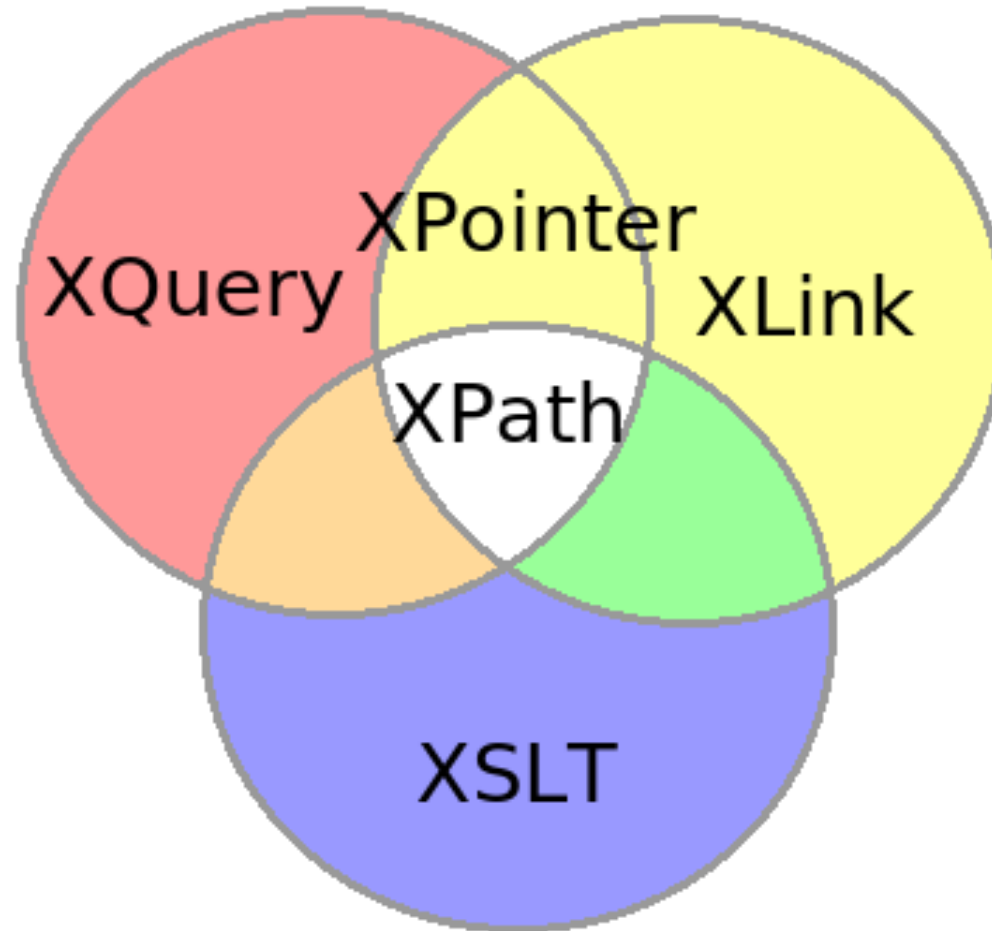


XML baze podataka

- Ponajviše zbog razvoja e-Poslovanja i ERP došlo je do razvoja XML jezika za razmjenu podataka, pa tako i XML temeljenih baza podataka

W3C XSLT standard



W3C XSLT standard

- **XSLT** – jezik koji se bavi transformacijom XML dokumenata
- **XQuery** – upitni jezik za postavljanje upita nad XML podacima
- **XLink** – jezik za stvaranje hiperveza u XML dokumentima
- **XPointer** – specifikacija za adresiranje komponenata
- **XPath** – jezik za selekciju dijelova XML dokumenata

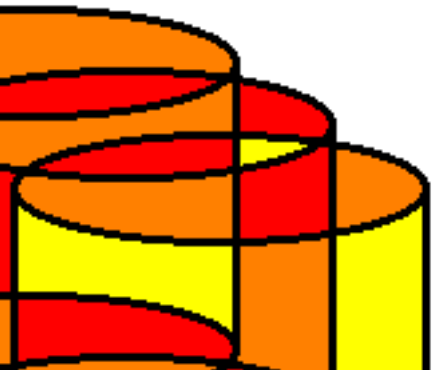
XPath

- Koristi se u svim prethodno navedenim dijelovima standarda te je osnova za XQuery
- Koristi se izrazima (putanjama) za navigaciju kroz XML dokument te sadrži biblioteku integriranih funkcija (preko 100)

Putanje



- Putanje omogućavaju označavanje čvorova ili skupova čvorova u XML dokumentu
- Vrlo su slične putanjama pri definiciji direktorija u nekom datotečnom sustavu



Priprema

- Za prikaz XML baza podataka koristit ćemo **eXist-DB** sustav koji se temelji na dokumentima (u ovom slučaju XML dokumentima) i kolekcijama dokumenata nad kojima je moguće izvršavati operacije kao nad bazom podataka
- Ako radite na računalu u dvorani sustav je instaliran te se pokreće automatski pri pokretanju računala (u slučaju da se ne pokrene, postoji ikona na brzoj traci za pokretanje)
- Ako radite na vlastitom računalu potrebno je instalirati eXist-DB (ako ste prethodno koristili skriptu s adrese <https://tinyurl.com/instalacija2019> eXist-DB je instaliran pod **\$HOME/software/exist-db** i nije potrebna dodatna instalacija). Sustav se pokreće skriptom **./startup.sh** koja se nalazi u **bin** direktoriju instalacije
- Upute ćemo isprobavati u razvojnoj okolini **eXide** koja je dio eXist-DB sustava i dostupna je na adresi **http://localhost:8080**

Upute

- Otvorite adresu **<http://localhost:8080/>** i kliknite na login (username je **admin**, a lozinka je prazna – *I know! I know! Možete to promijeniti! Google is your friend!*)
- Nakon logina kliknite na eXide IDE
- Kreirat ćemo novi XML dokument. Kliknite na New i pod type odaberite **XML**
- Za prikaz jezika XPath koristit ćemo datoteku dostupnu na adresi: **<https://tinyurl.com/knjige-xml>**
- Sadržaj datoteke kopirajte u eXide editor i kliknite na Save
- Dokument ćemo pohraniti pod **apps** kolekciju (direktorij) unutar koje ćemo kreirati kolekciju **knjige** (klikom na Create Collection i unosom naziva)
- Dokument ćemo nazvati **knjige.xml**

Upute

Putanja do dokumenta sada je sljedeća:

```
doc( '/db/apps/knjige/knjige.xml' )
```

Kliknite na New Xquery te ispod definicije verzije upišite tu putanju. Klikom na gumb Eval u donjem bi se dijelu prozora pod Adaptive output trebao pokazati XML dokument.

Primjer XML dokumenta

```
<?xml version="1.0"?>
<knjizara>
  <knjiga>
    <naslov jezik="hrvatski">
      Teorija i primjena baza podataka</naslov>
    <autori>
      <autor>Mirko Malekovic</autor>
      <autor>Markus Schatten</autor>
    </autori>
    <godina>2008</godina>
    <cijena>99.99</cijena>
    <kljucne-rijeci> ...
  </knjiga> ...
```

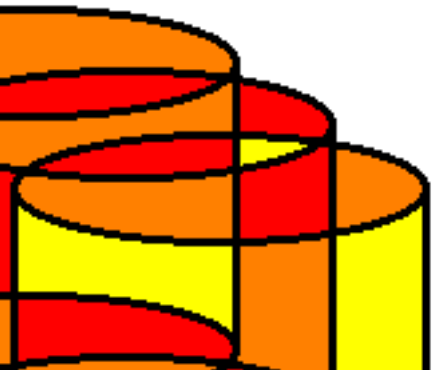
Čvorovi



<knjizara> - čvor dokumenta (dokument)

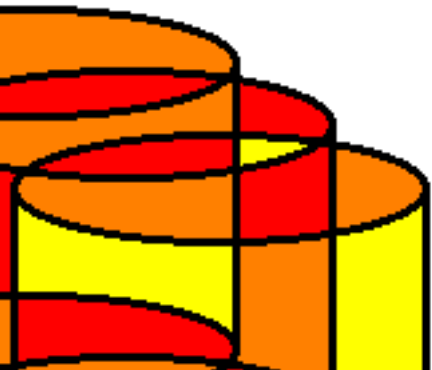
<cijena>49,99</cijena> - elementarni čvor
(element)

jezik="hrvatski" – čvor atributa (atribut)



Atomarne vrijednosti

Programiranje u Prologu engleski



Stavke

- Stavke (item) su ili čvorovi ili atomarne vrijednosti

Veze između čvorova

```
<knjiga>  
  <naslov jezik="hrvatski">  
    Teorija baza podataka</naslov>  
  <autori>  
    <autor>Mirko Malekovic</autor>  
  </autori>  
  <godina>2008</godina>  
  <cijena>99.99</cijena>  
</knjiga>
```

Veze između čvorova

<knjiga>

<naslov jezik="hrvatski">

Teorija baza podataka</naslov>

<autori>

<autor>Mirko Malekovic</autor>

</autori>

<godina>2008</godina>

<cijena>99.99</cijena>

</knjiga>

Veze između čvorova

<knjiga>

<naslov jezik="hrvatski">

Teorija baza podataka</naslov>

<autori>

<autor>Mirko Malekovic</autor>

</autori>

<godina>2008</godina>

<cijena>99.99</cijena>

</knjiga>

DJECA (CHILDREN)

Veze između čvorova

<knjiga>

RODITELJ (PARENT)

<naslov jezik="hrvatski">

Teorija baza podataka</naslov>

<autori>

<autor>Mirko Malekovic</autor>

</autori>

<godina>2008</godina>

<cijena>99.99</cijena>

</knjiga>

Veze između čvorova

<knjiga>

<naslov jezik="hrvatski">

Teorija baza podataka</naslov>

<autori>

<autor>Mirko Malekovic</autor>

</autori>

<godina>2008</godina>

<cijena>99.99</cijena>

</knjiga>

BRAĆA (SIBLINGS)

Prethci i potomci

- **Prethak** (ancestor) – roditelj, roditelj roditelja, ... nekog čvora
- **Potomak** (descendant) – dijete, dijete djeteta ... nekog čvora

Osnovni izrazi

naziv_cvora

označava svu djecu čvora pod zadanim nazivom

/

označava od korijenskog čvora

//

označava čvorove koji odgovaraju izrazu bez obzira gdje se nalaze

.

označava trenutni čvor

..

označava roditelja trenutnog čvora

@

označava attribute

Upute

- U nastavku ćemo prvo pokazati **XPath** (jezik putanja), a kasnije ćemo ga proširiti s jezikom **XQuery**
- Upute upisujete u nastavku funkcije **doc**, npr.:
`doc('/db/apps/knjige/knjige.xml')/knjizara/knjiga`
- Svaki upit i odgovarajuće rezultate kopirajte u datoteku **ime_prezime.txt**

Primjeri

- Označiti svu djecu korijenskog čvora knjizara

/knjizara

Primjeri

- Označiti svu djecu čvora knjiga koji je dijete korjenskog čvora knjizara

/knjizara/knjiga

Primjeri

- Označiti svu djecu čvora cijena bez obzira gdje se on nalazi

//cijena

Primjeri

- Označiti svu djecu čvora autori koji je dijete čvora knjiga

//knjiga/autori

Predikati

- Predikati se koriste za pronalaženje specifičnih čvorova odnosno čvorova sa specifičnim vrijednostima
- Mogu se koristiti razni operatori i veznici **+**, **-**, *****, **div**, **mod**, **=**, **!=**, **<**, **<=**, **>**, **>=**, **or**, **and**, **not**
- Predikati su uvijek unutar uglatih zagrada []

Primjeri

- Označiti prvi čvor knjiga koji je dijete korijenskog čvora knjizara

/knjizara/knjiga[1]

Primjeri

- Označiti zadnji čvor knjiga koji je dijete korijenskog čvora knjizara

/knjizara/knjiga[last()]

Primjeri

- Označiti predzadnji čvor knjiga koji je dijete korijenskog čvora knjizara

/knjizara/knjiga[last() - 1]

Primjeri

- Označiti samo prva dva čvora knjiga koji su dijeca korijenskog čvora knjizara

/knjizara/knjiga[position() < 3]

Primjeri

- Označiti samo one čvorove naslov koji imaju atribut jezik

//naslov[@jezik]

Primjeri

- Označiti samo one čvorove naslov čiji je jezik engleski

```
//naslov[ @jezik = 'engleski' ]
```

Primjeri

- Označiti samo one čvorove knjiga čija je cijena ispod 100.00 kn

```
//knjiga[ cijena < 100 ]
```


Primjeri

- Označiti naslove knjiga čija je cijena ispod 100.00 kn

```
//knjiga[ cijena < 100 ]/naslov
```

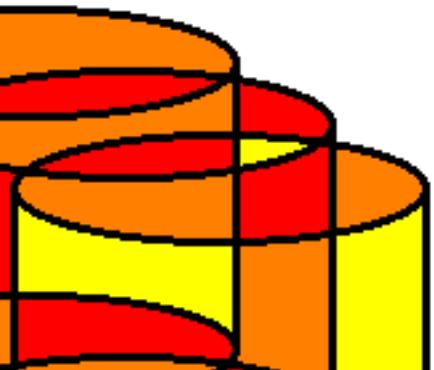
Označavanje nepoznatih vrijednosti



* - označava bilo koji elementarni čvor

@* - označava bilo koji atribut čvora

node () - označava bilo koji čvor



Primjeri

- Označiti djecu svih elementarnih čvorova koji su djeca čvora knjiga

/knjizara/knjiga/*

Primjeri

- Označiti sve elementarne čvorove u dokumentu

//*

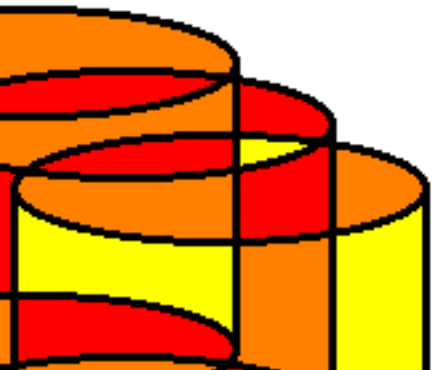
- 
- Označiti sve čvorove koji imaju bilo kakav atribut

//* [@*]

Označavanje više putanja



- Za označavanje više putanja koristi se operator | (pipe)



Primjeri

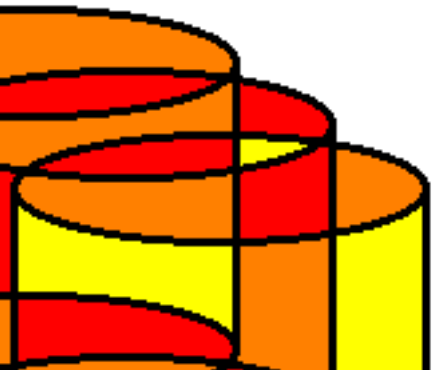
- Označiti sve elementarne čvorove naslov i cijena koji za roditelja imaju čvor knjiga

//knjiga/naslov | //knjiga/cijena

Osi



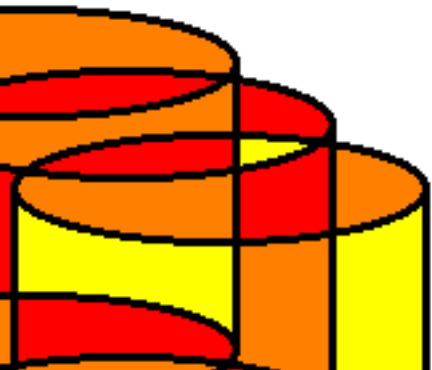
- Osi (axes) koriste se označavanje skupova čvorova koji su u nekom relativnom odnosu sa zadanim čvorom
- Uz definiciju osi koristi se operator ::



Osi



- **ancestor** – označava sve pretke trenutnog čvora
- **ancestor-or-self** – označava sve pretke skupa s trenutnim čvorom
- **attribute** – označava sve atribute trenutnog čvora
- **child** – označava svu djecu trenutnog čvora



Osi

- **descendant** – označava sve potomke trenutnog čvora
- **descendant-or-self** – označava sve potomke skupa s trenutnim čvorom
- **following** – označava sve što slijedi nakon zatvaranja trenutnog čvora
- **following-sibling** – označava svu braću/sestre nakon zatvaranja trenutnog čvora

Osi

- **namespace** – označava sve čvorove koji se odnose na prostor nazivlja (namespace) trenutnog čvora
- **parent** – označava roditelja trenutnog čvora
- **preceding** – označava sve prije otvaranja trenutnog čvora
- **preceding-sibling** – označava svu braću prije otvaranja trenutnog čvora
- **self** – označava trenutni čvor

Primjeri

- Označiti sve čvorove autor koji su djeca čvora autori koji za pretka imaju čvor knjižara

```
//autor i[ ancestor::knjizara ]/autor
```

Primjeri

- Označiti sve autore čiji je naslov djela na engleskom

```
//autori[ preceding-sibling::naslov/  
@jezik = 'engleski' ]/autor
```

Primjeri

- Označiti sve godine izdanja hrvatskih djela

```
//knjiga[ descendant::naslov/@jezik  
= 'hrvatski' ]/godina
```

Zadatak

- Implementirajte XPath upit koji će označiti sve čvorove cijena čija je godina izdanja prije 2000 godine.
- Upit i rezultat također kopirajte u datoteku **ime_prezime.txt**

XQuery

- XQuery (neki ga nazivaju i XQL) je jezik za postavljanje upita nad XML podacima
- Moglo bi se reći da je to SQL za XML podatke
- Temelji se na XPath izrazima

Koristi XQuery-a

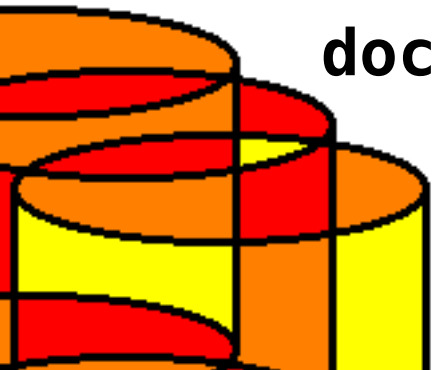
- Koristi se za
 - Ekstrakciju podataka za korištenje s web uslugama
 - Generiranje izvještaja iz podataka
 - Transformaciju XML podataka u XHTML
 - Pretraživanje web dokumenata ...

Funkcije



- XQuery koristi funkcije za ekstrakciju podataka iz XML dokumenata
- Primjerice funkcija **doc()** koristi se za otvaranje nekog XML dokumenta (koji ne mora biti nužno pohranjen na eXist-DB sustavu), npr.

```
doc( 'https://tinyurl.com/knjige-xml' )
```



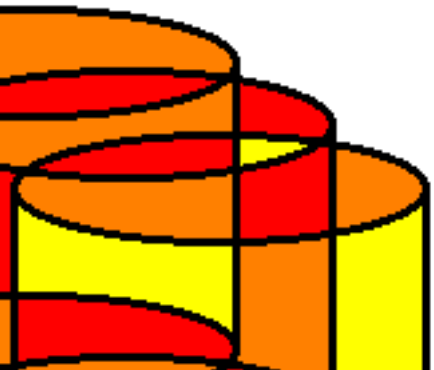
Putanje



- XQuery koristi putanje radi navigacije kroz XML dokumente

Primjer

```
doc( 'knjige.xml' )/knjizara/knjiga/godina
```



Napomena

- U nastavku ćemo umjesto pune putanje do dokumenta knjige.xml pisati samo knjige.xml
- Za isprobavanje upita koristite punu putanju.

Predikati

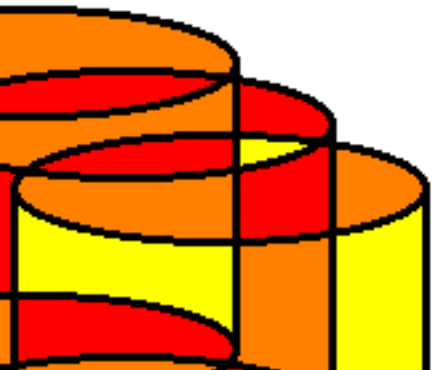
- Predikati se koriste za postavljanje uvjeta nad upitom

```
doc( 'knjige.xml' ) /knjizara/knjiga/godina[ text() < 2000 ]
```

FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

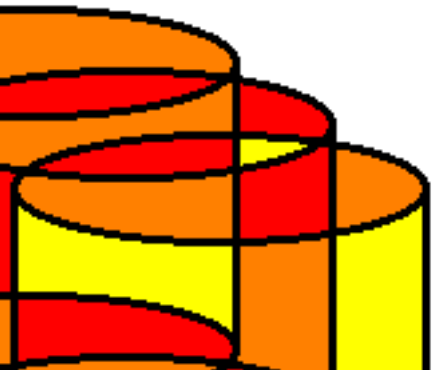


FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

**F
L
O
W
R**



FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

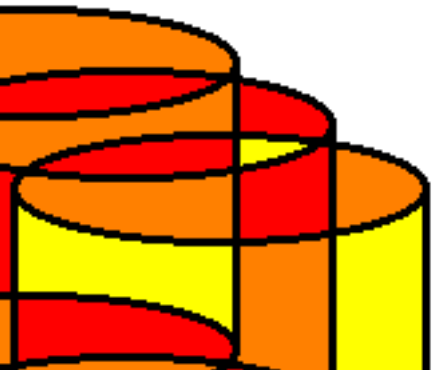
FOR

L

O

W

R



FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

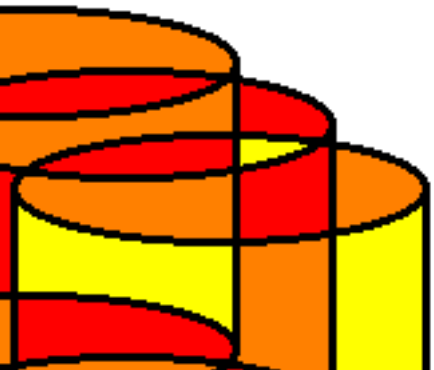
FOR

LET

O

W

R



FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

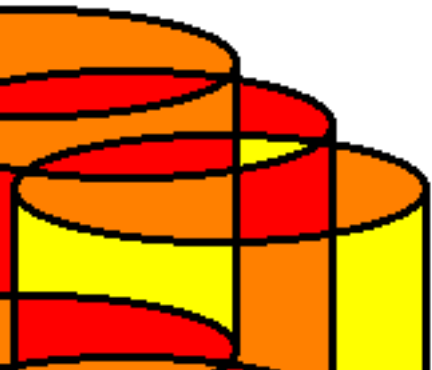
FOR

LET

ORDER BY

W

R



FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

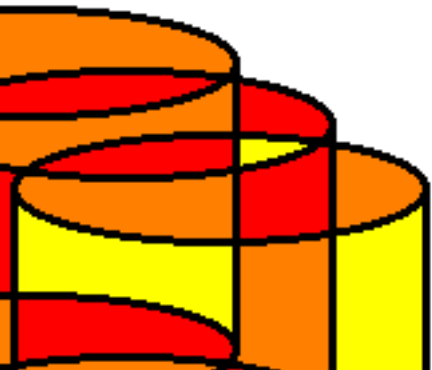
FOR

LET

ORDER BY

WHERE

R



FLOWR



- U XQuery jeziku koriste se tzv. FLOWR izrazi

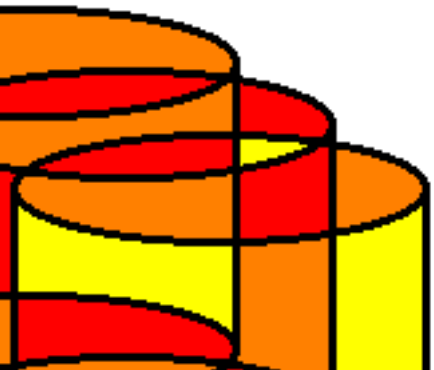
FOR

LET

ORDER BY

WHERE

RETURN



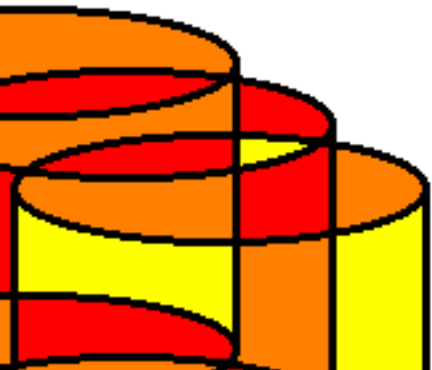
Primjer

```
for $x in doc( 'knjige.xml'  
  )/knjizara/knjiga  
where $x/godina < 2000  
order by $x/naslov  
return $x/autor/autor
```

XQuery i generiranje novih dokumenata



- XQuery se može iskoristiti da se iz postojećih podataka generiraju novi dokumenti



Primjer (generiranje HTML liste autora)

```
<ol>  
{  
for $x in doc( 'knjige.xml'  
  )/knjizara/knjiga/autori/autor  
order by $x  
return <li>{$x}</li>  
}  
</ol>
```

Kako bi dobili samo podatke

```
<ol>
{
for $x in doc( 'knjige.xml'
  )/knjizara/knjiga/autori/autor
order by $x
return <li>{$x/text()}</li>
}
</ol>
```


lli ...

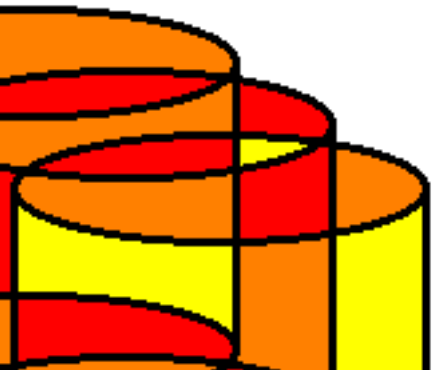
```
<ol>
{
for $x in doc( 'knjige.xml'
)/knjizara/knjiga/autori/autor
order by $x
return <li>{data($x)}</li>
}
</ol>
```

If-then-else uvjeti izlaza



- Moguće je postavljanje ako-onda uvjeta na izlaz

```
for $x in doc('knjige.xml')/knjizara/knjiga
return if( $x/naslov/@jezik = 'hrvatski' )
    then
    <hr_knjiga naslov="{ data( $x/naslov ) }" />
    else
    <en_knjiga naslov="{ data( $x/naslov ) }" />
```



For klauzula

- For klauzula ponaša se slično for petlji (ključna riječ **to**)

Primjer

```
for $i in (1 to 5)  
return <test i="{ $i}" />
```

For klauzula

- Ključna riječ **at**

```
for $x at $i in doc( 'knjige.xml'
  )/knjizara/knjiga
return <knjiga br="{ $i }"
  naslov="{ data( $x/naslov ) }" />
```

For klauzula

- Moguće je imati više od jednog izraza u for klauzuli

```
for $x in ( 1 to 3 ), $y in ( 'blue',  
    'red' )  
return  
<button class="{ $y }" width="{ $x }00" />
```

LET klauzula

- Omogućava postavljanje vrijednosti koje se neće iterirati u petlji

```
let $x := (1 to 5)
```

```
return <test param="{ $x }" />
```

Funkcije

- Postoji velik broj ugrađenih funkcija kojima je moguće raditi u XPath-u i XQuery-u

Primjeri

```
for $x in doc( 'knjige.xml'
  )/knjizara/knjiga/naslov/text()
return <naslov skraceno="{substring($x, 1, 10)}" />
```


Primjeri

```
for $x in doc( 'knjige.xml'
  )/knjizara/knjiga
return <knjiga cijena="{ round(
  $x/cijena ) }" naslov="{
  $x/naslov/text() }" />
```

Ažuriranje

- eXist implementira ekstenzije za ažuriranje XML dokumenata putem Update upita:
 - Insert
 - Replace
 - Value
 - Delete
 - Rename

Primjer - Insert

```
for $d in doc('knjige.xml')/knjizara/knjiga
where $d/naslov = 'Teorija i primjena baza podataka'
return update insert <autor>Chris Date</autor>
into $d/autori
```

Primjer - Replace

```
for $d in doc('knjige.xml')/knjizara/knjiga
where $d/naslov = 'Teorija i primjena baza podataka'
return update replace $d/cijena
       with <cijena>200.00</cijena>
```

Primjer - Value

```
for $d in doc('knjige.xml')/knjizara/knjiga
where $d/godina < 1998
return update value $d/cijena
       with 30.00
```

Primjer - Delete

```
for $d in doc('knjige.xml')/knjizara/knjiga
where $d/naslov = 'Teorija i primjena baza podataka'
return update delete
       $d/autori/autor[ . = 'Chris Date' ]
```

Primjer - Rename

```
for $d in doc('knjige.xml')/knjizara/knjiga  
return update rename $d/naslov as 'naziv'
```

Zadatak

- Svim knjigama koje su izdane prije 2000 godine i čiji je jezik naslova hrvatski, postaviti cijenu na 150.
- Upit također kopirajte u datoteku `ime_prezime.txt`

Zadatak



- Oblikujte novi XML dokument koji će sadržavati podatke o klijentima knjižare (**klijenti.xml**), pri čemu za svakog korisnika treba voditi računa o sljedećim podacima:
 - Korisničko ime
 - E-mail adresa
 - Država
 - Broj bankovne kartice
 - Interesi (popis ključnih riječi koje ih zanimaju)
- Kreirajte novi XQuery upit koji će uz svakog korisnika ispisati naslove onih knjiga koje imaju barem jednu zajedničku ključnu riječ s interesima korisnika
- Oblik XML dokumenta i oblik odgovora je proizvoljan
- Upit pridodajte u datoteku **ime_prezime.txt**

