

Skripta iz Dizajna i razvoja veb strana

Vlahović Nikola
2015.

01 - Osnovni koncepti

- *Internet* je „mreža svih mreža”. To je globalna računarska mreža koja je javno dostupna i sastoji se od drugih međusobno pozevanih mreža.
- *Internet* je globalni informacioni sistem koji je logički povezan sistemom adresiranja preko TCP/IP protokola.
- *ARPANet* je preteča Interneta realizovana 60-ih godina kao projekat Ministarstva odbrane SAD-a.
- *NSFNet* nastaje 1986. kada National Science Foundation osniva 6 akademskih računarskih centara i povezuje ih NSFNet mrežom.
- *World Wide Web* je servis Interneta koji je nastao 90-tih, omogućio je popularizaciju i najširu upotrebu Interneta.
- *Protokol* predstavlja skup pravila i normi koji opisuje postupke potrebne za uspostavljanje komunikacije između računara u mreži.
- *TCP/IP* – skup protokola na kojima se zasniva Internet.
 - * *IP (Internet Protocol)* – Obavlja zadatke vezane za usmeravanje (rutiranje) paketa u mrezi.
 - * *TCP (Transmission Control Protocol)* – Obavlja kontrolu ispravnosti podataka na krajevima veze.
 - * *FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (Hyper-Text Transfer Protocol).*
- *Servisi interneta* – skup usluga koje se koriste na Internetu. Najpoznatiji servisi su Email, WWW i FTP.
- *World Wide Web* – Najpristupačniji i najzastupljeniji Internet servis. Nastao u CERN-u.
- WWW – funkcionalne komponente
 - * *Veb klijent (browser)* – Korisnik kuca URL adresu ili klikće na link tipa www.raf.edu.rs/studije/1
 - Koristeći *DNS server*, browser razrešava deo URL adrese koja se odnosi na naziv domena (raf.edu.rs) u IP adresu => 217.24.20.10 . Potom kreira HTTP zahtev i šalje ga preko Interneta na dobijenu IP adresu.
 - * *Veb server* – Pronalazi traženi resurs (studije/1) i vraća ga veb pretraživaču u obliku HTTP odgovora.
 - Veb klijent na osnovu odgovora sa veb servera prikazuje traženu stranu.
- Osnovne funkcije veb servera su da čuva veb strane i sadržaje uključene u njih, prihvata zahteve klijenata za isporučivanje veb strana, isporučuje strane klijentima (download) i dodatni sadržaj (slike, css, skripte), kao i da prihvata sadržaj od klijenata (upload). Obično radi na portu 80. Primeri su Apache i IIS.
- Veb klijent je najčešće *veb browser* (softver koji preuzima i prezentuje resurse na vebu) ili *veb crawler* (poznat kao i spider, robot ili bot, softver koji na sistematičan način pregleda sadržaje na vebu i koriste se za indeksiranje. Takođe proveravaju ispravnost linkova i HTML koda. Najpoznatiji je GoogleBot.)
- *HTML* je predefinisani jezik za opis strukture veb strane i njenih elemenata.
- *Veb sajt* predstavlja grupu uzajamno povezanih veb strana sa sadržajem (tekst, slika, animacija, video).

Adresiranje na Internetu

- Dvojni sistem adresiranja

* Numeričke IP adrese su 32-bit adrese koje se dodeljuju svakom računaru i mogu biti *statičke* ili *dinamičke*.

Primer : 212.231.12.23

* *Simboličke (domenske adrese)* : raf.edu.rs .

- Oba načina adresiranja su ravnopravna.

- *DNS (Domain Name System)* prevodi simboličku u numeričku adresu i obrnuto.

- *Domen* predstavlja ime veb sajta (npr. raf.edu.rs). Sastoji se od domena drugog nivoa (SLD) i domena najvišeg nivoa (TLD). Domeni najvišeg nivoa se dele na generičke domene (gTLD – com, net, org, edu, gov) i nacionalne domene (ccTLD – rs, co.rs, org.rs, edu.rs, in.rs).

- *URL (Uniform Resource Locator)* adrese omogućavaju da se precizno definišu lokacije dokumenata na Internet adresi. *Linkovi (hiperveze)* ka drugim dokumentima definišu se korišćenjem URL adresa.

protokol	adresa (IP ili domen)	putanja	resurs
http://	www.vebsite.com	vesti/tema_dana	index.htm

- *ISP (Internet Service Provider)* je kompanija ovlašćena za registrovanje domena.

- *Hosting* je zakup fizičkog prostora na veb serveru na koji se smešta veb sajt.

- *W3C* je organizacija koja se bavi standardizacijom veb tehnologija tipa (X)HTML, CSS, XML, SVG...

02 – Proces dizajna i razvoja veb strana

Faze:

1. Prikupljanje informacija (svrha, ciljevi, ciljna grupa sajta i njegov sadržaj)
2. Planiranje (mape sajta, određivanje tehnologija i alata koji će biti korišćeni u izgradnji)
3. Dizajn (crtanje izgleda sajta, kreiranje prototipa i pojedinačnih elemenata sajta)
4. Razvoj (kreiranje početne strane, šablona (ljuske) za ostale strane i implementacija dodatnih elemenata)
5. Testiranje i isporučivanje (testiranje formi i skripti, provera u raznim browserima, provera koda...)
6. Održavanje (stavljanje novih sadržaja, SEO optimizacija)

- Tehnologije za razvoj veb strana su HTML jezik, CSS, JavaScript, PHP, baze podataka.

- Alati mogu biti Notepad++, Dreamweaver, za blogove WordPress, za CMS WordPress i Joomla...

- *HTML* je predefinisani jezik za kreiranje veb strana. On opisuje strukturu informacija u dokumentu označavanjem pojedinih elemenata teksta kao linkove, zaglavlja, liste i sl. Dopunjuje taj tekst ubacivanjem interaktivnih formi, slika i dr. objekata. Piše se pomoću tagova <>. HTML tagovi pokazuju browser-ima kako tekst tj. sadržaj treba da bude prikazan. HTML standard održava W3C.

- *XHTML (Extensible)* zamenjuje HTML, sastoji se od HTML 4.01 kombinovanog sa sintaksom XML jezika.

03 – XHTML

- XHTML dokumenti moraju da imaju koreni element *html*. Svi ostali elementi su podelementi njega.

```
<html>
<head>
  <title>XHTML strana</title>
</head>
<body>
  <h1>Naslov strane</h1>
  <p>Kratak tekst u prvom pasusu</p>
</body>
</html>
```

Korišćeni tagovi:

- html
- head
- title
- body
- h1
- p

- Tagovi se koriste za označavanje delova dokumenata. Završni tag ima / u odnosu na početni (</h1>).
- DOCTYPE – deklaracija, *head* element daje informacije o strani, dok *body* element definiše sadržaj strane.
- Atributi detaljnije definišu elemente. Pišu se isključivo unutar početnog taga (ime_atr. = vrednos_atr.).
- * width="25%", align="right" ... Vrednost atributa se mora nalaziti između navodnika.

XHTML

```
<a href="http://www.raf.edu.rs">Informacije o Računarskom fakultetu</a>
```

Browser

[Informacije o Računarskom fakultetu](http://www.raf.edu.rs)

- DOCTYPE deklaracija je definisana pomoću DTD – Document Type Definition.
- DTD precizno opisuje dozvoljenu sintaksu i gramatiku XHTML markup jezika i ukazuje browser-u korišćeni standard za kreiranje veb strane. Svaka verzija HTML jezika ima svoju DOCTYPE deklaraciju.

- Postoje 3 DTD tipa:

- Strict isključuje korišćenje prezentacionih elemenata i koristi se zajedno sa CSS elementima.
- Transitional se koristi za iskorišćenje HTML prezentacionih osobina i podršku non-CSS browser-a.
- Frameset se koristi kada želimo da delimo prozor browser-a u više okvira (frame-ova).

```
<!DOCTYPE html PUBLIC
```

```
"-//W3C//DTD XHTML 1.0 <Strict/Transitional/Frameset>//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-<strict/transitional/frameset>.dtd"?>
```

- *Title* element se piše u okviru head dela i prikazuje se u naslovnoj liniji pretraživača.

```
* <title>Računarski fakultet – Beograd</title>
```

- *Favicon* (favorite icon) se prikazuje u adresnoj liniji, favorites, meniju, tabovima...

- *Meta* elementi se nalaze u *head* delu, sadrže informacije koje čitaju browser-i i editori veb strana.
`<meta name="keywords" content="lista ključnih reči" />`
`<meta name="description" content="Kratak opis dokumenta." />`
`<meta name="author" content="ime" />`
`<meta name="robots" content="noindex,nofollow" />` - Onemogućava robotu indeksiranje dokumenta.
`<meta http-equiv="content-type" content="text/html; charset=utf-8" />` - Korišćeni skup karaktera.
`<meta http-equiv="refresh" content="n;url=<strana>.html" />` - n je broj sekundi, url definiše stranu.
- *Heading* elementi se koriste za definisanje naslova i podnaslova. Pišu se u okviru body tela i postoji ih 6.
- *Specijalni karakteri* počinju znakom & i završavaju se znakom ; , npr. ` ` - razmak.
- *p* element se piše u body-u i koristi se za kreiranje paragrafa – pasusa. `<p>Tekst pasusa. </p>`
- *div* element predstavlja odeljak ili sekciju veb strane i može da sadrži bilo koji drugi element.
- *span* element omogućava da se izdvoji deo teksta ili dokumenta.
- *br* element omogućava prelazak u novi red. On je prazan element te se piše `
`.
- *Komentari* su nevidljivi za korisnike. `<!-- Komentar -->`
- `<pre>` element omogućava da se njegov sadržaj prikaže u originalom formatu (formatting).
- `<blockquote>` element obezbeđuje da se njegov sadržaj prikaže uvučeno u odnosu na ostale elemente.

04 - XHTML

- Linkovi ka drugim resursima definišu se korišćenjem URL adresa, koje mogu biti apsolutne ili relativne. -
- Link koji sadrži relativni URL `href="bit.html"` u dokumentu na lokaciji `"http://www.loc.com/index.html"` ukazuje na dokument koji se nalazi na adresi `"http://www.loc.com/bit.html"`.
- *img* element mora da ima *src* atribut koji ukazuje na URL lokaciju slike. *Alt* atribut definiše alternativni tekst za sliku ukoliko je link na sliku pogrešan, slika nije u nekom podržanom formatu ili nije učitana.
- *title* atribut definiše mouse-over tekst. *Width* i *height* nisu obavezni atributi koji unapred def. veličinu sl.

XHTML

```

```

- `<object>` obezbeđuje umetanje eksternih resursa tipa audio, video, Java apleta, ActiveX, PDF, Flash...
- * Omogućava proširenje standardne funkcionalnosti browsera preko helper aplikacija (plug-in).
- *Linkovi* omogućavaju povezivanje veb strana (linkovanje). XHTML oznaka za link (hipervezu) je `<a>`.
- * `Tekst ili slika linka`
- *Bookmark* (obeleživač) je imenovana lokacija unutar veb strane i može postati odredište hiperveze.
- Postoje 3 vrste lista u XHTML-u : nenumerisane `` , numerisane `` i liste definicija `<dl>`.
- *Table* element se koristi za prikaz podataka u tabelarnom obliku. `<table><tr><td>..</td></tr></table>`.
- *colspan* atribut omogućava spajanje ćelija u kolonama, dok *rowspan* spaja ćelije po redovima.
- *cellpadding* predstavlja razmak između sadržaja ćelije i ivice ćelije. Nije podržan u HTML5.
- *cellspacing* predstavlja razmak između ćelija u tabeli. Nije podržan u HTML5.
- Za podelu tabele u horizontalne sekcije koriste se atributi *thead*, *tfoot* i *tbody*.
- *iframe* element definiše *inline frame* koji sadrži drugi resurs. Nije podržan u Strict DTD.

05 – XHTML

- *form* element se koristi za kreiranje XHTML forme za unos podataka i omogućava komunikaciju user/server.

```
<form action=URL method=POST|GET name=ime_obrasca>
    ...form polja
</form>
```

- *action* atribut sadrži URL na koji će biti poslani podaci sa obrasca.
- *method* atribut može imati vrednost POST ili GET.
- *_name* atribut je važan ako se koristi skript na strani klejnta.
- *fieldset* form element omogućava logičko grupisanje elemenata na formi. *Legend* tag definiše naslov.

```
<fieldset>
    <legend>PODACI</legend>
    Ime: <input type="text" size="30" /><br />
    Email: <input type="text" size="30" /><br />
    Datum rođenja: <input type="text" size="10" />
</fieldset>
```

- *textbox* form element omogućava unos linije teksta. Vrednost atributa *value* se prosleđuje do servera.

```
<input type="text"
    name="ime elementa"
    size="širina textbox-a u karakterima"
    maxlength="maksimalni broj karaktera u textbox-u"
    value="tekst koji će inicijalno biti prikazan"
/>
```

- *password* form element je isti kao textbox, pri čemu se tekst prikazuje kao niz zvezdica/tačaka.
- *radio* form element omogućava odabir jedne od više ponuđenih opcija.
- *checkbox* form element omogućava odabir više ponuđenih elemenata.

```
<input type="checkbox" ili "password" ili "radio"
    name="ime elementa"
    value="vrednost koja će biti poslata na server"
    checked="checked"
/>
```

- *select* form element omogućava da se korisniku ponudi skup opcija u obliku padajuće liste.
- *textarea* form element omogućava unos više redova teksta.
 - * `<textarea name="ime" rows="n" cols="n">Tekst.</textarea>`
- *file* form element omogućava da se datoteka prenese na server (upload). Form metoda mora biti *POST*.
- *hidden* form element omogućava čuvanje vrednosti, nevidljiv je korisniku.
- Postoje 3 button form elementa:
 1. *submit* dugme inicira prenos podataka sa forme kod klijenta na server.
 - * `<input type="submit" name="ime" value="Posalji" />`
 2. *reset* dugme vraća vrednost elemenata forme na njihove početne vrednosti.
 - * `<input type="reset" name="ime" value="Ponisti" />`
 3. *button* dugme se koristi kada je potrebno da se izvrši skript kao posledica pritiska na dugme.
 - * `<input type="button" name="ime" value="Izvrsi" onclick="sciprt" />`
- Preporuka je proveravati polje po polje i još jednom proveriti sve podatke posle klika na submit dugme.

06 – HTML5

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Osnovna struktura HTML5 dokumenta</title>
</head>
<body>
  <h1>....</h1>
  <p>...</p>
</body>
</html>

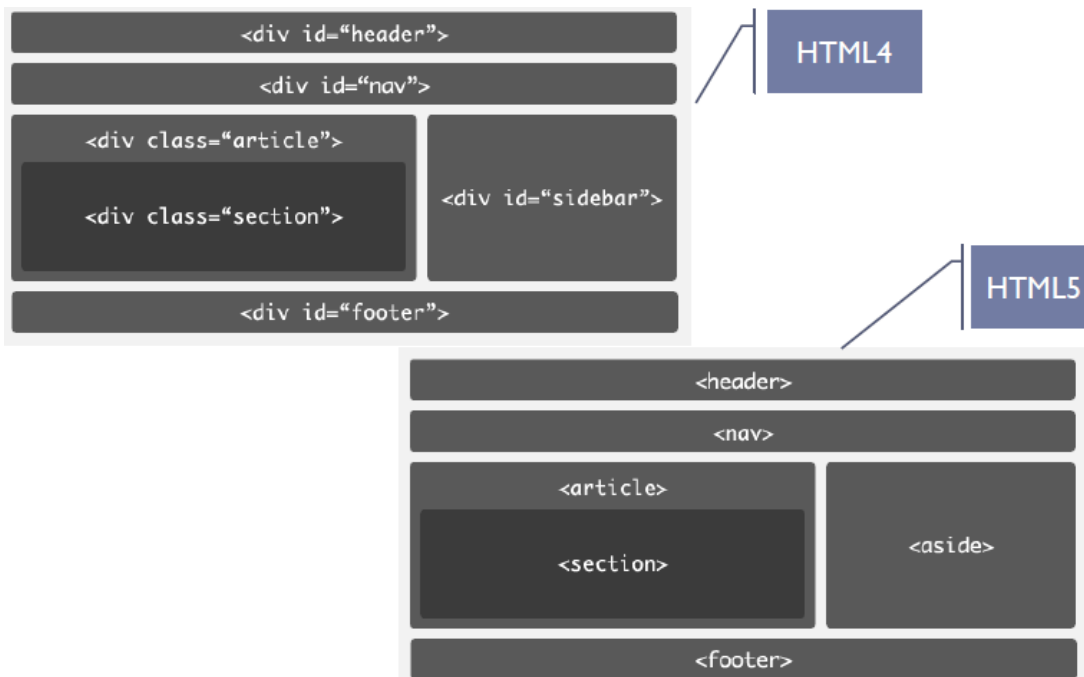
```

- Umesto: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- U HTML5: <!DOCTYPE html>

- Umesto: <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

- U HTML5: <meta charset="UTF-8">



- `<section>` omogućava grupisanje sadržaja kao `<div>`. Obično ima `<header>` i `<footer>` elemente.
- `<article>` element predstavlja deo strane koji može postojati samostalno tipa posta ili komentara na blogu.
- `<nav>` je rezervisan za deo dokumenta koji sadrži linkove ka drugim stranama ili na delove iste strane.
- `<aside>` predstavlja sadržaj povezan sa glavnim delom strane. Ovde mogu biti npr. ankete, related posts i sl.
- `<header>` element sadrži uvodne informacije za sekciju ili stranu.
- `<footer>` je element za označavanje podnožja i to ne samo cele strane, već i svakog dela koji se nalazi na str.
- `<video>` i `<audio>` elementi omogućavaju ugradnju video i audio elemenata.
- `Canvas` element omogućava crtanje u okviru definisane bitmap oblasti (Ostali API su npr. Drag and Drop i sl.)
- `polyfills` predstavljaju kod koji po potrebi emulira (oponaša) nepodržane HTML5 funkcije u browser-ima.

07 – CSS

- Cascading Style Sheets sadrži definicije stilova koji se primenjuju na elemente HTML dokumenata.
- CSS stilovi definišu kako će dokumenti biti prikazani korisnicima.
- CSS se predstavlja kao skup pravila. Svako pravilo uključuje selektor i deklaraciju (*atribut:vrednost*).

```
selektor
{
  atribut1: vrednost1;
  atribut2: vrednost2;
}
```

```
body
{
  background-color: #c0c0c0;
}
```

- Nasleđivanje je način prosleđivanja osobina roditelja na decu. Bez nasleđivanja bilo bi potrebno kreirati više CSS pravila za svaki HTML element na strani. Pravilo CSS-a je da poslednje podešavanje atributa zamenjuje prethodno.

- CSS boje se najčešće definišu pomoću količine crvene *R*, zelene *G* i plave *B* boje.

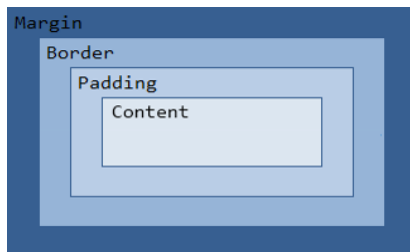
- * U procentima (0%-100%) : `color:rgb(35%,0%,50%)`
- * Pomoću decimalnih brojeva (0-255) : `color:rgb(89,0,127)`
- * Pomoću heksadecimalnih brojeva (00-FF) : `color:#59007f`
- * preko imena : `color:black`;

- Apsolutne jedinice mere su *in* (*inch*), *cm*, *mm*, *pt* (*point*), *pc* (*pica*) i *px* (*pixel*). Sve ostale tipa % su relativne.

- Web Safe fontovi su fontovi koji su instalirani na većini operativnih sistema.

- Browser tretira svaki element strane kao skup od 4 ugnježdene boksa. CSS boks model omogućava da se odredi veličina svakog boksa i na taj način menja izgled svakog elementa na strani.

* Svaki blok element u dokumentu ima sledeće osobine : *width*, *height*, *padding*, *margin* i *border*.



HTML elementi su ili blok ili inline elementi.

Karakteristike blok elemenata:

- Uvek počinju u novoj liniji.
- Širina je podrazumevano 100% u odnosu na roditeljski element.
- Mogu se podešavati margine i padding.
- Uobičajeno, mogu sadržati inline elemente i druge blok elemente.

Primeri : `<div>`, `<p>`, `<h1>`, `<form>`, ``, ``

Karakteristike inline elemenata:

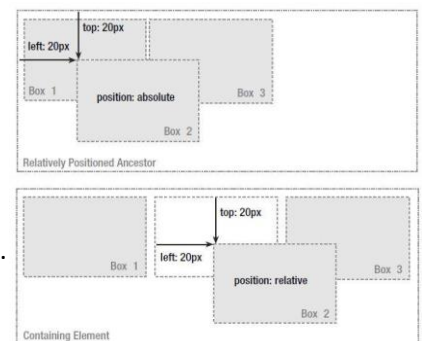
- Počinju u istoj liniji.
- Visina se ne može menjati.
- Širina elementa je širina teksta/slike i ne može se menjati.
- Uobičajeno, mogu sadržati podatke i druge inline elemente.

Primeri : ``, `<a>`, `<label>`, `<input>`, ``, ``, ``

- `display:block`; pretvara inline element u blok element, dok `display:inline`; radi obrnuto.

Grupe CSS atributa:

- * Font and text (`font-family`, `font-size`, `font-style`, `font-weight`, `text-align`, ...)
- * Color and background (`color`, `background-color`, `background-image`, ...)
- * Layout (`border`, `margin`, `padding`, ...)
- * Classification (`display`, `list-style`, `list-style-image`, ...)
- * Positioning (`position`, `visibility`, `height`, `width`, `z-index`, ...)
- Pozicioniranje elemenata može biti apsolutno (*gore*), relativno, fiksno i statično.
- (`float`; `clear`;) *overflow* – šta se vidi ako element prelazi veličinu okvira elementa.



08 – CSS

- HTML element kao selektor:

```
/* selektor je jedan HTML element */
body
{
    font-family: Arial, Helvetica, sans-serif;
    font-size: normal;
    background-color: #e5e5e5;
    margin: 0px;
    scrollbar-base-color: blue;
}

/* grupisanje selektora sa istim CSS atributima pomoću zareza */
h1, h2
{
    font-size: 14px;
    font-weight: bolder;
    color: #ff0000;
}

/* drugi nivo stavki u okviru nenumerisane liste */
ul li li
{
    color: red;
}

/* primenjuje se na sve p elemente čiji je roditelj div element */
div > p
{
    font-size: 10px;
}
```

- Klasa kao selektor (tačka ispred vrednosti *class* atributa) omogućava stilizovanje bilo kog HTML elementa.

```
.crvenaslova {
    color: #ff0000;
}
.pozadina {
    background-color: silver;
}
p.specijal {
    font-family: verdana, arial, sans-serif;
}

<p class="crvenaslova">Neki tekst 1</p>
<div class="crvenaslova pozadina">Neki tekst 2</div>
<p class="specijal">Neki tekst 3</p>
```

- Pseudo-klase su kreirane kako bi se omogućio opis različitih stanja koja se mogu pojaviti kod nekih elemenata.

```
/*
stilovi za linkove treba da budu definisani uvek u navedenom redosledu,
kako bi se izbeglo prepisivanje svojstava kada se link nalazi u više stanja
(na primer, visited i hover)
*/
a:link
{color: red;}
a:visited
{color: orange;}
a:focus
{color: purple;}
a:hover
{color: green;}
a:active
{color: blue;}
```

- ID kao selektor (# ispred vrednosti za *id* atribut). *#unique { font-size: 10px; }*

- Univerzalni selektor (oznaka je * - omogućava da se odjednom podese svi elementi koji su u okviru drugog).

```
/* biće primenjen na sve elemente u dokumentu */
*
{
    margin: 0; padding: 0;
}

/* svi elementi unutar div elementa imaju žutu pozadinu */
div *
{
    background-color: yellow;
}
```

Načini za primenu stilova:

- Povezivanjem : definicije stilova su u odvojenom CSS dokumentu – eksterni stilovi.

* <head> <linkrel="stylesheet" type="text/css" media="output" href="stil.css" /> </head>

- Ugrađivanjem : definicije stilova su sastavni deo HTML dokumenta – interni stilovi

```
<head>
  <style type="text/css">
    <!--
    body {
      font-family: Arial, Helvetica, sans-serif;
      font-size: normal;
      background-color: #e5e5e5;
      margin: 0px;
      scrollbar-base-color: blue;
    }
    -->
  </style>
</head>
```

- Inline : definicija stila se vezuje za pojedinačni element.

* <h1 class="crnaSlova">Tekst</h1> ; <h4 id="unique">Primer za ID selektor</h4>

CSS dozvoljava primenu vise stilova na isti document. CSS Hijerarhija je sledeća:

1. Stil označen sa **!important**
2. Inline stil (pomoću style atributa)
3. Ugrađeni stil (pomoću style elementa)
4. Eksterni stil (povezivanje pomoću link elementa)
5. Podrazumevano podešavanje browser-a

p {color: blue !important;}

čak iako je napisano <p style="color: red">

pasus će imati plavu boju, jer !important obezbeđuje da se pravilo ne prepíše

- Ukoliko postoji više stilova sa istom težinom, primenjuje se poslednji definisan.

* <style type="text/css"> p{color:purple;} p{color:green;}</style> => **Boja će biti zelena.**

- Ukoliko postoji više stilova za isti element sa različitom težinom, primenjuje se najspecifičniji stil :

* ID selektor > Class selektor > Kontekstualni selektori (pr. ul li, blockquote p) > Selektori za individ. elemente.

p {color: rgb(0,0,255);} blockquote p {color:rgb(255,0,0);} p.intro {color:rgb(0,255,0);} => **Boja je (0,255,0).**

- Layout – određivanje izgleda i rasporeda elemenata na strani: Koriste se %, px i em jedinice mere.

* Liquid (fluid) strane menjaju veličinu zajedno sa prozorom browser-a,

* Fixed strane gde se sadržaj postavlja u zonu strane koja ostaje fiksna bez obzira na dimenzija browser-a.

* Elastic strane sadrže delove koji postaju manji ili veći kako se menja veličina teksta.

- Pristupi veb strana za sve ekrane:

* Responsive gde se isti kod (HTML i CSS) isporučuje razl. uređajima preko jedne URL adrese, strana se prilagođava razl. veličinama ekrana dok se CSS koristi da promeni prikaz veb strane na uređaju.

* Dinamički gde se različit kod isporučuje razl. uređajima preko jedne URL adrese na osnovu zahteva.

* Posebni sajtovi gde se u odnosu na desktop verziju mobilna adresa sajta nalazi na drugoj URL adresi.

- CSS media query omogućava prilagođavanje CSS-a svakom uređaju, sastoji se od tipa medija i više izraza.

- Izrazi proveravaju uslove određenih karakteristika medija (npr. max-width: 400px) i ako je true koristi se određeni stil.

09 – CMS sistemi

- Content Management tj. upravljanje sadržajem predstavlja skup funkcija za kreiranje, pamćenje, uređivanje, upravljanje i objavljivanje sadržaja.

- CMS sistem predstavlja skup softverskih alata koji podržavaju funkcije za upravljanje sadržajem.

- Odlike sistema su jednostavnost, kontrola pristupa i izmene sadržaja, tekst i ostali sadržaji su u bazi podataka, strane se kreiraju korišćenjem sadržaja i šablona i laka prilagodljivost za razne svrhe korišćenja.

Vrste CMS-a:

* Sistemi za upravljanje dokumentima : Document Management System – DMS

- Automatizovana kontrola el. dokumenata (skenirani papirni dokumenti, el. dokumenti tipa Word, Excel...)

- F-je sistema su identifikacija dokumenta, smeštanje u repozitorijum, praćenje, kontrola verzija i prezentovanje.

* Sistemi za upravljanje digitalnim sadržajima – DAM

- Podskup sistema za upravljanje dokumentima, obrađuju uglavnom binarne dokumente (slike, audio i video).

* Sistemi za upravljanje obrazovnim sadržajima – LCM

- Upravljaču i isporučuju sadržaj za elektronsko učenje (eLearning). Kombinuju Word, PowerPoint, Flash, mp4.

* Sistemi za upravljanje veb sadržajima – web CMS

- Nadograđuju druge sisteme za upravljanje sadržajima, dodaju slot koji omogućava publikovanja sadržaja.

- Obuhvataju alate za kreiranje sadržaja, publikovanje i arhiviranje. (Na dalje pojam CMS podrazumeva web CMS.)

- What you see is what you get editori omogućavaju developeru da odmah vidi rezultat koda, dok ga piše.

Osnovne osobine CMS sistema:

- Korisnički interfejs za prikaz sadržaja.

- Veb interfejs za administraciju.

- Odvojenost sadržaja i interfejsa.

- Sadržaj se uglavnom čuva u bazi podataka.

- Interfejs se čuva u posebnim fajlovima.

Napredne osobine CMS sistema:

- Sistem šablona za lakše menjanje izgleda i rasporeda korisničkog i administratorskog interfejsa.

- Online uređivanje teksta pomoću WYSIWYG editora.

- Mogućnost modularnog proširenja sistema.

- Upravljanje slikama i ostalim datotekama iz administratorskog interfejsa.

- Prednosti CMS-a su brza isporuka sadržaja, njegov prikaz na razl. uređajima i platformama, unapređena optimizacija za robote, povećana produktivnost i posećenost sajta, unapređen integritet sajta.

- Mane su mu troškovi implementacije i održavanja, kao i problem sa kašnjenjem kod većih sistema.

Tipovi CMS sistema:

* Kompletan CMS sistem : Joomla!, Drupal.

* CMS za blogove : WordPress.

* CMS za wiki-je : MediaWiki.

10 – SEO – Search Engine Optimization

- Predstavlja skup metoda i tehnika koje olakšavaju mašinama za pretraživanje (Google, Bing) da pronađu, indeksiraju, kategorizuju i rangiraju veb sadržaj. SEO ne treba mešati sa SEM (Search Engine Marketing).
- SEM obezbeđuje pronalaženje veb strana preko plaćenih rezultata pretraživanja (sa desne strane Google).
- Cilj je pozicioniranje što bliže vrhu rezultata pretraživanja, što obezbeđuje veći broj posetilaca sajta.
- SEO se primenjuje zbog generisanja velikog saobraćaja bez trošenja novca za svakog posetioca.
- Mašine za pretraživanje na osnovu upita korisnika pretražuju svoju bazu podataka (indeks) i vraćaju listu sajtova koji odgovaraju zadatom upitu. One na razl. načine indeksiraju i rangiraju sadržaj strana (a ne sajtova).
- Pre nego što pretraživač može da da rezultat, on mora da ga pronađe. Mašine koriste specijalne softverske robote tj. paukove koji prave listu reči nađenih na veb stranama. Indeksi se dodaju u baze podataka i periodično ažuriraju. Kada korisnik unese pojam, pretražuju se indeksne baze podataka (be pretražuju se veb strane u realnom vremenu) i rezultati se rangiraju pomoću algoritama.

2 načina primene SEO optimizacije: on page optimizacija (levo) i off page optimizacija (desno).

- | | |
|---|---|
| ▶ pravljenje jedinstvenih, preciznih naslova veb strana | ▶ Prijavljivanje mašinama za pretraživanje |
| ▶ korišćenje "description" meta taga | ▶ Prijavljivanje direktorijumima za pretraživanje |
| ▶ unapređenje strukture URL adresa | ▶ Prijavljivanje članaka socijalnim mrežama |
| ▶ kreiranje mape veb sajta | ▶ Saopštenja za javnost |
| ▶ pružanje kvalitetnog sadržaja i usluga | ▶ Kreiranje Wiki profila |
| ▶ pisanje boljeg teksta za linkove | ▶ Strane na socijalnim mrežama |
| ▶ optimizacija korišćenja slika | ▶ Blog Hosting |
| ▶ odgovarajuće korišćenje "heading" tagova | ▶ Oglasi |
| ▶ efikasno korišćenje datoteke "robots.txt" | |

// Na dalje su samo preporuke šta raditi i izbegavati pri pisanju naslova, teksta, slika...

- Podešavanje atributa rel u okviru linka na vrednost "nofollow" govori robotima da ne prate link.

```
<a href="http://www.shadyseo.com/" rel="nofollow">Comment spammer</a>
```

- Google Search Console je alat koji vebmasterima omogućava da bolje optimizuju svoju stranicu za search.
- Google analytics omogućava analizu posećenosti sajta (kako korisnici stižu do sajta, najpopularniji sadržaj...).
- Google Website Optimizer omogućava izvršavanje eksperimenata na strani da bi došlo do bolje posećenosti.

11 – XML – Extensible Markup Language

- XML je jezik koji definiše strukturu po kojoj se podaci prosleđuju bilo kojoj aplikaciji za potrebe čuvanja, obrade ili prikaza. Može biti:
 - * Standardan dokument smešten na disku.
 - * Poruka koja se šalje preko HTTP-a.
 - * Znakovni niz u programskom jeziku.
 - * Objekat u bazi podataka.
 - * Bilo koji drugi način korišćenja tekstualnih podataka.
- XML je najviše korišćen standard za opis sadržaja i strukture dokumenata i njihovu razmenu na veb-u.
- XML omogućava razmenu podataka između nekompatibilnih sistema, podaci se čuvaju u običnom tekst (plain text) formatu, što omogućava razmenu podataka nazeavisnu od hardvera i softvera.
- Koristi se za čuvanje podataka u datotekama ili bazama podataka, kao i za kreiranje i opis novih jezika.

```
<?xml version="1.0" encoding="UTF-8" ?>
<studenti>
  <student>
    <brojDosijea>S-01/15</brojDosijea>
    <prezime>Jovanović</prezime>
    <imeOcalliMajke>Petar</imeOcalliMajke>
    <ime>Jovana</ime>
    <pol>ženski</pol>
    <godinaRodjenja>1996</godinaRodjenja>
  </student>
  <student>
    <brojDosijea>S-02/15</brojDosijea>
    <prezime>Petrović</prezime>
    <imeOcalliMajke>Dušan</imeOcalliMajke>
    <ime>Petar</ime>
    <pol>muški</pol>
    <godinaRodjenja>1996</godinaRodjenja>
  </student>
</studenti>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ...
</bookstore>
```

Title	Author
Everyday Italian	Giada De Laurentiis
Harry Potter	J. K. Rowling
XQuery Kick Start	James McGovern
Learning XML	Erik T. Ray

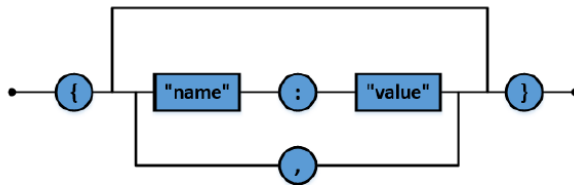
- XML je kreiran da bi se omogućio opis strukture teksta, dok je cilj HTML-a da omogući prikaz teksta.
- Tagovi i struktura HTML dokumenta su unapred definisani, dok XML nema utvrđen skup tagova.
- Element se sastoji od početnog taga, završnog taga i sadržaja (teksta) između.
- Svi moderni pretraživači imaju ugrađen XML parser koji konvertuje XML tagove u XML DOM.
- XML Document Object Modelom se manipuliše pomoću JavaScript-a. Izuzetak je CDATA.
- XSD (Schema Definition Language) jezik opisuje strukturu XML dokumenta. Pre njega koristio se DTD.
- XSD se piše korišćenjem XML jezika. XML elementi u XML semi mogu biti opisani pomoću:
 - * Prostih tipova koji sadrže samo podatke. <xs:element name="xxx" type="yyy"/> (string, decimal, INT..)
 - * Složenih tipova prazni elementi, elementi sastavljeni od drugih elemenata, elementi koji sadrže tekst...
- XML element koji sadrži atribut je complex tip, dok su atributi uvek tipa simple.
- XSD – veza sa XML dokumentom:
 - * Kao inline šema – XSD šema može biti sastavni deo XML dokumenta koji opisuje. (leva slika gore)
 - * Kao eksterna šema – XSD šema može biti u odvojenom dokumentu u odnosu na XML dok. koji opisuje.
- XML u HTML dokumentu se može prikazati korišćenjem CSS, JavaScript-a ili XSLT-a.

JSON – JavaScript Object Notation

- Jednostavan tekst format za razmenu podataka. Standard za razmenu između veb servera i veb klijenta.
- Za ljude je jednostavno čitanje i pisanje, dok je za mašine jednostavno generisanje i parsiranje.
- Posедуje isti potencijal za razmenu podataka kao i XML, ali su JSON dokumenti manji pa je dominantniji.

► Osnovna pravila:

- Podaci su definisani parovima *name:value*
- Parovi *name:value* razdvojeni su međusobno dvotačkom
- Parovi *name:value* razdvojeni su zarezima
- Sva imena i sve vrednosti moraju se nalaziti u okviru duplih navodnika
- Lista parova *name:value* mora se nalaziti unutar vitičastih zagrada



► JSON primer

```
{
  "studenti": {
    "student": [
      {
        "brojDosijea": "S-01/15",
        "prezime": "Jovanović",
        "imeOcalliMajke": "Petar",
        "ime": "Jovana",
        "pol": "ženski",
        "godinaRodjenja": "1996"
      },
      {
        "brojDosijea": "S-02/15",
        "prezime": "Petrović",
        "imeOcalliMajke": "Dušan",
        "ime": "Petar",
        "pol": "muški",
        "godinaRodjenja": "1996"
      }
    ]
  }
}
```

► XML primer

```
<?xml version="1.0" encoding="UTF-8" ?>
<studenti>
  <student>
    <brojDosijea>S-01/15</brojDosijea>
    <prezime>Jovanović</prezime>
    <imeOcalliMajke>Petar</imeOcalliMajke>
    <ime>Jovana</ime>
    <pol>ženski</pol>
    <godinaRodjenja>1996</godinaRodjenja>
  </student>
  <student>
    <brojDosijea>S-02/15</brojDosijea>
    <prezime>Petrović</prezime>
    <imeOcalliMajke>Dušan</imeOcalliMajke>
    <ime>Petar</ime>
    <pol>muški</pol>
    <godinaRodjenja>1996</godinaRodjenja>
  </student>
</studenti>
```

► I JSON i XML su:

- Jednostavni
- Otvoreni
- Obezbeđuju interoperabilnost
- Koriste Unicode standarde
- Ljudski čitljivi

► JSON

- Može da čuva samo tekst i brojeve
- Podaci se čuvaju kao nizovi
- Ima opcije jedino za transfer podataka bez njihovog formatiranja
- Za parsiranje se koristi JSON.Parse funkcija

► XML

- Može da čuva različite tipove podataka
- Moguće je zakačiti bilo koju datoteku u bilo kojem formatu (fotografije, audio, video datoteke)
- Podaci se čuvaju u obliku drveta
- Lako deljenje i transformacija dokumenata

12 – Semantički veb

Kako trenutno radi veb?

- * Ljudi kreiraju različite dokumente.
- * Adresiraju dokumente (koristeći URI) i na taj način ih čine dostupnim drugima na internet.
- * Generalno, veb se može posmatrati kao skup veb sajtova koji nude zbirke veb dokumenata, čiji je cilj da se korisnicima ponudi određeni sadržaj.
- * Veb se razvio u medijum komunikacije među ljudima, a ne među mašinama.
- * Kao posledicu toga, semantički sadržaj (značenje onoga što se nalazi na veb strani) zapisan je na način koji je dostupan samo ljudima.

Kako ljudi danas koriste veb?

- * Verovatno najčešći način korišćenja veba je njegovo pretraživanje.
- * Cilj je da se pronađu određene informacije ili resursi.
- * Ova aktivnost može biti veoma frustrirajuća, zbog dobijanja velikog broja rezultata kao odgovora na određeni upit.
- * Razlog za ovakvo stanje je što pretraživači sprovode svoju pretragu na osnovu koncepta "dokumenti koji sadrže date ključne reči" -i ako dokument sadrži ključnu reč, biće vraćen nazad korisniku kao rezultat pretraživanja.

Šta ljudi žele od veba?

- * Da imaju mogućnost da koriste veb na isti način kao što koriste dokumenta.
- Da mogu da povežu podatke
- Da koriste povezane podatke na način kako žele (pretraživanje, integracija podataka...)
- Da agenti, programi, skripte imaju sposobnost da tumače podatke.
- * Ljudi žele da prošire postojeći veb u veb podataka, da bi aplikacije mogle da direktno koriste podatke na vebu.

Kako od postojećeg veba napraviti veb podataka?

- * Objaviti podatke kako bi se na vebu znalo za njih.
- * Omogućiti linkovanje na URI sa drugih izvora (ne samo sa veb strana).
- Internet će na dalje raditi sve što treba.

- Semantički veb predstavlja kolekciju standarda i tehnologija za realizaciju podataka. Njegove ideje su dodavanje značenja trenutnom vebu, razumljivo mašinama sa ciljem da računari mogu da razumeju dokumente na vebu i automatski izvrše zadatke koji se danas u velikoj meri obavljaju ručno.
- Generalno, termin semantičko označavanje odnosi se na dodavanje podataka u neke druge podatke.
- Cilj semantičkog označavanja veb strana je davanje značenja podacima koji se nalaze u njima, ti mašine za pretraživanje, robot i browser-i mogu da izvuku i obrade označene podatke (a i bolje se indeksiraju).
- Načini za semantičko označavanje: Mikro formati, mikro podaci, RDFa, JSON-LD.

- **Mikro formati** omogućavaju dodavanje semantičkih informacija u veb strane. Informacije se dodaju pomoću atributa u HTML elemente. Primeri dodatih informacija su datum, broj telefona, e-mail adresa...
- Cilj je da se dodate informacije mogu lako pročitati pomoću nekog softvera, a potom i manipulirati njima.
- Kreiraju se dodavanjem *class* i *rel* atributa u HTML tagove.

- **Mikro podaci** predstavljaju sastavni deo HTML5 specifikacije i obezbeđuju standardizovanu sintaksu za dodavanje semantičkih oznaka na veb strane. Cilj je poboljšanje čitljivosti strana i pomoć automatizovanim programima tipa Googlebot-a da bolje razumeju sadržaj. Proširuje mikro formate sa ciljem otklanjanja njihovih nedostataka.
- HTML 5 mikro podaci omogućavaju dodavanje atributa u bilo koji HTML element koji mogu da čitaju botovi. Ne pravi problem kod browser-a koji ih ne podržavaju.

Postoji 5 atributa koji se mogu primeniti na bilo koji HTML5 element:

- * **itemscope** - ukazuje da je element mikro podatak i da potomci tog elementa sadrže informacije o njemu.
- * **itemtype** - definiše URL rečnika (šeme) koji mikro podaci koriste.
- * **itemid** - ukazuje na jedinstveni identifikator stavke, ukoliko je stavka definisana u rečniku.
- * **itemprop** - individualni element, pokazuje da tag sadrži vrednost osobine za stavku, pri čemu su osobine name i value opisane u rečniku.
- * **itemref** - osobine koje nisu potomci elementa sa *itemscope* atributom mogu biti povezani preko ovog atributa.

- Schema.org je opšti rečnik čijim pojmovima se omogućava precizno označavanje sadržaja veb strana.

- **JSON-LD** omogućava semantičko označavanje bez korišćenja atributa u HTML tagovima. Predstavlja semantički sloj na JSON formatom. Kao i mikro podaci, koristi rečnik schema.org.

- Prednost JSON-LD-a na JSON je mogućnost davanja semantičkog značenja parovima name-value.

- U veb stranu se ugrađuje preko taga `<script type="application/ld+json">`.

- **Open Graph (OG)** protokol kreirao je Facebook 2010. kao otvoreni standard. Cilj je povezivanje FB-a i drugih sajtova.

- Povezivanjem veb strane postaju objekat (čvor) u okviru FB socijalne mrežem sa istim funkcionalnostima koje imaju Facebook objekti.

- Korišćenje OG protokola ukazuje na koji način lajkovani ili deljeni sadržaji treba da budu prikazani na Facebook-u, odn. koji naslov, URL, slika i opis strane treba da se koriste

- Informacije za Facebook se postavljaju preko META podataka, koji se dodaju veb stranama unutar `<head>` elementa.

Osnovni meta podaci :

- * **og:title** - naziv objekta u okviru grafa, na pr. "Nicolas Cage".
- * **og:type** - tip objekta, na pr. "actor".
- * **og:image** - URL slike koja predstavlja objekat u grafu.
- * **og:url** - URL objekta koji će se koristiti kao stalni ID na grafu.

Opcioni meta podaci : * **og:audio** * **og:video** * **og:site_name** * **og:description...**

```

<div id="hcard-Snezana-R.-Popovic" class="vcard">
  <span class="fn n">
    <span class="given-name">Snezana</span>
    <span class="additional-name">R.</span>
    <span class="family-name">Popovic</span>
  </span>
  <div class="org">Racunarski fakultet</div>
  <a class="email" href="mailto:spopovic@ef.edu.rs">spopovic@ef.edu.rs</a>
  <div class="adr">
    <div class="street-address">Knez Mihailova 6</div>
    <span class="locality">Beograd</span>
    <span class="region">Srbija</span>
    <span class="postal-code">11000</span>
  </div>
  <div class="tel">011/123-123</div>
</div>

```

```

<section itemscope itemtype="http://schema.org/Person">
  Hello, my name is
  <span itemprop="name">John Doe</span>,
  I am a
  <span itemprop="jobTitle">Graduate research assistant</span>
  at the
  <span itemprop="affiliation">University of Dreams</span>
  My friends call me
  <span itemprop="additionalName">Johnny</span>
  You can visit my homepage at
  <a href="http://www.example.com" itemprop="url">www.example.com</a>
  <section itemprop="address" itemtype="http://schema.org/PostalAddress">
    I live at
    <span itemprop="streetAddress">1234 Peach Drive</span>
    <span itemprop="addressLocality">Warner Robins</span>
    <span itemprop="addressRegion">Georgiac</span>
  </section>
</section>

```

```

</div>
</section>
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Person",
  "name": "John Doe",
  "jobTitle": "Graduate research assistant",
  "affiliation": "University of Dreams",
  "additionalName": "Johnny",
  "url": "http://www.example.com",
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "1234 Peach Drive",
    "addressLocality": "Wonderland",
    "addressRegion": "Georgia"
  }
}
</script>
</body>

```

```

1 <html xmlns:og="http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
2 <head>
3 <title>Nicolas Cage - IMDb</title>
4 <meta property="og:site_name" content="IMDb" />
5 <meta property="og:title" content="Nicolas Cage" />
6 <meta property="og:type" content="actor" />
7 <meta property="og:url" content="http://www.imdb.com/name/nm0000115/" />
8 <meta property="og:image" content="http://ia.media-imdb.com/images/H/V/R36_0_630_1200_AI.jpg" />
9 <meta property="og:description" content="Nicolas Cage, Actor: Leaving Las Vegas. Nicolas Cage
10 was born in Long Beach, California, the son of comparative literature professor August Coppola
11 (a brother of director Francis Ford Coppola) and dancer/choreographer Joy Vogelsang.
12 He is of Italian (father) and German, English, and Polish (mother) descent.
13 Cage changed his name early in his career to make his own reputation,
14 succeeding brilliantly with a host of classic, quirky roles ..." />
15 ...
16 </head>
17 ...

```

Gore su m-format i m-data, dole su JSON-LD i Open Graph