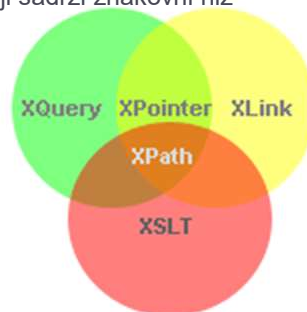


XPATH

Zoran Čirović

Značenje

- ▶ **XPath** = **XML Path Language**
- ▶ Jezik za identifikovanje delova XML dokumenta. Nije XML.
- ▶ Na primer, omogućava pisanje izraza koji upućuju na:
 - ▶ prvi element *osoba* u određenom dokumentu,
 - ▶ vrednost atributa ID prvog elementa osoba koji sadrži znakovni niz „Milica Jovanovic”
 - ▶ ...
- ▶ Na određeni čvor, upućuje se pomoću:
 - ▶ Položaja
 - ▶ Relativnog položaja
 - ▶ Tipa
 - ▶ Sadržaja
 - ▶



Zoran Čirović

Uvod

- ▶ *XPath* je standardizovan (W3C standard). Omogućava navigaciju do delova (čvorova) XML dokumenta (kao što su elementi, atributi, njihove vrednosti)
- ▶ XML dokument se posmatra kao stablo čvorova sa definisanim **korenom dokumenta**. Koren dokumenta je **bezimeni** čvor čije je dete **koreni element** XML dokumenta
- ▶ XPath definiše model stabla (*tree model*) za sve izraze u kojima se koristi.
- ▶ XPath sintaksno opisuje logičku strukturu XML dokumenta.



Struktura stabla XML dokumenta

- ▶ XML dokument je stablo koje se sastoji od čvorova.
- ▶ Čvorovi sadrže jedan ili više čvorova.
- ▶ Postoji tačno jedan korenski čvor.
- ▶ Sa stanovišta XPath izraza postoji 7 vrsta čvorova:
 1. Korenski
 2. Čvorovi elemenata
 3. Čvorovi teksta
 4. Čvorovi atributa
 5. Čvorovi komentara
 6. Čvorovi instr. za obradu
 7. Čvorovi prostora imena



Pažnja: čega nema u spisku vrsta čvorova?

1. Odeljak CDATA
 2. Referenci entiteta
 3. Deklaracija tipa dokumenta
- ▶ XPath obrađuje dokument tek pošto su sve navedene konstrukcije uključene u dokument.
 - ▶ Na primer: XPath ne može prepoznati odeljak CDATA, ne može prepoznati da li je određena vrednost atributa uključena neposredno u izvornom dokumentu ili je dobijena iz deklaracije u nekom DTDu.



Primer: XML dokument

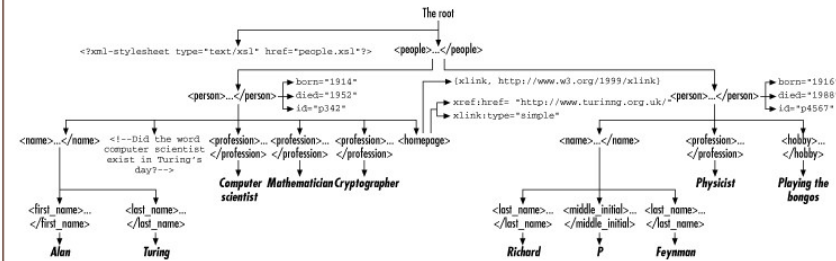
```
<?xml version="1.0"?>
<?xml-stylesheet type="application/xml"
  href="people.xml"?>

<!DOCTYPE people [
  <!ATTLIST homepage xlink:type CDATA
    #FIXED "simple"
    xmlns:xlink CDATA #FIXED
    "http://www.w3.org/1999/xlink">
  <!ATTLIST person id ID #IMPLIED>
]>

<people>
  <person born="1912" died="1954" id="p342">
    <name>
      <first_name>Alan</first_name>
      <last_name>Turing</last_name>
    </name>
    <!-- Did the word computer scientist exist in
      Turing's day? -->
    <profession>computer scientist</profession>
    <profession>mathematician</profession>
    <profession>cryptographer</profession>
    <homepage
      xlink:href="http://www.turing.org.uk"/>
    </person>
  <person born="1918" died="1988"
    id="p4567">
    <name>
      <first_name>Richard</first_name>
      <middle_initial>&#x50;</middle_initial>
      <last_name>Feynman</last_name>
    </name>
    <profession>physicist</profession>
    <hobby>Playing the bongoes</hobby>
    </person>
</people>
```



Primer: XML dokument - STABLO



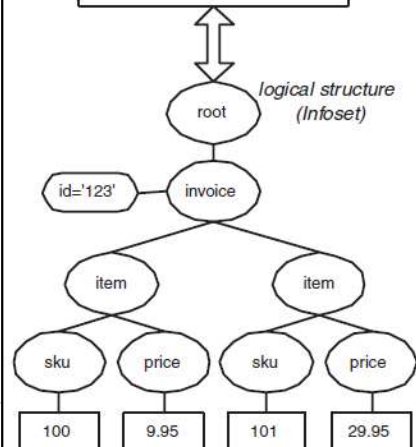
Važno!

- ▶ XPath model podataka ima nekoliko karakteristika koje nisu očigledne.
- 1. **Korenski čvor nije isto što i korenski element.** Korenski čvor stabla sadrži ceo dokument, uključujući i korenski element, kao i sve komentare i instrukcije za obradu smeštene pre početne oznake korenskog elementa.
- 2. Deklaracija XMLa, deklaracija tipa dokumenta (DOCTYPE) i razni delovi DTDa **ne mogu biti adresirani** pomoću XPath-a, iako XPath registruje attribute kojima je DTD dao podrazumevane vrednosti. (element *homepage* ima vrednost *xlink:type* kome je DTD dodelio vrednost)
- 3. **Atributi *xmlns* i *xmlns:prefix* ne smatraju se atributskim čvorovima.** Pridružuju se svakom čvoru za koji važi neka deklaracija, a ne samo za koji je prostor imena deklarisan.

Primer:

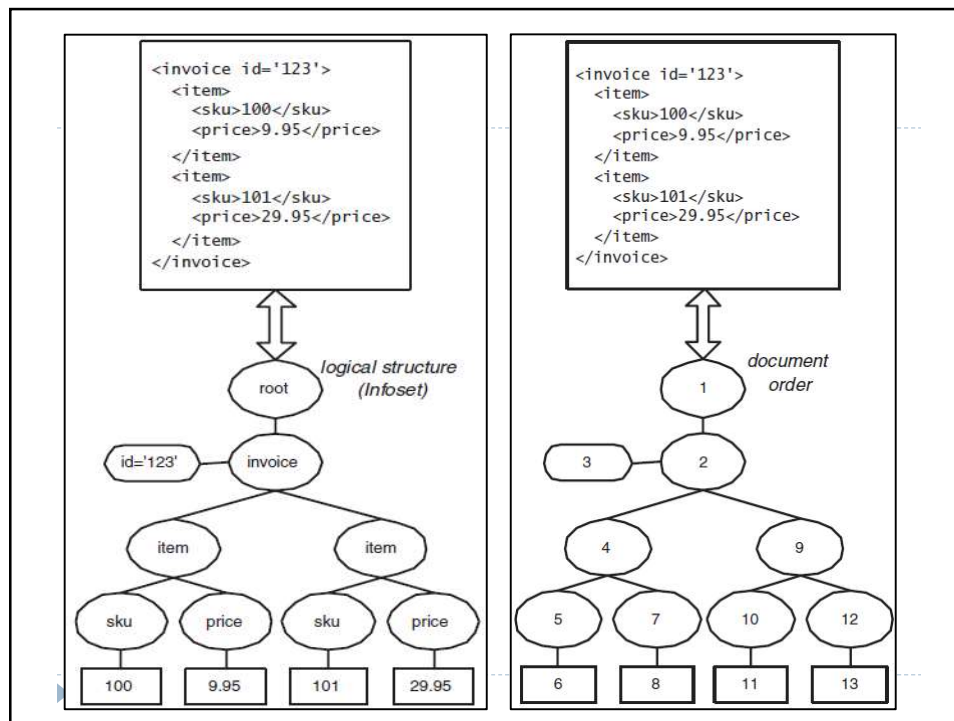
- ▶ XPath izrazi identifikuju skup čvorova u stablu.
- ▶ Na narednom primeru XPath izraz identifikuje dva elementa sa cenama:
- ▶ **/invoice/item/price**
- ▶ Ovaj tip izraza se naziva **lokacijska putanja**.
- ▶ Lokacijska putanja izgleda kao putanja u fajl sistemu, a pri tome se vrši navigacija kroz XPath stablo kako bi se identifikovao skup čvorova. Više detalja kasnije.

```
<invoice id='123'>
  <item>
    <sku>100</sku>
    <price>9.95</price>
  </item>
  <item>
    <sku>101</sku>
    <price>29.95</price>
  </item>
</invoice>
```



Redosled u dokumentu

- ▶ XPath ispituje document tj. stablo dokumenta kroz određeni redoslet tzv. **dokument redosled** (engl. *document order*).
- ▶ Dokument redosled je redosled u kom su čvorovi dokumenta serijalizovani. (Nekada je potrebno ispitati čvorove u suprotnom redosledu.)
- ▶ Pojam serijalizacije?



Programiranje i XPath

- Primer primene XPath u programiranju je u jeziku JavaScript zajedno sa XML DOM modelom:

```
var nl =
  doc.selectNodes("/invoice/item/price");
for (i=0; i<nl.length; i++) {
  ... // process price element here
}
```

- Kod se može napisati koristeći različite jezike i koristeći različite XML procesore.

Podsećanje: relacije čvorova

- ▶ **Roditelj/dete – Parent/ Children**
- ▶ *book* je roditelj za title, author, year, price
- ▶ **<book>**
 - <title>**Harry Potter**</title>**
 - <author>**J K. Rowling**</author>**
 - <year>**2005**</year>**
 - <price>**29.99**</price>**
 - </book>**
- ▶ **Blizanci – Siblings**
- ▶ Čvorovi koji imaju istog roditelja.
- ▶ **Preci - Ancestors**
- ▶ Čvorovi koji su u roditeljskom delu stabla. Roditelj, roditelj roditelja itd.
- ▶ **Potomci - Descendants**
- ▶ Čvorovi koji su deca, deca dece itd. u odnosu na tekući čvor.



Putanja lokacije (engl. *location path*)

- ▶ Identifikuje skup čvorova u dokumentu.
- ▶ Taj skup čvorova može biti prazan ili sa više elemenata.
- ▶ Skup može sadržati bilo koju vrstu (od 7) čvorova ili njihove kombinacije.
- ▶ Putanja lokacije se sastoji od lokacijskih koraka (engl. *location steps*)
- ▶ Svaki korak lokacije se izračunava u odnosu na određeni čvor u dokumentu koji nazivamo **konteksni čvor** (engl. *context node*)



Putanja lokacije **korena**

- ▶ Nju označava kosa crta /
- ▶ `<xsl:template match="/">`
 - ▶ `<html>`
 - ▶ `<xsl:apply-templates/>`
 - ▶ `</html>`
- ▶ `</xsl:template>`
- ▶ Kosa crta je **apsolutna putanja** lokacije - uvek isto znači.

▶

Koraci lokacije elementa potomaka

- ▶ Druga najjednostavnija putanja lokacije je ime jednog elementa. Ta putanja bira sve elemente čije ste ime zadali,
- ▶ ali koji su potomci konteksnog čvora.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Tr
>

<xsl:template match="people">
  <xsl:apply-templates select="person"/>
</xsl:template>

<xsl:template match="person">
  <xsl:value-of select="name"/>
</xsl:template>

</xsl:stylesheet>
```

Obratite pažnju na šablonsko pravilo za element **person**. XSLT to pravilo aktivira 2 puta! Po jednom za svaki čvor *person* u dokumentu. Prvi put je prvi element *person*, a drugi put za drugi element.

▶

Koraci lokacije atributa

- ▶ Da bi adresirali neki atribut kontekstnog elementa koristi se karakter **@** i iza njega ime atributa.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>
  <xsl:template match="/">
    <html>
      <xsl:apply-templates select="people"/>
    </html>
  </xsl:template>

  <xsl:template match="person">
    <table>
      <xsl:apply-templates select="person"/>
    </table>
  </xsl:template>

  <xsl:template match="person">
    <tr>
      <td><xsl:value-of select="name"/></td>
      <td><xsl:value-of
        select="@born"/></td>
      <td><xsl:value-of
        select="@died"/></td>
      </tr>
    </xsl:template>
  </xsl:template>

  <xsl:template match="people">
    <table>
      <xsl:apply-templates select="person"/>
    </table>
  </xsl:template>
```

Prvo pravilo ima *match="/"* koji odgovara korenskom čv. Drugo se primenjuje na sve elemente *people* (takvih ima samo jedan). Treće se primenjuje samo elemente *person*, kontekstnog čvora tj. *people*.. Konačno se izdvajaju vrednosti za svaki *person*.

Koraci lokacije komentara, teksta, inst. za obr

- ▶ Za ove čvorove se koriste:

1. `comment()`
2. `text()`
3. `processing-instruction()`

Odgovaraju svim delovima koji su deo tekućeg čvora. Svaki komentar čini zaseban čvor.

```
<xsl:template match="comment()">
  <i>Komentar izbačen</i>
</xsl:template>
```

Podržani tipovi podataka

▶ XPath podržava **tri tipa podataka**:

1. **boolean,** **true / false**
2. **number,** ***floating point* broj (IEEE 754)**
3. **string,** **sekvenca karaktera**
- ▶ **(node-set)** **kolekcija čvorova bez duplikata**

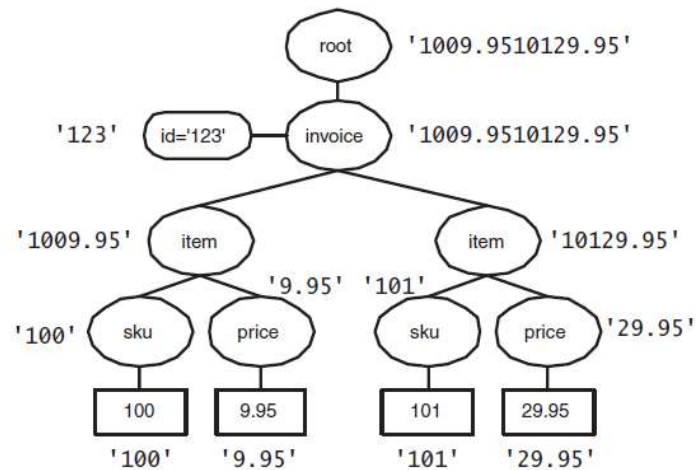
▶ Zoran Čirović

String tip kao vrednost čvorova

- ▶ Svaki čvor ima svoju tekstualnu tj. stringovsku reprezentaciju. Ona omogućava poređenje i rad sa objektima tj. elementima koji su različitog tipa.
- ▶ Stringovska (tekstualna) vrednost:
1. Korenski – povezani svi tekstovi čvorova dece
 2. Čvorovi elemenata - povezani svi tekstovi čvorova dece
 3. Čvorovi teksta – Karakteri podataka
 4. Čvorovi atributa – normalizovana vrednost atributa
 5. Čvorovi komentara – karakteri podataka unutar komentara
 6. Čvorovi instr. za obradu – karakteri podataka praćeni ciljnom instrukcijom
 7. Čvorovi prostora imena – prostor imena URI

▶ Zoran Čirović

Primer



Zoran Čirović

Džokerski znakovi

*	odgovara svim čvorovima elemenata bez obzira na njihova imena. Ne podudara se sa atributima, tekstom, komentarima i instrukcijama za obradu. Ispred može da stoji prefiks za prostor imena, na pr: svg:* i tada se odnosi na elemente tog prostora imena.
@*	odgovara svim čvorovima atributa sve attribute
node()	odgovara i korenskom i svim ostalim
text()	Sel. tekstualni sadržaj
comment()	Sel. komentare

Osnovni izrazi

nodename	Selektovanje svih čvorova sa zadatim imenom
/	Selektovanje od korenog čvora
//	Selektovanje čvorova u dokumentu od konteksnog čvora , uključujući i kontekсни, koji odgovaraju selekciji bez obzira gde su unutar. Kada je na početku izraza odnosi se na sve čvorove dokumenta.
.	Selektovanje konteksnog čvora
..	Selektovanje roditeljskog čvora
@	Selektovanje atributa

Apsolutna putanja počinje sa kosom crtom (*engl. forward slash*) / , dok relativna ne.

Putanja se može sastojati od više lokacijskih koraka, razdvojenih kosom crtom.

//@id/.. Označava sve elemente koji imaju atribut id.

//last_name/..name Označava sve elemente *name* koji su braća elementa last_name u dokumentu, a potomci su istog roditeljskog čvora.

Primeri pretrage / , džoker znaci

/bookstore/*	Selektuju se svi podelementi elementa bookstore
//*	Selektuju se svi potomci od tekućeg elementa u dokumentu
//title[@*]	Selektuju se svi title elementi koji imaju bilo koji atribut počev od tekućeg

Opšti XPath izrazi

- ▶ Putanje lokacija nisu jedina moguća vrsta izraza. Rezultat izraza mogu biti brojevi i znakovni nizovi. Na primer, mogući su izrazi:
 - 3.141529
 - 2+2
 - 'Rosalind Franklin'
 - true()
 - 32.5 < 76.2
 - position()=last()
- ▶ XPath izrazi koji nisu skupovi čvorova ne mogu da se koriste u atributu **match** elementa **xsl:template**.
- ▶ Mogu se koristiti kao vrednosti atributa **select** elementa **xsl:value-of elements**, kao i u predikatima putanja lokacija.



Vrste izraza

Lokacijska putanja	/ //
Bulov izraz	or and
Izrazi jednakosti	= !=
Relacioni izrazi	<= < >= >
Numerički izrazi	+ - div mod * - (unarni)



Zoran Čirović

Numerički izrazi

- ▶ XPath nema prave brojeve. Svi njegovi brojevi zauzimaju 8 bajtova memorije i predstavljaju brojeve dvostruke preciznosti po standardu IEEE 754 (**double**).
- ▶ Mogu se koristiti 5 osnovnih binarnih operacija i jedna dodatna unarna za promenu znaka.
- ▶ Kosa crta, /, nije mogla da bude korišćena za deljenje!
- ▶ `<xsl:template match="person">`
- ▶ `<century>`
- ▶ `<xsl:value-of select="(@born - (@born mod 100)) div 100"/>th`
- ▶ `</century>`
- ▶ `</xsl:template>`



Znakovni izrazi

- ▶ To su uređene sekvence Unicode znakova. Tekstualne vrednosti zadaju se u navodnicima ili polunavodnicima. Ukoliko znakovni niz sadrži navodnike, mora biti oivičen polunavodnicima i obrnuto.
- ▶ Za poređenje ovih sekvenci koriste se znaci jednakosti = ili nejednakosti !=. Mogu se koristiti i znaci za manje i veće da bi se vršilo poređenje tj sortiranje, ali to nema smisla ako nije u kontekstu određenog jezika (koristi se Unicode).



Logički izrazi

- ▶ Ima dva stanja: **true** i **false**.
- ▶ Koristi se pri poređenjima.
- ▶ Najčešće se koriste u predikatima unutar putanja lokacija.
- ▶ Postoje svi operatori poređenja:
- ▶ **= != <= < >= >**
- ▶ Uz ove operatore koriste se operatori **and** i **or** za kombinovanje više celina.
- ▶ **not()** – ova funkcija menja logičku vrednost u suprotnu.

```
<xsl:template match="profession">
  <xsl:if test=".='computer scientist' or.='physicist'">
    <xsl:value-of select="."/>
  </xsl:if>
</xsl:template>

<xsl:template match="profession">
  <xsl:choose>
    <xsl:when test=".='computer scientist'">
      <i><xsl:value-of select="."/></i>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="."/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```



Funkcije nad skupom čvorova

- ▶ **position()** – pozicija čvora
- ▶ **last()** – vraća broj čvorova konteksnog čvora, što je istovremeno i pozicija poslednjeg čvora. **Pozicije kreću od 1, ne od 0.**
- ▶ **count(.)** – broj čvorova u skupu koji je primljen kao argument (ne za kontekсни čvor). Na primer **count(//name)** vraća broj elemenata *name* u dokumentu.
- ▶ **id(.)** – prima kao argument znakovni niz. Kao rezultat daje skup svih čvorova u dokumentu čiji su identifikatori jednaki ulaznim. To su atributi tipa id deklarirani u DTDu.



Funkcije znakovnih nizova

- ▶ **string(.)** – pretvara argument proizvoljnog tipa u znakovni niz
- ▶ **starts-with(. , .)** – prima 2 argumenta, a rezultat je logička vrednost *true* ako prvi počinje drugim. (Ne postoji ends-width)
- ▶ **contains(. , .)** – prima 2 argumenta. Ako prvi sadrži drugi, vraća se *true*
- ▶ **substring(. , . , .)** – prima 3 arg. Prvi je string koji se ispituje. Drugi je pozicija od koje se izdvaja rezultat. Treći je broj karaktera koji se izdvajaju.
- ▶ **string-length(.)** – vraća broj jednak dužini znakovnog niza koji prima

▶

Funkcije znakovnih nizova - primeri

- | | |
|--|-------|
| ▶ starts-with('Richard', 'Ric') | true |
| ▶ starts-with('Richard', 'Rick') | false |
| ▶ contains('Richard', 'ar') | true |
| ▶ contains('Richard', 'art') | false |
| ▶ substring('MM/DD/YYYY', 7) | YYYY |
| ▶ substring('MM/DD/YYYY', 4, 2) | DD |
| ▶ string-length(//name[position()=1]) | 29 |

▶

Numeričke funkcije

- ▶ **number()** pretvara argument u broj. Ako argument nedostaje, konvertuje kontekсни čvor. Čvor se konvertuje prvo u string, a zatim u broj. Na primer:
 - ▶ *true/false* 1
 - ▶ "7.5" 7.5.
 - ▶ "Fred" NaN
- ▶ **round()** – vrši zaokruživanje (naviše)
- ▶ **floor()** – vraća najveći ceo broj manji od argumenta
- ▶ **ceiling()** – vraća najmanji ceo broj veći od argumenta



Primer

```
<?xml version="1.0"
  encoding="ISO-8859-1"?>
```

```
<bookstore>
```

```
<book>
  <title lang="eng">Harry
  Potter</title>
  <price>29.99</price>
</book>
```

```
<book>
  <title lang="eng">Learning
  XML</title>
  <price>39.95</price>
</book>
```

```
</bookstore>
```

XPath	značenje
bookstore	Selektuju se elementi <i>bookstore</i> sa svim podelementima
/bookstore	Selektuje se koreni elemenat bookstore sa svim podelementima <i>bookstore</i> . Ako putanja startuje sa (/) to znanči da je reč o apsolutnoj putanje!
bookstore/book	Selektuju se svi <i>book</i> elementi koji su podelementi od svih <i>bookstore</i>
//book	Selektuju se svi book elemeni bez obzira gde su u dokumentu
bookstore//book	Selektuju se svi <i>book</i> elementi koji su potomci <i>bookstore</i> elementa, bez obzira gde su u <i>bookstore</i> elementu
//@lang	Selektuju se svi atributi koji imaju ime <i>lang</i>



Predikati

- ▶ Obično izrazi upućuju na više čvorova.
- ▶ **Predikati se koriste za pronalaženje specifičnog čvora koji sadrži specifičnu vrednost.**
- ▶ Predikat **sadrži logički izraz koji se izračunava za svaki čvor u spisku konteksnog čvora.** Ukoliko je rezultat izraza netačan, taj čvor se izbacuje iz rezultujuće liste, inače ostaje u njoj. Predikati su uvek uokvireni četvrtastim zagradama.
- ▶ Mogu se koristiti i polunavodnici.
 - ❑ `//profession[. = "physicist"]`
 - ❑ `<xsl:apply-templates select="//profession[.= 'physicist']" />`



Predikati: primeri:

<code>/bookstore/book[1]</code>	Selektuje se prvi element book koji je podelement od bookstore elementa.
<code>/bookstore/book[last()]</code>	Selektuje se poslednji element book koji je podelement od bookstore elementa.
<code>/bookstore/book[last()-1]</code>	Selektuje se predposlednji element book koji je podelement od bookstore elementa.
<code>/bookstore/book[position()<3]</code>	Selektuje se element book koji je podelement od bookstore elementa i na poziciji manjoj od 3.
<code>//title[@lang]</code>	Selektuju se svi title elementi koji imaju atribut lang
<code>//title[@lang='eng']</code>	Selektuju se svi title elementi koji imaju atribut lang sa vrednošću 'eng'
<code>/bookstore/book[price>35.00]</code>	Selektuje se elementi book koji su podelementi od bookstore elementa a čija je cena veća od 35.00
<code>/bookstore/book[price>35.00]/title</code>	Selektuje se elementi title koji su podelementi od book elementa a čija je cena veća od 35.00 a koji je podelement od bookstore



Selektovanje nekoliko putanja

//book/title //book/price	Selektuju se svi title i price elementi u svim book elementima
//title //price	Selektuju se svi title i price elementi u dokumentu
/bookstore/book/title //price	Selektuju se svi title elementi elementa book koji je podelement od bookstore elementa i svi price elementi u dokumentu



PREGLED....

- ▶ Lokacijska putanja daje kao rezultat skup čvorova. Putanja može biti
 - ▶ apsolutna ili
 - ▶ relativna

- ▶ XPath definiše dva pojma kako bi objasnio proces rada:
 - ▶ *Konteksni skup* – engl. *context node-set*
 - ▶ *Konteksni čvor* – engl. *context node*
- ▶ *Konteksni skup* se definiše kao tekući skup čvorova koji se nalaze iznad date tačke u izrazu. Tekući čvor je onaj koji se obrađuje.



Zoran Čirović

- ▶ Lokacijski koraci se obrađuju u redosledu, **s leva na desno, i to samo jednom.**
- ▶ Lokacijske putanje se mogu **spajati** “unioned” pomoću operatora |
- ▶ Kombinacija dva skupa čvorova **isključuje duplikate.**
- ▶ **/ invoice / item** – identifikuje dete elemente od korenskog *invoice* elementa.
- ▶ **item / sku** – identifikuje dete sku elemente od deteta kontekstnog elementa, item.
- ▶ **sku | price | desc** – identifikuje kontekstne deca elemente sku, price i desc



Pregled: XPath operatori

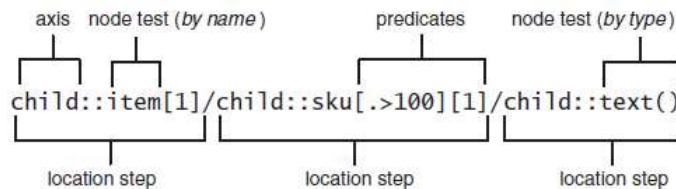
operator	značenje	primer	rezultat
	Računa se dva skupa čvorova	//book //cd	Returns a node-set with all book and cd elements
+	Sabiranje	6 + 4	10
-	Oduzimanje	6 - 4	2
*	Množenje	6 * 4	24
div	Deljenje	8 div 4	2
=	Jednakost	price=9.80	true ako je price is 9.80 false ako je price is 9.90
!=	Nejednakost	price!=9.80	true ako je price is 9.90 false ako je price is 9.80
<	Manje od	price<9.80	true ako je price is 9.00 false ako je price is 9.80
<=	Manje ili jednako	price<=9.80	true ako je price is 9.00 false ako je price is 9.90
>	Veće	price>9.80	true ako je price is 9.90 false ako je price is 9.80
>=	Veće ili jednako	price>=9.80	true ako je price is 9.90 false ako je price is 9.70
or	ili	price=9.80 or price=9.70	true ako je price is 9.80 false ako je price is 9.50
and	i	price>9.00 and price<9.90	true ako je price is 9.80 false ako je price is 8.50
mod	Ostatak pri deljenju	5 mod 2	1



Šema za lokacijsku putanju

- ▶ Lokacijska putanja sadrži nekoliko delova i ključne reči:

- ▶ *Axis*
- ▶ *Node test*
- ▶ *Predicates*



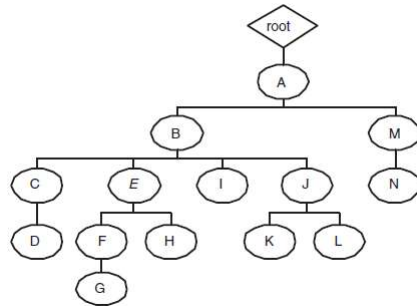
Axis

ancestor	Selektuju se svi preci (roditelj, roditelj roditelja, itd.) u odnosu na tekući čvor
ancestor-or-self	Selektuju se svi preci (roditelj, roditelj roditelja, itd.) u odnosu na tekući čvor i sam tekući čvor
attribute	Selektuju se svi atributi tekućeg čvora
child	Selektuju se sva deca u odnosu na tekući čvor
descendant	Selektuju se svi potomci (deca, deca dece, itd.) u odnosu na tekući čvor
descendant-or-self	Selektuju se svi potomci (deca, deca dece, itd.) u odnosu na tekući čvor i sam tekući čvor
following	Selektuje se sve u dokumentu posle završnog taga tekućeg čvora
following-sibling	Selektuju se svi blizanci (elementi u istom nivou) nakon tekućeg čvora
namespace	Selektuju se svi čvorovi prostora imena tekućeg čvora
parent	Selektuje se roditelj tekućeg čvora
preceding	Selektuje se sve u dokumentu što je pre početnog taga tekućeg čvora
preceding-sibling	Selektuju se svi blizanci pre tekućeg čvora
self	Selektuje se tekući čvor



Primer

- Tekući čvor je "E"



Axis	Nodes (relative to E)
self	E
parent	B
child	F,H
descendant	F,G,H
descendant-or-self	E,F,G,H
ancestor	B,A,root
ancestor-or-self	E,B,A,root
preceding	D,C
preceding-sibling	C
following	I,J,K,L,M,N
following-sibling	I,J



Primer 2

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<bookstore>

  <book>
    <title lang="eng">Harry
    Potter</title>
    <price>29.99</price>
  </book>

  <book>
    <title lang="eng">Learning
    XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

child::book	Selektuju se svi book čvorovi koji su deca tekućeg čvora
attribute::lang	Selektuje se lang atribut tekućeg čvora
child::*	Selektuju se sva deca tekućeg čvora
attribute::*	Selektuju se svi atributi tekućeg čvora
child::text()	Selektuju se svi tekstualni deca čvorovi od tekućeg čvora
child::node()	Selektuju se svi deca čvorovi tekućeg čvora
descendant::book	Selektuju se svi book potomci tekućeg čvora
ancestor::book	Selektuju se svi book preci tekućeg čvora
ancestor-or-self::book	Selektuju se svi book preci tekućeg čvora kao i tekući ako je book
child::* / child::price	Selektuju se sve cene unuka tekućeg čvora



Primer 3

```
<?xml version="1.0"?>
<Employees>
  <Person>
    <FirstName>A</FirstName>
    <LastName>B</LastName>
    <DateOfBirth>2008-12-12
  </DateOfBirth>
  </Person>
  <Person>
    <FirstName>C</FirstName>
    <LastName>D</LastName>
    <DateOfBirth>2008-11-11
  </DateOfBirth>
  </Person>
  <Person>
    <FirstName>E</FirstName>
    <LastName>F</LastName>
    <DateOfBirth>2008-10-10
  </DateOfBirth>
  </Person>
</Employees>
```

sibling axis demo.

LastName koji sadrži tekst "B".
DateOfBirth koji sadrži tekst "2008-12-12".

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>

  <xsl:template match="/">
    <html>
      <head>
        <title>following-sibling axis.</title>
      </head>
      <body>
        <h3>sibling axis demo.</h3>
        <xsl:apply-templates
select="/Employees/Person[1]/FirstName" />
        </body>
      </html>
    </xsl:template>

    <xsl:template match="FirstName">
      <xsl:for-each select="following-sibling::*">
        <paragraph>
          <xsl:value-of select="name(.)" />
          koji sadrži tekst
          "<xsl:value-of select="." />".
        </paragraph>
      </xsl:for-each>
    </xsl:template>
  </xsl:stylesheet>
```

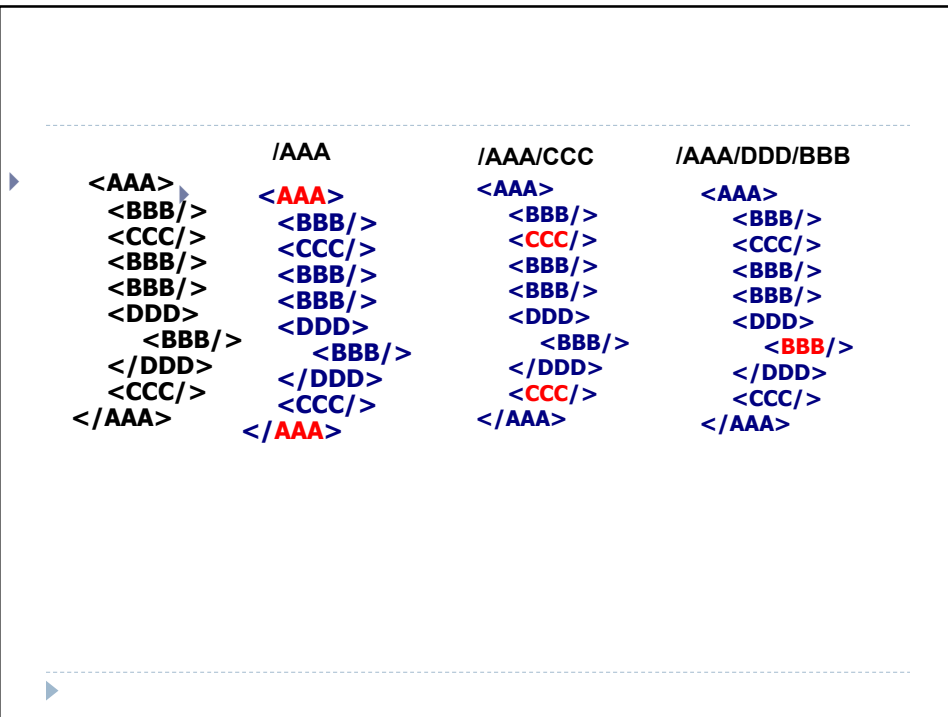


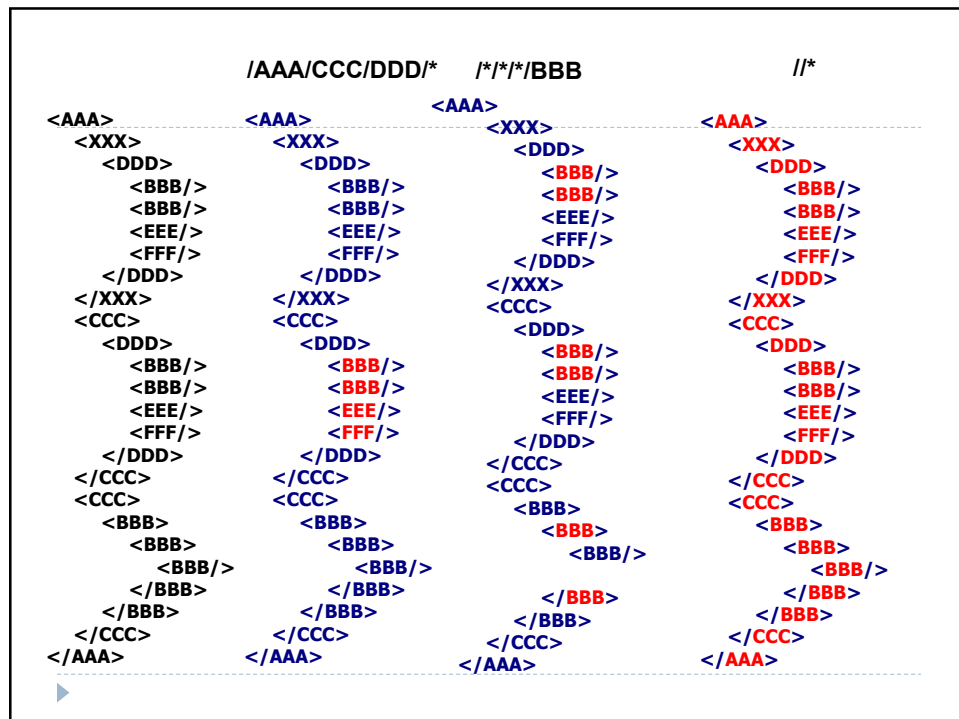
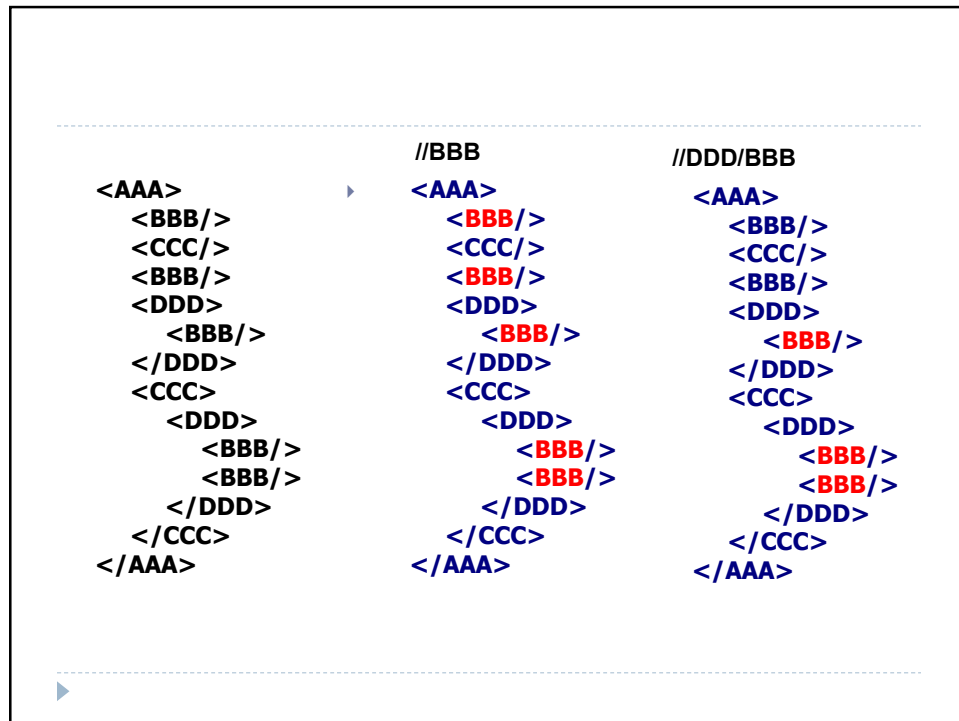
Primer

- ▶ Pogledati primer
- ▶ http://www.w3schools.com/xpath/xpath_examples.asp



Primeri





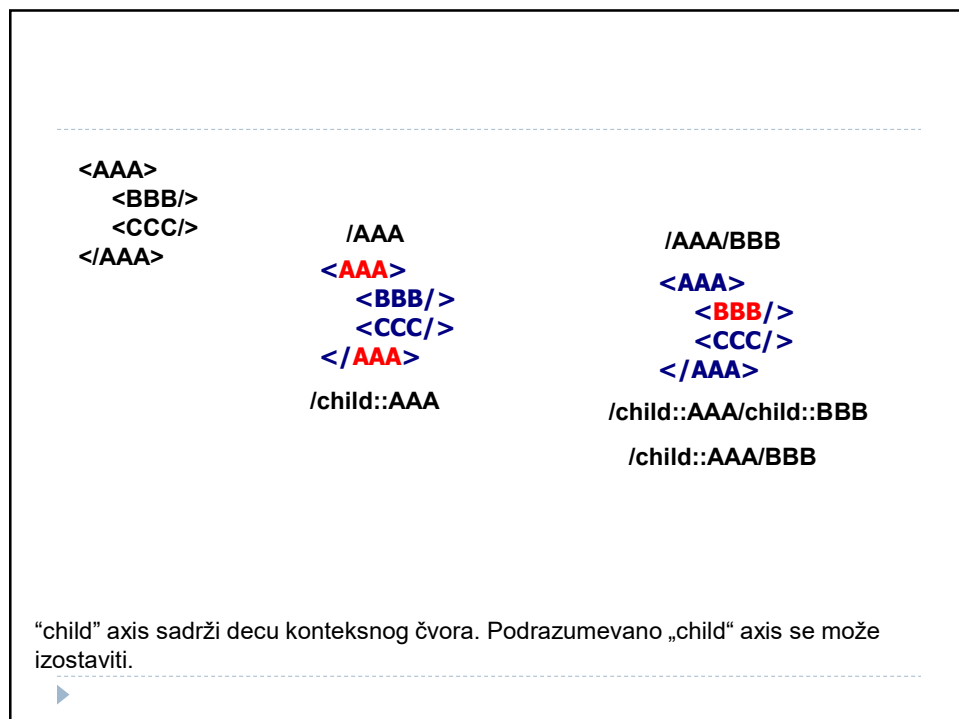
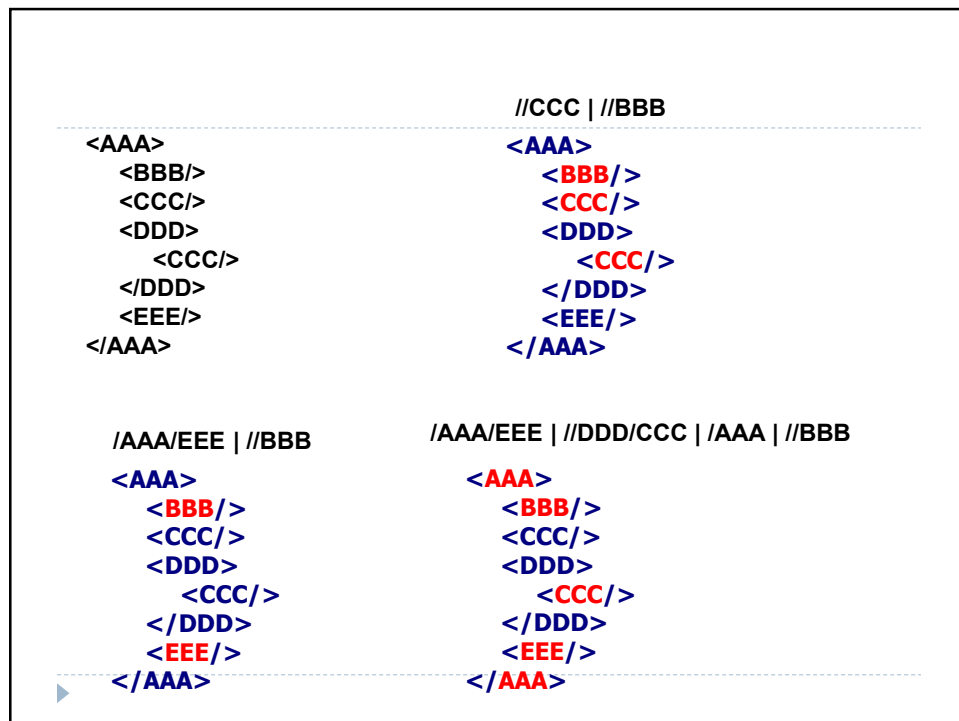
	/AAA/BBB[1]	/AAA/BBB[last()]
<AAA>	<AAA>	<AAA>
<BBB/>	<BBB/>	<BBB/>
<BBB/>	<BBB/>	<BBB/>
<BBB/>	<BBB/>	<BBB/>
<BBB/>	<BBB/>	<BBB/>
</AAA>	</AAA>	</AAA>

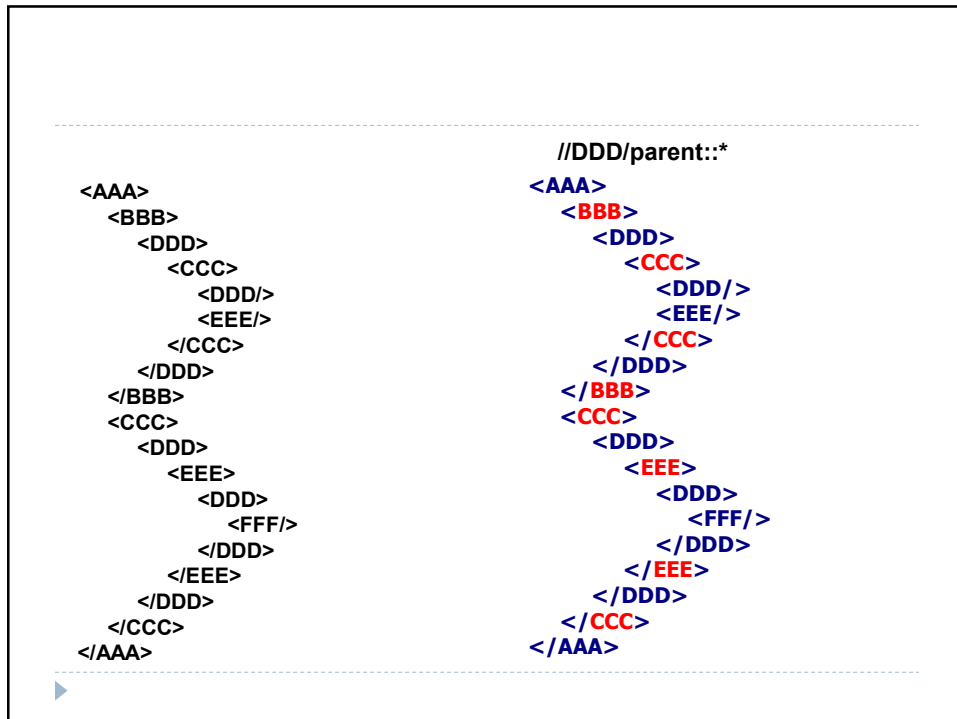
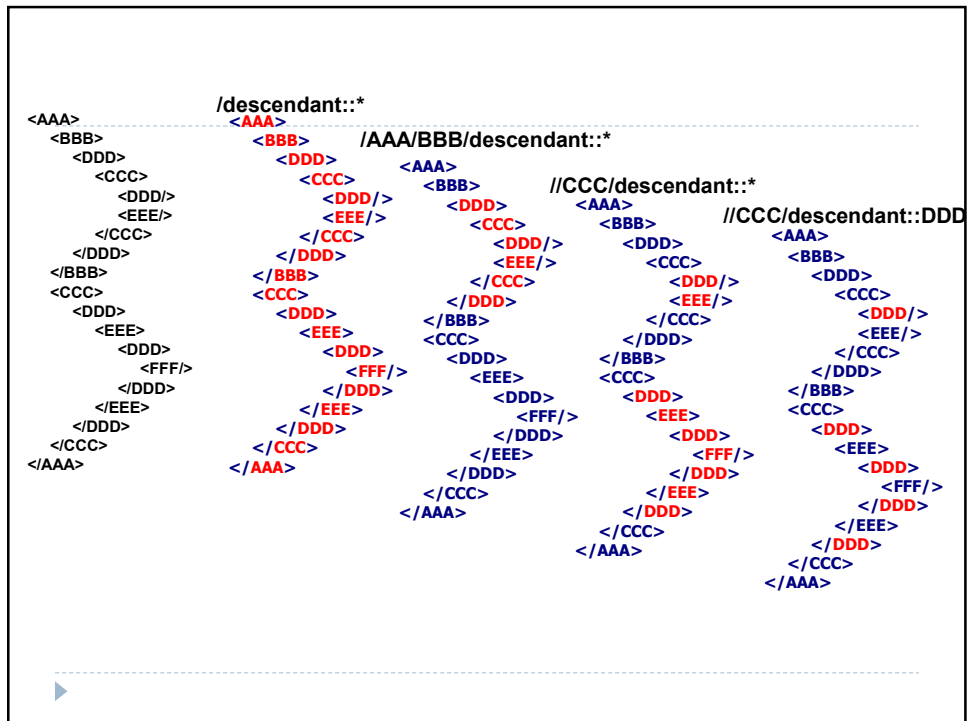
	//@id	//BBB[@id]
<AAA>	<AAA>	<AAA>
<BBB id = "b1"/>	<BBB id = "b1"/>	<BBB id = "b1"/>
<BBB id = "b2"/>	<BBB id = "b2"/>	<BBB id = "b2"/>
<BBB name = "bbb"/>	<BBB name = "bbb"/>	<BBB name = "bbb"/>
<BBB/>	<BBB/>	<BBB/>
</AAA>	</AAA>	</AAA>

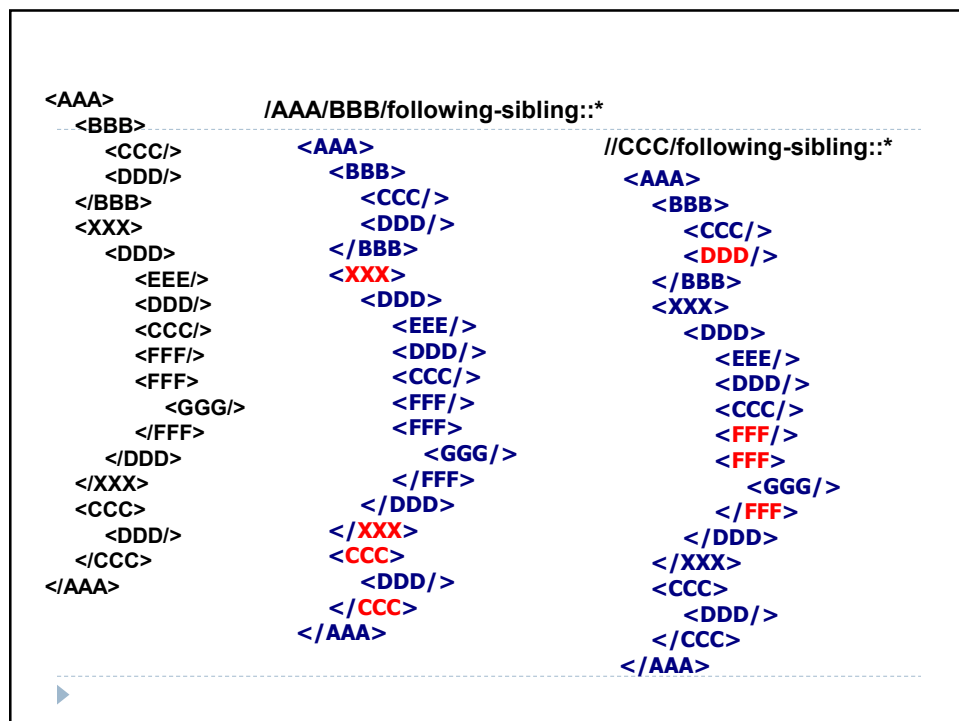
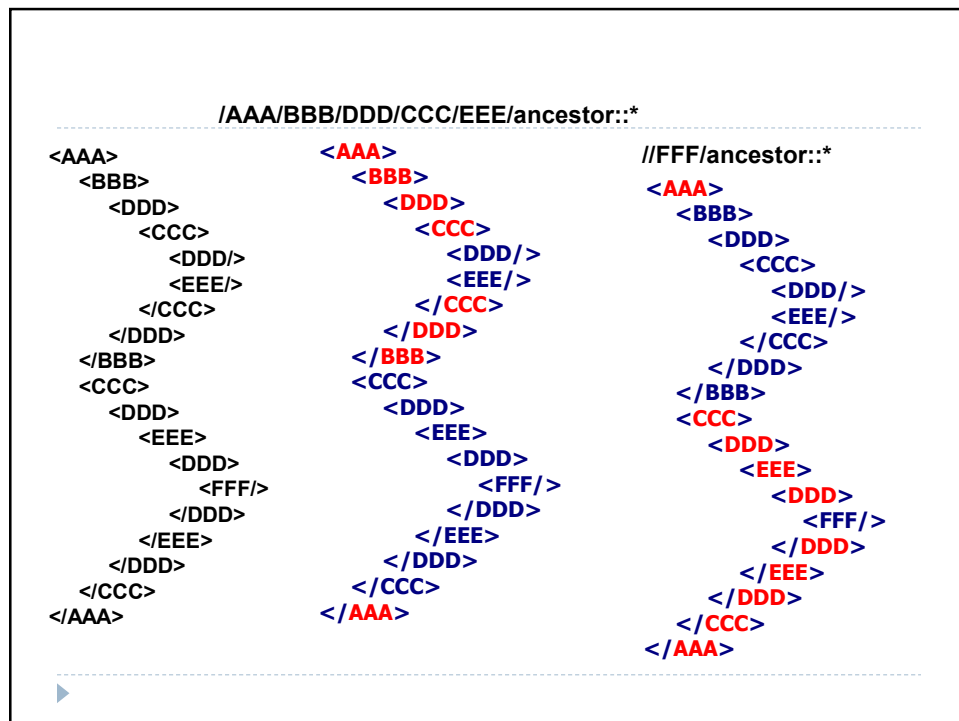
//BBB[@name]	//BBB[@*]	//BBB[not(@*)]
<AAA>	<AAA>	<AAA>
<BBB id = "b1"/>	<BBB id = "b1"/>	<BBB id = "b1"/>
<BBB id = "b2"/>	<BBB id = "b2"/>	<BBB id = "b2"/>
<BBB name = "bbb"/>	<BBB name = "bbb"/>	<BBB name = "bbb"/>
<BBB/>	<BBB/>	<BBB/>
</AAA>	</AAA>	</AAA>

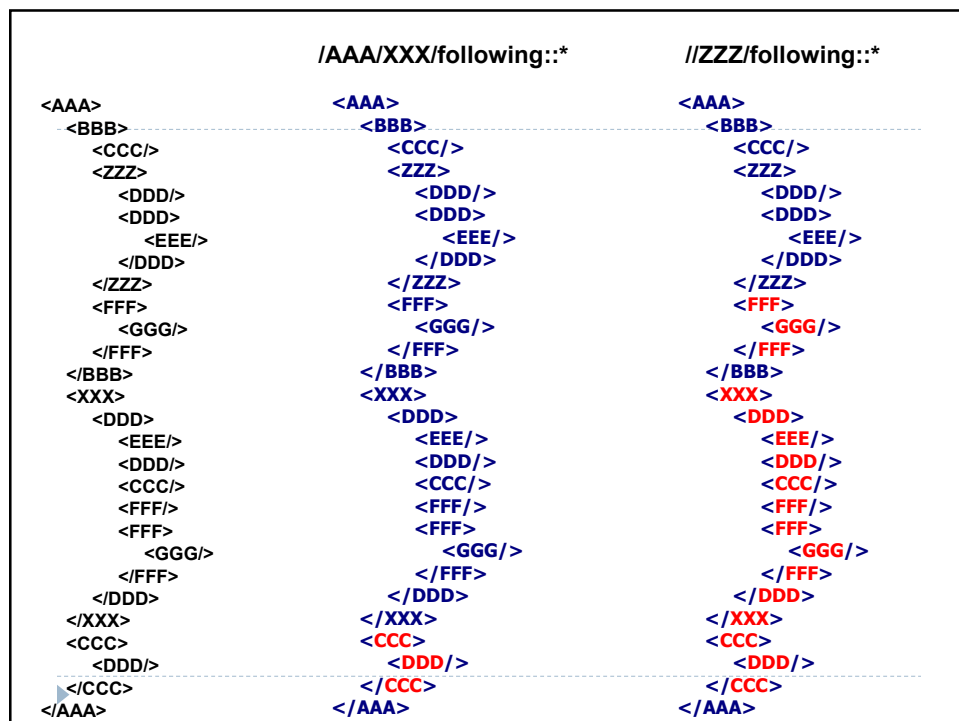
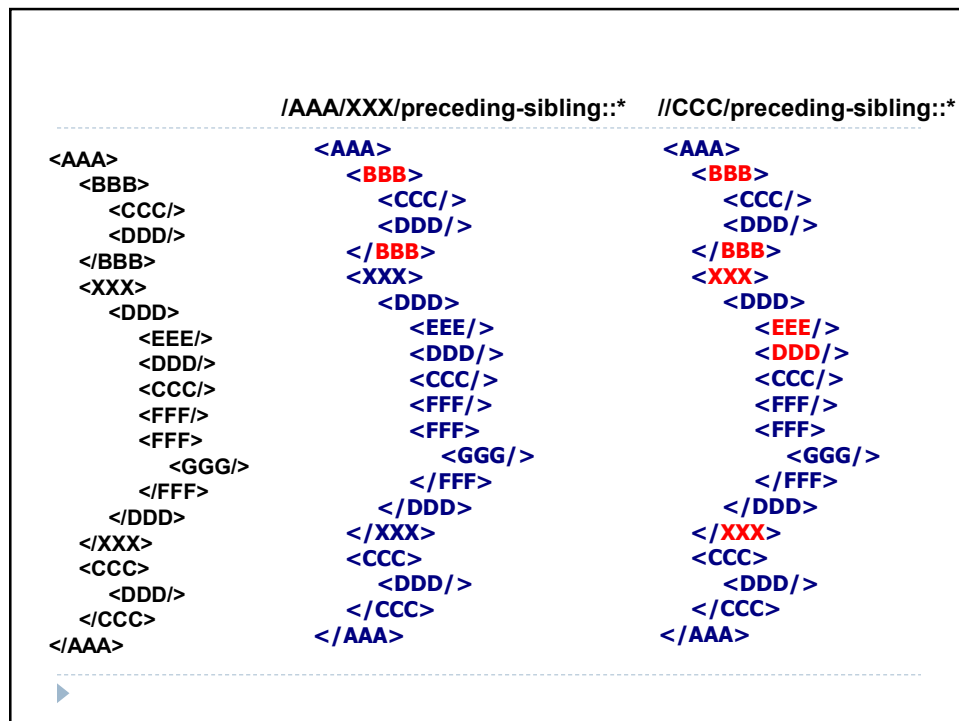
<pre> <AAA> <BBB id = "b1"/> <BBB name = " bbb "/> <BBB name = "bbb"/> </AAA> </pre>		<pre> //BBB[@id='b1'] <AAA> <BBB id = "b1"/> <BBB name = " bbb "/> <BBB name = "bbb"/> </AAA> </pre>	
<pre> //BBB[@name='bbb'] <AAA> <BBB id = "b1"/> <BBB name = " bbb "/> <BBB name = "bbb"/> </AAA> </pre>		<pre> //BBB[normalize-space(@name)='bbb'] <AAA> <BBB id = "b1"/> <BBB name = " bbb "/> <BBB name = "bbb"/> </AAA> </pre>	

	//*[count(BBB)=2]	//*[count(*)=2]	//*[count(*)=3]
<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>









	/AAA/XXX/preceding::*	//GGG/preceding::*
<AAA>	<AAA>	<AAA>
<BBB>	<BBB>	<BBB>
<CCC/>	<CCC/>	<CCC/>
<ZZZ>	<ZZZ>	<ZZZ>
<DDD/>	<DDD/>	<DDD/>
</ZZZ>	</ZZZ>	</ZZZ>
</BBB>	</BBB>	</BBB>
<XXX>	<XXX>	<XXX>
<DDD>	<DDD>	<DDD>
<EEE/>	<EEE/>	<EEE/>
<DDD/>	<DDD/>	<DDD/>
<CCC/>	<CCC/>	<CCC/>
<FFF/>	<FFF/>	<FFF/>
<FFF>	<FFF>	<FFF>
<GGG/>	<GGG/>	<GGG/>
</FFF>	</FFF>	</FFF>
</DDD>	</DDD>	</DDD>
</XXX>	</XXX>	</XXX>
<CCC>	<CCC>	<CCC>
<DDD/>	<DDD/>	<DDD/>
</CCC>	</CCC>	</CCC>
</AAA>	</AAA>	</AAA>

„preceding“ axis sadrži sve čvorove koji su pre konteksnog čvora, izuzimajući sve ... pretke i izuzimajući attribute i prostore imena.

	/AAA/XXX/descendant-or-self::*	//CCC/descendant-or-self::*
<AAA>	<AAA>	<AAA>
<BBB>	<BBB>	<BBB>
<CCC/>	<CCC/>	<CCC/>
<ZZZ>	<ZZZ>	<ZZZ>
<DDD/>	<DDD/>	<DDD/>
</ZZZ>	</ZZZ>	</ZZZ>
</BBB>	</BBB>	</BBB>
<XXX>	<XXX>	<XXX>
<DDD>	<DDD>	<DDD>
<EEE/>	<EEE/>	<EEE/>
<DDD/>	<DDD/>	<DDD/>
<CCC/>	<CCC/>	<CCC/>
<FFF/>	<FFF/>	<FFF/>
<FFF>	<FFF>	<FFF>
<GGG/>	<GGG/>	<GGG/>
</FFF>	</FFF>	</FFF>
</DDD>	</DDD>	</DDD>
</XXX>	</XXX>	</XXX>
<CCC>	<CCC>	<CCC>
<DDD/>	<DDD/>	<DDD/>
</CCC>	</CCC>	</CCC>
</AAA>	</AAA>	</AAA>

