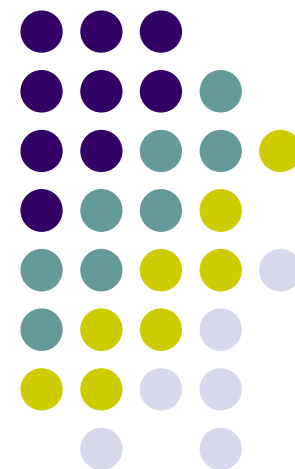
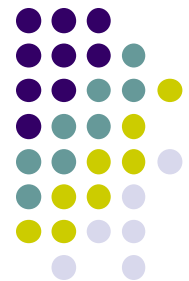


Klijentske tehnologije

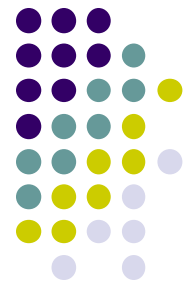
HTML, DTD, CSS
Maja Štula
ak. god. 2011/2012



HTML



- HTML (*HyperText Markup Language*) je jezik koji se koristi za publiciranje informacija na World Wide Web-u.
- *Markup* jezik je jezik koji kombinira sadržaj koji se objavljuje sa dodatnim podacima o tome sadržaju. Ti dodatni podaci mogu biti, npr. boja teksta, veličina slova, položaj informacije u pregledniku i sl., i izražavaju se kao *markup*, izmiješan s osnovnim sadržajem dokumenta.
- Prihvaćena verzija HTML standarda je 4.01 (24.12.1999) (<http://www.w3.org/TR/html401/>), a 2008 je definirana i nova verzija HTML 5 (<http://www.w3.org/html/wg/html5/>).
- Standardizacijom HTML-a bavi se W3 konzorcijum.



W3 konzorcijum

- W3C (*World Wide Web Consortium*) je osnovan 1994 kroz suradnju MIT-a i CERN-a uz podršku DARPA-e i Europske Komisije.
- Cilj W3C je razvoj Web standarda, otvorenost Web-a tj. mogućnost pristupa svima bez obzira na resurse, obrazovanje, ...
- Razvoj weba ne može biti nečije vlasništvo – standardi trebaju biti otvoreni, a to je cilj i kompanijama vezanim uz web pa su tako članovi W3C i IBM, Microsoft, Sun i još preko 400 kompanija zainteresiranih za razvoja weba.
<http://www.w3.org/Consortium/Member/List>
- W3C standardi se nazivaju W3C preporuke (*recommendations*).
- W3C koordinira i rad drugih organizacija vezan uz web poput IETF-a.



Povijest HTML-a

- HTML je “izmislio” Tim Berners-Lee postavljajući temelje www-a, a širenje HTML-a dovelo je do postupka standardizacije.
- HTML 2.0 (prva specifikacija HTML standarda) razvijen je pod vodstvom IETF (kraj 1995).
- Danas se razvojem HTML (od 1996) bavi W3 konzorcijum.
- 2000. HTML je definiran i ISO standardom (ISO/IEC 15445:2000).

Verzija	Godina
HTML 3.2	1997
HTML 4.0	1998
HTML 4.01	1999
HTML 5	Još uvijek draft



HTML standard

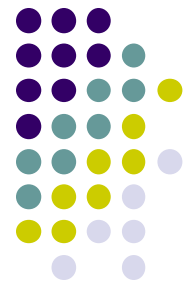
- Standard osigurava korišćenje HTML-a na različitim platformama i preglednicima.
- Svaka verzija HTML-a razvija se uz pokušaj postizanja što većeg konsenzusa među kompanijama iz područja www-a. Postizanje konsenzusa osigurava interoperabilnost.
- HTML se stalno razvija zbog razvoja novih tehnologija (i softverskih i hardverskih).



Markup jezik

- Pojam “markup” je izveden iz tradicionalne izdavačke prakse “označavanja, obilježavanja” (*mark up*) rukopisa, kojima su se dodavale simboličke oznake za tiskanje na marginama rukopisa. To je kombiniranje sadržaja sa dodatnim oznakama tj. podacima o tome sadržaju koji se nazivaju *markup* i izmiješani su s osnovnim sadržajem dokumenta.
- Najpoznatiji *markup* jezik je HTML za publiciranje na Web-u.
- XML (*Extensible Markup Language*), XHTML (*eXtensible Hypertext Markup Language*), LaTeX....

SGML



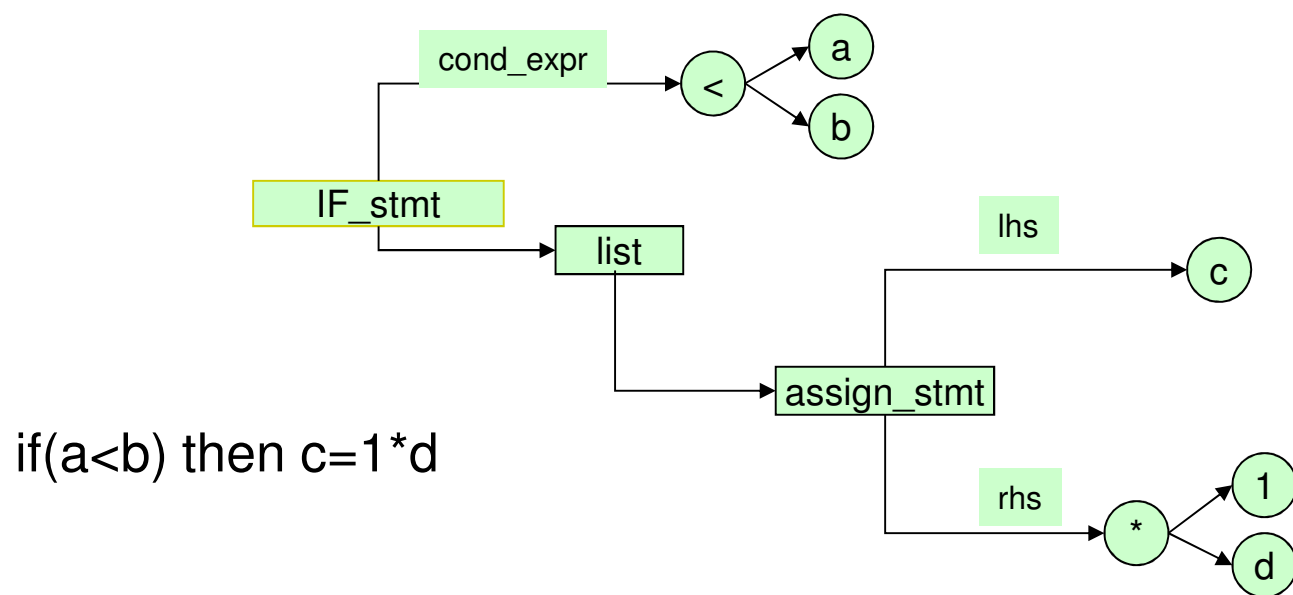
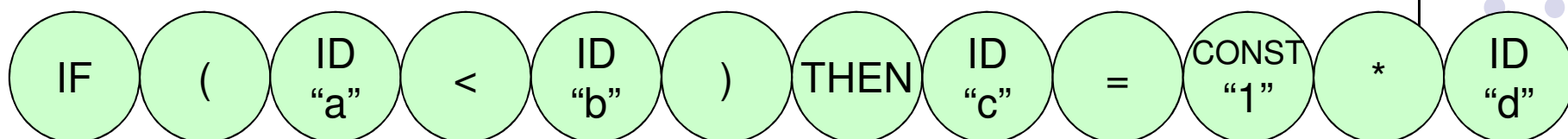
- Za definiranje drugih markup jezika koristi se SGML (*Standard Generalized Markup Language*). SGML je također markup jezik (ISO standard 8879). HTML je definiran preko SGML-a.
- Markup jezik definiran u SGML-u zove se SGML aplikacija. SGML aplikacija sadržava:
 - SGML deklaraciju – SGML deklaracija sadrži leksičku osnovu SGML dokumenta (koji se karakteri mogu pojavljivati u SGML aplikaciji, kako se označavaju markup delimiteri i sl.).
 - DTD (*Document Type Definition*) – DTD definira sintaksu markup jezika tj. skup pravila koja dokument treba poštivati da bi bio ispravan (*valid*) prema shemi određenoj DTD-om. (npr. <http://www.w3.org/TR/html4/loose.dtd>)
 - Specifikaciju koja opisuje semantiku jezika (npr. <http://www.w3.org/TR/html401/>).
 - Instance dokumenta koji sadrže podatke i oznake (*markup*). Svaka instanca sadrži referencu na DTD koji se koristi za interpretiranje dokumenta.



SGML ↔ DTD

- Na početku svakog HTML dokumenta trebalo bi specificirati DTD prema kojem je taj dokument napisan.
- DTD (*Document Type Definition*) je jedan od XML shema jezika kojim se opisuje dokument. (XML Schema (W3C), RELAX NG).
- DTD koristi skup deklaracija u skladu sa sintaksom *markup* jezika dokumenta.
- Na taj način SGML parser zna prema kojem DTD-u treba provjeriti da li je dokument ispravno napisan.
- Sintaktičkom analizom ili *parsiranjem*, tokeni (znakovi dobiveni leksičkom analizom) se organiziraju (obično u formi stabla) u gramatički ispravnu strukturu.

sintaktička analiza (*parsiranje*)





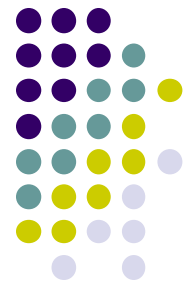
DTD

- DTD se definira na početku dokumenta u jednoj liniji u DOCTYPE (*document type*) deklaraciji. Početna namjera je bila da se DTD koristi i za određivanje tipa dokumenta (html, xml, ...).
- Tipična deklaracija izgleda:

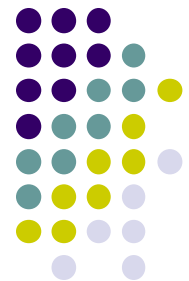
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

- Znak <! se koristi za označavanje SGML deklaracije.
- DOCTYPE opisuje da se radi o SGML deklaraciji tipa dokumenta.
- Oznaka HTML se odnosi na tip dokumenta i oznaku prvog elementa u dokumentu.
- PUBLIC je način pristupa DTD-u koji se koristi u dokumentu.

DTD

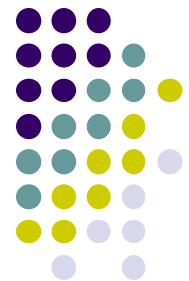


- "-//W3C//DTD HTML 4.0//EN" je FPI (*Formal Public Identifier*) koji parser koristi za pronalaženje odgovarajućeg DTD-a prilikom obrade dokumenta. FPI je SGML javni identifikator (nečega) tj. komad posebno formatiranog teksta koji jedinstveno identificira proizvod, specifikaciju ili dokument. (URI?)
- *Formal Public Identifier* ima format:
- (ISO | + | -)//schema's owner//schema description//language id
- tj. "*Owner//Keyword Description//Language*"
- Opcionalno se može na kraju navesti i čitav URI do DTD dokumenta.
- PRIMJER DTD-a: <http://dtd.nlm.nih.gov/publishing/>



HTML ↔ DTD

- Ispravan (*valid*) HTML dokument sadrži deklaraciju verzije HTML korištene u dokumentu. Deklaracija tipa dokumenta sadrži DTD koji se koristi u stranici.
- HTML 4.01 specificira tri DTD.
 - [HTML 4.01 Strict DTD](#) : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
 - [HTML 4.01 Transitional DTD](#) : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
 - [HTML 4.01 Frameset DTD](#) : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
- Autor stranice bi trebao uključiti jedan u stranicu.



DTD

- DTD opisuje dopuštenu, moguću strukturu dokumenta na način da definira elemente koje dokument može sadržavati, vrijednosti tih elemenata, attribute koji mogu ići uz neki element i vrijednosti koje atributi mogu poprimati.
- Struktura dokumenta definirana je preko:
 - Elementa
 - Atributa
 - Entiteta – entiteti su varijable, mogu sadržavati bilo kakav tip podatka
 - PCDATA (Parsed Character DATA) – to je tekst koji će parser isparsirati i ukoliko sadrži tagove razložiti će ga na elemente.
 - CDATA (Character DATA) – to je tekst koji se ne parsira.

DTD

<!ELEMENT element-name category>

ili

<!ELEMENT element-name (element-content)>



- Struktura DTD dokumenta je opisana preko deklaracija elemenata i njihovih atributa.
- Deklaracija elemenata navodi elemente koji se mogu pojaviti u dokumentu i specificira kako su elementi deklarirani i što mogu sadržavati.
- Deklaracija liste atributa definira dopušteni skup atributa za određeni element uključujući i tip vrijednosti atributa.

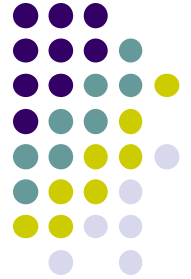
```
<!ELEMENT META - O EMPTY -- generic metainformation -->
<!ATTLIST META
  %i18n;           -- lang, dir, for use with content --
  http-equiv NAME #IMPLIED -- HTTP response header name --
  name NAME #IMPLIED -- metainformation name --
  content CDATA #REQUIRED -- associated information --
  scheme CDATA #IMPLIED -- select form of content -- >
```

Atribut je opcionalan

Atribut je obavezan

<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=ISO-8859-2">

DTD



```
<!ENTITY % i18n
"lang %LanguageCode; #IMPLIED -- language code
dir (ltr|rtl)          #IMPLIED -- direction for weak/neutral text --" >
```

```
<!ENTITY % LanguageCode
"NAME" -- a language code, as per [RFC1766] -->
```

```
<!ENTITY % coreattrs
"id      ID          #IMPLIED -- document-wide unique id
class    CDATA       #IMPLIED -- space-separated list of classes
style    %StyleSheet; #IMPLIED -- associated style info
title    %Text;       #IMPLIED -- advisory title --" >
```

HTML ↔ DTD



- Vrijednosti koje atributi mogu poprimati označavaju se ili SGML tipovima ili nekim drugim predefiniranim tipovima. Tako je npr. CDATA SGML niz karaktera (string) iz skupa karaktera dokumenta. Korisnički agent treba vrijednosti tipa CDATA interpretirati na sljedeći način:
 - Zamijeniti entitete karaktera (*character entities*) sa “pravim” karakterima. Određeni karakteri u HTML su specijalni i rezervirani (npr. < i > omeđuju tag). Ako želimo prikazati te karaktere u browseru moramo ih zamijeniti u HTML kodu sa entitetima karaktera koji se označavaju sa `&entity_name;` ili `&#entity_number;` (npr. ` ` tj. ` ` = razmak; `£` tj. `£` = £ karaktera). [Primjer](#)
 - Ignorirati prelazak u novi red (*line feed*, `\n`, 10, HEX 0A).
 - Zamijeniti oznaku kraja retka (*carriage return*, `\r`, 13, HEX 0D) i tab s jednim razmakom (*space*).
- Provjera validacije HTML dokumenta:
<http://validator.w3.org/>

DTD



FONT element je *deprecated* tj. ne preporuča se njegovo korištenje i definiran je samo u loose.dtd.

```
<!ELEMENT FONT - - (%inline;)*          -- local change to font -->
<!ATTLIST FONT
  %coreattrs;      -- id, class, style, title --
  %i18n;           -- lang, dir --
  size CDATA #IMPLIED -- [+|-]nn e.g. size="+1", size="4" --
  color %Color; #IMPLIED -- text color --
  face CDATA #IMPLIED -- comma-separated list of font names --
>
```

(Arrows from 'size', 'color', and 'face' in the DTD point to the corresponding attributes in the example below)

```
<font size=4 color="red">neki tekst</font>
```



HTML

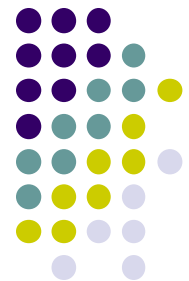
- HTML dokument sastoji se od tri dijela:
 - deklaracije tipa dokumenta,
 - deklarativnog zaglavlja ograničenog elementom HEAD,
 - tijela koje uključuje stvarni sadržaj dokumenta, ograničenog BODY elementom ili FRAMESET elementom.
- Prazna mjesta i linije dozvoljeni su bilo gdje u HTML dokumentu.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>My first HTML document</TITLE>
</HEAD>
<BODY>
<P>Hello world!</P>
</BODY>
</HTML>
```



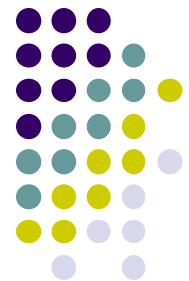
HTML Elementi

- Popis elemenata prema [HTML4.01 specifikaciji](#) sadrži 91 element od kojih su neki *deprecated* (neodobreni) (sličan termin kao i *obsolete* (zastario))
- Elementi su osnovni građevni blokovi HTML dokumenta. Imaju dva osnovna svojstva:
 - attribute
 - sadržaj
- Element obično ima početnu oznaku (tag) i krajnju. Npr. `<html></html>`
- Neki elementi (`
`) ne trebaju krajnju oznaku.



HTML Elementi

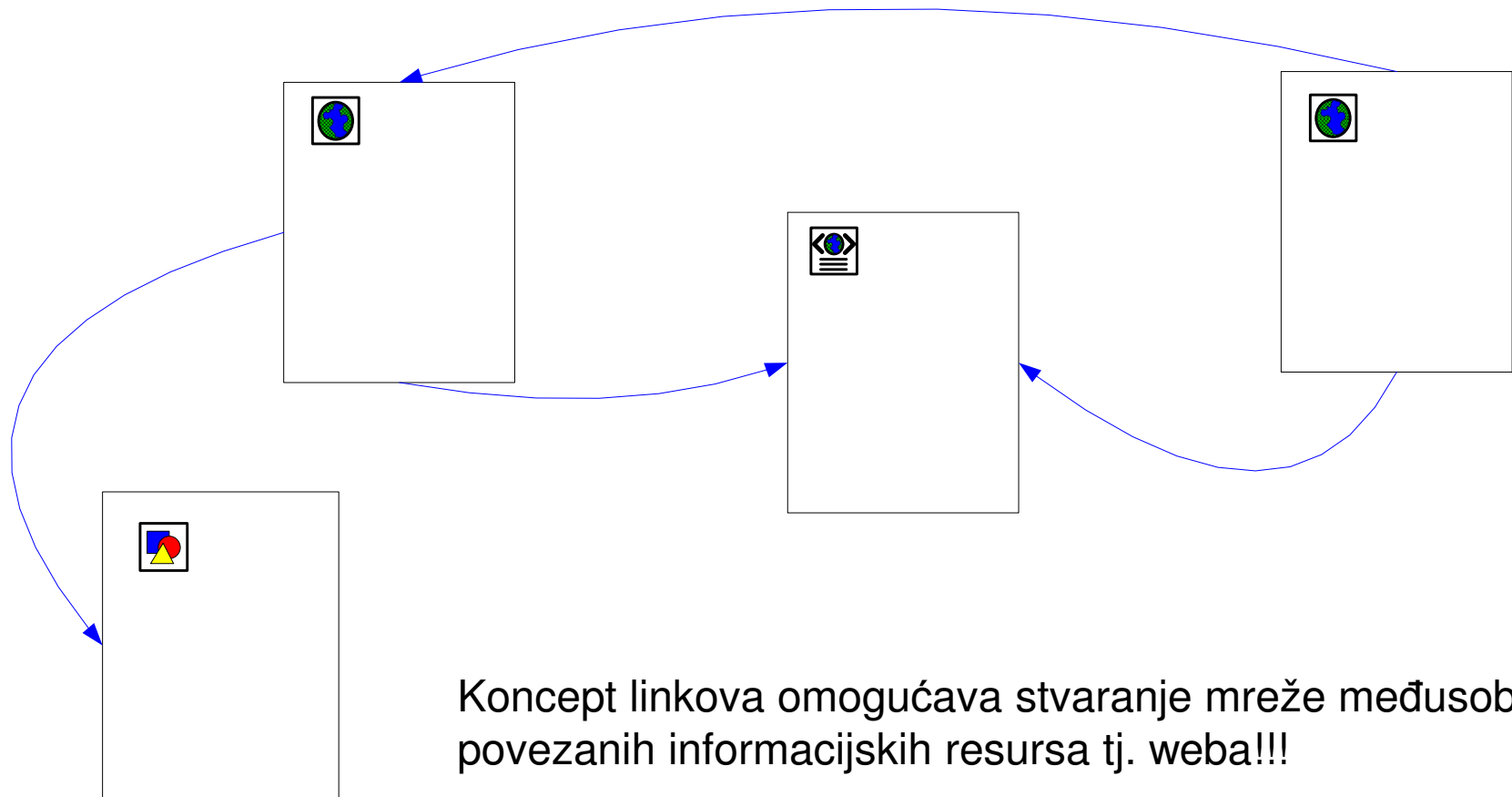
- HTML elementi se mogu podijeliti na strukturalne, prezentacijske i hipertekst elemente.
 - Strukturalni elementi opisuju strukturu sadržaja dokumenta. Npr. `<h2>Naslov dva</h2>` , element `<h2>` opisuje sadržaj *Naslov dva* kao naslov druge razine u dokumentu.
 - Prezentacijski elementi opisuju izgled odnosno iscrtavanje sadržaja. Npr. ` Boldani tekst `
 - Hipertekst elementi kreiraju linkove prema drugim dokumentima. Npr. `Link`



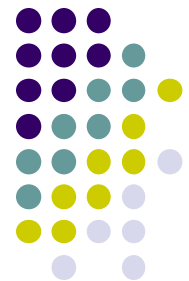
Linkovi

- Hipertekst je način organiziranja velikih količina informacija odnosno način rješavanja problema informacijskog preopterećenja (*information overload*).
- Premda je jednostavan, koncept linka bio je jedan od osnova uspjeha Web-a.
- Link uvijek ima dva kraja. Počinje od izvorišnog sidra i upućuje na odredišno sidro što može biti bilo koji web resurs (HTML stranica, slika, ...) adresiran URI-jem. Krajevi linka se nazivaju sidra.

Linkovi

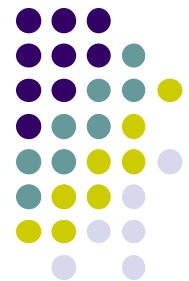


Koncept linkova omogućava stvaranje mreže međusobno povezanih informacijskih resursa tj. weba!!!



Linkovi

- Osnovno ponašanje linka se očituje kao dohvaćanje resursa na koje link upućuje.
- Osim osnovnog ponašanja link može ukazivati i na drugačiju vezu među dokumentima.
- Tipovi linkova mogu biti: Alternate, Stylesheet, Start, Next, Prev, Contents, Indeks, Glossary, Copyright, Chapter, Section, Subsection, Appendix, Help, Bookmark. Npr. `<link rel="stylesheet" href="boja_pozadina.css" type="text/css">`
- Atributi rel i rev koriste se za navođenje tipa linka.
 - rel %LinkTypes; #IMPLIED -- forward link types --
 - rev %LinkTypes; #IMPLIED -- reverse link types --



Linkovi

- HTML elementi LINK i A omogućavaju povezivanje dokumenata odnosno kreiranje hiperteksta (premda se link definira i preko IMG elementa () i preko FORM elementa (<form method= "GET" action= "program">)).
- Atribut href (i LINK i A elementa) sadrži vrijednost odredišta linka tj. URI.

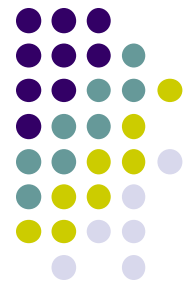
```
<A href="http://www.fesb.hr/">FESB</A>
```

- URI može biti apsolutan URI (http://www.fesb.hr/~kiki/moja_stranica.htm), relativan URI (/~kiki/moja_stranica.htm), kao i link definiran unutar iste stranice (URI fragment #nešto).

```
<A href="#bio">BIOGRAPHICAL NOTE</A>
```

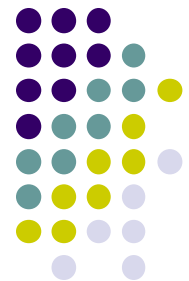
```
<a name="#bio"><FONT size=4>Professional skills:</FONT></a>
```

name	CDATA	#IMPLIED	-- named link end --
href	%URI;	#IMPLIED	-- URI for linked resource --



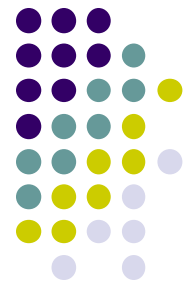
Veličina HTML elementa

- HTML specificira tri tipa vrijednosti veličine HTML elemenata:
 - Piksel (*pixels*): Vrijednost (%Pixels; u DTD-u) je cjeli broj koji predstavlja broj piksela kojim će biti iscrtan element. (npr. font-size: 19px;)
 - Duljina (*length*): Vrijednost (%Length; u DTD-u) može biti ili piksel ili postotak vertikalne ili horizontalne dimenzije. Tako vrijednost "50%" znači da će veličina elementa biti pola od raspoložive veličine.
 - Višestruka duljine (*multilength*): Vrijednost (%MultiLength; u DTD-u) može biti ili duljina ili relativna duljina u formi zapisa "i*", gdje je "i" cijeli broj. Prilikom iscrtavanja elementima se veličina određuje na osnovu raspoloživog prostora * i. Npr. 20 piksela * 2 = 40 piksela. [Primjer](#)





Boje HTML elementa


- Vrijednost atributa tipa boja (%Color;) odnosi se na boju specificiranu u ["A Standard Default color Space for the Internet - sRGB"](#).
- HP i Microsoft (1996) su predložili sustav upravljanja bojama (*color managment system*) kroz standardni prostor boja (*standard color space* – sRGB).
- To je metoda za rad sa bojama koje Microsoft i HP koriste u svojim proizvodima, ali je metoda dobro prilagođena uređajima poput monitora, skenera, printera i sl. tako da je i drugi proizvođači softvera i hardvera lako implementiraju u svoje proizvode.
- Osim toga metoda je jednostavna i neovisna o uređaju na kojem se boja i prikazuje te omogućava laku konverziju u neki drugi format definiranja boja.



Boje

- Prostor boja (*color space*) je model za predstavljanje boja numerički sa tri ili više koordinata npr. RGB prostor boja je određen sa Crvenom, Zelenom i Plavom koordinatom.
- Vrijednost boje se zapisuje kao heksadecimalni broj (na početku s hash znakom) ili nekim od 16 definiranih imena boja. Imena boja su *case-insensitive*. [Primjer](#)

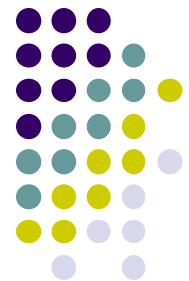
	Black = "#000000"
	Silver = "#C0C0C0"
	Gray = "#808080"
	White = "#FFFFFF"
	Maroon = "#800000"
	Red = "#FF0000"
	Purple = "#800080"
	Fuchsia = "#FF00FF"

	Green = "#008000"
	Lime = "#00FF00"
	Olive = "#808000"
	Yellow = "#FFFF00"
	Navy = "#000080"
	Blue = "#0000FF"
	Teal = "#008080"
	Aqua = "#00FFFF"



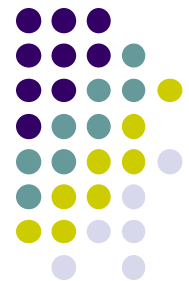
Boje

- Korišćenje elemenata i atributa boja u HTML dokument je zastarjelo (*deprecated*).
- Preporučeno je definiranje boja u CSS-u (*Cascading Style Sheets*).



Imena HTML okvira

- Kada u stranici koristite okvire (*frame*) kako biste mogli pristupati pojedinom okviru trebate atributu NAME dodijeliti vrijednost preko koje možete pristupati tom okviru.
- Ime okvira (%FrameTarget; u DTD-u) mora počinjati sa slovima (a-z, A-Z). Inače ga korisnički agent treba ignorirati.
- Ova se vrijednost također koristi u HTML elementima A, LINK, AREA, FORM, ... kao vrijednost atributa TARGET kako bi se definiralo u kojem okviru ili prozoru se dohvaća informacijski resurs identificiran u elementu.



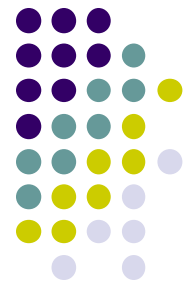
Imena okvira

- Sljedeća imena su rezervirana i imaju specijalno značenje :
 - `_blank` – korisnički agent bi trebao dohvatiti navedeni dokument u novom, neimenovanom prozoru.
 - `_self` - korisnički agent bi trebao dohvatiti navedeni dokument u istom okviru (prozoru) u kojem se nalazi i element koji upućuje na dokument.
 - `_parent` - korisnički agent bi trebao dohvatiti navedeni dokument u prvom FRAMESET elementu promatranog elementa. Ako element nema FRAMESET element tada je ova vrijednost ekvivalenta `_self` vrijednosti.
 - `_top` - korisnički agent bi trebao dohvatiti navedeni dokument u istom, čitavom prozoru u kojem se nalazi i promatrani element.

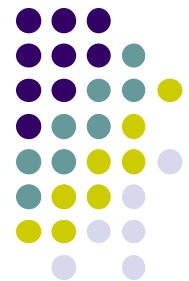
PRIMJER

```
<a href="jedna.htm" target="_blank">Otvori prvu stranicu u novom prozoru</a>  
<a href="jedna.htm" target="_top">Otvori prvu stranicu u ovom prozoru</a>  
<a href="druga.htm" target="jedna">Otvori drugu stranicu u prvom prozoru</a>  
<a href="jedna.htm" target="druga">Otvori prvu stranicu u drugom prozoru</a>
```

XHTML



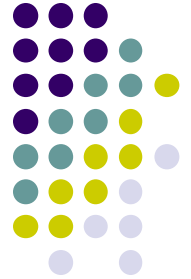
- XHTML (*Extensible HyperText Markup Language*) je skup trenutnih i budućih specifikacija koji proširuju HTML tipove dokumenata i redefiniiraju ih pomoću XML-a.
- XHTML se sastoji od svih elemenata iz HTML 4.01 kombiniranih sa sintaksom XML-a (*eXtensible Markup Language*). XHTML je baziran na XML-u (nasljednik HTML-a).
- W3 konzorcijum pored dvije verzije standarda XHTML-a (1.1 iz 2001. godine i 1.0 iz 2000. godine) i dalje preporučuje i HTML 4.01 standard za publiciranje informacija na webu.
- XHTML postavlja stroža pravila nego HTML.
- Cilj XHTML je smanjiti opterećenje parsera i korisničkih agenata kako bi se web sadržaji lakše prilagodili novim zahtjevima poput npr. korištenja mobilnih uređaja za pristup web sadržajima.



XHTML ↔ HTML

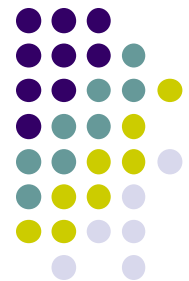
- XHTML postavlja strože zahtjeve prema formatiranju dokumenta tako da svi elementi moraju biti eksplicitno zatvoreni (
).
- U XML-u imena elemenata i atributa su case-sensitive pa XHTML definira da se koriste samo mala slova za imena elemenata i atributa dok HTML dopušta i mala i velika slova.
- Vrijednosti atributa obavezno trebaju biti navedene u jednostrukim ili dvostrukim navodnicima.
- Trenutno svi preglednici ne podržavaju XHTML (ili ga ne podržavaju u potpunosti).

XHTML



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
  lang="en">
<head>
<title>Virtual Library</title>
</head>
<body>
<p>Moved to <a href="http://vlib.org/">vlib.org</a>.
</p>
</body>
</html>
```

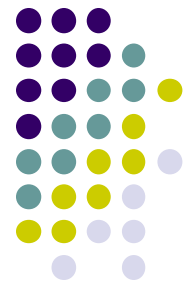
XHTML



- Osnovni (*root*) element XHTML je `<html>`. Mora sadržavati deklaraciju imenskog prostora. Za XHTML je <http://www.w3.org/1999/xhtml> definirani imenski prostor.
- Imenski prostor je identificiran sa URI-jem, a sadrži kolekciju imena koja se u XHTML dokumentu koriste kao imena elemenata i atributa. Npr:

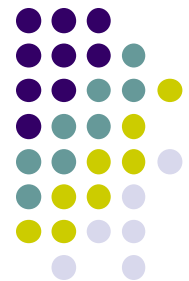
```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```
- Imenski prostor se deklarira preko atributa `xmlns` ili atributa koji kao prefiks imaju `xmlns:`.
- Deklaracija imenskog prostora se odnosi na element gdje je navedena kao i na sve elemente koji su ugniježđeni unutar promatranog elementa ukoliko nije pregažena sa drugom deklaracijom imenskog prostora.

CSS



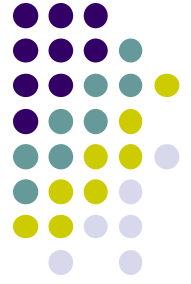
- CSS (*Cascading Style Sheets*) je jezik stila (*stylesheet language*) koji se koristi za opisivanje prezentacije tj. izgleda dokumenta napisanog u *markup* jeziku.
- Jezik stila je računalni jezik za opisivanje stila elemenata u dokumentu napisanom u *markup* jeziku. (XSL – *Extensible StylSheet Language*, ...)
- CSS se obično koristi s HTML-om i XHTML-om za opisivanje stila stranice, ali može se primijeniti i na dokumente napisane u drugim *markup* jezicima npr. XML-u.

CSS



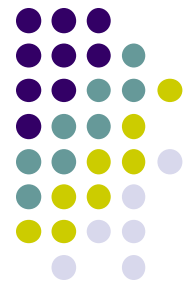
- HTML je originalno dizajniran za definiranje sadržaj dokumenta.
- Pretraživač se trebao baviti izgledom dokumenta, bez korištenja ikakvih tagova za formatiranje.
- Kako su dodavani novi HTML tagovi i atributi koji su se odnosili na formatiranje izgleda sadržaja () originalnoj HTML specifikaciji, postalo je sve teže odvajati sadržaj HTML dokumenata od izgleda dokumenta.
- Da bi riješili taj problem W3C je stvorio stilove kao dodatak za HTML 4.0. Svi glavni pretraživači podržavaju CSS.

CSS



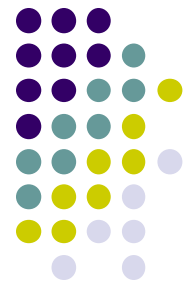
- Specifikacija CSS standard je u nadležnosti W3 konzorcijuma.
- Trenutno aktualna verzija je CSS2.1
<http://www.w3.org/TR/CSS21/>
- Provjera validacije CSS (*Cascading Style Sheets*) dokumenta:
<http://jigsaw.w3.org/css-validator/>

CSS



- Stilovi se obično spremaju u vanjske .css datoteke.
- Vanjske stranice stilova omogućavaju korisnicima upravljanje izgledom više web stranica preko samo jednog CSS dokumenta.
- Uključivanje CSS dokumenta u HTML dokument:
`<link rel="stylesheet" href="example.css" type="text/css">`
- ili
`<style type="text/css">`
`@import "example.css";`
`</style>`

CSS



- CSS kôd se može umetati u HTML kôd na dva načina.
- Jedan je ograničavanje CSS koda unutar HEAD elemenata tagovima `<style>`.

```
<style type="text/css">  
p {color: red; }  
body {background-color: green; }  
</style>
```
- Drugi način je umetnuti CSS, koji se preko atributa `style` može umetati u bilo koji HTML element.

```
<p style="background: blue; color: white;">Umetnuti CSS</p>
```
- PRIMJER

CSS

```
vanjski.css
body {
    p {color: teal; }
    h1 {color: teal;}
}
```



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> New Document </TITLE>
```

```
</HEAD>
```

```
<link rel="stylesheet" href="vanjski.css" type="text/css">
```

```
<style type="text/css">
```

```
p {color: red; }
```

```
</style>
```

```
<BODY>
```

Ovo je stranica sa css-om.

```
<p>Vanjski CSS se neće primjeniti jer postoji interni CSS koji je "bliži" tj. "jači"</p>
```

```
<h1>Vanjski CSS će se primjeniti jer ne postoji interni CSS</h1>
```

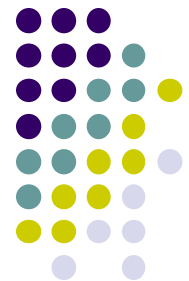
```
<p>Interni CSS</p>
```

```
<p style="background: blue; color: white;">Umetnuti CSS</p>
```

```
</BODY>
```

```
</HTML>
```


CSS



- Što se zbiva ako se više pravila stila odnosi na neki element?
- Da bi odredio koje će se stil primjeniti na HTML element CSS primjenjuje pravila:
 - kaskade,
 - nasljeđivanja,
 - specifičnosti ,
 - lokacije.
- Nasljeđivanje određuje što će se dogoditi ako sam element nema definirani stil. U odsutnosti pravila, djeca nasljeđuju od roditelja brojne stilove.
- Pravilo kaskade određuje da se kombiniraju stilovi koji su definirani na različitim mjestima (u vanjskom css-u, umetnutom css-u itd.) ako se stilovi međusobno ne poništavaju.

CSS



- Lokacija pravila se odnosi na mjesto gdje je pravilo definirano. Što je pravilo bliže elementu (najbliže je umetnuti css) to je jače tj. "pobjeđuje" najbliže pravilo.
- Pravilo specifičnosti govori da što je selektor pravila specifičniji, to je jače pravilo. Pri srazu dva pravila, specifičnije je pravilo važnije – bez obzira na lokaciju. Ako su oba pravila jednako specifična, posljednje pravilo pobjeđuje.



CSS specifičnost

- Specifičnost pravila se definira korištenjem atributa. *id* atributi se smatraju najspecifičnijima jer u dokumentu moraju biti jedinstveni, dok prisutnost atributa *class* čini selektor specifičnijim od onoga koji ga nema.
- Naslijeđena se svojstva smatraju najopćenitijima i nadjačat će ih svako drugo pravilo.
- [PRIMJER](#)

CSS specifičnost



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
</HEAD>
<style type="text/css">
p {
color: red;
font-size:16pt;
}
/* ovo je definicija svojstava elemenata klase moja */
.moja {
color: blue;
font-size:28pt;
}
</style>
<BODY>
<p>Ovo je p bez class atributa</p>
<p class="moja">Ovo je p sa class atributom</p>
</BODY>
</HTML>
```

CSS KOMENTAR!!!

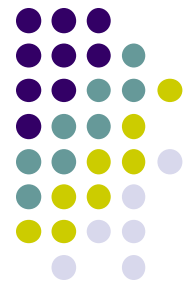
CSS nasljeđivanje



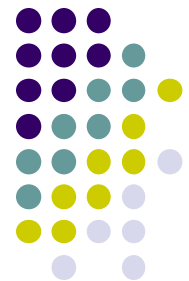
PRIMJER:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
</HEAD>
<style type="text/css">
body {
    font-family: verdana, arial, sans-serif;
}
p {
    color: red;
    font-size: 16pt;
}
</style>
<BODY>
<p>Ovo je p tag sa naslijeđenim svojstvima iz body taga
<br>
<br>
<b>Ovo je b tag unutar p taga <b>
</p>
<b>Ovo je b tag izvan p taga <b>
</BODY>
</HTML>
```

CSS



- Osnovno je pravilo da je među ravnopravnim stilovima najjači onaj koji je na posljednjem mjestu. To znači da lokalni stilovi imaju najveću važnost i nadjačat će sve druge prije primijenjene stilove.
- Koji stil će se koristiti kada je više od jednog stila specificirano za jedan HTML element? Općenito, svi stilovi će se kaskadno povezati u novu virtualnu stranicu stilova prema sljedećim pravilima, gdje broj četiri ima najveći prioritet:
 - 1) unaprijed zadano u pretraživaču;
 - 2) vanjska stranica stilova;
 - 3) unutarnja stranica stilova (unutar `<head>` taga);
 - 4) lokalna primjena stilova (unutar HTML elementa).



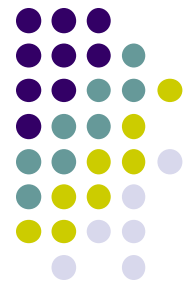
CSS sintaksa

- CSS ima jednostavnu sintaksu, koristi riječi engleskog jezika za definiranje vrijednosti stila.
- CSS sadrži niz pravila. Svako pravilo sadrži jedan ili više selektora (*selectors*) međusobno razdvojenih zarezom i deklaracijskih blokova (*declaration block*).
- Deklaracijski blok se sastoji od liste deklaracija odvojenih točka-zarezom (;) i zatvorenih u zagradama ({}).
- Svaka deklaracija se sastoji od svojstva, znaka (:) i vrijednosti svojstva.



CSS sintaksa

- selektor {svojstvo: vrijednost} ↔ p {color: red; }
- Selektor je HTML element/tag koji se definira dok je svojstvo atribut tog elementa koje se želi promijeniti. Svojstvo i vrijednost se odvajaju dvotočkom.
- Ako se želi navesti više od jednog svojstva, međusobno se odvajaju ;.
p {
color: red;
font-size:16pt;
}
- Istovremeno se može postaviti svojstvo za više različitih elemenata koji se međusobno odvajaju zarezom:
- h1,h2,h3,h4,h5,h6 { color: green }



CSS sintaksa

- Selektor klase omogućava definiranje različitih stilova za isti tip HTML elementa.
- PRIMJER:

```
<style type="text/css">
p.jedan {color: red; }
p.dva {color: blue;}
</style>
<BODY>
<p class="jedan">p klasa jedan</p>
<p class="dva">p klasa dva</p>
</BODY>
```
- Može se preskočiti pisanje imena elementa i pisati:

```
<style type="text/css">
.jedan {color: red; }
.dva {color: blue;}
</style>
```



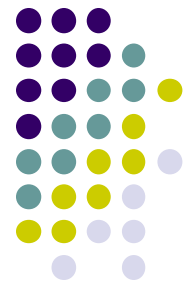
CSS sintaksa

- Id selektor koji se odnosi na HTML element sa vrijednošću id atributa navedenog u selektoru se definira sa prefiksom #.

- PRIMJER:

```
<style type="text/css">
p#jedan {color: red; }
/*ili sintaksa*/
#dva {color: blue;}
</style>
<BODY>
<p id="jedan">p id jedan</p>
<p id="jedan">p id jedan</p>
<p id="dva">p klasa dva</p>
</BODY>
```

CSS



- Prednost korištenja CSS:
 - Prezentacijske informacije o kolekciji stranica su na jednom mjestu i lako se mogu mijenjati.
 - Za različite korisnike može se koristiti različiti CSS (personalizacija izgleda).
 - Veličina i složenost HTML dokumenta su smanjeni jer je prezentacijska informacija pohranjena u zasebnom dokumentu.