Primijetimo da u html-kodu nema nikakvih oblikovanja.

Kompletno oblikovanje možemo definirati u css-datoteci koju smo u html-u nazvali "prazno.css":

### css-datoteka "prazo.css"

```
#container {width:300px; text-align:center;
  margin:auto; padding10px;
  background-color:aqua;
  border:1px dashed black;
  ← Širina stranice je manja od 100% prozora browsera,
  ← zato za centriranje stranice koristimo margin:auto
  ← Kontejner smo obojali samo da se zornije prikaže padding
  Ako se deklaracija izostavi, neće biti obruba

#header, #content, #footer{background:white;
  font-family:verdana; font-style:italic;
  text-align:center; margin-top:6px;
  padding:5px; border:1px solid black}
  ← Stavljamo bijelu pozadinu za navedene sastavnice stranice
  ← svi fontovi su skošena verdana pretpostavljene visine 11px
  ← svi su tekstovi centrirani, razmak među sekcijama je 6px.
  ← definira razmak od sadržaja do obruba i izgled obruba
  ako ne želimo obrub, deklaraciju border: treba izostaviti.

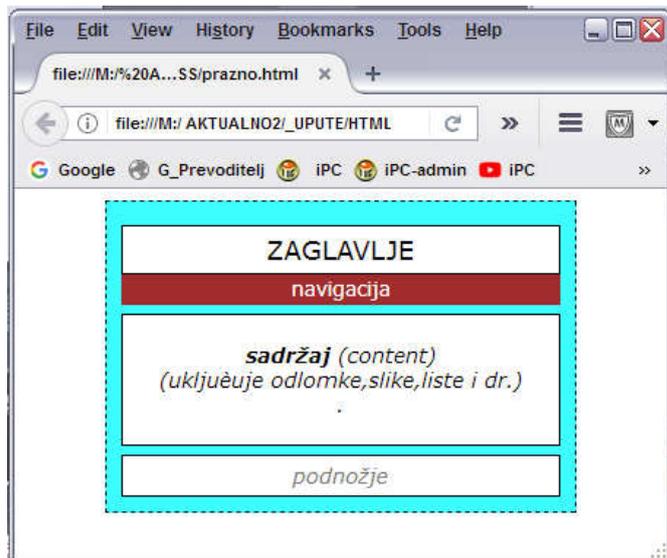
#nav{background-color:brown; color:white;
  padding:1px 0px 3px 0px}
  ← prostor izbornika je smeđ, a poveznice menua su bijele
  ← visina menu-trake se prilagođava s atributma paddinga

#header{font:x-large arial}
  ← definira masni i povečan ispis zaglavlja (tj.naziva stranice)
.naslov{font-weight:bold; font:Arial 1,3em}
  ← svi naslovi će biti podebljani i nešto povećani
#footer{color:gray}
  ← sadržaj podnožja bit će blijeđe ispisan od ostalih tekstova
  (tj. sivo). Ako ne želimo to, deklaraciju treba izostaviti.
```

Evo izgleda takve web-stranice u pregledniku (sa css-datotekom smo postigli ciljano oblikovanje):

Vidimo da je prostor stranice centriran u odnosu na raspoloživ prostor monitora, tj. lijeva i desna margina su jednake, što je postignuto nalogom **margin:auto**. Da smo širinu kontejnera postavili na 100% to bi bilo suviše, jer se u tom slučaju koriste pretpostavljene jednake sve četiri minimalne margine i paddingi kontejnera.

Preglednici automatski povećavaju visine elemenata koliko treba, zavisno od količine sadržaja u elementu i po potrebi dodaju bočni klizač kontejneru ako čitav sadržaj nije stao u prozor preglednika. No dizajner može predvidjeti zadanu visinu bilo kojem elementu nalogom **height: XXXpx**. U tom će slučaju preglednici automatski ugraditi klizače za dohvatanje sadržaja i tom elementu ako sadržaj nije u njega stao.



U ovom primjeru, svi su sadržaji web-stranice smješteni u jednom stupcu, što se u praksi često koristi za jednostavne pod-stranice, odnosno html-dokumente. Uvodne stranice web-sjedišta međutim u pravilu imaju složeniju strukturu, sadrže više izbornika, a središnji dio s glavnim sadržajima raspoređen je u više stupaca. U nastavku su date upute za izradu takvih web-stranica.

## WEB STRANICE S VIŠE STUPACA I RAZMJEŠTAJ ELEMENATA NA WEB-STRANICI

Središnji segment stranice, sadržavat će u pravilu dva ili tri stupca. Više od toga su rijetke iznimke. Lijevi stupac najčešće sadrži uski "sekundarni" vertikalno orijentirani izbornik, a desni stupac, ako se predvidi, može sadržavati još jedan vertikalni izbornik, ili posljednje objave, kakav reklamni sadržaj i slično. Svi su ti sadržaji najčešće ustvari također neke vrste izbornici, jer obično prikazuju sažetke u obliku teksta, slike ili njihove kombinacije, koje su ustvari poveznica na interni ili vanjski detaljni prikaz ili dokument. Slikovni sadržaji mogu biti animirani, najčešće u obliku reklamnog banera, kolekcije slika, slide-show-a koje se izmjenjuju automatski ili na klik i dr.

Desni ili središnji stupac se nastoji održati što širim za glavni sadržaj, koji na uvodnoj stranici može držati listu stalnih priloga ili detaljne udarne informacije, a u njega će se učitavati i svi sadržaji pozvani putem poveznica (doc, xls, pdf ili drugi dokumenti, slike, video sadržaji i dr.) ako se ne prikazuju u novom Internetskom prozoru ili na izvornoj lokaciji, o čemu odlučuje vlasnik stranice.

Prigodno, savjetujemo slijedeće:

- *Pretrpavanje web-stranice mnoštvom animiranih sadržaja s jedne strane stvara utisak zbrke koja otežava usredotočenost na predmet interesa, a s druge strane nepotrebno povećava vrijeme učitavanja i promet prema mreži i time smanjuje resurse za korisni sadržaj (npr. kvalitet video reprodukcije). Ne zaboravimo da je cilj web-stranice da pruži što kvalitetniju i ažurnu informaciju što većem broju posjetitelja, bez njihovog maltretiranja suvišnim radnjama, podmetanja neželjenog sadržaja i zbunjivanja nejasnim ili nesuvislim slijedom postupaka pri izboru ili preuzimanju sadržaja, kupovini ili plaćanju i dr. To su elementi pri koncipiranju layout-a i funkcioniranja web-sjedišta od presudne važnosti za popularnost, tj. posjećenost, te rejting vlasnika i ranking (položaj u rezultatima tražilica) web-stranice.*
- *Svakome je stalo da tražilice njegovu web-stranicu bolje rangiraju, tj. postave što višlje u rezultatima pretraživanja. Treba znati da programi koji skupljaju podatke na temelju kojih se određuje ranking (npr. Google-bot) ne "vide" slike i grafički dizajn, već se ocjenjivanje vrši na temelju tekstualnih sadržaja, ispravnosti poveznica, broju i važnosti posjetilaca, povezanosti s društvenim mrežama i ostalih statističkih pokazatelja. Stoga je potpuno pogrešno uvjerenje da će stranice s puno šarenila i animacije biti bliže vrhu rezulata pretraživanja. U pravilu, ako se stranica sviđa posjetiteljima, svidjet će se i tražilicama.*

No, da se vratimo na temu. Ovaj puta želimo web-stranicu sa sekcijama prema prethodnoj slici, samo u središnji - glavni prostor želimo smjestiti dvije kolone, odnosno shemi dodati lijevi menu. Možemo to i skicirati, no nakon kratkog iskustva dovoljna će biti i skica u mašti. Ta dodana lijeva kolona također dobiva status sekcije, koju jednostavno umećemo i u **kosturnu shemu**. U njoj sekcije trebaju imati redosljed kakav želimo u pregledniku, pa će kosturna shema izgledati ovako:

```

<!DOCTYPE HTML 5>
<html>
  <head><title>HOME</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="2stupca.css" type="text/css">
  </head>
  <div id="container">
    <div id="header"></div>
    <div id="nav"></div>
    <div id="left"></div>
    <div id="content"></div>
    <div id="footer"></div>
  </div>
</html>

```

- ← propisuje željenu razvojnu verziju HTML-a
- ← početni tag html-dokumenta (tj. web-stranice)
- ← početni tag zaglavlja s definicijom naslova stranice
- ← zajedno sa !DOCTYPE definira hrvatske znakove
- ← naznačuje pripadajuću css-datoteku
- ← zaključuje zaglavlje (ono se ne vidi u pregledniku)
- ← početni tag container-a (koji uključuje sve sekcije)
- ← zaglavlje stranice (uključuje naslov iz tittle i zaglavnu sliku)
- ← horizontalni izbornik
- ← ovo će biti lijev kolona (vertikalni menu i dodatni sadržaji)
- ← ovo će biti desna kolona (glavni sadržaji web-stranice)
- ← podnožje stranice (posebne napomene, poveznice i sl.)
- ← završni tag container-a (tj. svih sadržaja stranice)
- ← završni tag izvornog koda html-dokumenta

Žuti dio kosturne sheme definira sekcije stranice pa će se one u pregledniku unutar kontejnera nizati jedna ispod druge, redoslijedom iz sheme. Ako želimo da se dvije sekcije (kod nas **left** i **content**) postave u isti red, obje moramo "podignuti" kao da lebde iznad ravnine monitora nalogom **float**, i tako stvore prostor u kome se mogu smjestiti u isti "redak", poravnate gornjim rubovima. No, one će se u tom slučaju preklapati, jer se obje izvorno smještaju lijevo. Nalog **float** zato ima opcije **left** i **right**, pa jednoj "lebdećoj" sekciji nalogom **float:left** možemo propisati lijevo, a drugoj sa **float:right** desno poravnanje. Ako zbir njihovih širina ne prelazi širinu kontejnera, one će se postaviti jedna uz drugu s nezauzetom širinom između sebe. U protivnom, dijelom će se prekrivati.

Nadalje, ako nekoj koloni propišemo visinu 100%, protegnut će se do dna containera pa dolaze u koliziju sa 100% širokim podnožjem. Nalog **clear:left** ili **right** ili **both** unutar definicija podnožja zaustavlja širenje kolona prema dolje, pa omogućuje da footer zauzme punu širinu containera.

Za primjer ćemo kreirati stranicu **2stupca.html**, koristeći strukturu menu-a iz stranice **index.html** sa strane 10. U zaglavlje ćemo ubaciti "zaglavnu sliku", jer je slika zaglavlja praktički pravilo na suvremenom layout-u. Sekundarni, bočni izbornik smjestit ćemo u lijevi stupac, sa nekim dodatnim sadržajima, a razmještam sekcija omogućit ćemo postavu tamnosmeđe pozadine ispod oba menua. Izgled nove stranice u pregledniku prikazuje slika, a na sljedećim stranicama je html-kod i pripadna css-datoteke s objašnjenjima naloga.

slika zaglavlja

horiz. menu (**nav**)

verikalni izbornik

lijeva kolona (**left**)

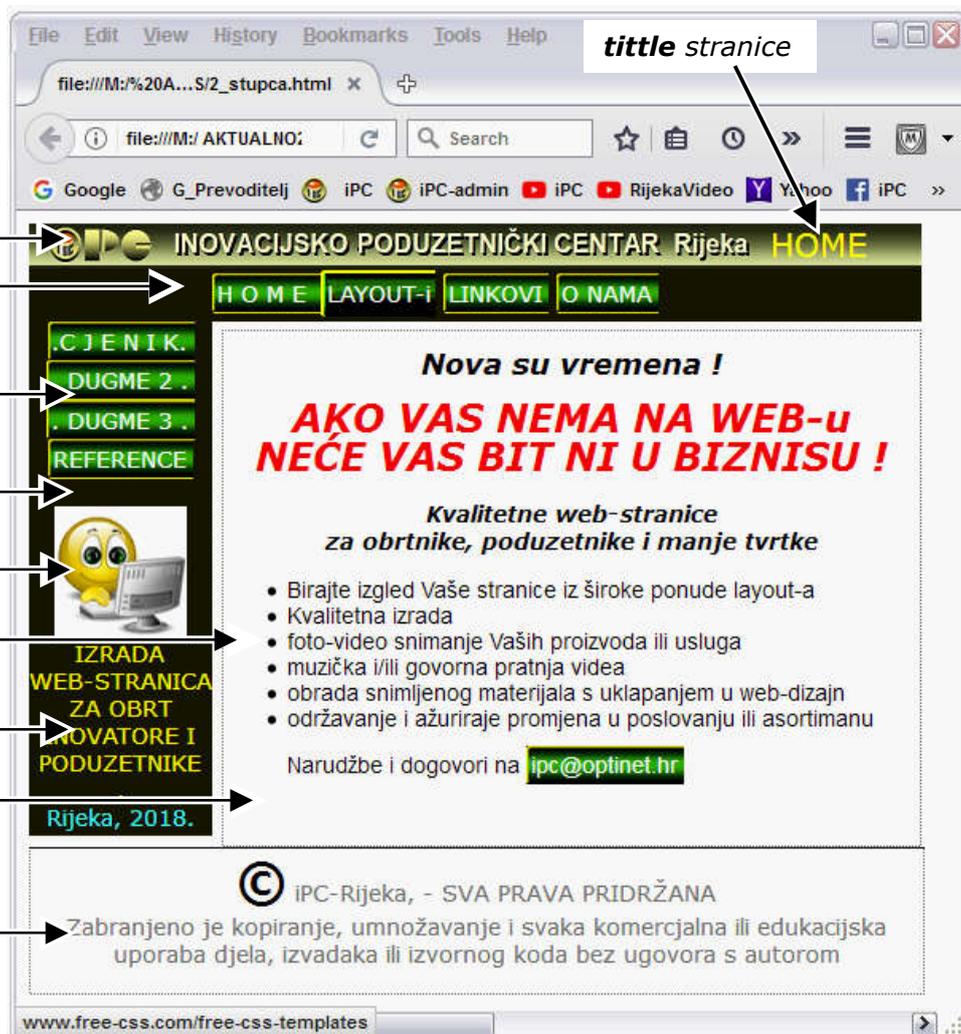
baner (GIF-animacija)

lista

reklamni odlomak

desna kolona (**content**)

podnožje stranice (**footer**)



Slika zaglavlja i naziv stranice (tittle) čine zaglavlje stranice u pregledniku. Linkove na još nepostojeće pod-stranice usmjerit ćemo privremeno na stranicu **uRadu.html**, koju treba posebno kreirati. U html-datoteci pokazuje se i kako se kreira lista, ma da se i ona mogla definirati u css-u.

U okviru je dio kosturne sheme koji definira redosljed sekcija stranice, a van okvira je detaljiran taj kostur sa uvodnom sekcijom, što skupa čini "izvorni kod" (source code), odnosno kompletan html-dokument. **Ni ovdje nije sadržano oblikovanje stranice.** Plavo su sadržaji, tj. tekstovi i podaci (poveznice) korisnika. Crveno su selektori, za koje se u css-datoteci definiraju oblikovni elementi. Uz **nazive sekcija** stranice, to su i uobičajene html-oznake kao **p** (odlomak), **h1**, **h2** i **h3** (naslovi i podnaslovi), ali i proizvoljne znakovne kombinacije kao **pv**, **pv1**, **pv2** (također označuju odlomke). Oblici naslova i podnaslova su preddefinirani, ali smo im u css-datoteci korigirali svojstva

Lista između tagova `<ul>` i `</ul>` za marker ima kružić, a svaki red se stavlja u tagove `<li>` `</li>`. Također je preddefinirana. Brojčane liste bi `<li>` `</li>` redove imale između tagova `<ol>` i `</ol>` i automatski bi mijenjale brojeve pri umetanju ili brisanju redaka u listi.

```

<!DOCTYPE HTML 5>
<html>
  <head>
    <title>HOME</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="2stupca.css" type="text/css">
  </head>
  <div id="container">
    <div id="header"></div>
    <div id="nav">
      <a href="2stupca.html">H O M E </a>
      <a href="http://www.free-css.com/free-css-templates">LAYOUT-i</a>
      <a href="uRadu.html">LINKOVI</a>
      <a href="onama.html">O NAMA</a></div>
    <div id="left">
      <a href="uRadu.html">.C J E N I K.</a></br>
      <a href="uRadu.html">. DUGME 2 .</a></br>
      <a href="uRadu.html">. DUGME 3 .</a></br>
      <a href="uRadu.html">REFERENCE</a></br>
      <p></p>
      <pv1>IZRADA </br>WEB-STRANICA</br>
        ZA OBRT I </br>PODUZETNIKE</br>.</br></pv1>
      <pv2>Rijeka, 2018.</p.v2></div>
    <div id="content">
      <h2>Nova su vremena !</h2>
      <h1>AKO VAS NEMA NA WEB-u</br>
        NEĆE VAS BIT NI U BIZNISU !</h1>
      <h3> Kvalitetne web-stranice</br> za obrtnike, poduzetnike i manje tvrtke</h3>
      <ul><li>Birajte izgled Vaše stranice iz široke ponude layout-a</li>
        <li>Kvalitetna izrada</li>
        <li>foto-video snimanje Vaših proizvoda ili usluga</li>
        <li>muzička i/ili govorna pratnja videa</li>
        <li>obrada snimljenog materijala s uklapanjem u web-dizajn</li>
        <li>održavanje i ažuriranje promjena u poslovanju ili asortimanu</li></ul>
      Narudžbe i dogovori na <a href="mailto:ipc@optinet.hr">ipc@optinet.hr</a>
    </div>
    <div id="footer"> <span id="vz">©;</span> IPC-Rijeka, -
      SVA PRAVA PRIDRŽANA</br>
      Zabranjeno je kopiranje, umnožavanje i svaka komercijalna
      ili edukacijska uporaba djela, izvadaka ili izvornog koda
      bez ugovora s autorom</div>
  </div>
</html>

```

postava hr znakova

```

<div id="container">
  <div id="header"></div>
  <div id="nav"></div>
  <div id="left"></div>
  <div id="content"></div>
  <div id="footer"></div>
</div>

```

← podvlakama je odmaknut menu u desno (ne reagira na log padding)

horizontalni izbornik

vertikalni izbornik

← baner

odlomci

naslovi i podnaslovi

LISTA

← formatiranje sa span

podnožje

← Završni tag containera je iza svih sadržaja, pa će oblikovanja pod identifikatorom **#container** važiti za elemente u svim sekcijama.

Oblikovanje stranice u cijelosti je riješeno u datoteci **2stupca.css** . kako slijedi:

```
#container{width:560px; height:320px}
#header {width:560px;height:28px; background-image:url(iPCglava.jpg)}
tittle    {width:560px; font-size:20px; font-style:arial; color:yellow;
           float:left; padding-left:465}

#nav{padding:3 0 0 115; width:445px; height:30px}      (px se podrazumijeva)
#nav, #left {color:white; background-color:rgb(36 36 18)}
a          {line-height:21px; color:white; padding:4 4 4 3;
           background-image:url(dugmeD.jpg);text-decoration:none}

a:hover   {padding:6 3 6 4; background-image:url(dugmeG.jpg)}

#left     {float:left; width:115px; height:100%; text-align:center}
p         {width:100%}

pv1       {font-size:12px; color:yellow}
pv2       {font-size:14px;color:aqua}
#content  {float:right; width:437px;height:100%; text-align:center;
           margin-top:6px; border: 1px dotted gray; font-style:italic;
           font-family:verdana}
h2        {line-height:10px;padding:0}
h1        {font:arial, sans-serif; color:red;line-height:26px}
ul        {text-align:left; font:0.6em verdana, sans-serif}

#footer   {width:560px; border:1px dotted gray;border-top:1px solid black;
           clear:both; font-size:11px;color:gray; text-align:center;
           padding-bottom:16px}
#vz       {font-size:300%; font-weight:bold; color:black}
.naslov   {font-weight:bold; font:Arial 1,3em} ljeno, bljeđi (sivi) ispis.
```

- ← Zadane su širina i visina web-stranice (containera)
- ← dimenzionirano je zaglavlje na 100% širine stranice, slika zaglavlja postavljena je kao pozadinska grafika
- ← postavljene karakteristike naslova. Širina 100%.  
Nalogom float-left naslov polopljen s retkom zaglavlja, i lijevim paddingom pomaknut na desnu prazninu slike.
- ← horizontalni menu, paddingom pomaknut u desno 115px
- ← za oba menua postavljena tamno smeđa pozadina
- ← Poveznice (oba izbornika) su bijele bez podvlake, na dugmetu ugrađenom u vidu pozadinske grafike. Visinu dugmadi definira visina retka (line-height), širinu definira dužina natpisa na dugmetu + lijevi i desni padding.
- ← Dugme nad kojim je kursor je "povećano" gornjim i donjim paddingom, sa pozadinskom slikom "dugme gore".
- ← lebdeća kolone široka 115px, visoke 100%, centriran tekst
- ← baner se postavlja u odlomak širok 100% širine kolone.  
Po visini zauzima prostor proporcionalan visini slike, pa je dovoljno navesti širinu. Proširit će se do paddinga kolone.
- ← formatira veličinu i boju teksta ispod baner-a
- ← formatira tekst na dnu lijeve kolone
- ← glavni sadržaj (content) kao lebdeća desna kolona postaviti će se uz lijevu klonu s menu-om. Visina je 100% stranice.
- ← text odmaknut od gornjeg ruba 5 px, lagano naznačen okvir, skošeno, centrirano pismo verdana
- ← Podnaslov h2 modificirane visine retka (10px) padding 0px
- ← crvena slova veličine naslova, smanj. visine redaka (26px)
- ← znakovna lista - lijevo poravnata verdana veličine 0.6em
- ← podnožje po cijeloj širini stranice, lagani obrub, gore puna crta. Clear:both prekida širenje obih kolona prema dolje.
- ← Lagani obrub, padding 16 na dnu stranice, središnje centr.
- ← Copyright znak povećan 300% crn, ostli tekst bljeđi 11 px.

Ovako pripremljena web-stranica može se također koristiti kao uzorak za izradu novih stranica, samo se plavi podaci trebaju korigirati i eventualno dopuniti konkretnim sadržajima vlasnika ili naručitelja

**Napomena: Materijal je zaštićen autorskim pravom, pa se svaka komercijalna (uklj. edukativna) uporaba mora ugovoriti s autorom.**



## Na kraju, još nekoliko napomena za praksu:

- Svi su primjeri u ovom tutorialu pisani u Notepad-u, dakle to se može, ali se preporuča za izradu web-stranica koristiti neki tome namijenjen teksteditor ili jošbolje alat. Za to ne treba koristiti Word ili slične sofisticirane tekstprocesore, jer bi na rezultat mogli utjecati macro-i i napredne opcije samog editora, pa bi neke funkcije koje rade pri kreiranju dokumenta mogle zakazivati u primjeni gotove web-stranice, oblikovanja u Word-u mogu se izgubiti pri otvaranju datoteka u drugim editorima itd.

No, zbog loše interpretacije hrvatskog znakovlja, nije uputno ni korištenje Notepad-a za iole ozbiljniji rad. Nedostatak tabulatora u Notepadu također je zapreka sažetijem strukturiranju izvornog koda, koje nam u početku olakšava snalaženje u html-u i css-u.

Specijalizirani alati za izradu web-stranica imaju ugrađene rutine za ispis učetalih slogova i naloga, mogu bojom razlikovati pojedine sastavnice koda, provjeriti uradak na više odabranih preglednika, "pročistiti" kod, uploadati uradak na server, ugraditi muzičku pratnju itd, pa sve u svemu olakšavaju i ubrzavaju kreiranje stranica. Profesionalci koriste relativno skupi **Adobe Dreamweaver**, no za početnike je više nego dovoljna besplatna verzija web-kreatora **CoffeeFreeHTMLEditor.exe** dostupnog na web-adresi: <http://www.coffeecup.com/free-editor>. Razumije se, da u njega možete učitati i bilo koji dokument, ili njegove ili tuđe predloške koje onda možete preraditi za svoje potrebe, no i za njegovo korištenje treba Vam barem znanje iz ovog tutoriala. Ovdje smo koristili samo najvažnija svojstva i naloge, a u Coffe-u ćete naći još mnoge druge, pa je to šansa i za usavršavanje u vlastitoj režiji (dakako uz konsultiranje Coffe-ovog Help-a).

- Strukturirani ispis html i css koda vrlo je pregledan i u početku će nam pomoći da se lakše snađemo u njemu. Ako međutim namjeravate koristiti uzorke s Interneta, preuzete stranice, tj. njihove css-datoteke neće biti strukturirane, već se uobičajeno daju u zbijenom obliku, bez ikakvih razmaka među redovima i kodovima, pa se na to treba naviknuti. Preglednici naime, kakosmo već rekli, ignoriraju praznine preko jednog blenk-a, pa one nisu potrebne zbog funkcionalnosti koda, nego samo zbog našeg lakšeg snalaženja u source-code-u.

Za primjer evo "kondenzirane" verzije css-datoteke našeg posljednjeg primjera. Ona je jednako funkcionalna kao i izvorna verzija sa prethodnih stranica, samo traži izvježbano oko za obradu.

```
#container{width:560px; height:320px} #header{width:560px; height:28px; background-image:url(iPCglava.jpg)} tittle{width:560px; font-size:20px; font-style:arial; color:yellow; float:left; padding-left:465} #nav{padding:3 0 0 115; width:445px; height:30px} #nav, #left {color:white; background-color:rgb(36 36 18)} a{line-height:21px; color:white; padding:4 4 4 3; background-image: url(dugmeD.jpg);text-decoration:none} a:hover {padding: 6 3 6 4; background-image:url(dugmeG.jpg)} #left{float:left; width:115px; height: 100%; text-align:center} p{width:100%} pv1{font-size:12px; color:yellow} pv2{font-size:14px;color:aqua} #content{float:right; width:437px; height: 100%; text-align:center; margin-top:6px; border: 1px dotted gray; font-style:italic; font-family: verdana} h2{line-height:10px; padding:0} h1{font:arial, sans-serif; color:red; line-height: 26px} ul{text-align:left; font:0.6em verdana, sans-serif} #footer {width:560px; border:1px dotted gray; border-top:1px solid black; clear:both; font-size:11px; color:gray; text-align:center; padding-bottom: 16px} #vz{font-size:300%; font-weight:bold; color:black}
```

- Na kraju, ako Vam ne uspijeva napraviti neki nalog tako da ga preglednici prihvate, možete ga ubaciti uz pomoć navedenog CoffeeFreeHTMLEditor-a, bolje nego uz pomoć kakvog objektnog programa poput npr. Kompozer-a. Ovi naime ne rade uvijek optimalizirane kodove, tj. znaju imati suvišnih deklaracija (npr. deklariraju i sve pretpostavljene naloge i sl.) a mogu Vam i drastično "preraditi" Vaš kod, čim ste ga učitali u njih, čak i ako ne spremite izmjene. Coffe-ov alat to neće napraviti. U svakom slučaju prije takvih intervencija spremite si sigurnosnu kopiju Vašeg koda s drugim imenom, za svaki slučaj.

Prilažemo pregledan popis najvažnijih svojstava koja se mogu oblikovati u css-datotekama, a velik broj drugih, može se naći u spomenutom Coffe-ovom alatu.

Ako Vam nešto od svega nije najbolje "sjelo", dobro je to pogledati i u nekom drugom tutorialu. Jedan od njih dostupan je na adresi: <http://www.izrada-webstranica.com>.

## ŠALABAKTER ZA OBLIKOVANJE

Umetanje slike u html: `<p></p>` (mail i link ide sa `<href="">`)

Povezivanje html-a sa css-datotekom: `<link rel="stylesheet" href="imeStila.css" type="text/css">`

U slijedećim izrazima: **S** je selektor, a **imeKlase** i **identifikator** su proizvoljne riječi.

Def. Klase u **html-u**: `<S.class="imeKlase">.....</S>`

Def. identifikatora: `<S id="identifikator">..... </S>`

Span: `<S>normalan tekst<span class="imeKlase">drukčiji</span>`

Div: `<div id="identifikator"><p> . . . . . </p></div>`

href Selekcija: oblikovanje se primjenjuje na href nalog ako je abc sadržan

u **CSS-u**: `S.imeKlase{oblikovanje}`  
`identifikator{oblikovanje}`  
`.imeKlase{oblikovanje}`  
`#identifikator{oblikovanje}`  
`S[href="abc"]{oblikovanje}`

### DEFINIRANJE PROSTORA

`<P></p>` odlomak. Dimenzioniranjem praznog odlomka može se npr. "rezervirati" prostor za sliku.

**border:** obrub (sam prima više svojstava)

**-weight:** debljina (px ili 100-700%)

**-color:** boja (opisno, rgb ili heksadecim)

**-style** vrsta linije obruba

**:solid** puna crta

**:dashed** crtkana

**:dotted** točkana

**:double** dvostruka

**-top, right, bottom, left** i kombinacije

`hr{width:50%}` horiz. crta (argumenti kao gore)

**margin** udaljenost do susjednog objekta ili ruba prostora (px ili %prostora) samo margin: definira sve jednake.

**-top:** gornja margina (T)

**-right:** desna (R)

**-bottom:** donja (B)

**-left:** lijeva (L)

**padding** odmak sadržaja od okvira (px)

**-top: -right: -bottom: -left:** kao gore

drugi oblik: **padding:(T R B L)**

px se podrazumijeva

### PISMO

**tittle** naslov web-stranice, dokumenta

**h** naslov (preddefinirana svojstva)

**h1** do **h9** podnaslovi (h3 je norm. velič.)

**font**

**-family:** oblik pisma (arial, tahoma i dr.) i grupe **serif** i **sans-serif**

**-size:** veličina (px ili % norm. ili em)

**-weight:** debljina px ili % norm, ili opisno **normal, bold, light, bolder, lighter**

**-style:** **none, underline, overline, italic**

**-color:** boja (opisno, rgb ili heksadecim)

**line-height:** visina retka (px)

**letter-spacing:** razmak među slovima

**word spacing:** raznakmeđu riječima

**list** liste ul znakovne, ol brojčane (u html-u redovi počinju s `<li>`)

**-style:** prihvaća više svojstava

**-style-type:** oznaka ispred retka

**-style-image:** stavlja sliku ispred retka

### text-align

poravnanje teksta:

**:left** lijevo

**:right** desno

**:center** centrirano

**-decoration** dopunsko oblikovanje:

**:underline** podvučen

**:overline** pod crtom

**:linetrough** precrtan

**:blink** bljeskajući

**:none** ništa ili prekid oblikovanja

**-indent** uvučen početak (px)

### poveznice

**a** poveznice (prihvaća više svojstava)

**:link** neposjećena veza

**:hover** veza ispod kursora

**:visited** posjećena veza

**:active** izgled dok je veza aktivna

### BOJE

**color:** boja opisno RGB heksadecimalno

**white** bjela **rgb(0,0,0)** **#ffffff**

**black** crna **rgb(255,255,255)** **#000000**

**gray** siva

**blue** plava

**navy** morn.pl.

**red** crvena **rgb(255,0,0)**

**silver** srebrena

**green** zelena

**yellow** žuta

**brown** smeđa

**orange** narančasta

**viola** ljubičasta

**transparent** prozirna

### TABELA

**table**

tabela

**.nav** naslovni red tabele

**.rest** ostatak tabele

**th** naslov tabele

**tr** redak tabele

**td** ćelija tabele

Ugadnja linijskog stila u html-datoteci:

`<selektor style="oblikovanje kao u css bez">`

Ugrađeni stil (ide između `<head>` tagova u html):

`<style type="text/css">S{oblikovanje}</style>`

**S** je (ili su) selektor(i). Stil djeluje na `<body>`