

Y36XML – Technologie XML

Přednáší:

Irena Mlýnková (mlynkova@ksi.mff.cuni.cz)

Martin Nečaský (necasky@ksi.mff.cuni.cz)

ZS 2009

Stránka přednášky:

<http://www.ksi.mff.cuni.cz/~mlynkova/Y36XML/>

Osnova předmětu

- ☐ Úvod do principů formátu XML, přehled XML technologií, jazyk DTD
 - ☐ Datové modely XML, rozhraní DOM a SAX
 - ☐ Úvod do jazyka XPath
 - ☐ Úvod do jazyka XSLT
 - ☐ XPath 2.0, XSLT 2.0
 - ☐ Úvod do jazyka XML Schema
 - ☐ Pokročilé rysy jazyka XML Schema
 - ☐ Přehled standardních XML formátů
 - ☐ Úvod do jazyka XQuery
 - ☐ Pokročilé rysy jazyka XQuery, XQuery Update
 - ☐ Úvod do XML databází, nativní XML databáze, číslovací schémata, structural join
 - ☐ Relační databáze s XML rozšířením, SQL/XML
-

XPath 1.0 a 2.0

Opakování XPath 1.0 na příkladech

Představení XPath 2.0

XPath 1.0 – Brief Tutorial

- Navigace v XML dokumentech pomocí cest
 - XML dokument chápán jako strom s uzly reprezentujícími dokument, elementy, atributy, texty, ...
 - Výsledkem vyhodnocení cesty je množina uzlů nebo nebo nějaká hodnota
 - node-set, boolean, string, number
-

XPath 1.0 – Brief Tutorial

- ☐ Cesta je složena z kroků

- ☐ Krok:

`osa::test-uzlu predikát1 ... predikátN`

- Osa

- ☐ Určuje směr pohybu v daném kroce

- Test uzlu

- ☐ Propustí pouze ty uzly vybrané osou, které mají zadaný typ a případně název

- Predikát

- ☐ Logická podmínka v hranatých závorkách ([]), která dále filtruje výsledek kroku

- ☐ Zkratky usnadňující zápis

XPath 1.0 – Brief Tutorial

`/objednavka ⇔ /child::objednavka`

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

`/objednavka/zakaznik/jmeno`

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

/objednavka/polozky/polozka/jmeno

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```


XPath 1.0 – Brief Tutorial

`/objednavka/polozky/polozka/@kod`
⇔ `/objednavka/polozky/polozka/attribute::kod`

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//jmeno ⇔ /descendant-or-self::node()/jmeno
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

`//polozka/jmeno`

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

`//polozka/* ⇔ //polozka/child::*`

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
/*/*/jmeno
```

```
<objednavka cislo="0233">  
  <zakaznik>  
    <jmeno>Martin Nečaský</jmeno>  
    <email>martinnec@gmail.com</email>  
  </zakaznik>  
  <polozky>  
    <polozka kod="V289348">  
      <jmeno>Jméno výrobku 289348</jmeno>  
      <barva>blue</barva>  
      <kusu>1</kusu><cena-kus>1234</cena-kus>  
    </polozka>  
    <polozka kod="V99381">  
      <jmeno>Jméno výrobku 99381</jmeno>  
      <kusu>5</kusu><cena-kus>98</cena-kus>  
    </polozka>  
  </polozky>  
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//polozky/polozka[1] ⇔ //polozky/polozka[position()=1]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//polozky/polozka[last()]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//polozka[barva]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```


XPath 1.0 – Brief Tutorial

```
//polozka[@kod]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//objednavka[.//polozka/barva]
```

```
<objednavka cislo="0233">  
  <zakaznik>  
    <jmeno>Martin Nečaský</jmeno>  
    <email>martinnec@gmail.com</email>  
  </zakaznik>  
  <polozky>  
    <polozka kod="V289348">  
      <jmeno>Jméno výrobku 289348</jmeno>  
      <barva>blue</barva>  
      <kusu>1</kusu><cena-kus>1234</cena-kus>  
    </polozka>  
    <polozka kod="V99381">  
      <jmeno>Jméno výrobku 99381</jmeno>  
      <kusu>5</kusu><cena-kus>98</cena-kus>  
    </polozka>  
  </polozky>  
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//objednavka[.//polozka[3]]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//polozka[@kod = "V289348"]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//objednavka[.//polozka/barva = "blue"]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//objednavka[zakaznik/jmeno = "Martin Nečaský" and  
              ((.//polozka = "blue") or (.//polozka = "red"))]
```

```
<objednavka cislo="0233">  
  <zakaznik>  
    <jmeno>Martin Nečaský</jmeno>  
    <email>martinnec@gmail.com</email>  
  </zakaznik>  
  <polozky>  
    <polozka kod="V289348">  
      <jmeno>Jméno výrobku 289348</jmeno>  
      <barva>blue</barva>  
      <kusu>1</kusu><cena-kus>1234</cena-kus>  
    </polozka>  
    <polozka kod="V99381">  
      <jmeno>Jméno výrobku 99381</jmeno>  
      <kusu>5</kusu><cena-kus>98</cena-kus>  
    </polozka>  
  </polozky>  
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//objednavka[count(../polozka) > 2]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

XPath 1.0 – Brief Tutorial

```
//polozka[count(*) > 3]
```

```
<objednavka cislo="0233">
  <zakaznik>
    <jmeno>Martin Nečaský</jmeno>
    <email>martinnec@gmail.com</email>
  </zakaznik>
  <polozky>
    <polozka kod="V289348">
      <jmeno>Jméno výrobku 289348</jmeno>
      <barva>blue</barva>
      <kusu>1</kusu><cena-kus>1234</cena-kus>
    </polozka>
    <polozka kod="V99381">
      <jmeno>Jméno výrobku 99381</jmeno>
      <kusu>5</kusu><cena-kus>98</cena-kus>
    </polozka>
  </polozky>
</objednavka>
```

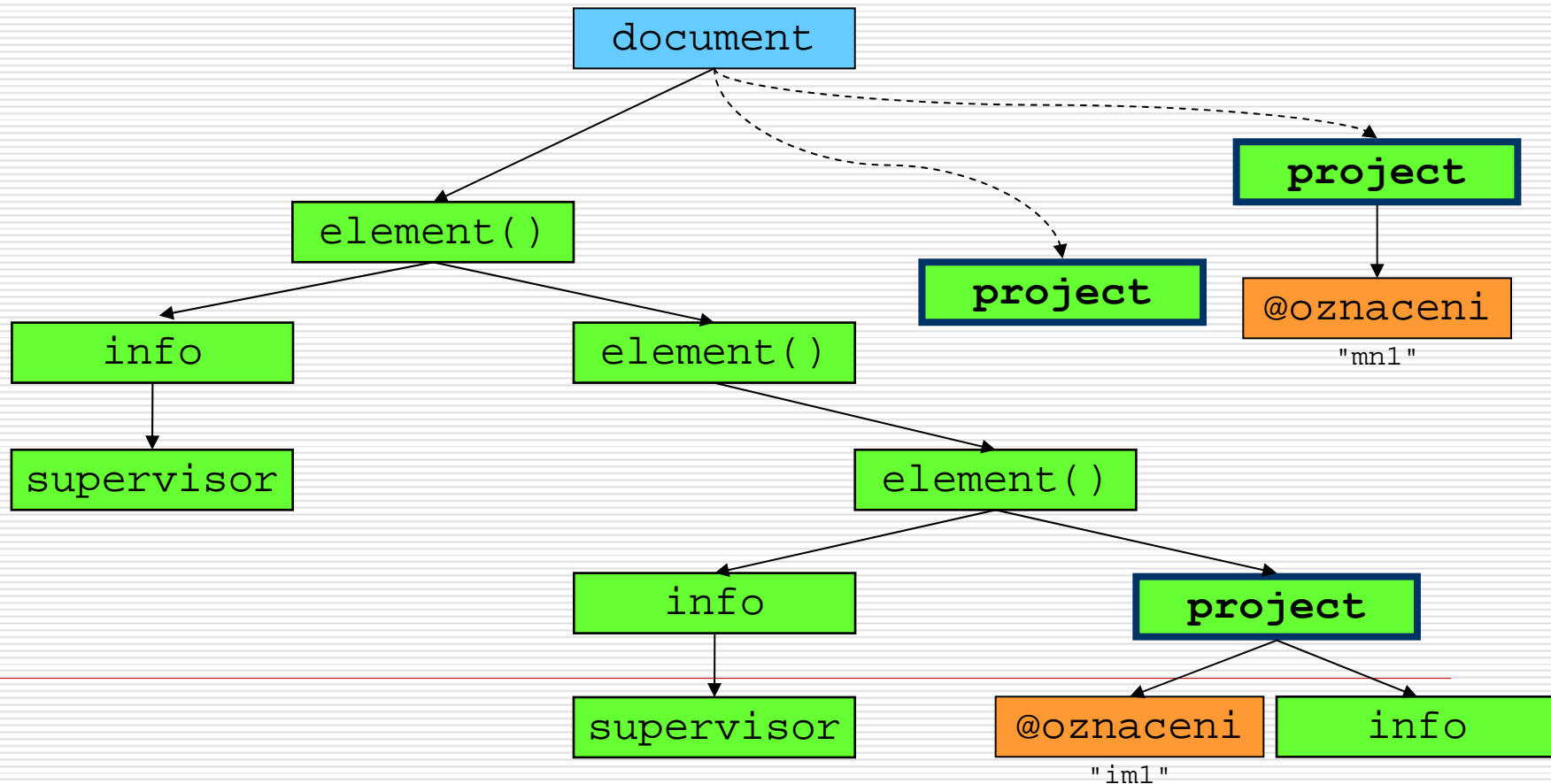

XPath 1.0 – Brief Tutorial

```
//zakaznik[not(jmeno = preceding::zakaznik/jmeno)]
```

```
<objednavky>  
  <objednavka cislo="0233">  
    <zakaznik><jmeno>Martin Nečaský</jmeno></zakaznik>  
  </objednavka>  
  <objednavka cislo="0243">  
    <zakaznik><jmeno>Josef Novák</jmeno></zakaznik>  
  </objednavka>  
  <objednavka cislo="0289">  
    <zakaznik><jmeno>Martin Nečaský</jmeno></zakaznik>  
  </objednavka>  
</objednavky>
```

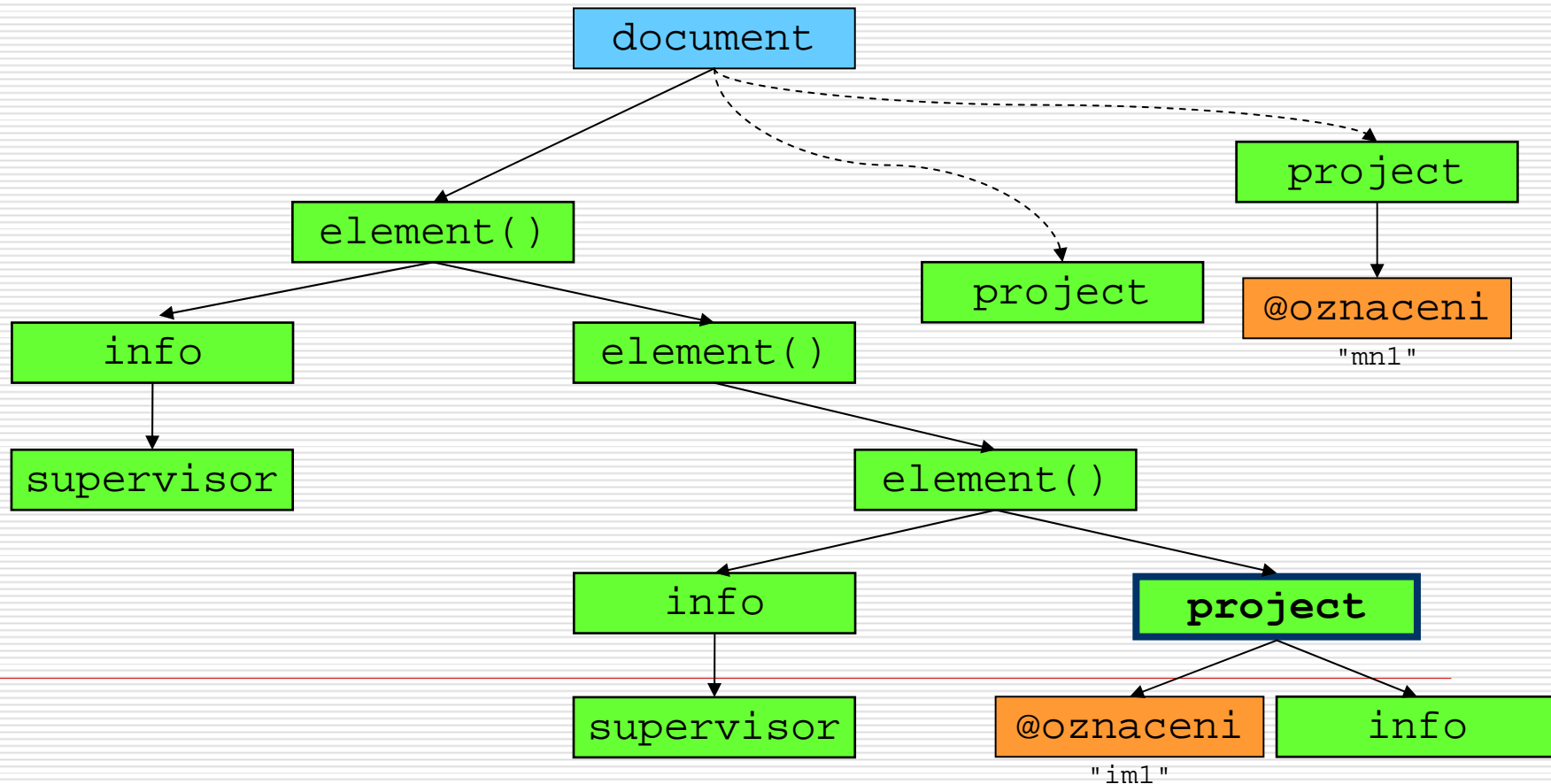
XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::* /info/supervisor) [last()]
```



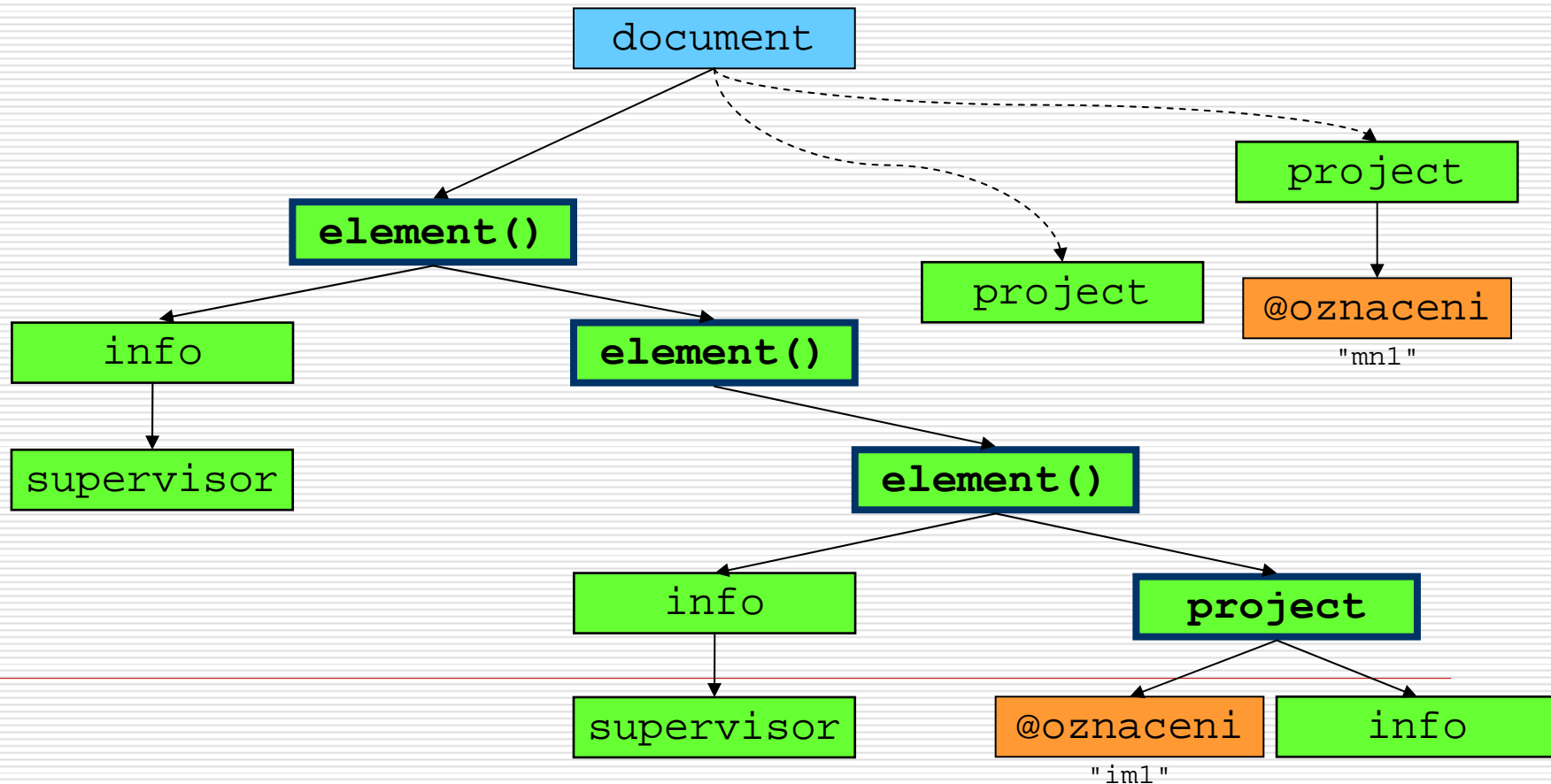
XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::* /info/supervisor) [last()]
```



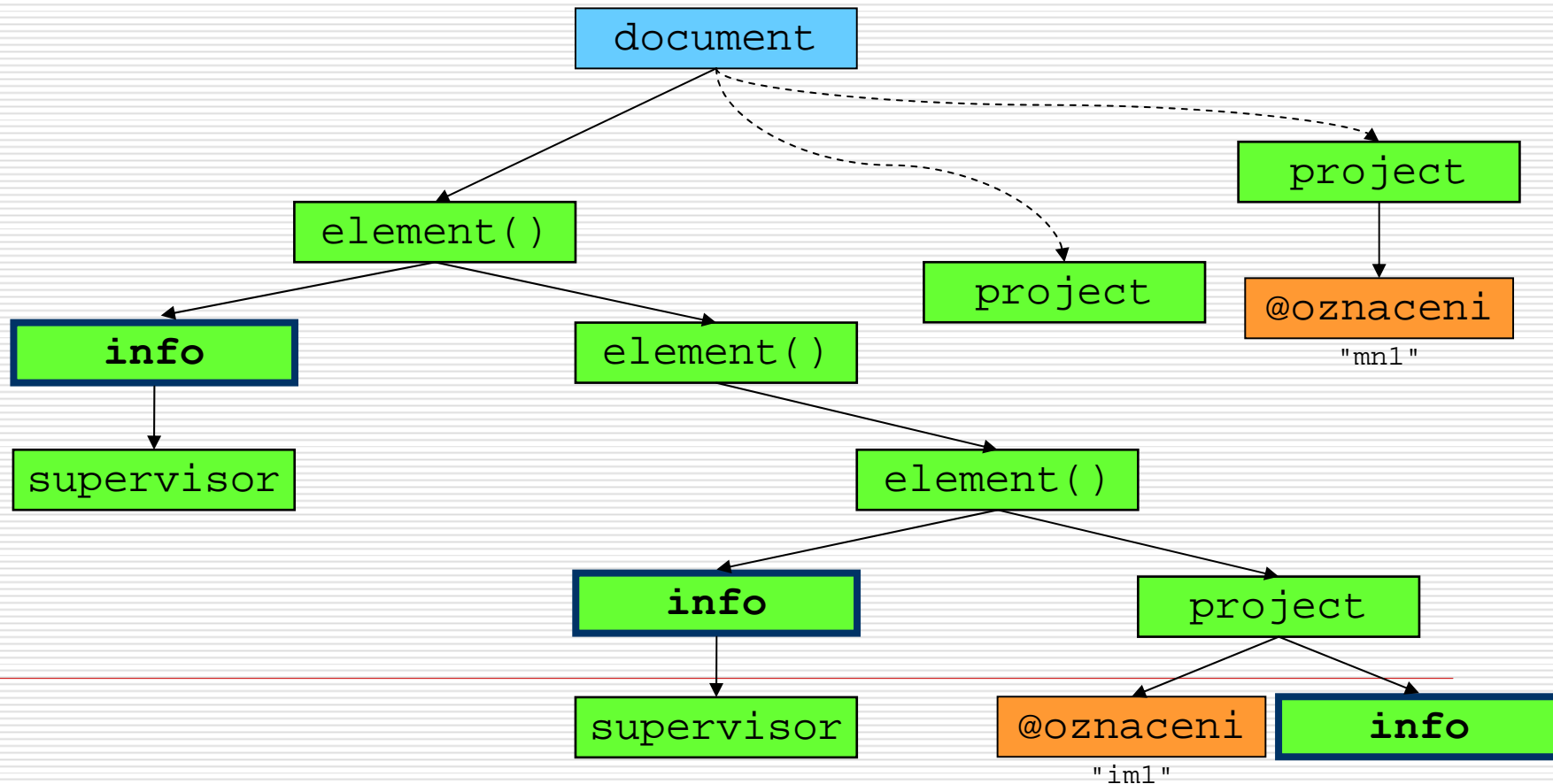
XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::* /info/supervisor) [last()]
```



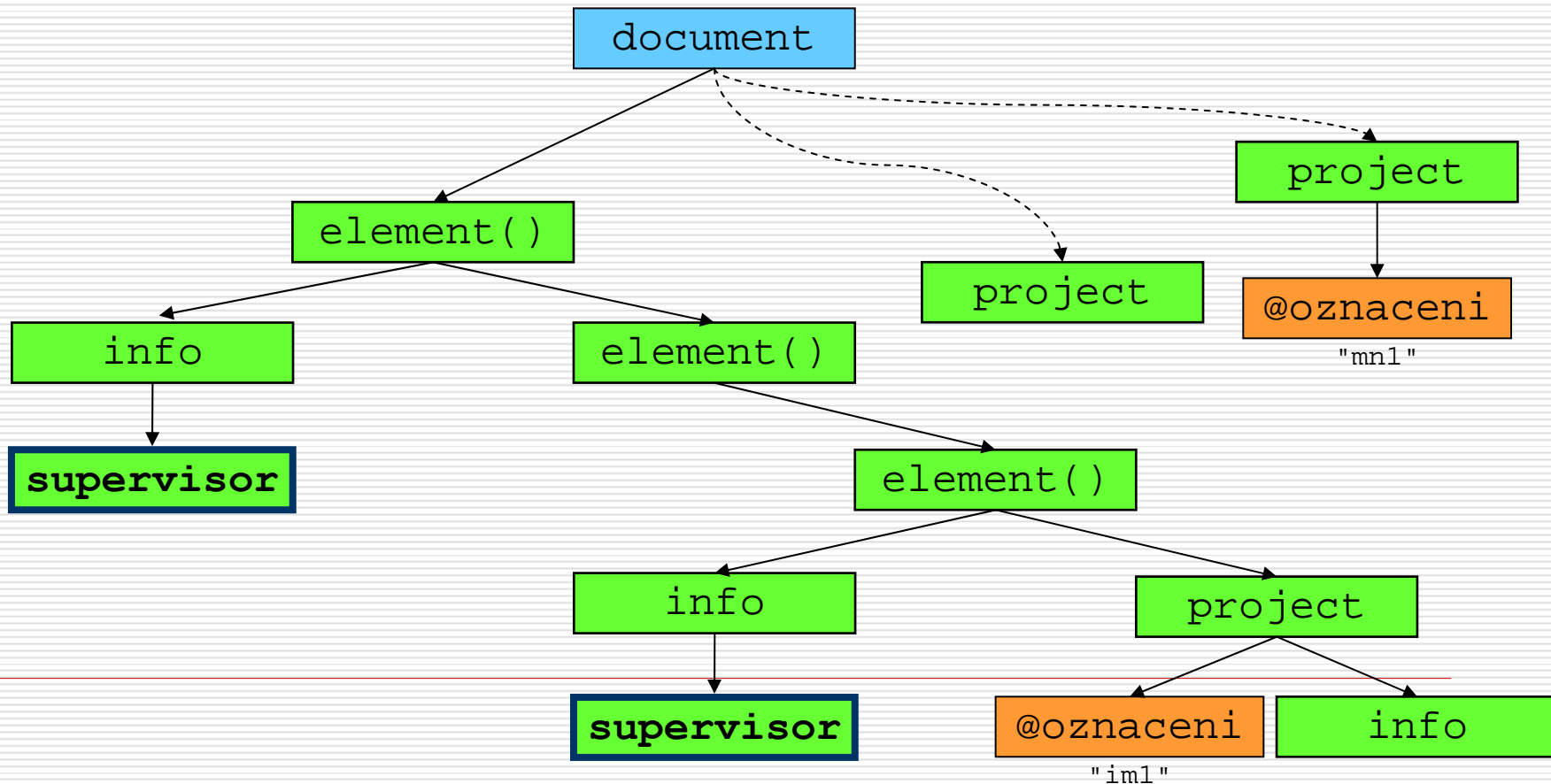
XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::* /info /supervisor) [last()]
```



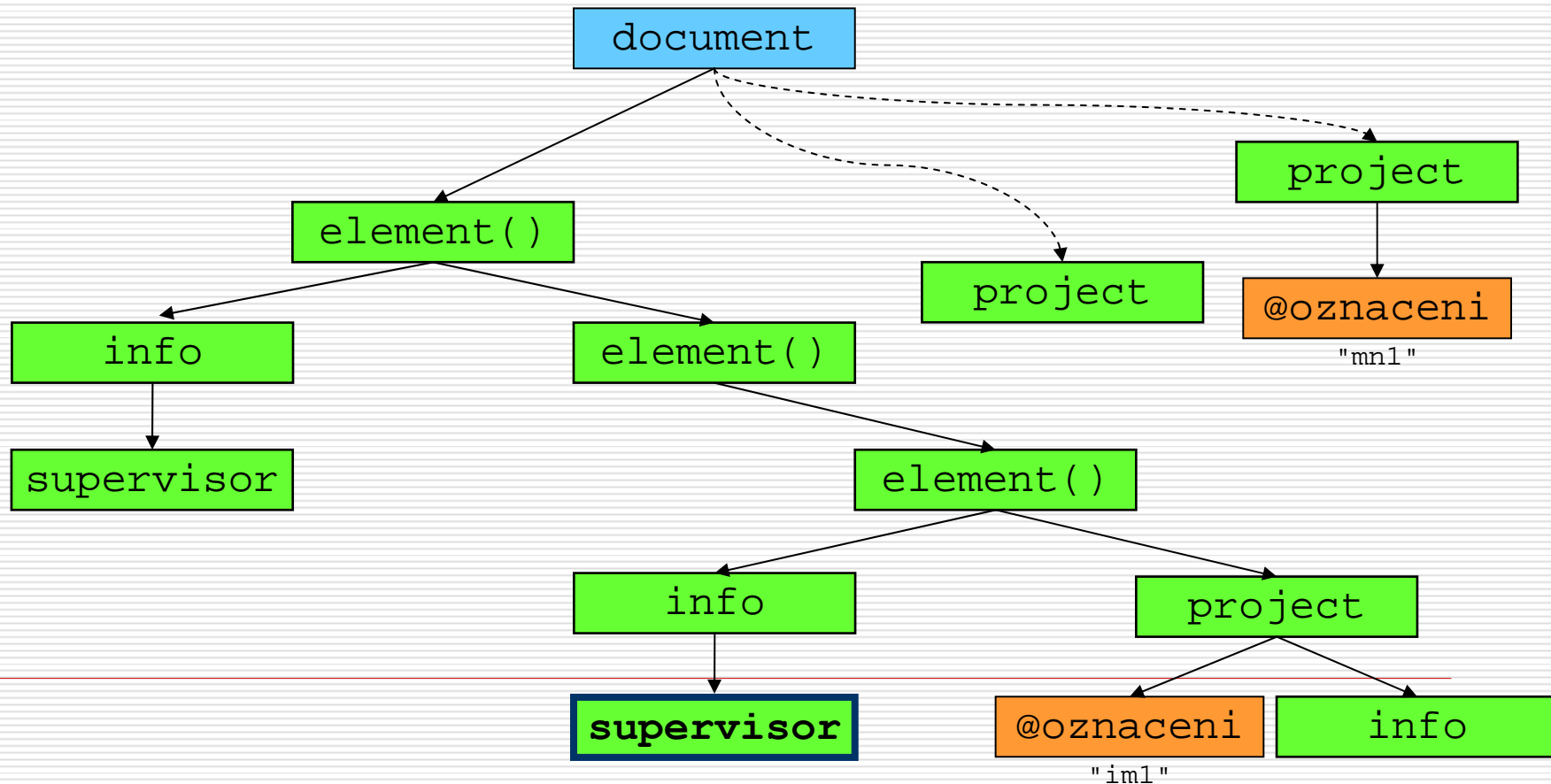
XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::*/info/supervisor)[last()]
```



XPath 1.0 – Brief Tutorial

```
(//project[@oznaceni="im1"]/ancestor-or-self::* /info/supervisor) [last()]
```



XPath 2.0

- ❑ Pracuje na uspořádaných kolekcích
 - ❑ Přidává nové konstrukce
 - Iterace sekvencí, slučování sekvencí, podmíněné výrazy, kvantifikované výrazy
 - ❑ Vazba na XML schémata
 - XML uzly mají přiřazeny datové typy ve smyslu jazyka XML Schema
-

XPath 2.0

☐ XPath 1.0

- node-set, boolean, number, string

☐ XPath 2.0

- sequence, datové typy jazyka XML Schema
-

XPath 2.0 – Datový model

- **sequence** (**sekvence**) je *uspořádaná* kolekce položek
 - Výsledkem každého XPath 2.0 výrazu je sekvence
 - **položka** je buď atomická hodnota nebo uzel
 - **atomická hodnota** je hodnota nějakého jednoduchého datového typu jazyka XML Schema
 - **uzel** je instance nějakého typu uzlů (element, atribut, ...)
-

XPath 2.0 - Uzly

☐ uzel má

- identitu
 - datový typ (jednoduchý či složený typ ve smyslu jazyka XML Schema)
 - typovanou hodnotu
 - ☐ Hodnota podle datového typu (fn:data())
 - řetězcovou hodnotu
 - ☐ Typovaná hodnota zkonvertovaná na xs:string (fn:string())
-

XPath 2.0 - Sekvence

- ❑ Konstruktor ()
 - (1, 2, 3, 4)
 - ❑ Konstruktor to
 - (1, 5 to 8) = (1, 5, 6, 7, 8)
 - ❑ Konstruktor může obsahovat XPath výrazy
 - (//kniha, //cd)
 - ❑ Konstruktor může být krok v XPath výrazu
 - (1 to 100)[. mod 5 = 0]
 - //polozka/(cena,mnozstvi)
-

XPath 2.0 - Sekvence

- ❑ Vše je sekvence
 - $1 = (1)$
 - ❑ Sekvence jsou mělké (tj. neobsahují podsekvence)
 - $(1, (2, 3), 4) = (1, 2, 3, 4)$
 - ❑ Může obsahovat duplicitní prvky
 - $(1, 2, 1 \text{ to } 2) = (1, 2, 1, 2)$
 - ❑ Může obsahovat atomické hodnoty i uzly
 - $(1, 2, //kniha)$
-

XPath 2.0 - Iterace

- Konstrukce **for** pro iteraci sekvencí uvnitř výrazů

```
for $i in (1,2,3)
return $i
```

□ (1,2,3)

```
for $i in (10,20),
    $j in (1,2)
return ($i+$j)
```

□ (11,12,21,22)

```
for $varname1 in (expression1),
    ...,
    $varnameN in (expressionN)
return expression
```

XPath 2.0 – Iterace

```
fn:sum( for
    $polozka in //polozka
return
    $polozka/mnozstvi * $polozka/cena )
```

XPath 2.0 – Slučování

- ❑ Sjednocení sekvencí
 - **union** nebo znak | (| již v XPath 1.0)
 - ❑ Průnik sekvencí
 - **intersect**
 - ❑ Rozdíl sekvencí
 - **except**
 - ❑ Pouze pro sekvence uzlů
 - Pokud sekvence obsahuje položku, která není uzel - chyba
 - ❑ Všechny operátory eliminují duplicity
 - Dva uzly jsou duplicitní, pokud mají stejnou identitu
-

XPath 2.0 - Slučování

```
expression1 union expression2
```

```
expression1 intersect expression2
```

```
expression1 except expression2
```

XPath 2.0 – Slučování

```
for  
    $polozka in objednávka//polozka  
return  
    $polozka/* except $polozka/cena
```

XPath 2.0 – Slučování

```
//polozka[barva="blue"]  
intersect  
//objednavka[poradovecisko>1000]**
```

```
<objednavka cislo="0233" poradovecisko="2911">  
  <zakaznik>  
    <jmeno>Martin Nečaský</jmeno>  
    <email>martinnec@gmail.com</email>  
  </zakaznik>  
  <polozky>  
    <polozka kod="V289348">  
      <jmeno>Jméno výrobku 289348</jmeno>  
      <barva>blue</barva>  
      <kusu>1</kusu><cena-kus>1234</cena-kus>  
    </polozka>  
    ...  
  </polozky>  
</objednavka>
```

XPath 2.0 – Porovnávání

- Již jsme viděli dva typy porovnávacích operatorů:
 - =, !=, ... a &eq;, ≠, ...
 - Nový typ: porovnávání uzlů
 - expression1 **is** expression2
 - true, pokud se oba operandy vyhodnotí na jeden a ten samý uzel
 - expression1 **<<** expression2, resp. expression1 **>>** expression2
 - true, pokud uzel identifikovaný levým operandem předchází, resp. následuje za uzlem identifikovaným pravým operandem (bere se pořadí uzlů v dokumentu)
 - Pokud se některý z operandů vyhodnotí na sekvenci delší než 1 – chyba
 - Pokud se některý z operandů vyhodnotí na prázdnou sekvenci, je výsledkem prázdná sekvence
-

XPath 2.0 – Porovnávání

```
//objednavka[
  zakaznik/jmeno = "Martin Nečaský"
and
  . << (
    //objednavka[
      */jmeno = "Martin Nečaský"
and
      fn:sum(
        for $p in .//polozka
        return
          $p/mnozstvi * $p/cena
      ) > 100000
    ]
  ) [1]
]
```

XPath 2.0 – Podmíněné výrazy

```
if (expression1)
  then (expression2)
  else (expression3)
```

```
for $vyrobek in /katalog//vyrobek
return
  if ($vyrobek/sleva = "ano")
  then return $vyrobek/cena-po-sleve
  else return $vyrobek/plna-cena
```

XSLT 1.0 a 2.0

Opakování XSLT 1.0 na příkladech

Pokročilejší konstrukce XSLT 1.0

Představení XSLT 2.0

Ukázka XSLT skriptu

XSLT 1.0 - Třídění

☐ Pomocí **xsl:sort**

- Uvnitř xsl:apply-templates nebo xsl:for-each

- Příznak **select**

 - ☐ Podle čeho třídíme

- Příznak **order**

 - ☐ **ascending** (vzestupně), **descending** (sestupně)

XSLT 1.0 - Třídění

```
<xsl:for-each select="//polozka">  
  <xsl:sort select="./jmeno" />  
  ...  
</xsl:for-each>
```

XSLT 1.0 – Klíče

- Pomocí **xsl:key**
 - Deklaruje klíč v XML datech
 - Příznak **name** : Název klíče
 - Příznak **match**
 - XPath výraz identifikující elementy, pro které klíč deklarujeme
 - Příznak **use** : XPath výraz identifikující prvky klíče
 - Funkce **key(název-klíče, hodnota-klíče)**
 - Nalezne uzel podle klíče **název-klíče** s hodnotou **hodnota-klíče**
-

XSLT 1.0 - Třídění

```
<xsl:key name="vyrobek-key"
          match="vyrobek"
          use="./kod-vyrobku" />

<xsl:for-each select="//polozka">
  <xsl:variable name="vyrobek"
                select="key('vyrobek-key', ./@kod)" />
  <xsl:value-of select="$vyrobek/jmeno" />
  <xsl:value-of select="$vyrobek/vyrobce" />
</xsl:for-each>
```

XSLT 1.0 – Módy

- ❑ Zpracování uzlů vícekrát, v různých módech
 - ❑ Příznak **mode** elementů `xsl:template` a `xsl:apply-template`
 - pro `xsl:template` pouze ty s příznakem `match`
-

XSLT 1.0 - Mody

```
<xsl:template match="/">
  <xsl:apply-templates mode="prehled" />
  <xsl:apply-templates mode="plny-vypis" />
</xsl:template>

<xsl:template match="polozka" mode="prehled">
  ...
</xsl:template>

<xsl:template match="polozka" mode="plny-vypis">
  ...
</xsl:template>
```

XSLT 1.0 – Kombinace skriptů

☐ Vložení existujícího skriptu pomocí

■ **xsl:include**

- ☐ Příznak href se odkazuje na vkládaný skript
- ☐ Pravidla z vkládaného skriptu jsou překopírována

■ **xsl:import**

- ☐ Příznak href se odkazuje na vkládaný skript
 - ☐ Pravidla z vkládaného skriptu jsou překopírována
 - ☐ Navíc platí, že pravidla z aktuálního skriptu mají přednost před vloženými
-

XSLT 1.0 - Kombinace skriptů

```
<!-- stylesheet A -->  
<xsl:stylesheet ...>  
  
  <xsl:include href="B.xsl" />  
  <xsl:import href="C.xsl" />  
  
</xsl:stylesheet>
```


XSLT 1.0 – Kombinace skriptů

- Určení přednosti při importu
 - A importuje B a potom C
 - B importuje D
 - C importuje E
 - Přednost (od nejmenší)
 - A, C, E, B, D
-

XSLT 2.0

- ❑ Používá XPath 2.0
 - ❑ Nové konstrukce
 - Výstup do více různých dokumentů
 - **xsl:for-each-group** pro seskupování uzlů
 - Uživatelsky definované funkce
 - ❑ Mohou být volány z XPath výrazů
 - ❑ ... mnoho dalších věcí
 - Nestihneme
 - Viz. dokumentace
-

XSLT 2.0 – výstup

☐ instrukce **xsl:result-document**

■ Příznak **href**

- ☐ Výstupní soubor (URL)

■ Příznak **format**

- ☐ Formát výstupního souboru
- ☐ Odkaz na nějakou instrukci xsl:output

☐ instrukce **xsl:output**

■ Příznak **name**

- ☐ Pojmenování, abychom se na něj mohli odkazovat
-

XSLT 2.0 – výstup

```
<xsl:output name="objednavky-report-format" method="xhtml" .../>

<xsl:output name="objednavka-format" method="xhtml" ... />

<xsl:template match="/">
  <xsl:result-document href="objednavky-report.html"
                      format="objednavky-report-format">
    <html>
      <body><xsl:apply-templates /></body>
    </html>
  </xsl:result-document>
  <xsl:for-each select="document('objednavky.xml')//objednavka">
    <xsl:result-document href="objednavka{./@cislo}.html"
                      format="objednavka-format">
      <xsl:apply-templates select="." />
    </xsl:result-document>
  </xsl:for-each>
</xsl:template>
```

XSLT 2.0 – výstup

□ exampleN01.xsl

XSLT 2.0 – seskupování uzlů

- ❑ Seskupování uzlů do skupin podle daných podmínek
 - ❑ Konstrukce **xsl:for-each-group**
 - Příznak **select**
 - ❑ Stejný jako u xsl:for-each
 - Příznak **group-by**
 - ❑ XPath výraz specifikující hodnoty, podle kterých máme seskupovat
 - Další příznaky pro jiné typy seskupování
 - ❑ Viz. Specifikace
 - Funkce **current-group()** vrací prvky v aktuální skupině
-

XSLT 2.0 – seskupování uzlů

```
<xsl:template match="/">
  <xsl:for-each-group select="document('vyrobky.xml')//vyrobek"
                    group-by="./kategorie">
    <h1><xsl:value-of select="./kategorie" /></h1>
    <p>
      <xsl:value-of select="current-group()/jmeno" separator="," />
    </p>
  </xsl:for-each-group>
</xsl:template>
```

XSLT 2.0 – seskupování uzlů

□ exampleN02.xsl

XSLT 2.0 – uživatelské funkce

- ❑ Pomocí instrukce **xsl:function**
 - Příznak **name**
 - ❑ Jméno funkce
 - Příznak **as**
 - ❑ Návratový typ funkce
 - Instrukce **xsl:param**
 - ❑ Specifikace parametrů
 - Podobný mechanismus jako pojmenované šablony
 - Funkce lze ale volat z XPath výrazů
-

XSLT 2.0 – uživatelské funkce

```
<xsl:function name="mf:cena-s-dph" as="xs:anyAtomicType">
  <xsl:param name="cena" as="xs:anyAtomicType"/>
  <xsl:value-of select="$cena * 1.19" />
</xsl:function>

<xsl:template match="/">
  <html>
    <body>
      <xsl:value-of select="mf:cena-s-dph
        (sum(for $p in //polozka return $p/cena * $p/mnozstvi))" />
    </body>
  </html>
</xsl:template>
```

XSLT 2.0 – uživatelské funkce

□ exampleN03.xsl



Konec

