

---

# Technologie XML

Úvod: XPATH

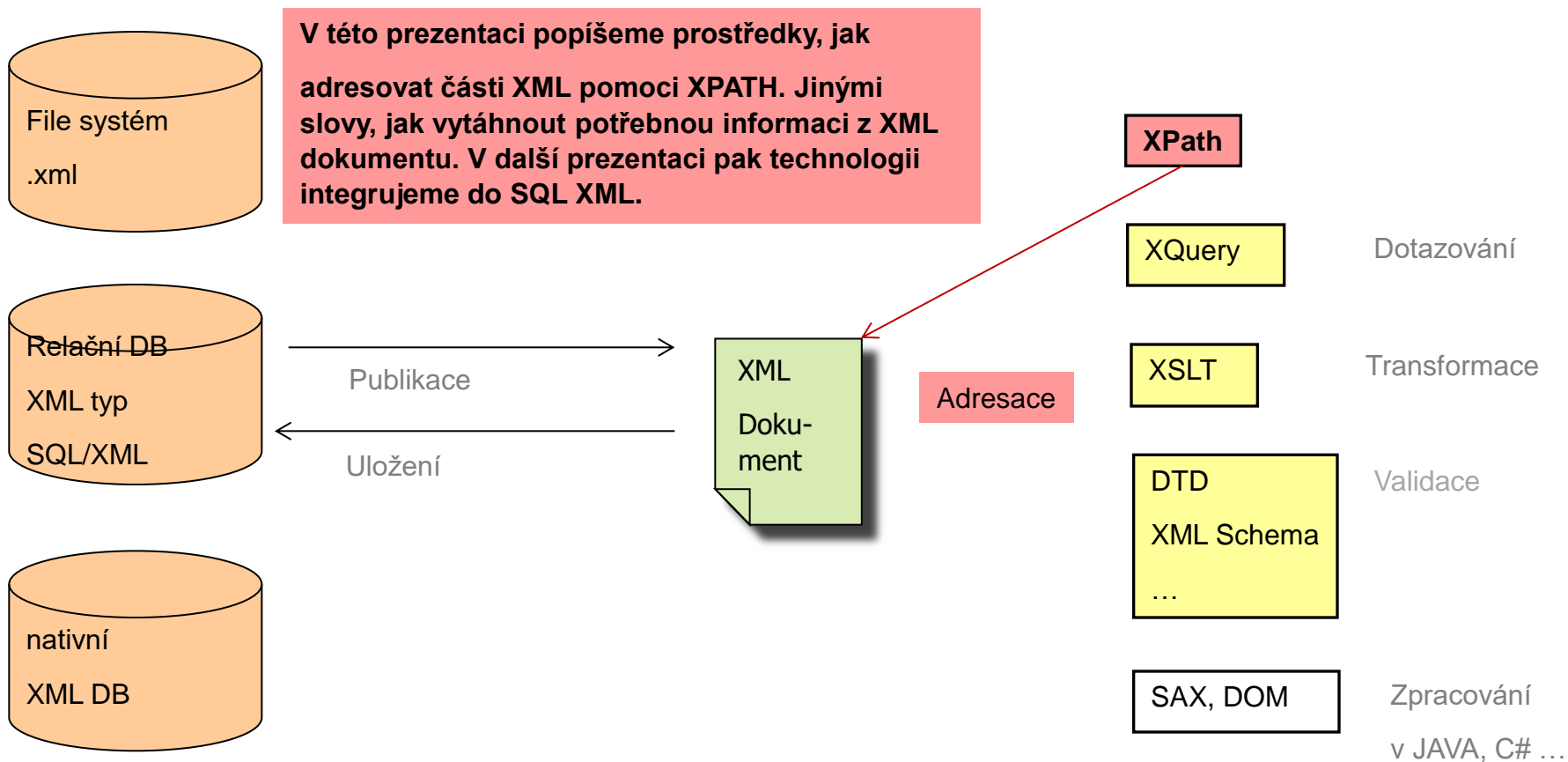
Jiří Měska (jiri.meska@gmail.com)

MIB008, Datové a procesní modely

MFF UK Praha, 2020

## Perzistence XML dat

## Zpracování XML dat



# Adresace v XML

```
<?xml version="1.0" encoding="windows-1250"?>
<panovnici>
  <panovník rod="přemyslovec">
    <jmeno>Vratislav II</jmeno>
    <tituly>
      <titul>král český</titul>
      <titul>správce hradu</titul>
    </tituly>
    <panoval titul="král český">
      <od>1061</od>
      <do>1092</do>
    </panoval>
    <smrt></smrt>
  </panovník>
```

```
<panovník rod="přemyslovec">
  <jmeno>Přemysl Otakar I.</jmeno>
  <tituly>
    <titul>král český</titul>
  </tituly>
  <panoval titul="král český">
    <od>1192</od>
    <do>1193</do>
  </panoval>
</panovník>
</panovnici>
```

## Odpověď pomocí XPATH:

```
/panovnici/panovník[2]/@rod
```

```
/panovnici/panovník[1]/tituly/titul/text()
```

**Problém:** Hledáme způsob se odkázat na:

- text „**přemyslovec**“ v atributu rod druhého krále?
- text „**král český**“, který je obsahem elementu titul prvního krále

# XPath: Základní charakteristika technologie

- XPath je jazyk, pomocí kterého lze adresovat části XML dokumentu.
- Jazyk XPath se obvykle používá jako adresovací jazyk v XML technologiích jako XSLT, XQuery, XML/SQL .... Při práci s XML je přirozené odkazovat na data uložená v XML pomocí metainformace, tj. struktury stromu a názvů elementů a atributů.
- Forma zápisu XPath výrazů nepoužívá XML a je navržena tak, aby mohla být hodnotou uložena jako atribut v XML, element v XML nebo URI.
- Poskytuje základní prostředky pro práci s výrazy typu
  - znakový řetězec,
  - číslo
  - logická hodnota
- Jazyk XPath podporuje jmenné prostory, tj. umí vyhledávat s požadovaným jmenným prostorem

# Infoset: Datový model well-formed XML dokumentu

Specifikace jazyk XPath je definována v termínech abstraktního datového modelu XML, který se dívá na XML jako na stromovou strukturu uzlů.

Datový model použitý standardem XPath 1.0 vychází ze specifikace „XML Infoset“, lze najít na stránkách konsorcia W3C ([www.w3c.org/TR/xml-infoset](http://www.w3c.org/TR/xml-infoset)).

Model se dívá na XML jako na strom, jehož uzly jsou: **dokument jako celek, elementy, atributy, texty, procesní instrukce, komentáře, jmenný prostor.**

Model definuje vlastnosti jednotlivých uzlů.

Například uzel elementu a atributu má jméno a textovou hodnotu.

Abstraktní model si ukážeme na následujícím jednoduchém XML dokumentu:

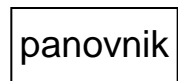
```
<?xml version="1.0" encoding="windows-1250"?>
<panovník rod="přemyslovec">
  <jmeno>Přemysl Otakar II.</jmeno>
  <tituly>
    <titul>král český</titul>
  </tituly>
</panovník>
```

# Infoset: Datový model XML dokumentu

## Legenda:



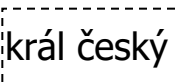
... Uzel dokumentu



... Uzel elementu



... Uzel atributu

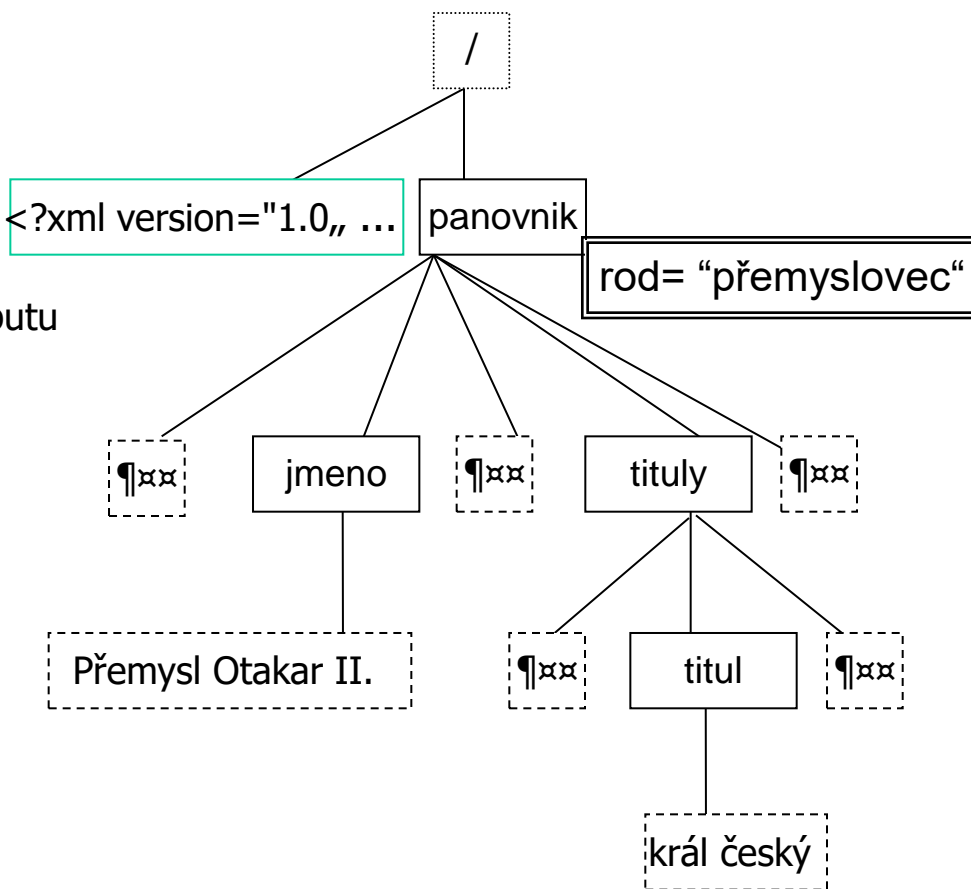


... Textový uzel



... Uzel instrukce

## Stromová struktura dokumentu:



## Infošet: typy uzlů (=note, =information item)

**Uzel dokumentu** – vstupní bod dokumentu, označuje se /. Uzel dokumentu má vždy alespoň jednoho potomka, v našem případě element `<panovník>` a instrukci pro zpracování `<?xml...`

Pozor! Neplést si s **kořenovým elementem** XML dokumentu, kterým je uzel `<panovník>` .

**Uzel elementu** – popisuje element dokumentu, např. `<panovník>`, `<jmeno>` ..., má vlastnosti:

**jméno uzlu elementu**

**textovou hodnotu uzlu elementu**

Potomkem uzlu element mohou být: **jiné elementy, textové uzly nebo komentáře.**

K uzlu elementu mohou být připojeny další uzly: **atributy, jmenný prostor**

**Textovou hodnotou** uzlu element je zřetězení všech bezprostředních potomků, které jsou textovými uzly

**Uzel atributu** – připojen vždy k nějakému uzlu elementu. Jeho rodičem je element, který atribut obsahuje. Uzel atributu má

**jméno uzlu atributu**

**hodnotu uzlu atributu**

## Infošet: typy uzlů

**Textový uzel** – Koncové listové uzly, ve kterých jsou uloženy textové hodnoty elementů.

Případné odřádkování je převedeno na <LF>, XML zápis je &#xA.

Pokud se v XML objeví entita &lt;, je nahrazena znakem <.

Speciálními textovými uzly jsou ¶|xx, ¶|. Představují odřádkování mezi elementy. Je třeba na ně brát zřetel,

**Uzel komentáře** – Vždy koncový uzel bez názvu.

Obsahem je komentář. <!-- xxx -->

**Uzel instrukce pro zpracování** – Vždy koncový uzel.

Názvem je cíl instrukce, v našem případě XML.

**Uzel jmenného prostoru** - Připojen vždy k nějakému uzlu elementu. Jeho rodičem je element, který je do jmenného prostoru zařazen.



# XPath: Adresace pomocí cesty dokumentem

Jazyk XPath popisuje adresaci v XML dokumentu pomocí cesty infosestem dokumentu.

**Cesta se logicky skládá z jednotlivých skoků:** jednotlivé skoky jsou odděleny lomítkem:

`/skok1 /skok2 /skok3`

`/child::panovnici/child:panovnik[2]/@rod`

XPath vyhodnocuje skoky postupně zleva doprava.

Každý další skok je vyhodnocen na všechny výsledky vyhodnocení předchozích skoků.

**Skok je tvořen:**

`/child::panovnik[2]`

**Identifikátorem osy** – který říká, kterým směrem postupujeme,

v našem případě k potomkům: `child ... (všechny děti)`

**Výběrem uzlu (test uzlu)**, zde vybereme uzel na označené ose,

v našem případě provedeme výběr zadáním jména: `panovnik ... (všechny panovníky)`

**Množinou predikátů** (podmínek), zde zadáme další podmínky pro výběr

v našem případě druhý v pořadí: `[2] ... (druhý panovnik ze všech panovníků)`

**Syntaxe skoku:**

`Identifikátor_osy :: test_uzlu [predikat1][predikat2]... [predikatN]`

**Kolika skoky je tvořen následující XPath?**

`/child::panovnici/child::panovnik[2]/child::tituly/child::titul/child::text()`

## XPath – identifikátor osy výběru

Xpath definuje 12 os výběru vzhledem k zvolenému aktuálnímu uzlu uvnitř dokumentu

1. **Self::** (aktuální), označuje sebe
2. **Parent::** (rodič). Označuje vždy jediného rodiče
3. **Ancestor::** (předci). Označuje předky od rodiče výše. Pořadí od rodiče ke kořenovému elementu.
4. **Ancestor-or-self::** (aktuální a předci).
5. **Child::** (děti). Označuje přímé potomky
6. **Descendant::** (potomci). Označuje všechny potomky (děti, vnuky, pravnuky ...)
7. **Descendant-or-self::** (aktuální a potomci)
8. **Preceding-sibling::** (předchozí sourozenci). Sourozenci od aktuálního, tedy v opačném pořadí než v XML.
9. **Preceding::** (předchůdci): preceding-sibling včetně svých potomků
10. **Following-sibling::** (následní sourozenci): Sourozenci od aktuálního, ve stejném pořadí jako v XML.
11. **Following::** (následníci): following-sibling včetně svých potomků
12. **Attribute::** (atributy aktuálního uzlu)

```

<A>
  <AB1>
    <AB1C1/>
    <AB1C2>
      <AB1C2D1/>
      <AB1C2D2/>
      <AB1C2D3>
        <AB1C2D3E1>
          <AB1C2D3E1F1/>
        </AB1C2D3E1>
        <AB1C2D3E2/>
      </AB1C2D3>
      <AB1C2D4>
        <AB1C2D4E1/>
      </AB1C2D4>
    </AB1C2>
    <AB1C3/>
  <AB1>
</A>

```

## Příklad

Osy výběru vzhledem k aktuálnímu uzlu <AB1C2D3>

1. **Self::** (aktuální)  
AB1C2D3
2. **Parent::** (rodič).  
AB1C2
3. **Ancestor::** (předci).  
AB1C2, AB1, A
4. **Ancestor-or-self::** (aktuální a předci):  
AB1C2D3, AB1C2, AB1, A
5. **Child::** (děti).  
AB1C2D3E1, AB1C2D3E2
6. **Descendant::** (potomci).  
AB1C2D3E1, AB1C2D3E1F1, AB1C2D3E2

```

<A>
  <AB1>
    <AB1C1/>
    <AB1C2>
      <AB1C2D1/>
      <AB1C2D2/>
      <AB1C2D3>
        <AB1C2D3E1>
          <AB1C2D3E1F1/>
        </AB1C2D3E1>
        <AB1C2D3E2/>
      </AB1C2D3>
      <AB1C2D4>
        <AB1C2D4E1/>
      </AB1C2D4>
    </AB1C2>
    <AB1C3/>
  <AB1>
</A>

```

## Příklad

Osy výběru vzhledem k aktuálnímu uzlu <AB1C2D3>

7. **Descendant-or-self::** (aktuální a potomci):  
AB1C2D3, AB1C2D3E1, AB1C2D3E1F1, AB1C2D3E2
8. **Preceding-sibling::** (předchozí sourozenci):  
AB1C2D2, AB1C2D1
9. **Preceding::** (předchůdci):  
AB1C2D2, AB1C2D1
10. **Following-sibling::** (následní sourozenci):  
AB1C2D4
11. **Following::** (následníci):  
AB1C2D4, AB1C2D4E1
12. **Attribute::** (atributy aktuálního uzlu):  
empty

# Formy syntaxe XPath

XPath definuje plnou a zkrácenou syntaxi:

Chceme vybrat titul druhého panovníka: **král český**

**Plná syntaxe:** `/child::panovnici/child::panovník[2]/child::tituly/child::titul/child::text()`

**Zkrácená syntaxe:** `/panovnici/panovník[2]/tituly/titul/text()`

Chceme vybrat rod druhého panovníka: **přemyslovec**

**Plná syntaxe:** `/child::panovnici/child::panovník[2]/attribute::rod`

**Zkrácená syntaxe:** `/panovnici/panovník[2]/@rod`

Zkrácená syntaxe předpokládá, že pokud nic nevedeme, jedná se o bezprostředního potomka, tj **child::**.

Přístup na atribut `attribute::jmeno` zkracujeme na `@rod`

# XPath: přehled zkrácené syntaxe

XPath umožňuje zkrácenou syntaxi pro nejčastěji používané osy.

Pokud není nic uvedeno, např /panovník

Pokud je uvedeno @, např /panovník/@rod

Pokud je uvedeno //, např //panovník

Pokud je uvedeno ., např //panovník[./panoval/od<1200]

Pokud je uvedeno .., např //panovník/panoval/../jmeno

osa **child::**

osa **attribute::**

osa **descendant::**

osa **self::**

osa **parent::**

## XPath: Výběr uzlu

### Výběr uzlu může být test na jméno elementu:

- Jméno elementu
- namespace:jmeno elementu
- \* nebo namespace:\* (všechny elementy)

### Výběr uzlu může být test na typ uzlu:

- comment()
- text()
- processing-instruction()
- node() – všechny uzly
- element() – všechny elementy

Co je výsledkem následujících výrazů?

`/child::panovnici/child::panovnik[1]`

`/child::panovnici/child::panovnik[1]/child::jmeno/child::text()`

`/child::processing-instruction()`

`/child::panovnici/child::comment()[2]`

`/child::panovnici/child::panovnik[1]/attribute::rod`

```
<?xml version="1.0" encoding="windows-1250"?>
<panovnici>
<!-- vloženo 1.1.2000 -->
<!-- opraveno 1.1.2001 -->
  <panovnik rod="přemyslovec">.
    <jmeno>Vratislav II</jmeno>
    <tituly>
      <titul>král český</titul>
    </tituly>
    <panoval titul="král český">
      <od>1061</od> <do>1092</do>
    </panoval>
    <smrt></smrt>
  </panovnik>
```

## XPath: Predikát

U každého skoku může být uveden jeden nebo více predikátů, které snižují počet vybraných uzlů.

Predikáty se zapisují do hranatých závorek [] a vyhodnocují se postupně tak jak jsou zapsány. Pokud je vyhodnocení true, uzel zůstává ve množině uzlů, které vyhovují cestě, jinak z ní vypadá.

Při vyhodnocování predikátů je **aktuálním uzlem** uzel, pro který se predikát vyhodnocuje. Například `panovník[2]` se predikát [2] vyhodnocuje vzhledem k aktuálnímu uzlu `panovník`

Interpretace logické hodnoty v závislosti na typu výrazu uvedeného v predikátu, např:

- Číselná hodnota – číslo se interpretuje jako pozice v množině uzlů na ose, po které se pohybujeme (`/panovníci/panovník[1]` )
- Množina uzlů – Pokud je výsledkem množina uzlů, potom je hodnota true právě tehdy, když je neprázdná (`/panovníci/panovník[ smrt ]`)

(Poznámka: `smrt` je odkaz na podelement `panovník`)

Příklady predikátů:

```
/panovníci/panovník[1]
```

```
/panovníci/panovník/panoval[ od<1192 ]
```

```
/panovníci/panovník[ count(tituly/titul)>1 ]
```

```
/panovníci/panovník[ smrt ]
```



# XPath: Absolutní a relativní adresace

## Cesta je:

### Absolutní adresace:

Adresace vzhledem k uzlu dokumentu.

Absolutní cesta vždy začíná uzlem dokumentu /.

### Relativní adresace:

Adresace se vztahuje se k nějaké pozici v dokumentu.

např. na pozici uzlu panovnici je možno odkázat na panovníka: panovník[1]

## Příklady:

Příklad relativní cesty tvořené jedním skokem:

**child::panovník[1]**

Příklad absolutní cesty tvořené dvěma skoky:

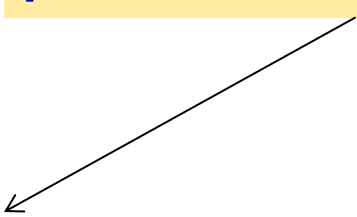
**/child::panovnici/child::panovník[1]**

Kombinovaný příklad

**/child::panovnici/child::panovník[**child::panoval/child::od<1620**]**

vybere všechny uzly panovník a to panovníků,  
kteří započaly vládu před rokem 1620

**child::panoval/child::od<1620**  
je relativní cesta vzhledem k uzlu  
**panovník**



# XPath: Návrátová hodnota XPath

## Návrátovou hodnotou může být (viz dále výraz):

jednoduchá hodnota

elementární textová hodnota

uzel (element, procesní instrukce, poznámka)

logická hodnota

sekvence (uspořádaný seznam jednoduchých hodnot)

**`/child::panovnici/child::panovník[child::panoval/child::od<1620]`**

## Příklady návratových hodnot (absolutní adresace)

Příklady jsou vztaženy vzhledem k ilustračnímu XML

Plná syntaxe	Zkrácená syntaxe	Typ návratové hodnoty Hodnota
/child::panovnici/ <b>child::panovník</b>	/panovnici/ <b>panovník</b>	sequence elementů (panovník[1], panovník[2])
/child::panovnici/ <b>child::panovník</b> [2]	/panovnici/ <b>panovník</b> [2]	Element panovník[2]
/child::panovnici/ <b>child::</b> panovník[2]/ <b>child::jmeno/</b> <b>child::text()</b>	/panovnici/ <b>panovník</b> [2]/ <b>jmeno/</b> <b>text()</b>	Text hodnota elementu Přemysl Otakar I
/child::panovnici/ <b>child::panovník</b> [2]/ <b>attribute::rod</b>	/panovnici/ <b>panovník</b> [2]/ <b>@rod</b>	textová hodnota atributu přemyslovec
/child::panovnici/ <b>child::node()</b>	/panovnici/ <b>node()</b>	Sequence uzlů a textů: (text()[1], panovník[1], text()[2], panovník[2] text()[3])

## Příklady návratových hodnot (relativní adresace)

Vzhledem k ilustračnímu XML. Relativní cesta vzhledem k uzlu: **/panovnici/panovník[2]**

Plná syntaxe	Zkrácená syntaxe	Typ návratové hodnoty Hodnota
child::jmeno/child::text()	jmeno/text()	Textová hodnota Přemysl Otakar I
attribute::rod	@rod	Textová hodnota přemyslovec
child::panoval/child::od/child::text() )	panoval/od/text()	Textová hodnota (číslo) 1192
self::node()	.	uzel elementu panovník[2]
parent::node()	..	Uzel elementu panovnici
parent::node()/child::panovník[1]	../panovník[1]	Uzel elementu panovník[1]

# XPath syntaxe: Výraz

Výraz je nejobecnější XPath konstrukce.

XPath cesta je sama výrazem.

Výraz se může vyskytnout v predikátu (tj. v hranatých závorkách).

Přesná syntaxe je nad rámec této přednášky.

XPath výraz může vrátet hodnotu jednoho z následujících typů:

- Logická hodnota
- Číslo
- Textový řetězec
- Množina uzlů
- Část stromu vstupního dokumentu, kterou lze podle potřeby vložit do výstupu (tento typ přidává až XSLT)

# XPath: Operátory

Při specifikaci cesty se mohou v predikátech (hranaté závorky) vyskytovat výrazy s operátory:

<p>Operátor sjednocení (operátor mezi výrazy typu sequence uzlů, tj. může se vyskytnout i mezi cestami)</p>	<p> </p>	<p>/panovnici/panovnik[ jmeno='Vratislav II' ]   /panovnici/panovnik[ jmeno='Přemysl Otakar I.' ]</p>
<p>Logické operátory</p>	<p><b>not(), or, and</b></p>	<p>/panovnici/panovnik[ <b>jmeno='Vratislav II' or jmeno='Přemysl Otakar I.'</b> ]</p>
<p>Relační operátory</p>	<p><b>=, !=,</b> <b>&gt;, &lt;,</b> <b>&gt;=, &lt;=</b></p>	<p>/panovnici/panovnik/panoval[ <b>od&lt;1192</b> ] Začal panovat před rokem 1192</p>
<p>Matematické operátory</p>	<p><b>+, -, *,</b> <b>sum(), count(),</b> <b>mod</b></p>	<p>/panovnici/panovnik[panoval[ <b>(number(do)-number(od))&gt;20</b>]] Panoval více než 20 let</p>

# XPath: Funkce

Součástí XPath jsou zabudované funkce:

- Logické funkce
- Funkce pro práci s čísly
- Funkce pro práci se znakovými řetězci
- Funkce pro práci s množinami uzlů
- Agregační funkce

Podrobnější informace viz dokumentace.

## XPath: Logické funkce (návratová hodnota boolean)

	boolean boolean(object)	Konverze do logické hodnoty. Číslo je true tehdy a jen tehdy, když je různé od 0. Množina uzlů je true tehdy a jen tehdy, když je neprázdná. Textový řetězec je true tehdy a jen tehdy, když delka>0

Poznámka: Implicitně jsou logické funkce aplikovány na výsledek vyhodnocení predikátu.



## XPath: Funkce pro práci s čísly (návrátová hodnota number)

number ceiling(number)	Vrací nejmenší celé číslo větší než argument. ceiling(-4.5) = -4
number floor(number)	Vrací největší celé číslo menší než argument. ceiling(-4.5) = -5
number number(object?)	Konverze argumentu na číslo. Logická hodnota: true je 1, false je 0 Znakový řetězec je konvertován na číslo, pokud jde o whitespace následovaný znaménkem a číslem, jinak je konvertováno na NaN. Množina uzlů je konvertována nejprve na znakový řetězec a potom na číslo.
number round(number)	Zaokrouhlení na nejbližší celé číslo
number sum(node-set)	Summa přes všechny hodnoty uzlů – u každého uzlu proběhne konverze na číslo

# XPath: Funkce pro práci s čísly

Poznámka: XPath odkazuje na standard IEEE 754, který definuje floating point aritmetiku.

Standard zahrnuje definice:

- záporné a kladné nekonečno
- NaN (not a number). NaN reprezentuje výsledek některých operací, např. dělení 0.

Funkce v následujících tabulkách jsou popsány:

***<typ návratové hodnoty> <jméno\_funkce>( <parametry> )***

Parametry jsou popsány jako seznam typů parametrů. Pokud je uvedena hvězdička, mohou se opakovat.

## XPath: Funkce pro práce se znakovými řetězci I

string concat(string, string, string*)	Zřetězení dvou a více znakových řetězců
boolean contains(string, string)	Test, zdali první řetězec obsahuje druhý. Porovnání je case sensitive.
string normalize-space( string?)	Funkce ořízne úvodní a koncové whitespace (mezera, LF,CR, tab) a vnitřní posloupnosti whitespace nahradí mezerou. Pokud není specifikován, je implicitně nahrazen textem aktuálního kontextového uzlu.
boolean start-with(string, string)	Test, zdali první řetězec začíná druhým. Porovnání je case sensitive.
string string(object?)	Konverze na string.  Jde-li o množinu uzlů, vrací se text prvého z uzlů ve výskytu v dokumentu.  Číselné hodnoty neobsahují vedoucí nuly. NaN je „NaN“  Logické hodnoty true a false jsou konvertovány do „true“ a „false“

## XPath: Funkce pro práce se znakovými řetězci II

number string-length(string?)	Počet znaků v řetězci. Pokud není specifikován, je implicitně nahrazen textem kontextového uzlu.
string substring(string, number, number?)	Vybírá znakový podřetězec z pozice dané druhým argumentem včetně. Délka je volitelným parametrem. Pozice ve znakovém řetězci jsou 1,2...
string substring-after(string, string)	Vrací znakový řetězec z prvního argumentu následující první výskyt druhého argumentu
string substring-before(string, string)	Vrací znakový řetězec z prvního argumentu předcházející první výskyt druhého argumentu
string translate(string, string, string)	Ve znakovém řetězci prvního argumentu se nahradí veškeré výskyty znaků druhého argumentu odpovídajícími znaky třetího argumentu. Pokud třetí argument je kratší, potom ty znaky, které nemají odpovídající znak v třetím argumentu jsou odstraněny. Využívá se například ke konverzi na velká písmena

Co je výsledkem?

substring( „0123456789“, 3 )

Co je výsledkem?

translate(„++420-2-61911213“, „+-“, „0“)

## XPath: Funkce pracující s množinou uzlů

	number count(node-set)	Vrací počet uzlů v množině uzlů. Příklad: count(//panovník) vrací počet elementů panovník
	number last()	Vrací velikost kontextu v jehož rámci je funkce vyhodnocována. Příklad: //panovník[last()] vrací posledního panovníka dokumentu
	string local-name(node-set?)	Vrací lokální část jména prvního uzlu.
	string name(node-set?)	Vrací kvalifikované jméno prvního uzlu, tj. jmenný prostor (Alias+URI)+lokální jméno
	string namespace-uri(node-set?)	Vrací jmenný prostor prvního uzlu
	number position()	Vrací pro vyhodnocovaný uzel jeho pozici v kontextu. Příklad: //panovník[2] je totéž jako //panovník[position()=2]

## XPath: Agregáční funkce

max	max((arg,, arg..))	Vrací největší argument Example: max((1,2,3)) Result: 3 Example: max(('a', 'k')) Result: 'k'
min	min((arg,, arg..))	Vrací nejmenší argument Example: min((1,2,3)) Result: 1 Example: min(('a', 'k')) Result: 'a'
sum	sum((arg,, arg..))	Vrací součet numerických hodnot
avg	avg((arg,, arg..))	Vrací průměr Example: avg((1,2,3)) Result: 2

# Využití XPath

**XQuery** - dotazovací jazyk vzhledem k XML

**SQL/XML** – rozšíření SQL o prostředky práce s XML

**xmlexist()** funkce

**xpath()** funkce

**XSLT** – jazyk pro popis transformací XML do XML, text aj.

# XPath výraz se zadáním zdrojového XML:

Následující vstupní funkce jsou součástí definice XPath

**doc()** – vrací dokument identifikovaný URL, přesněji uzel dokumentu (document node)

**collection()** – vrací kolekci, která je sekvencí uzlů, z nichž každý je identifikován URL

Co vrací následující XPath výrazy?

```
doc("panovnici.xml")/panovnici/panovnik
```

```
doc("panovnici.xml")//panovnik
```

**Jaká je podmínka, aby předchozí dva výrazy vracely stejnou množinu uzlů?**

```
doc("panovnici.xml")//panovnik/@rod
```

```
doc("panovnici.xml")//panovnik[@rod="přemyslovec"]
```

```
doc("panovnici.xml")//panovnik[jmeno="Vratislav II"]/panoval/od/text()
```



## Technologie pro práci s XML a jejich standardizace

- **XML (Extensible Markup Language)**– Obecný značkovací jazyk vyvinutý a standardizovaný konsorciem W3C. Umožňuje snadné vytváření odvozených značkovacích jazyků (aplikací) pro široké spektrum účelů a dat.
- **XPath (XML Path Language)** – Jazyk pomocí kterého lze adresovat části XML. Pomocí XPath lze vybírat elementy, atributy. XPath je standardem W3C.
- **XSLT (Extensible Stylesheet Language Transformation)** – Deklarativní jazyk o odvozený z XML umožňující popsat transformaci XML dokumentů. Nejčastěji používaný pro konverzi XML dat opět do XML, HTML, XHTML, TXT nebo PDF.
- **XQuery (XML Query Language)**. Jazyk určený pro dotazy na data obsažená v XML souborech, „obdoba SQL vzhledem k relační databázi“

XPath je společný technologie pro adresaci v XML pro XSLT, XQuery

