

# የመሠረታዊና ቴክኖሎጂ ጥናት

**ԴՐՈՄԻ – ԱՃԹՈՅՈՒՄԻՃԹՈՒՎՃԻՐՅՈՒՆ,  
ՄԵԿԻՐԳՓՅԱՆՈՒՎՃԻՐՅՈՒՆ Զ ՓՃՄԻՄԵՆՃԱՆԻՐԻՐԻ  
ՐԱՅՄԵՆՈՍՈՒԿԻ**

**മുതലം, തൊഴിലുറക്ക്, തൊഴിലുറക്ക്, തൊഴിലുറക്ക്,  
തൊഴിലുറക്ക്, തൊഴിലുറക്ക്**

**ጠሐይ ሽሕዳር**

# Otvoreno računarstvo

## XML - vizualizacije, transformacije i filtriranja podataka

- XHTML, DOM
- XSL (XPath, XSLT, XSL-FO)

Mario Žagar





XHTML

# Što je XHTML?

---



- Extensible HyperText Markup Language
- HTML, ali po XML sintaksi [?]
- Značajke
  - Dobra oblikovanost
    - Pravilno gniježđenje, zatvaranje elemenata, uporaba navodnika, ...
  - Postojanje kompletne strukture (npr. <body>)
  - Uporaba XML alata za manipulaciju i uređivanje

# Zašto XHTML?

---



- Potreba za “striktnijim” i “čišćim” HTML-om
  - problemi s nezgrapnim, pogrešno strukturiranim HTML dokumentima
  - pogodno za preglednike na malim uređajima
- Smanjenje pogrešaka
  - automatskom validacijom pri izradi stranica
- Korišćenje imeničkih prostora (namespaces)
  - dodavanje biblioteka specifičnih oznaka

# Primjer “lošeg” HTML-a

---

- “Loš” HTML znači da ne poštuje sintaksu
- Preglednici ga svejedno prikazuju – podilaženje lošem kodu ☐
- Zašto je “loš”?
  - Nedosljedni programeri rade greške i ne ispravljaju ih
  - HTML je (bio) predviđen za prikaz, a ne za opis podataka
- Primjer “lošeg” HTML-a:

```
<html> <head>  
  <title>Loš HTML</TITLE>  
  <body> <h1>Jako loš... </body>
```

# Najčešće pogreške

---

- Nepravilno ugniježđivanje
  - Primjer: `<b><i>tekst</i></b></i>` umjesto `<b><i>tekst</i></b>`
- Nezatvoreni element
  - Primjer: `<b>tekst` umjesto `<b>tekst</b>`
  - Primjer: `<br>` umjesto `<br />`
- Oznake i atributi se pišu i velikim slovima
  - Primjer: `<B>tekst</b>` umjesto `<b>tekst</b>`
  - Primjer: `<a HREF="...">tekst</a>`
- Dokument ne sadrži samo jedan korijenski element
  - Primjer: Nepostojanje samo jednog `<html>` elementa

# Najčešće pogreške #2

---

- Navodnici za vrijednosti atributa
  - Primjer: `<table width=100%>` umjesto `<table width="100%">`
- Minimizacija vrijednosti atributa
  - Primjer: `<input checked>` umjesto `<input checked="checked">`
- Uporaba zastarjelog name atributa kod oznaka:
  - a, applet, frame, img, ...
- Korištenje specijalnih znakova
  - Primjer: Ti & ja umjesto Ti & amp; ja
- Nepostojanje DOCTYPE deklaracije



# Usporedba XHTML – HTML



- XHTML 1.0 – preformulirani HTML 4.01
  - 3 podverzije (isto kao i za HTML 4.01)
    - XHTML 1.0 Strict
    - XHTML 1.0 Transitional
    - XHTML 1.0 Frameset
- XHTML 1.1
  - Čišćenje od zastarjelih oznaka (deprecated)
  - Modularizacija
  - Kontrola prezentacije putem CSS



# Tipovi XHTML dokumenata

---

- **XHTML 1.0 Strict**
  - “Čiste” oznake, podržava CSS
    - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
- **XHTML 1.0 Transitional**
  - Podrška i za zastarjele oznake i ako preglednik ne podržava CSS
    - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- **XHTML 1.0 Frameset**
  - Podrška i za korištenje okvira (frames)
    - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`

# XHTML 1.1 modularizacija

---

- XHTML je velik i kompleksan
- Manji, dobro definirani skupovi elemenata
  - Koriste se po potrebi i za određene namjene
- 28 modula:
  - Jezgreni (core) elementi (oznake)\*
    - Structure, Text, Hypertext, List
  - Osnovni elementi (oznake)\*
    - Presentation, Basic Forms, Basic Tables, Image
  - Ostali elementi (oznake)
    - Applet, Base, Bi-directional Text, Client Image Map, Edit, Forms, Frames, Iframe, Intrinsic Events, Legacy, Link, Metainformation, Name Identification, Object, Scripting, Server Image Map, Style Attribute, Style Sheet, Tables, Target

\* elementi ovih modula opisani su detaljnije u poglavlju HTML

# Struktura XHTML dokumenta



- Osnovna struktura XHTML dokumenta

## Primjer:

```
<!DOCTYPE ...>
```

```
<html>
```

```
  <head>
```

```
    <title>... </title>
```

```
  </head>
```

```
  <body> ... </body>
```

```
</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
  XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/  
  xhtml1-strict.dtd">
```

```
<html>
```

```
  <head>
```

```
    <title>Naslov</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>odlomak teksta</p>
```

```
  </body>
```

```
</html>
```




- Potpuno
  - Neki manji preglednici za PDA/mobilne uređaje
- Djelomično - većina preglednika podržava, ali...
  - Microsoft Internet Explorer
    - Verzija 7 još uvijek nema mogućnost prikaza ako je isporučen kao XML dokument
    - Ispravno prikazuje kao HTML dokument
  - Firefox
    - Verzija 2 još uvijek ne zna inkrementalno prikazivati XHTML

Tim Berners-Lee - otac HTML-a, 27.10.2006.: *“The **attempt to get the world to switch to XML**, including quotes around attribute values and slashes in empty tags and namespaces **all at once didn't work**. The large HTML-generating public did not move, largely because **the browsers didn't complain**.”*

# Povijest (i budućnost)

---



- Razvoj paralelan s razvojem HTML standarda
- Inačice:
  - XHTML 1.0 W3C preporuka 26.01.2000.
    - Drugo izdanje 1.8.2002.
    - Temeljen na HTML 4.0 W3C preporukama iz 1997. i 1998.
  - XHTML Basic W3C preporuka 19.12.2000.
  - XHTML Basic 1.1 W3C kandidat za preporuku 13.7.2007.
  - XHTML 1.1 W3C preporuka 31.05.2001.
  - XHTML 1.1 drugo izdanje W3C radni prijedlog 16.02.2007.
  - XHTML 2.0 W3C radni prijedlog 26.07.2006.
    - Nekompatibilan s XHTML 1.1
  - HTML 5 (i XHTML 5) W3C radni prijedlog 14.1.2008.
    - Različit od XHTML 2.0 – drugi pravac

# Linkovi

---



- W3C XHTML specifikacija
  - [www.w3.org/TR/xhtml1](http://www.w3.org/TR/xhtml1)
  - [www.w3.org/TR/2001/REC-xhtml11-20010531](http://www.w3.org/TR/2001/REC-xhtml11-20010531)
  - [www.w3.org/TR/xhtml11](http://www.w3.org/TR/xhtml11)
  - [www.w3.org/TR/xhtml2](http://www.w3.org/TR/xhtml2)
- Validator
  - [validator.w3.org](http://validator.w3.org)



XML - DOM

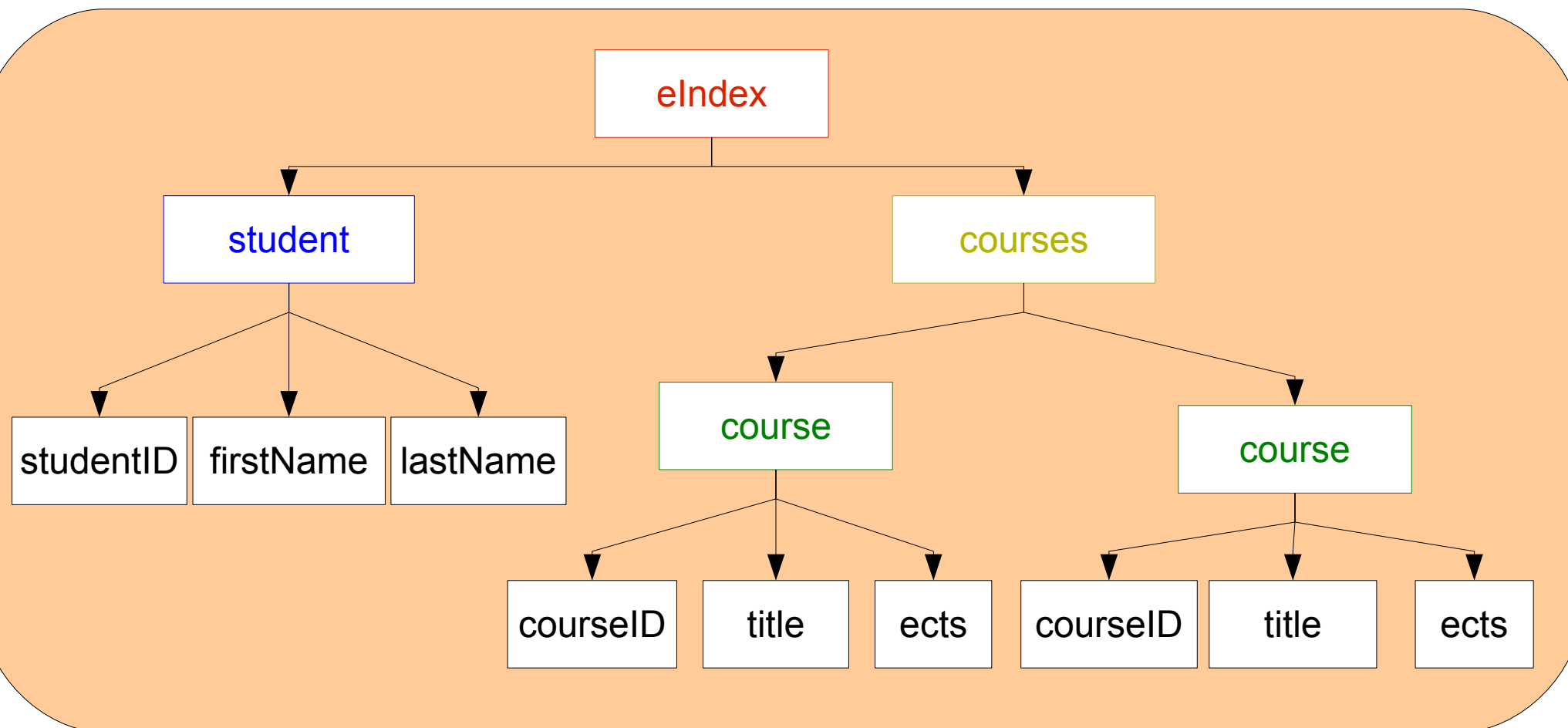


# eIndex XML dokument



```
<?xml version="1.0" ?>
<!DOCTYPE eindex SYSTEM "eIndex.dtd">
<!-- eIndex XML dokument -->
<eindex>
  <student status="exchange">
    <studentID>00362342433</studentID>
    <firstName>Ole Gunnar</firstName>
    <lastName>Solskjær</lastName>
    ...
  </student>
  <!-- popis predmeta -->
  <courses>
    <course status="module">
      <courseID>361</courseID>
      <title lang='hr'>Otvoreno računarstvo</title>
      <ects>4</ects>
      ...
    </course>
    <course status="orientation">
      <courseID>225</courseID>
      <title lang='hr'>Baze podataka</title>
      <ects>4</ects>
      ...
    </course>
  </courses>
</eindex>
```

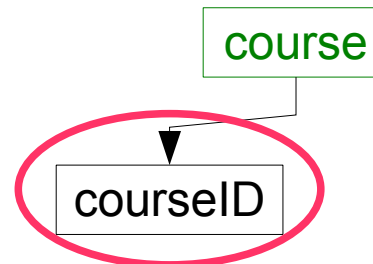
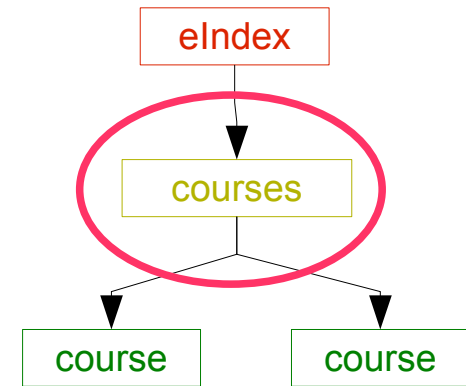
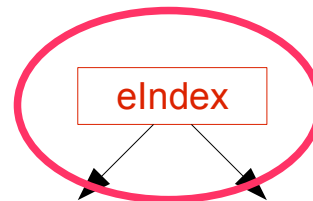
# Struktura eIndex dokumenta



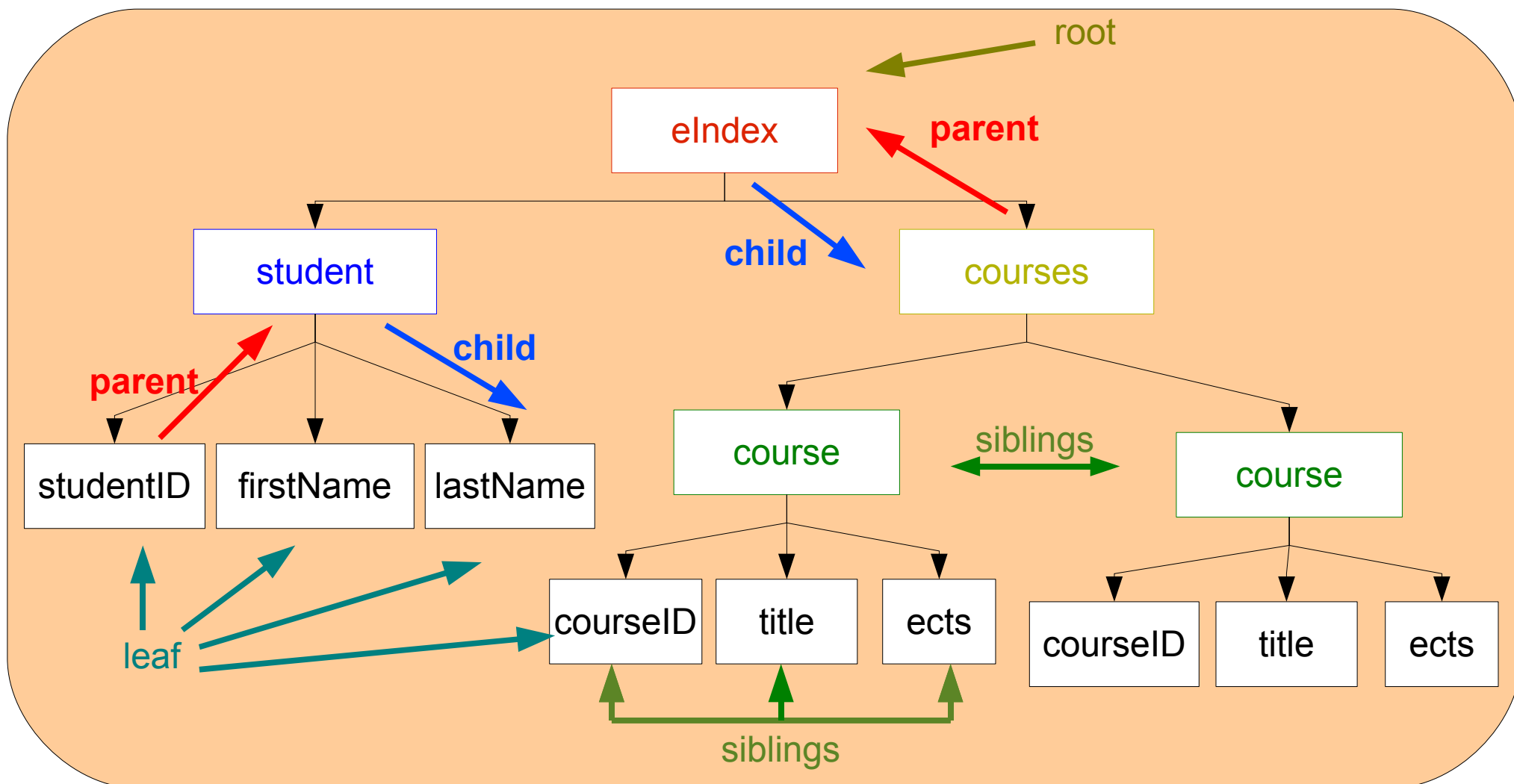
Nisu prikazani: XML deklaracija, deklaracija tipa dokumenta, komentari, atributi, ...

# Svojstva stabla

- **Stablo:** skup povezanih čvorova, čine aciklički povezani graf
- **čvorovi:**
  - mogu sadržavati vrijednost
  - 0..n čvorova djece
  - čvor roditelj
- **korijen**
  - nema roditelja
  - postoji put od korijena do svakog čvora stabla
- **listovi** – nemaju čvorova djece



# Odnosi čvorova u stablu

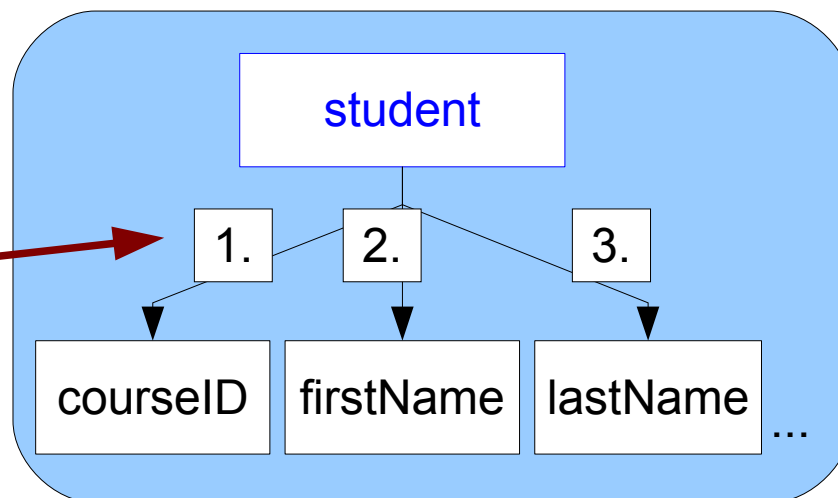


# Uređeno stablo



- **Neuređeno stablo**

- nema poretka između čvorova djece pojedinog čvora



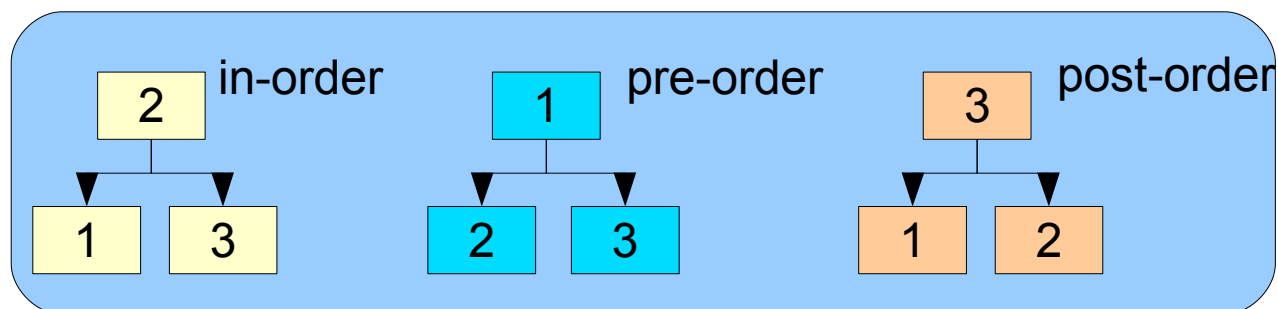
- **Uređeno stablo**

- postoji uređenost (poredak) čvorova djece pojedinog čvora

# Obilazak stabla



- Algoritmi nad strukturom podataka stablo
  - započinju s korijenom
  - koriste rekurziju
- Obilazak stabla:



- pre-order: čvor roditelj se obilazi prije obilaska čvorova djece
  - in-order, post-order: čvor roditelj se obilazi nakon obilaska čvorova djece
- Operacije nad stablom:
  - enumeracija elemenata, traženje elemenata, dodavanje elementa, dodavanje podstabla, brisanje podstabla, ...

# Document Object Model

---



*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

<http://www.w3.org/DOM/>

# Povijest i verzije DOM-a

---

- Početci razvoja sredinom 90-ih godina (W3C)
- Nulta verzija u sklopu HTML 4
- Specifikacije razdvojene u razine, svaka razina sastoji se od obveznih i izbornih modula
- DOM razina 1 – listopad 1998.
  - navigacija u strukturi (HTML i XML), promjena sadržaja i strukture
- DOM razina 2 – studeni 2000.
  - XML prostori imena, filtrirani pogledi, događaji
- DOM razina 3 – travanj 2004.
  - 6 specifikacija (core, load&save, XPath, pogledi i formatiranje, zahtjevi, validacija)



# XML DOM

---

- Objektni model XML dokumenta (koristi objektnu paradigmu za reprezentaciju komponenata dokumenta)
- Platformno i jezično neutralan (ne pogoduje specifičnom programskom jeziku ili operacijskom sustavu, postoje implementacije za sve važnije programske jezike i platforme)
- DOM definira standardni skup razreda koji reprezentiraju tipove komponenata XML dokumenta (elemente, attribute, komentare...)
- DOM definira standardne funkcije za rad s dokumentom (obilazak, pretraživanje, dodavanje, brisanje, mijenjanje vrijednosti ...)

# Komponente XML DOM-a

---

Document

– dokument u cjelini

DocumentType

– deklaracija tipa dokumenta

Element

– element

Attr

– atribut

ProcessingInstruction

– naredba obrade

Text

– jednostavan sadržaj (tekst)

EntityReference

– referenca na entitet

Comment

– komentar

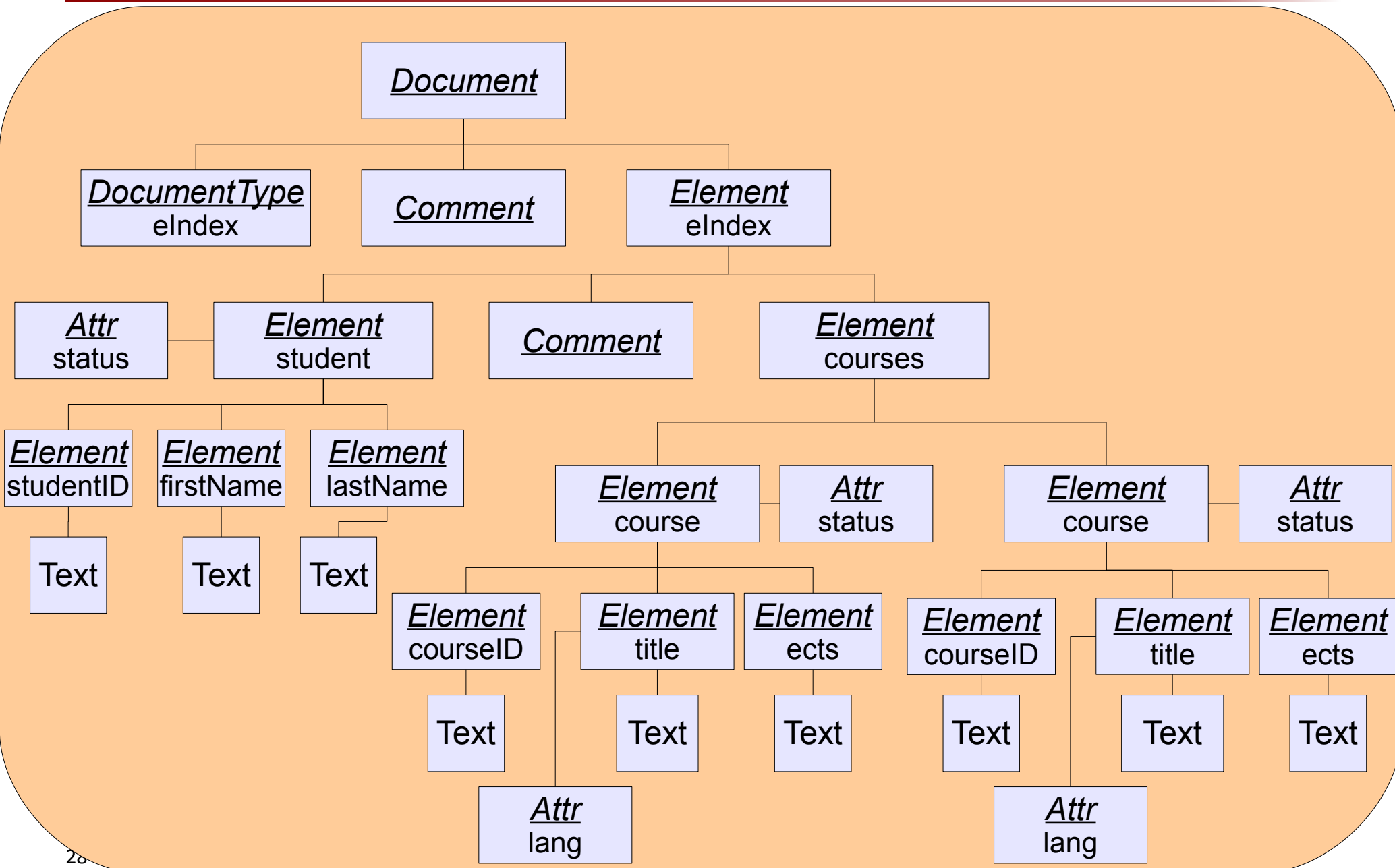
...

# XML dokument kao stablo

---

- Svaka komponenta je čvor stabla (tip Node)
- Komponente se kombiniraju u uređeno stablo koje reprezentira strukturu XML dokumenta
- Između komponenata dokumenta vrijede odnosi istovjetni odnosima u uređenom stablu
- Odnosi čvorova i vrijednosti čvorova izraženi vrijednostima svojstava pojedinog čvora

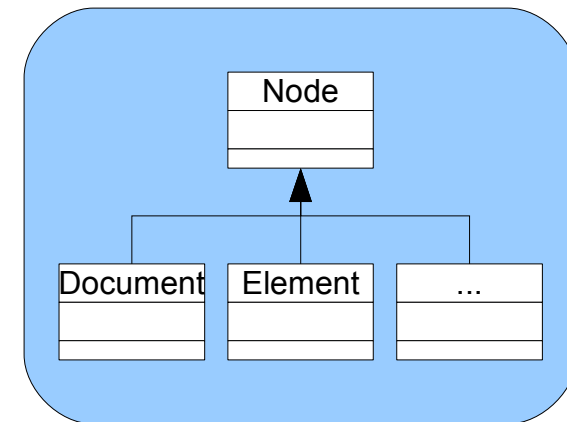
# DOM za primjer dokumenta eIndex



# Tipovi podataka specifični DOM-u



- Osnovni tip podatka: Node
- Tipovi podataka koji nasljeđuju Node:
  - komponente modela (Document, Element, Attr, ...) => nasljeđuju svojstva i metode
- NodeList
  - uređena lista elemenata tipa Node (poredak bitan)
  - elementu se pristupa preko pozicije u listi (indeksa)
  - koristi se za čuvanje liste elemenata djece
- NamedNodeMap
  - neuređen skup elemenata tipa Node
  - elementu se pristupa navođenjem njegova
  - koristi se za čuvanje liste atributa elementa



# Svojstva tipa *Node*

## strukturna

```
ownerDocument : Document
parentNode   : Node
childNodes   : NodeList
firstChild    : Node
lastChild     : Node
previousSibling : Node
nextSibling   : Node
attributes*   : NamedNodeMap
```

(\* - samo čvor *Element*)

## podatkovna

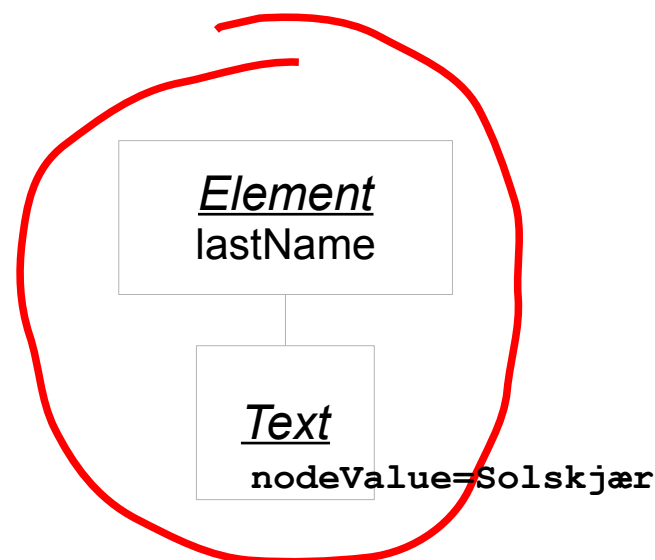
```
nodeName : string
noteType  : number
nodeValue : string
```



# Jednostavan sadržaj elementa



- Čvor tipa Element ne čuva jednostavan sadržaj izravno!
- Jednostavan sadržaj je u modelu reprezentiran tipom čvora Text
- Čvor tipa Attr izravno sadržava vrijednost atributa





# Pregled vrijednosti svojstava

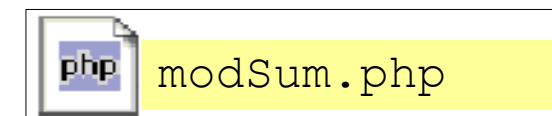
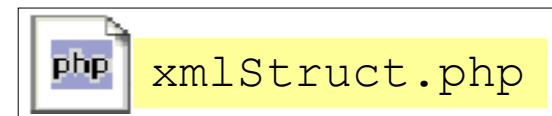


Tip čvora	nodeType	nodeName	nodeValue	Tipovi čvorova djece
Document	9	#document	null	Element (samo 1), ProcessingInstruction, Comment, DocumentType
DocumentType	10	<i>ime tipa</i>	null	Nema čvorova djece
ProcessingInstruction	7	<i>instrukcija</i>	<i>argumenti</i>	Nema čvorova djece
EntityReference	5	<i>ime ref. na entitet</i>	null	Element, Text, Comment, ProcessingInstruction, CDATASection
Element	1	<i>ime elementa</i>	null	Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference
Attr	2	<i>ime atributa</i>	<i>sadržaj</i>	Text, EntityReference
Text	3	#text	<i>sadržaj</i>	Nema čvorova djece
CDATASection	4	#cdata-section	<i>sadržaj</i>	Nema čvorova djece
Comment	8	#comment	<i>sadržaj</i>	Nema čvorova djece

# Kretanje po modelu

---

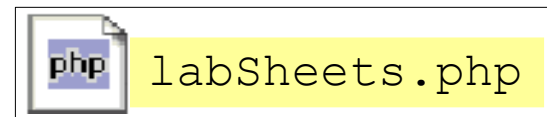
- U skladu s hijerarhijom elemenata modela
  - koriste se svojstva tipa čvora Node (parentNode, childNodes, nextSibling, attributes ...)
  -
- Izravan pristup elementima modela
  - metode GetElementsByTagName(ime-elementa) i GetElementsByTagNameNS(namespace, ime-elementa)
  - pronađeni čvorovi zapisani u tipu NodeList
  - mjesto u listi odgovara položaju čvora u modelu kod pre-order obilaska stabla



# Stvaranje novih komponenata

---

- Metode za stvaranje novih komponenata (čvorova) modela definirane samo na tipu čvora Document
  - `createAttribute`, `createAttributeNS`, `createElement`, `createElementNS`, `createComment`, `createTextNode`, `createProcessingInstruction` ...
- Novostvoreni čvor je nužno dodati na željeno mjesto unutar stabla modela
  - nakon stvaranja, čvor ne postaje automatski i dio modela!

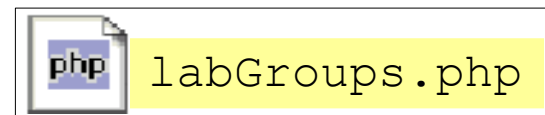


# Dodavanje komponenata u model

---



- Metode definirane na čvoru tipa Node
  - appendChild() - dodaje na kraj liste djece
  - insertBefore() - dodaje ispred navedenog čvora djeteta
- Metode definirane na čvoru tipa Element
  - setAttribute() - dodaje atribut u listu atributa elementa
  - setAttributeNode() - dodaje čvor atribut u listu atributa elementa

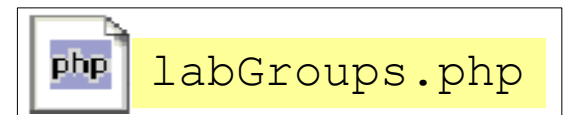


# Promjena vrijednosti komponenata

---



- Samo na tipovima čvorova koji izravno sadrže vrijednosti:
  - Text, CDATASection, Comment (data, nodeValue, appendData(), deleteData(), insertData(), replaceData() ...)
  - Attr (value, nodeValue)

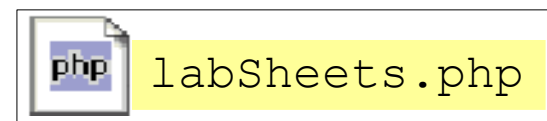
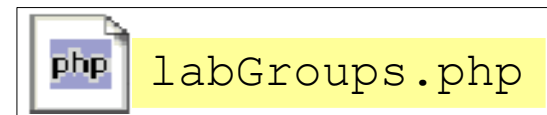


# Brisanje komponentata iz modela

---



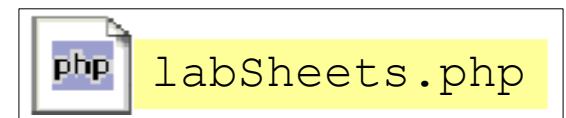
- Metode definirane na čvoru tipa Node
  - `removeChild()` - briše čvor dijete
  - `replaceChild()` – zamjenjuje čvor dijete s drugim
  -
- Metode definirane na čvoru tipa Element
  - `removeAttribute()` - briše atribut iz liste atributa elementa
  - `removeAttributeNode()` - briše čvor atribut iz liste atributa elementa



# Ostale akcije

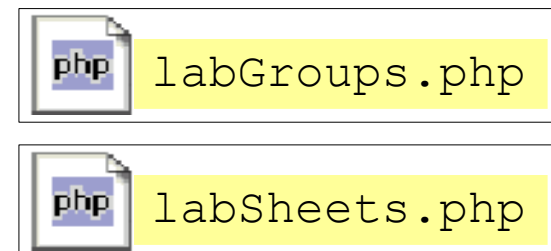
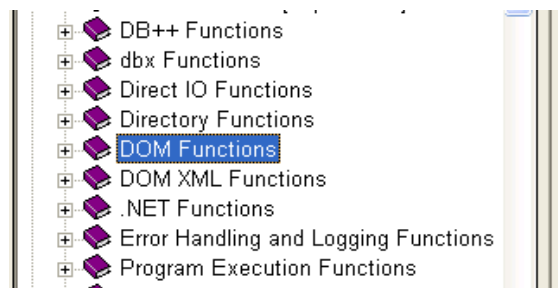
---

- Kloniranje čvorova – `cloneNode()` - novi čvor kopija originalnog čvora:
  - plitka kopija: klonira se samo navedeni čvor
  - duboka kopija: klonira se čitavo podstablo
- Metode za prepoznavanje i rad sa prostorima imena
- Metode za ispitivanje postojanja čvorova djece i atributa
- Metode za ispitivanje postojanja čvorova djece i atributa
- ...



# Pretvorba XML <-> DOM

- Uvoz i parsiranje postojećeg XML dokumenta
  - XML dokument mora biti dobro oblikovan
  - moguća provjera valjanosti (DTD ili shema moraju biti raspoloživi)
- Programsko stvaranje novog stabla dokumenta
  - DocumentImplementation objekt, metode:
    - createDocument(nsURI, ime-dokumenta, tip-dokumenta)
    - createDocument(ime-dokumenta, pubID, systemID)
- Izvoz DOM modela u XML
- PHP DOM:





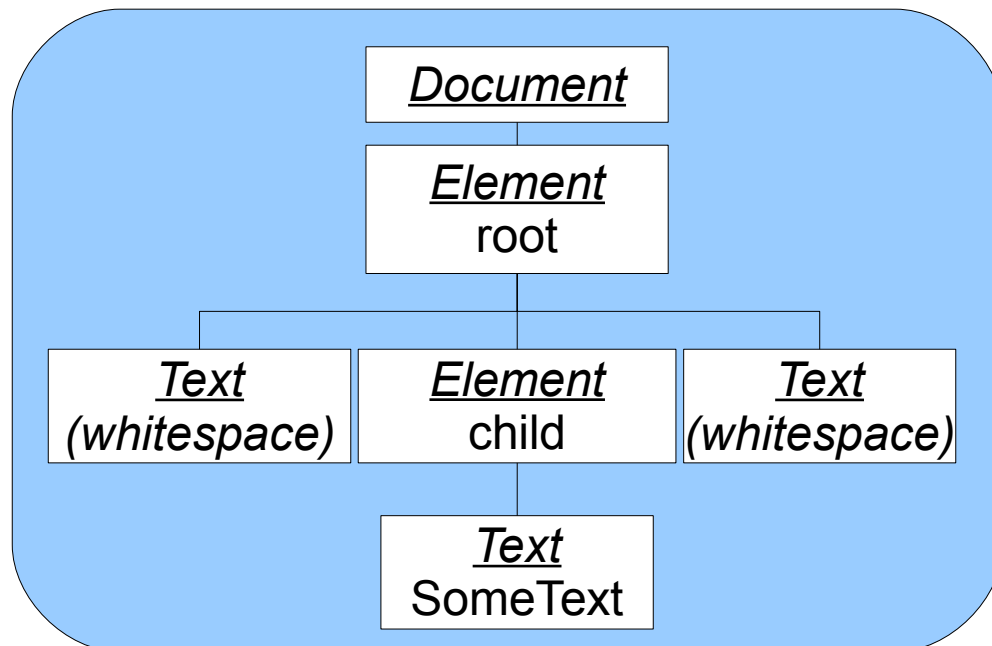
# Whitespace (I)



XML dokument za uvoz unutar aplikacije (xmlStruct.xml):

```
<?xml version='1.0' ?>
<root>
  <child>Some text</child>
</root>
```

Kako izgleda DOM ovog dokumenta nakon parsiranja?

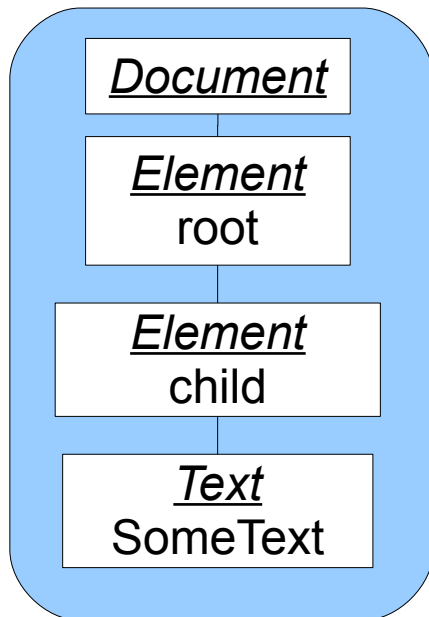


```
#document
  root
    #text x [
    ]
    child
      #text [Some
text]
      #text x [
]
```

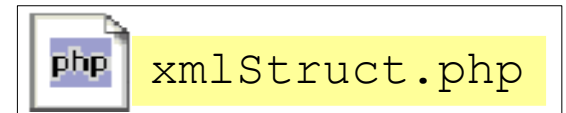
# Whitespace (II)



A kada bismo htjeli da DOM izgleda ovako:



```
#document
  root
    child
      #text [Some text]
```



```
<?xml version='1.0' ?>
<root><child>Some text</child></root>
```



XSL

XPath

# Što je XSL?

---



- XSL ili EXtensible Stylesheet Language
- Porodica preporuka za definiranje transformacija i prezentacije XML dokumen(a)ta
- Sastoji se od 3 dijela:
  - Jezik izraza za pristup i lociranje (adresiranje) dijelova XML dokumenta
    - XPath
  - Jezik za transformacije
    - XSLT (EXtensible Stylesheet Language Transformations)
  - Rječnik za formatiranje XML dokumenta za prezentaciju
    - XSL-FO (XSL Formatting Objects)
      - Formalni naziv za XSL-FO je zapravo “XSL”
- W3C o XSL: [www.w3.org/Style/XSL](http://www.w3.org/Style/XSL)

# Što je XPath?

---



- XPath ili XML Path Language
- Jezik za specifikaciju upita nad XML podacima
  - Kao SQL za upite nad relacijskim bazama
- Lociranje (adresiranje) dijela XML dokumenta
  - Pronalaženje elemenata i atributa koji odgovaraju određenom kriteriju
  - Traženje u bilo kom smjeru (unaprijed, unatrag, u oba smjera)
- Npr. pronaći sva poglavlja knjige koja govore o XPath-u

# Zašto naziv XPath?

---

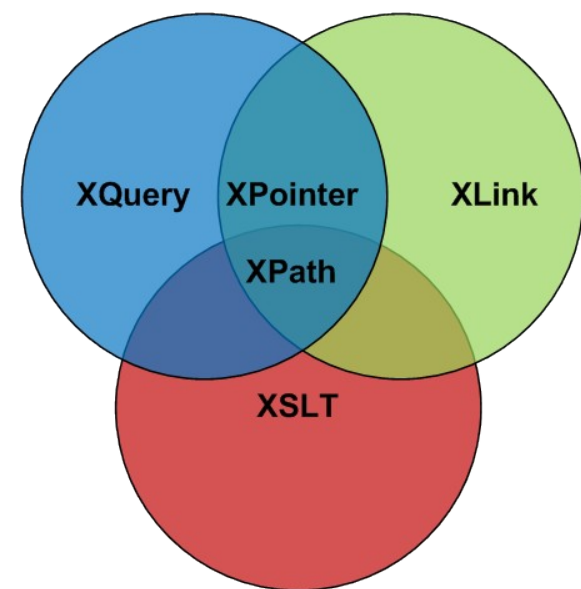


- X dolazi od XML
- Path dolazi od...
  - XML dokumenti se najčešće prezentiraju kao stablo čvorova (hijerarhija)
  - Izrazi opisuju put (path) do određenog čvora ili skupa čvorova
  - Sličnosti:
    - DOS/Windows path – c:\My Documents\temp
    - UNIX – /docs/temp
    - URI – http://www.fer.hr/docs/temp.html
- ZA RAZMIŠLJANJE: Koja je razlika apsolutnog (potpunog) i relativnog puta?

# Povijest XPath-a



- Inačice:
  - XPath 1.0 W3C preporuka 16.11.1999.
  - XPath 2.0 W3C preporuka 23.01.2007.
- Zašto je nastao?
  - XPath je kod uporabe ugrađen u neki drugi jezik
  - Definiran tijekom razvoja XSLT standarda (XML Stylesheet Language Transformation) i XPointer-a
    - XSLT koristi podskup XPath-a
    - XPointer koristi XPath i dodatnu sintaksu
  - Standardni način upita koji druge tehnologije nasljeđuju
    - XQuery koristi i nadograđuje XPath



# XPath 2.0



- Definicija: Jezik izraza (*expression language*) koji dozvoljava procesuiranje nad podatkovnim modelom
- Podatkovni model
  - Apstraktna, logička struktura XML dokumenta
  - XDM ili XQuery/XPath Data Model
  - Stablata reprezentacija XML dokumenata
  - Sadrži i atomarne vrijednosti
    - Cijele brojeve, znakovne nizove, istinosne vrijednosti i sljedove koji mogu sadržavati reference na druge čvorove ili atomarne vrijednosti



# Zašto XPath 2.0?

---

- XPath 2.0 je nadskup XPath 1.0
  - Podržava bogatiji skup podatkovnih tipova
  - Iskorištava tip informacije kod validacije putem XML Scheme
  - Većinom kompatibilan unatrag s XPath 1.0
- Rezultat XPath izraza može biti skup čvorova ili atomarnih vrijednosti, odnosno bilo koji slijed u podatkovnom modelu
- XPath 1.0 još uvijek šire rasprostranjen od XPath 2.0

# XML primjer



```
<?xml version="1.0"?>
```

```
<knjiga>
```

```
  <autor>Marko</autor>
```

```
  <naslov>XPath</naslov>
```

```
  <cijena>1</cijena>
```

```
</knjiga>
```

**ROOT** - adresa `"/`

**Knjiga** - adresa `"/knjiga"`

**Autor** - adresa `"/knjiga/autor"`

**"Marko"** - adresa `"/knjiga/autor/text()"`

**Naslov** - adresa `"/knjiga/naslov"`

**"XPath"** - adresa `"/knjiga/naslov/text()"`

**Cijena** - adresa `"/knjiga/cijena"`

**"1"** - adresa `"/knjiga/cijena/text()"`

# Izrazi, ključne riječi, funkcije

---

- Osnovni blok je **XPath izraz** (*expression*)
  - Znakovni niz (Unicode)
  - Može biti gniježđen
- Ključne riječi – malim slovima
- Sadrži biblioteku standardnih funkcija
  - 100-njak funkcija
  - Znakovne, numeričke, istinosne i vremenske
  - Za manipulaciju čvorovima i slijedovima

# XPath čvorovi

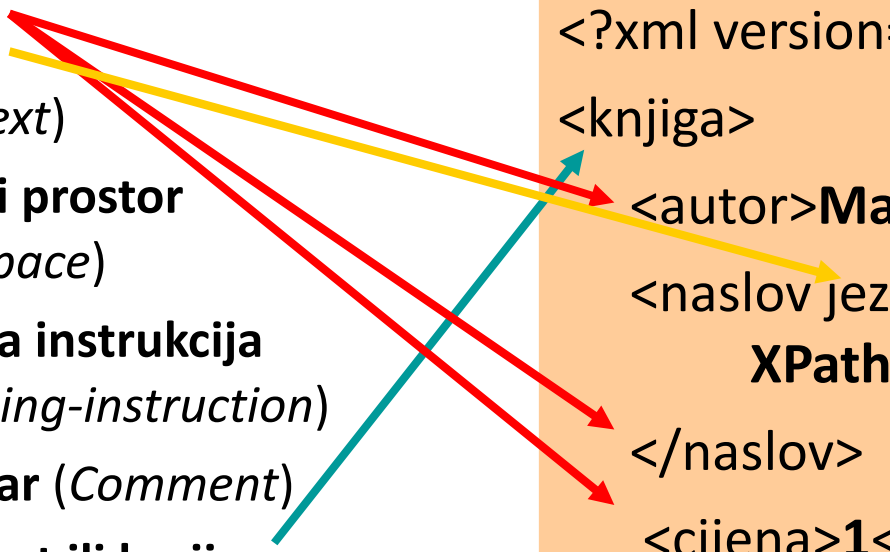


- Sedam tipova čvorova:

- **Element**
- **Atribut**
- **Tekst** (*Text*)
- **Imenički prostor** (*Namespace*)
- **Procesna instrukcija** (*Processing-instruction*)
- **Komentar** (*Comment*)
- **Dokument ili korijen** (*Document – root*)

*Primjer:*

```
<?xml version="1.0"?>
<knjiga>
  <autor>Marko</autor>
  <naslov jezik="hr">
    XPath
  </naslov>
  <cijena>1</cijena>
</knjiga>
```



# XPath terminologija

---



- Atomarna vrijednost (*atomic value*)
  - Čvor koji nema djecu ni roditelja
- Članak (*item*)
  - Članak je čvor ili atomarna vrijednost
- Roditelj (*parent*)
  - Svaki element i atribut ima jednog roditelja
- Dijete (*child*)
  - Elementi mogu imati djecu
- Braća, sestre, djeca istog roditelja (*sibling*)
  - Čvorovi koji imaju istog roditelja
- Predak (*ancestor*)
  - Roditelj čvora ili roditelj roditelja ili ...
- Potomak (*descendant*)
  - Dijete čvora ili dijete djeteta ili ...

# Primjeri



- Atomarna vrijednost
  - Marko ili hr su atomarne vrijednosti
- Roditelj
  - Knjiga je roditelj naslova, autora i cijene
- Dijete
  - Naslov, autor i cijena su djeca knjige
- Braća
  - Naslov, autor i cijena su braća
- Predak
  - Knjižnica je predak autora
  - Knjižnica je predak knjige
- Potomak
  - Naslov je potomak knjige
  - Naslov je potomak knjižnice

*Primjer:*

```
<?xml version="1.0"?>
```

```
<knjižnica>
```

```
  <knjiga>
```

```
    <autor>Marko</autor>
```

```
    <naslov jezik="hr">
```

**XPath**

```
  </naslov>
```

```
    <cijena>1</cijena>
```

```
  </knjiga>
```

```
</knjižnica>
```

# Evaluacija XPath izraza

---

- XPath izraz je slijed koraka (u stablu)
  - Svaki korak je izjava za pretragu
- XPath izraz sadrži trenutni kontekst
  - Čvor u stablu je početna točka koraka
- Svaki korak osim zadnjeg se evaluira u skup čvorova u stablu
- Zadnji korak vraća: broj, istinosnu vrijednost, znakovni niz ili skup čvorova

# Odabir čvor(ov)a

---

- Specijalni znakovi

/	odabir od korijenskog čvora
//	odabir od trenutnog čvora bez obzira na položaj
.	odabir trenutnog čvora
..	odabir roditelja trenutnog čvora
@	odabir atributa
*	bilo koji element čvora
@*	bilo koji atribut čvora
	odabir izraza prije i izraza iza ovog znaka



# Primjer odabira čvorova



knjižnica	odabir sve djece elementa knjižnica
/knjižnica	odabir korijenskog elementa knjižnica
knjižnica/knjiga	odabir svih knjiga koji su djeca knjižnice
//knjiga	odabir svih knjiga bez obzira gdje su u dokumentu
knjižnica//knjiga	odabir svih elementa knjiga, bilo gdje, ali potomaka elementa knjižnica
//@jezik	odabir svih atributa jezik bilo gdje u dokumentu
/knjižnica/*	odabir sve djece elementa knjižnica
//*	odabir svih elemenata u dokumentu

- Filtar
- Služe za odabir čvora sa specifičnom vrijednosti
- Uokvireni uglatim zagradama – [ ]
- Primjeri:
  - knjiga[1]
  - knjiga[last()]
  - knjiga[position()-1]
  - //naslov[@jezik='hr']

# Trenutni kontekst

---

- Označava “gdje sam” trenutno
- Aktivni element u XPath-u adresira korak
  - /Root/.../predak/roditelj/ČVOR/dijete/potomak
- ČVOR je uvijek jednostruki čvor
  - Može sadržavati samo jednog roditelja i jedan korijen
- Apsolutni put
  - /korak/korak/korak...
- Relativni put
  - korak/korak/korak...

# Sintaksa koraka

---

- XPath lokacija puta sadrži jedan ili više koraka odijeljenim kosom crtom (“/”)
- Svaki korak sadrži
  - Os – relacija u stablu čvorova odabranog čvora u odnosu na trenutni kontekst (čvor)
  - Čvor – čvor u odabranoj osi
  - Predikat – filter nad odabranim čvoro(vi)m(a)
- Sintaksa: /os::čvor[predikat]/

# Primjeri korištenja osi



child::knjiga	Svi čvorovi tipa <b>knjiga</b> koji su djeca trenutnog čvora
attribute::jezik	Svi atributi tipa <b>jezik</b> trenutnog čvora
child::*	Sva djeca trenutnog čvora
child::text()	Svi tekstualni čvorovi djeca trenutnog čvora
child::node()	Svi čvorovi djeca trenutnog čvora
descendant::knjiga	Svi potomci tipa <b>knjiga</b> trenutnog čvora
ancestor::knjiga	Svi preci tipa <b>knjiga</b> trenutnog čvora
ancestor-or-self::knjiga	Svi preci tipa <b>knjiga</b> trenutnog čvora i on sam ako je <b>knjiga</b>
child::* / child::cijena	Svi unuci tipa <b>cijena</b> trenutnog čvora

# XPath operatori



	Pripajanje dva skupa čvorova
+	Zbrajanje
-	Oduzimanje
*	Množenje
div	Cjelobrojno dijeljenje
=	Logička usporedba jednakosti
!=	Logička usporedba nejednakosti
<	Logičko manje
<=	Logičko manje ili jednako
>	Logičko veće
>=	Logičko veće ili jednako
or	Logičko ili
and	Logičko i
mod	Ostatak pri cjelobrojnem dijeljenju

# Linkovi

---



- W3C XPath specifikacija
  - [www.w3.org/TR/xpath](http://www.w3.org/TR/xpath)
  - [www.w3.org/TR/xpath20](http://www.w3.org/TR/xpath20)
- XPath Tutorial
  - [www.w3schools.com/xpath](http://www.w3schools.com/xpath)
  - [www.zvon.org/xxl/XPathTutorial/General/examples.html](http://www.zvon.org/xxl/XPathTutorial/General/examples.html)
- XPath popis funkcija
  - [www.w3schools.com/xpath/xpath\\_functions.asp](http://www.w3schools.com/xpath/xpath_functions.asp)
- XPath osi
  - [www.xml.com/pub/a/2000/12/20/xpathaxes.html](http://www.xml.com/pub/a/2000/12/20/xpathaxes.html)
  - [www.xml.com/pub/a/2001/01/03/xpathaxes.html](http://www.xml.com/pub/a/2001/01/03/xpathaxes.html)



XSLT



# Što je XSLT?

---



- **XSLT** ili **EX**tensible **S**tylesheet **L**anguage **T**ransformations
- Jezik temeljen na XML
- Služi za **transformacije XML dokumenata** u druge XML dokumente ili neki drugi oblik
- Iz originalnog XML dokumenta se na temelju transformacijskih pravila stvara novi dokument

# XSLT značajke

---



- Tipovi odredišnog dokumenta:
  - XML dokument
  - HTML ili XHTML dokument
  - Čisti tekstualni dokument (*plain text*)
- Mogućnosti:
  - Dodavanje i prikaz elemenata i atributa
  - Brisanje i sakrivanje nepoželjnih dijelova
  - Uređivanje, oblikovanje i sortiranje podataka
  - Testiranje i uvjetno izvršavanje pravila

# XSLT kao jezik

---

- Zasnovan na temeljima:
  - Funkcionalnih jezika
    - Npr. Erlang, Mathematica, R, ML, ...
  - Jezika usporedbe tekstualnih izraza
    - Npr. SNOBOL, awk, regularni izrazi, ...
  - Procesora predložaka (template processor)
    - Npr. JSP ili ASP, kao i podrška u C, Javi, Perlu, ...
  - Turingovog stroja
- Predak – jezik DSSSL (Document Style Semantics and Specification Language)
  - Ista funkcija kao XSLT, samo nad SGML dokumentima
- Koristi XPath za navigaciju u XML podacima

# Povijest XSLT i značajke

---

- Verzije:
  - XSLT 1.0 W3C preporuka 16.11.1999.
  - XSLT 1.1 - nije uspio
  - XSLT 2.0 W3C preporuka 23.01.2007.
- Zašto je nastao?
  - Potreba za transformacijom XML dokumenata
- Deklarativni jezik
  - Ne navodi konkretne akcije nad nekim stanjima
  - Navodi pravila koja upućuju što raditi s određenim čvorom (XML elementom)

# XSLT model

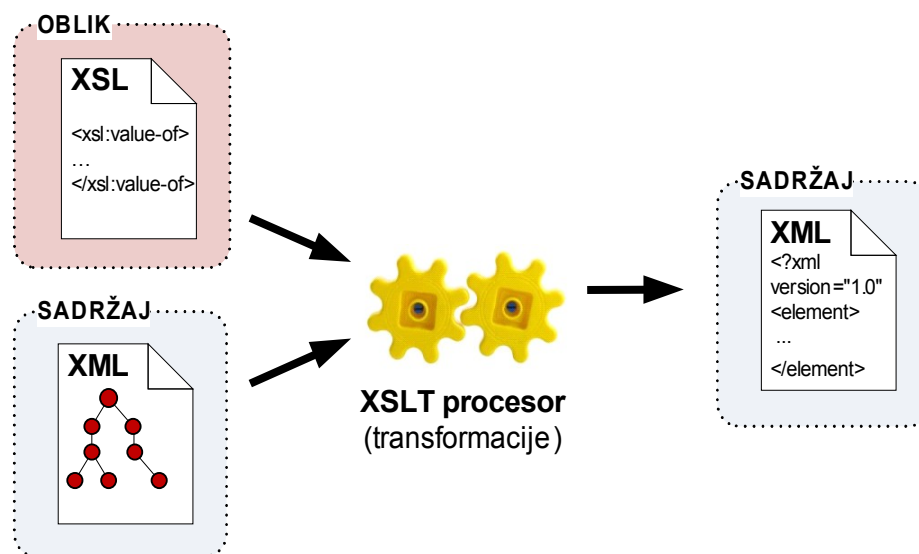
---

- XSLT model uključuje:
  - XML izvorišni dokument
  - XSLT predložak stila (*stylesheet*)
  - XSLT procesor (*processor, processing engine*)
  - Posljedični (rezultantni) dokument
- XSLT predložak stila sadrži niz pravila i drugih direktiva pomoću kojih daje upute procesoru kako napraviti novi dokument
  - Po strukturi to je isto XML dokument

# XSLT procesor



- Ulaz
  - XML izvorišni dokument (npr. datoteka “.xml”)
  - XSLT predložak stila (npr. datoteka “.xsl”)
- Izlaz
  - Proizvodi novi XML dokument
    - Može biti XML, HTML, XHTML, ...



# Postupak procesuiranja

---

- Uzima se izvorišni XML dokument
- Iz njega se gradi stablasta struktura
- Ponavlja se postupak:
  - Putem XPath izraza se pronalaze odgovarajući čvorovi
  - Nad pronađenim čvorovima se primjenjuju pravila
  - Stvaraju se novi elementi (čvorovi) u odredišnom dokumentu

# XSLT procesor

---



- Može se nalaziti na:
  - Poslužitelju (*server-side*)
  - Klijentu (*client-side*)
- Može biti:
  - Samostalna aplikacija
  - Sadržan u Web pregledniku
  - Sadržan u aplikacijskom poslužitelju
  - Sadržan u programskom okruženju (*framework*)
  - Sadržan u operacijskom sustavu



# Podrška kod preglednika i alati

---



- Podrška kod preglednika:
  - MS Internet Explorer – verzija 6 podržava XSLT 1.0
  - Mozilla FireFox – od verzije 1.0.2 podržava
  - Mozilla – djelomično
  - Netscape – od verzije 8 koristi Mozilla engine
  - Opera – od verzije 9 podržava
- Alati
  - Niz specijaliziranih alata za manipulaciju XML i XSL dokumentima
    - Npr. Altova XMLSpy

# Poznatiji XSLT procesori

---

- Poznatiji XSLT procesori
  - Saxon – besplatan, Java, podržava i XSLT 2.0
  - Xalan – besplatan, Java i C++, dio Apache XML Project
  - xsltproc – besplatan, C, dio libxml2 biblioteke
  - XT – besplatan, Java
  - MSXML – samo na Windowsima
  - Sablotron – C++, ugrađen u PHP
  - 4XSLT – Python

# Deklaracija predloška stila

---



- Korijenski element
  - `<xsl:stylesheet>` ili `<xsl:transform>`
  - Potrebna deklaracija verzije i imeničkog prostora
    - Primjer:  
`<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">`
- Spajanje izvorišnog XML dokumenta i XSLT predloška stila
  - Umetanje povezivanja nakon deklaracije, a prije korijenskog dokumenta u izvorišnom (XML) dokumentu
    - `<?xml-stylesheet type="text/xsl" href="predlozak.xsl"?>`
  - Otvaranjem spojenog XML dokumenta s XSLT predloškom stila u pregledniku, transformacija će se izvršiti automatski

# Predlošci

---



- XSL predložak stila (*stylesheet*) sadrži jedan ili više pravila, tzv. predložaka (*templates*)
- Svaki predložak sadrži pravila koja se izvršavaju kada se locira odgovarajući čvor
- Predložak koristi `<xsl:template>` element i `match` atribut
  - Match atribut se koristi za pronalaženje XML podatka
  - Vrijednost match atributa određuje pronalazak čvora

# <xsl:value-of> element

---

- **<xsl:value-of>** služi za dohvaćanje vrijednosti odabranog čvora
- Za vrijednost **select** atributa koristi se XPath izraz
- Primjer:  
`<xsl:value-of select="knjiznica/knjiga/naslov">`

# Primjer korištenja <xsl:value-of>



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <h1>Tablica</h1>
    <table border="1">
      <tr>
        <th>Naziv</th>
        <th>Autor</th>
      </tr>
      <tr>
        <td><xsl:value-of select="knjiznica/knjiga/naziv"/></td>
        <td><xsl:value-of select="knjiznica/knjiga/autor"/></td>
      </tr>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

**SAMO JEDNA KNJIGA!**

# <xsl:for-each> element

---

- <xsl:for-each> služi za pristup svim elementima u određenom skupu čvorova
- Za vrijednost **select** atributa koristi se XPath izraz (opet)
- Primjer: želimo prikazati više knjiga iz knjižnice

# Primjer korištenja <xsl:for-each>



...

```
<tr>
  <th>Naziv</th>
  <th>Autor</th>
</tr>
<xsl:for-each select="knjiznica/knjiga">
  <tr>
    <td><xsl:value-of select="naziv"/></td>
    <td><xsl:value-of select="autor"/></td>
  </tr>
</xsl:for-each>
```

...



# Filtriranje

---



- Dodatno je moguće na `<xsl:for-each>` elementu napraviti filtriranje dodavanjem uvjeta na `select` atribut
- Koriste se logički operatori (`=`, `!=`, `&lt;`, `&gt;`;)
- Primjer:

```
<xsl:for-each select="knjiznica/knjiga[autor='August  
Šenoa']">
```

...

```
</xsl:for-each>
```

# Uvjetni izlaz - <xsl:if> element

---

- Uvjet može biti i nezavisan od petlje
- Postavljanje uvjeta nad izlaz pomoću <xsl:if> elementa
  - Sadrži atribut test u koji se upisuje uvjet ispitivanja

- **Primjer:**

```
<xsl:for-each select="knjiznica/knjiga">
```

```
...
```

```
  <xsl:if test="autor='August Šenoa'">
```

```
    ...
```

```
  </xsl:if>
```

```
...
```

```
</xsl:for-each>
```

# Višestruki uvjet – <xsl:choose> element

- Za više uvjeta ne koristimo <xsl:if>
- **<xsl:choose>** element se koristi zajedno s **<xsl:when>** i **<xsl:otherwise>** elementima
  - <xsl:when> element sadrži atribut **test** u koji se upisuje uvjet ispitivanja

Primjer:

```
<xsl:for-each
  select="knjiznica/knjiga">
  ...
  <xsl:choose>
    <xsl:when
      test="autor='August Šenoa'"/>
    ...
    </xsl:when>
    <xsl:when test="autor='Marija
      Jurić Zagorka'"/>
    ...
    </xsl:when>
    <xsl:otherwise>
    ...
    </xsl:otherwise>
  ...
  </xsl:choose>
</xsl:for-each>
```

# Sortiranje – <xsl:sort> element

---

- Moguće je i sortirati izlaz pomoću <xsl:sort> elementa
  - Umeće se unutar <xsl:for-each> elementa
  - Sadrži atribut select u kojem se zadaje po čemu se sortira
- Primjer:  

```
<xsl:for-each select="knjiznica/knjiga">  
  <xsl:sort select="autor"/>  
  ...  
</xsl:for-each>
```

# Više predložaka

---

- Na isti dokument može se primijeniti više raznih predložaka
- Svaki predložak (*template*) bi trebao odgovarati različitom čvoru
- Koristi se niz `<xsl:template>` elemenata
- Unutar `<xsl:template>` elementa koristi se `<xsl:apply-templates>` element
- Primjer:
  - Jedan osnovni predložak za cijelu knjižnicu
  - Posebni predložak za knjigu
  - Posebni predložak za naslov

# Primjer s više predložaka

---

- Primjer:

```
<xsl:template match="/">
  <html><body> <h1>Tablica</h1>
    <xsl:apply-templates/>
  </body> </html>
</xsl:template>
<xsl:template match="knjiga">
  <p>
    <xsl:apply-templates select="naslov"/>
  </p>
</xsl:template>
<xsl:template match="naslov">
  Naslov: <span style="color:#000000">
    <xsl:value-of select="."/></span> <br />
</xsl:template>
```

# Linkovi

---



- W3C XSLT specifikacije
  - [www.w3.org/TR/xslt](http://www.w3.org/TR/xslt)
  - [www.w3.org/TR/xslt20](http://www.w3.org/TR/xslt20)
- XSLT Tutorial
  - [www.xmlfiles.com/xsl](http://www.xmlfiles.com/xsl)
  - [www.zvon.org/xxl/XSLTutorial/Output/index.html](http://www.zvon.org/xxl/XSLTutorial/Output/index.html)



XSL-FO



# Što je XSL-FO?

---



- XSL-FO ili EXtensible Stylesheet Language Formatting Objects
- Jezik temeljen na XML-u
- Služi za formatiranje XML dokumenata u oblik sa stranicama koji se može ispisivati
  - Dokument za ispis na papir
  - Dokument za pregled na zaslonu u obliku stranica
- Formalni naziv za XSL-FO je zapravo XSL

# Povijest XSL-FO

---

- Verzije:
  - **XSL-FO** kao dio XSL 1.0 W3C preporuke 15.10.2001.
  - **XSL-FO** kao dio XSL 1.1 W3C preporuke 05.12.2006.
- Zašto je nastao?
  - Potreba za formatiranjem XML dokumenata u obliku sa stranicama koji se može i ispisivati
  - Unificirani prezentacijski jezik

# XSL-FO značajke

---

- Koristi FO oznake kod stvaranja izlaznog dokumenta
- Najčešći izlazni oblici:
  - PDF (Portable Document Format)
  - PS (PostScript)
  - RTF (Rich Text Format)
- Omogućava izradu raznih naprednih elemenata i sekcija
  - Npr. sadržaj knjige, indeks, reference, ...

# Dizajn uvjetovan sadržajem

---



- Dizajn uvjetovan sadržajem (*content-driven*)
  - Standardno se koristi za knjige, članke, pravne dokumente i sve što se ispisuje s prelomljenim stranicama
  - Suprotnost dizajnu uvjetovanom razmještajem (*layout-driven*)
    - Kod magazina i novina sadržaj mora stati u “okvir”, višak uglavnom nije dozvoljen i ne prelama se na drugu stranu, već se ne prikazuje
  - Uključuje opseg jednostrukog toka (*single-flow span*) sadržaja
    - Najčešće dulji tekst s ponavljajućim marginama, zaglavljem i sl.
    - Tekst (ili sadržaj) se prelama po stranicama izlaznog dokumenta po XSL-FO pravilima
- XSL-FO radi po principu dizajna uvjetovanog sadržajem
  - Različito od PDF ili PS dokumenata
  - Ne opisuje se definitivni izgled stranica, već relativni razmještaj područja s elementima sadržaja

# Struktura XSL-FO dokumenta

---

- XSL-FO dokument je XML dokument
- Sintaksa poštuje XSL-FO specifikaciju
  - Ali ne postoji DTD ili XML Schema
- Dvije obavezne sekcije:
  - **Skup imenovanih rasporeda stranica**
  - **Skup podataka dokumenta s oznakama**
    - Koriste razne rasporede stranica

# Struktura XSL-FO dokumenta 2

---



- Raspored stranice (*page layout*) definira značajke stranice
  - Smjer toka teksta, veličinu stranice, margine, razlike parnih i neparnih stranica, ...
- Sadržajni dio sadrži sljedove tokova
  - Svaki tok je pridružen rasporedu stranice
- Tok sadrži listu blokova koji sadrže tekstualne podatke i/ili oznake
- Blokovi se ponašaju slično kao kod CSS

# XSL-FO mogućnosti

---

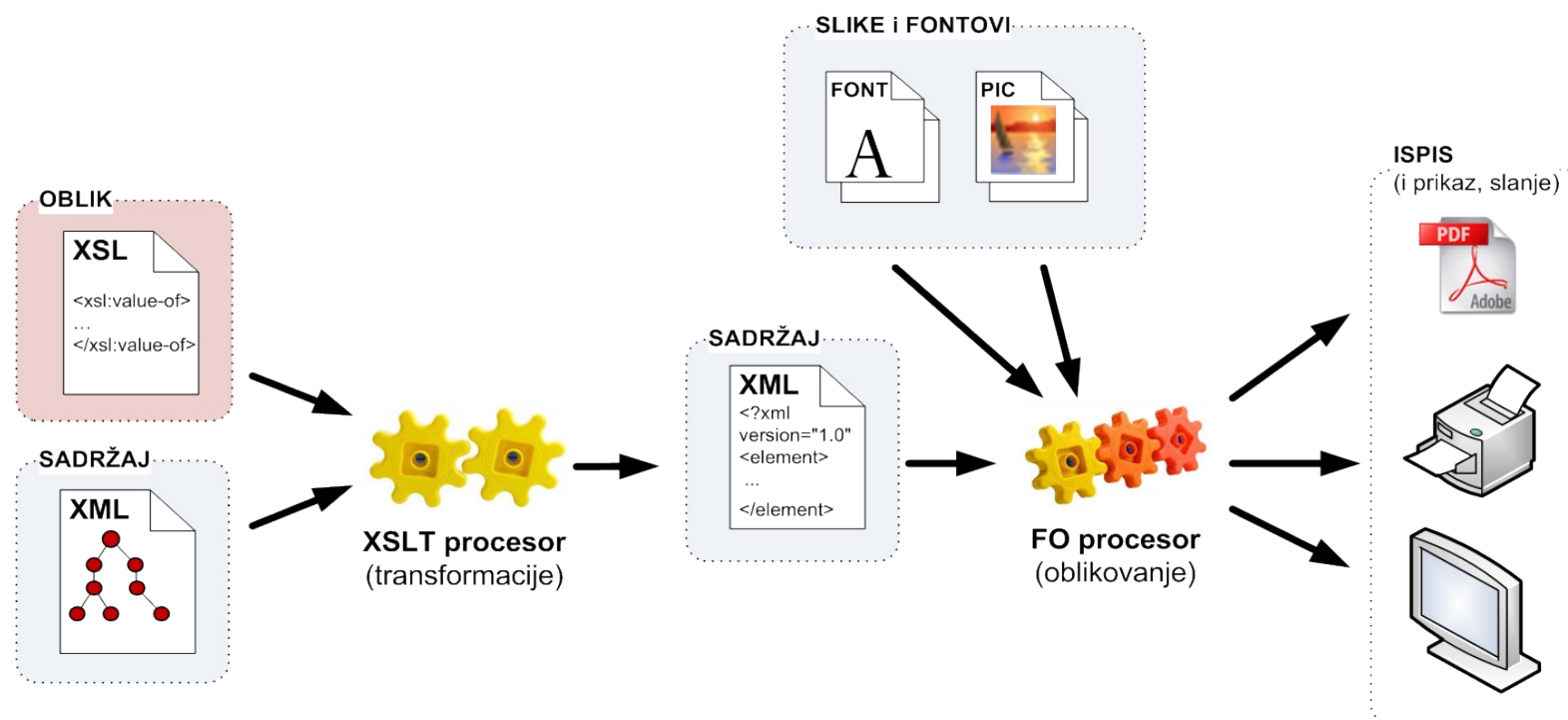


- XSL-FO 1.0 mogućnosti:
  - Višestruke kolone, liste, kontrola prijeloma stranica, zaglavlja, tablice, orijentacija teksta, boje pozadine, slike, fontove, ...
- XSL-FO 1.1 mogućnosti (dodatno):
  - Višestruke tokove, knjižne oznake (*bookmark*), indeksiranje, citate, napredno upravljanje grafičkim elementima

# Kako radi XSL-FO



- Dva koraka:
  - **1. korak:** Iz XML dokumenta s podacima i XSLT predložka stila se pomoću XSL transformacija stvara novi XSL-FO dokument
  - **2. korak:** Iz XSL-FO dokumenta, zajedno sa slikama i fontovima se pomoću XSL formatiranja stvara izlazni dokument





# Primjer osnovne strukture

---



```
<?xml version="1.0" encoding="ISO-8859-2"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

  <fo:layout-master-set>
    <fo:simple-page-master master-name="Naziv">
      <!-- Ovdje idu postavke predloška stranice -->
    </fo:simple-page-master>
    <!-- Ovdje mogu ići drugi simple-page-master predlošci -->
  </fo:layout-master-set>

  <fo:page-sequence master-reference="Naziv">
    <!-- Ovdje ide sadržaj stranice -->
  </fo:page-sequence>
  <!-- Ovdje mogu ići drugi page-sequence sadržaji-->

</fo:root>
```

# XSL-FO područja

---



- XSL-FO model definira pravokutna područja za prikaz sadržaja
  - Sadržaj se (bez obzira na tip) smješta u područja
- Područja uključuju:
  - Stranice (*pages*)
  - Regije (*regions*)
  - Blokove (*block areas*)
  - Linije (*line areas*)
  - Područje unutar linije (*inline areas*)

# Prednosti i nedostaci

---



- Prednosti:
  - Jednostavnost transformacija korištenjem XSLT
  - Sličnost s CSS
  - Manji troškovi kod dobavljanja i održavanja
  - Podrška za sve jezike
  - Zrelost standarda i manjak konkurencije
- Nedostaci:
  - Ograničenja kod kompliciranog dizajna i tipografije
  - Ograničenja jednostrukog prolaska (*single-pass*)
  - Postoje bolji, komercijalni produkti (QuarkXPress, InDesign, ...)

# Poznatiji XSL-FO procesori

---

- **Apache FOP** (Formatting Objects Processor), *The Apache XML Graphics Projects* – besplatan, Java, dio Apache
- **Unicorn Formatting Objects**, *Unicorn Enterprises SA* – nije besplatan, za Windows-e
- **XEP**, *RenderX* – nije besplatan, Java
- **XML2PDF**, *Altsoft* – nije besplatan, za Windows-e
- **XML Professional Publisher**, *XyEnterprise* – nije besplatan, Windows i Unix
- **xmlroff** – besplatan, C, temeljen na libxml2
- **XSL Formatter**, *Antenna House* – nije besplatan, Windows, Unix, Linux

- W3C XSL-FO specifikacije kao dio XSL
  - [www.w3.org/TR/xsl11](http://www.w3.org/TR/xsl11)
  - [www.w3.org/TR/2001/REC-xsl-20011015](http://www.w3.org/TR/2001/REC-xsl-20011015)
- XSL-FO Tutorial
  - [www.w3schools.com/xslfo](http://www.w3schools.com/xslfo)
  - [www.renderx.com/tutorial.html](http://www.renderx.com/tutorial.html)
  - [www.xml.com/pub/a/2002/03/20/xsl-fo.html](http://www.xml.com/pub/a/2002/03/20/xsl-fo.html)
- XSL-FO Reference
  - [www.w3schools.com/xslfo/xslfo\\_reference.asp](http://www.w3schools.com/xslfo/xslfo_reference.asp)
  - [www.zvon.org/xxl/xslfoReference/Output/index.html](http://www.zvon.org/xxl/xslfoReference/Output/index.html)
- Apache FOP
  - [xmlgraphics.apache.org/fop](http://xmlgraphics.apache.org/fop)
- xmlroff
  - [xmlroff.org](http://xmlroff.org)
- XEP
  - [www.renderx.com/tools/index.html](http://www.renderx.com/tools/index.html)

# Zaključak

