

GeoWeb: Značkovací jazyky pro geomatiku

Otakar Čerba

poslední aktualizace: 7. 11. 2017

Osnova

- ▶ Základy značkovacích jazyků (koncept, historie...)
- ▶ Extensible Markup Language (XML)
- ▶ Značkovací jazyky v geomatice
 - ▶ Datové formáty
 - ▶ Schémové jazyky
 - ▶ Transformační a stylové jazyky

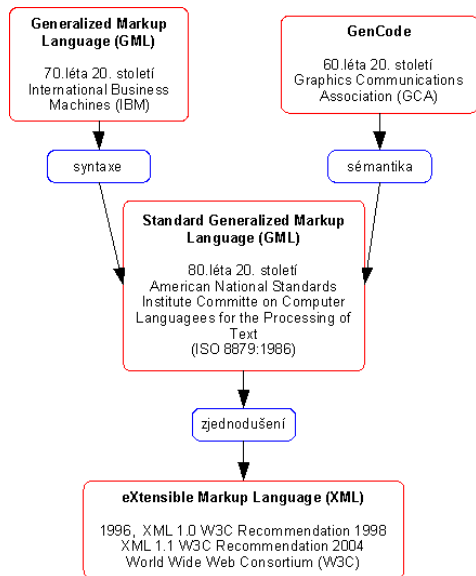
Značkovací jazyky – diskuze

- ▶ Jak je jejich základní princip?
- ▶ Proč a kdy vlastně vznikly?

Značkovací jazyky

- ▶ Markup languages
- ▶ Metajazyky, které označují význam jednotlivých částí dokumentů a nikoli jejich vzhled
- ▶ Tzv. samopopisné jazyky – kromě vlastního dokumentu dokáží popsat i jeho strukturu

Stručná historie značkovacích jazyků



Extensible Markup Language

- ▶ 1996 (začátek aktivit) – 1998 (první standard)
- ▶ Standard W3C
- ▶ Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation 26 November 2008
- ▶ XML 1.1 (Second Edition), W3C Recommendation, 16 August 2006
- ▶ Namespaces in XML 1.1 (Second Edition), W3C Recommendation, 16 August 2006

Co vlastně je XML?

Charles F. Goldfarb* označil XML za „svatý grál“ počítačových technologií, který řeší problém univerzální výměny dat mezi různými systémy.

*Otec SGML (Standard Generalized Markup Language) a dědeček HTML

Co vlastně je XML?

Podle Erica Raymonda* (2001) můžeme XML chápat ve třech úrovních.

1. Jako protokol pro uchovávání a správu informací.
2. Jako rodinu technologií, které dokáží vše od formátování dokumentů až po filtrování dat.
3. Jako XML filosofii pro výměnu informací, která poskytuje maximální použitelnost a flexibilitu a zároveň striktní strukturovanost dat.

*Autor knihy Katedrála a tržiště (modely vývoje open source software), správce The New Hacker's Dictionary. . .

Výhody XML

- ▶ Jednoduchost
- ▶ Otevřenost
- ▶ Forma zápisu
- ▶ Čtení a porozumění
- ▶ Propojení mezi XML a jinými aplikacemi
- ▶ Modifikace XML podle potřeb uživatele
- ▶ Malé množství pravidel pro práci s XML soubory

Nevýhody XML

- ▶ Zápís značkovacích jazyků pomocí tagů je komplikovaný a často poměrně zdlouhavý
- ▶ Soubory XML mají ve srovnání s binárními formáty nebo formátem JSON větší velikost
- ▶ Nadbytečná komplexnost
- ▶ Nutnost striktního dodržování standardů
- ▶ Pro používání XML je nutné plné porozumění základům

Součásti XML dokumentu

- ▶ Tagy
- ▶ Elementy
- ▶ Atributy
- ▶ Znakové a textové entity
- ▶ CDATA – dlouhé texty se znakovými entitami (kódy...)
- ▶ Komentáře
- ▶ Procesní instrukce – připojení stylů, skriptů...
- ▶ Hlavička dokumentu

Součásti XML dokumentu

```
<?xml version="1.0"?>  
<sbirka_map>  
  <mapa id="Arentin1619">  
    <nazev>Regni Bohemia nova et exacta descriptio</nazev>  
    <meritko>1:504 000</meritko>  
  </mapa>  
  <mapa id="Vetter1668">  
    <nazev>Bohemiae Rosa</nazev>  
    <meritko>1:1 268 450</meritko>  
  </mapa>  
</sbirka_map>
```

atribut

element

obsah elementu

počáteční tag

koncový tag

XML tagy

- ▶ Tag – značka pro strukturování XML dokumentu
- ▶ Tagy jsou uzavřeny do ostrých závorek
- ▶ Počáteční a koncové tagy – koncový tag obsahuje znak „lomítko” (/) bezprostředně před názvem tagu
- ▶ Jména tagů (a obecně jména v XML)
 - ▶ musí začínat písmenem nebo podtržítkem
 - ▶ mohou obsahovat čísla, tečky, dvojtečky a pomlčky, ale nikoli mezery
 - ▶ kromě anglické abecedy je možné využívat i tzv. ideografických znaků, do kterých patří i znaky české abecedy
 - ▶ XML je case-sensitive

XML elementy

- ▶ Základní stavevní kámen XML
- ▶ Ohraničeny tagy (na rozdíl od HTML jsou vyžadovány počáteční i koncové tagy)
- ▶ Zkrácený zápis pro element bez obsahu – vypustí se koncový tag a lomítko se doplní za jméno elementu
- ▶ Jméno elementu by mělo být voleno s ohledem na význam prvku (sémantika)
- ▶ **Celý XML dokument je uzavřen do jediného kořenového elementu**
- ▶ **Jednotlivé elementy musí být do sebe vnořené a nesmí se křížit**

Atributy v XML

- ▶ Doplňková informace k elementům – element může obsahovat i více různých atributů
- ▶ Zapisují se do počátečního tagu elementu ve tvaru jméno atributu, rovnítko a hodnota atributu zapsaná do uvozovek nebo apostrofů
- ▶ V jednom elementu se nesmí vyskytovat více stejných atributů

```
<odstavec autor="J.Novák" software="XML Mind">  
    Text odstavce...  
</odstavec>
```

Komentáře (nejen) v XML

- ▶ Zapisují se mezi znaky `<!--` a `-->`
- ▶ Nejsou součástí kódu = nejsou zpracovávány
- ▶ Nesmí být součástí ostatního značkování (např. nelze například „zakomentovat“ atributy)
- ▶ Neměly by obsahovat dvojici po sobě jdoucích znaků – (spojovník)

Znakové entity

- ▶ Některých znaky (například <, >, &) není možné do XML zapsat (součást zápisu značkování) → znakové entity.
- ▶ Příklady: <, >, &, ', " (možnost používání uvozovek i apostrofů zároveň)
- ▶ Jako znakovou entitu lze vložit libovolný znak pomocí Unicode kódu – před vlastní kód je nutné umístit prefix
 - ▶ decimální kód – prefix #
 - ▶ hexadecimálního kódu – prefix #x

Hlavička dokumentu / XML deklarace

- ▶ První řádka XML dokumentu
- ▶ Ohraničená ostrými závorkami a otazníky
- ▶ Klíčové slovo `xml` a povinné deklarace verze XML – `version`
- ▶ Nepovinné prvky
 - ▶ Použité kódování – `encoding` (nemusí se používat pokud je soubor v UTF-8)
 - ▶ Parametr `standalone` – možnost používání externích souborů (`yes` = defaultní hodnota / `no`)
- ▶ Atributy hlavičky XML dokumentu musí být přesném pořadí

Jmenné prostory / XML Namespaces

- ▶ Jeden dokument – více typů značení (více značek pro elementy a atributy definovaných v různých schématech)
- ▶ Seskupení příbuzných prvků XML dokumentů z důvodů snazšího rozeznávání

```
xmlns:prefix="URI sady značek"
```

```
...xmlns:svg="http://www.w3.org/2000/svg"  
xmlns:mapa="http://www.mojemapa.net/mapa"...
```

```
<svg:circle cx="0" cy="0" r="100px"/>  
<mapa:linie meta:titulek="Linie1">  
...nějaký obsah elementu...  
</mapa:linie>
```

Automatická kontrola XML dokumentů

- Správně formátovaný dokument (well-formed document) splňuje veškeré náležitosti XML dokumentu (správné vnořování prvků, hlavička. . .) → **syntaxe** → **parser**
- Platný (validní) dokument (valid document) odpovídá příslušnému souboru zapsaném v jazyce pro popis dokumentů → **syntaxe + sémantika** → **validátor**

XML vs. JSON

- ▶ Oba formáty jsou
 - ▶ samoopisné
 - ▶ hierarchické
 - ▶ parsovatelné
 - ▶ dostupné přes XMLHttpRequest
- ▶ JSON – kratší (nemá koncový tag), rychlejší na zpracování, používá pole
- ▶ XML – další součásti – transformační jazyk, schémový jazyk, dotazovací jazyk

Proč XML v geomatice?

- ▶ Pro většinu operací s prostorovými daty se používá jeden druh technologií
- ▶ Otevřenost, ze které vyplývají široké možnosti modifikace, integrace do vlastních aplikací a harmonizace datových sad. . .
- ▶ XML technologie jsou běžně používány v řadě oborů
- ▶ XML je standardizované, stabilní a léty prověřené → nehrozí náhlý zánik nebo prudké změny
- ▶ Vytvořené styly, šablony a algoritmy lze použít vícekrát pro různá data
- ▶ Existuje velké množství datových souborů přímo v XML
- ▶ (Obvykle) není potřeba žádný speciální komerční software
- ▶ Na internetu je zdarma k dispozici velké množství návodů, tutoriálů, šablon, specifikací, diskuzních fór a emailové konference

Aplikace XML v geomatice

- ▶ Ukládání a kódování dat (prostorová data, vektorové grafické formáty. . .)
- ▶ Popis dat a modelů (metadata, schémata, ontologie)
- ▶ Transformace (prostorových) dat, včetně vizualizace (transformační a stylové jazyky)

XML pro správu a ukládání dat

- ▶ Obecná data – XML ... RDF
- ▶ Hypertext – (X)HTML, DocBook, MathML...
- ▶ Vektorová grafika – SVG, VML, PGML, X3D...
- ▶ Multimédia – SMIL, VoiceML...
- ▶ Prostorová (geografická) data – GML, KML; G-XML, CityGML, IndoorGML, SOTF, LandXML, OMF, XMML, NVML...

Geography Markup Language (GML)

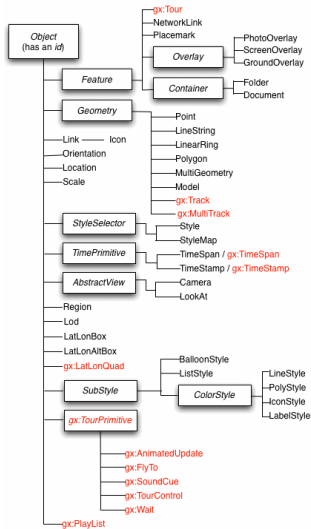
- ▶ První verze 1999, OGC standards (GML 3.3), ISO 19134
- ▶ Definováno pomocí mnoha XSD (geometrie, prvky, vazby...)
- ▶ Základ INSPIRE / WFS
- ▶ Velmi komplexní – základní typy objektů (bod, linie, polygon...) + kruhové oblouky, spline křivky, parametrické a Beziérovky křivky, Solid Geometry, TIN, grid...
- ▶ Topologie – 2D, 3D, časová
- ▶ Další prvky – affinní transformace, agregace...

GML – příklad (SPOI)

```
<gml:featureMember>
  <ogr:Týnec fid="Týnec.0">
    <ogr:geometryProperty>
      <gml:Point srsName="EPSG:4326">
        <gml:coordinates>13.2674117,49.3450856</gml:coordinates>
      <ogr:Název>Týnec č.p. 9|cs</ogr:Název>
      <ogr:Zeměpisná_délka>13.2674117</ogr:Zeměpisná_délka>
      <ogr:Zeměpisná_šířka>49.3450856</ogr:Zeměpisná_šířka>
      <ogr:Popis>Dům s popisným číslem 9 v Týnci u Klatov|cs</ogr:Popis>
      <ogr:Typ_objektu>tourism_attraction</ogr:Typ_objektu>
      <ogr:Stát>http://www.geonames.org/3077311</ogr:Stát>
      ...
      <ogr:Obec>http://www.geonames.org/3063812</ogr:Obec>
      <ogr:Adresa>Týnec 9, 340 21 Janovice na Úhlavou</ogr:Adresa>
      <ogr:Email xsi:nil="true"/>
      ...
      <ogr:What3words>contracting.sobs.regretfully</ogr:What3words>
    </ogr:Týnec>
  </gml:featureMember>
```

Keyhole Markup Language (KML)

- ▶ Jednoduchý jazyk
- ▶ Původně API pro virtuální glóbus of společnosti Keyhole, Inc.
→ Google (2004) → KML 2.2 Standard OGC (2008) → KML 2.3 (2015)
- ▶ Souřadnice – WGS 84, EGM96 (výšky)
- ▶ KML – XML syntaxe; KMZ – komprimovaný soubor, doplněný o další prvky (například textury, obrázky, ikony, 3D objekty – COLLADA)



Podle KML Reference

KML – minimální příklad

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>Týnec</name>
  <description>Týnec č.p. 9</description>
  <Point>
    <coordinates>13.2674117,49.3450856</coordinates>
  </Point>
</Placemark>
</Document>
</kml>
```

KML – příklad (SPOI)

```
<Placemark>
  <ExtendedData><SchemaData schemaUrl="#Týnec">
    <SimpleData name="Název">Týnec č.p. 9|cs</SimpleData>
    <SimpleData name="Zeměpisná délka">13.2674117</SimpleData>
    <SimpleData name="Zeměpisná šířka">49.3450856</SimpleData>
    <SimpleData name="Popis">Dům s popisným číslem 9 v
    <SimpleData name="Typ objektu">tourism_attraction</SimpleData>
    <SimpleData name="Stát">http://www.geonames.org/307
    <SimpleData name="Kraj">http://www.geonames.org/333
    <SimpleData name="Okres">http://www.geonames.org/306
    <SimpleData name="Obec">http://www.geonames.org/306
    <SimpleData name="Adresa">Týnec 9, 340 21 Janovice
      ...
    <SimpleData name="What3words">contracting.sobs.regi
  </SchemaData></ExtendedData>
  <Point><coordinates>13.2674117,49.3450856</coordinates>
</Placemark>
```

Ostatní XML datové formáty na přednáškách KGM/GW

- ▶ Scalable Vector Graphics (SVG) – pravděpodobně 5. prosince
- ▶ Resource Description Framework (RDF) a na něm založeném formáty – pravděpodobně 19. prosince, SGG
- ▶ 3D – Ing. Karel Jedlička, Ph.D.

Schémové jazyky

- ▶ Jazyky pro popis dokumentu / XML schémata / XML Schema Languages
- ▶ DTD (Document Type Definition)
- ▶ XML Schema
- ▶ RELAX NG (Regular Language Description for XML - Next Generation)

RELAX (Regular Language Description for XML), Schematron, TREX (Tree Regular Expressions for XML), XML-Data, XDR (XML-Data Reduced), DCD (Document Content Description), DSD (Document Structure Description), SOX (Schema for Object-oriented XML), DDML (Document Definition Markup Language, formerly XSchema), XDuce, DTD++, DTD4DTD (Data Types for DTD), Exampatron, Hook, NRL (Namespace Routing Language)...

Schémové jazyky

- ▶ Formální definice nového značkovacího jazyka = definování podmnožiny XML
- ▶ Definice
 - ▶ jednotlivých prvků XML dokumentu (elementy, atributy, entity...)
 - ▶ způsobů jejich použití (přípustné kombinace, omezení, datové typy, vzájemné vazby...)
- ▶ Schémata umožňují plné využití všech možností značkovacího jazyka, především kontroly dokumentu z hlediska sémantiky (syntaxi zajišťuje především parser)
- ▶ Kontrolu dokumentu z hlediska schématu zajišťují validátory
- ▶ Schémata mohou být chápány jako filtry chránící systémy a především uživatele před širokou různorodostí XML formátů
- ▶ Funkce metadat i datových modelů

Proč schémata?

- ▶ Schémata = modely
- ▶ Formální definice značkovacích jazyků
- ▶ Validace dokumentů
- ▶ Lepší manipulace s dokumenty prostřednictvím XML editorů
- ▶ Dokumentace pro značkovací jazyk
- ▶ Automatické vytěžování a zpracování dat → interoperabilita

Úrovně validace dokumentů

- ▶ Struktura – testování prvků XML dokumentu (elementů, atributů. . .), ale nikoli jejich obsahu
- ▶ Datové typy – vytvoření vazeb mezi značkováním a obsahem – kontroluje se obsah elementů a atributů bez ohledu na jejich vzájemné vazby
- ▶ Integritní omezení – kontrola všech vazeb (např. odkazů) uvnitř dokumentu nebo mezi dokumenty
- ▶ Business rules – kontrolují se omezení typu „datum narození musí být menší než datum úmrtí“ apod.

DTD

- ▶ Podpora SGML – není potřeba. . .
- ▶ Podpora velkého množství aplikací – klesá. . .
- ▶ Kompaktní, přehledné a čitelné – jak pro koho. . . – zápis je nestandardní
- ▶ Neodpovídá současným potřebám – bez jmenných prostorů, definice povoleného obsahu elementů, většiny datových typů
- ▶ Na DTD se nedají aplikovat transformační XSLT styly. . .

XML Schema

- ▶ W3C standard
- ▶ Výhody
 - ▶ Podpora datových typů a jmenných prostorů
 - ▶ Referenční integrita, zajištění klíčových hodnot
 - ▶ Objektová orientace, modularizace schémat, tvorba dokumentace schématu
- ▶ Nevýhody: složitá specifikace, komplexnost

RELAX NG

- ▶ Dvě formy zápisu – XML a RNC (kompaktní textová syntaxe)
- ▶ Výhody: jednoduchost, propojení s jazykem Schematron
- ▶ Nevýhody: slabá podpora v aplikacích

Jaký schémový jazyk

- ▶ DTD – nepotřebujeme-li využívat jmenné prostory a datové typy
- ▶ W3C XML Schema – potřebujeme-li pracovat se jmennými prostory a datovými typy a zároveň chceme využívat převážně komerční aplikace
- ▶ RELAX NG – potřebujeme-li pracovat se jmennými prostory a datovými typy a zároveň chceme využívat převážně open source aplikace

Existuje jen jedna validace?

- ▶ Dokumenty s více jmennými prostory → validace pomocí více schémat = pomocí různých schémových jazyků (pravděpodobně)
- ▶ Další uživatelské potřeby – kontrola gramatiky, „business rules“, datových typů, integrity mezi jednotlivými prvky dokumentu, znakových sad. . .

→ prostředí (rámec) pro komplexní validaci XML dokumentů

Document Schema Definition Languages (DSDL)

- ▶ <http://dSDL.org/> – stránka nefuguje
- ▶ Standard ISO – ISO/IEC 19757
- ▶ Prostředí a množina jazyků pro kontrolu kvality a správnosti XML dokumentů
- ▶ Typy kontrol – struktura dokumentu (RELAX NG), logická pravidla (Schematron), datové typy, mapování XML struktur. . .
- ▶ Hlavní přínos – využívání předností jednotlivých validačních technologií
- ▶ Nevýhoda – vypadá to, že proces vývoje DSDL ustrnul. . .

I přes výhody schémových jazyků se v praxi většinou vytváří jednoúčelové (aplikačně závislé) validátory často jako přímé součásti zpracování dat.

Rychlost, implementace uživatelských požadavků...

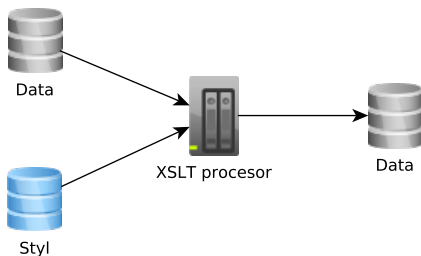
Stylové jazyky

- ▶ Jazyky pro transformování a formátování dokumentů
- ▶ CSS (Cascading StyleSheet) – webové stránky
- ▶ XSL (eXtensible Stylesheet Language)
 - ▶ XSLT (eXtensible Stylesheet Language Transformation)
 - ▶ XSL-FO (eXtensible Stylesheet Language - Formatting Objects)
 - zpracování textu
- ▶ SLD (Styled Layer Descriptor)

DSSSL (Document Style Semantics and Specification Language), FOSI (Formatted Output Specification Instance), STX (Streaming Transformations for XML)...

XSLT

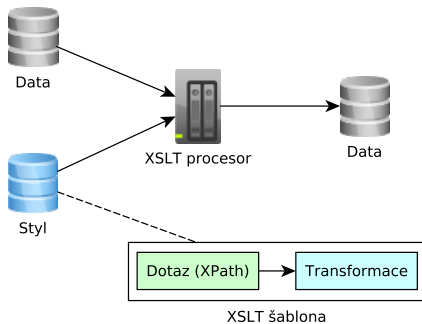
- ▶ Na pomezí mezi SQL (dotazovacími jazyky) a programovacími jazyky
- ▶ Obsahuje dotaz na data, ale také možnost úpravy (modifikace) těchto dat



XSLT standardy – XSLT 2.0 (2007)

- ▶ XSL Transformations (XSLT) Version 2.0
- ▶ XML Path Language (XPath) 2.0
- ▶ XQuery 1.0 and XPath 2.0 Functions and Operators
- ▶ XQuery 1.0 and XPath 2.0 Data Model (XDM)
- ▶ XQuery 1.0: An XML Query Language
- ▶ XSLT 2.0 and XQuery 1.0 Serialization
- ▶ XML Syntax for XQuery 1.0 (XQueryX)
- ▶ XQuery 1.0 and XPath 2.0 Formal Semantics

XSLT styl

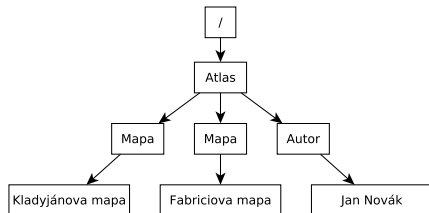


XML Infoset

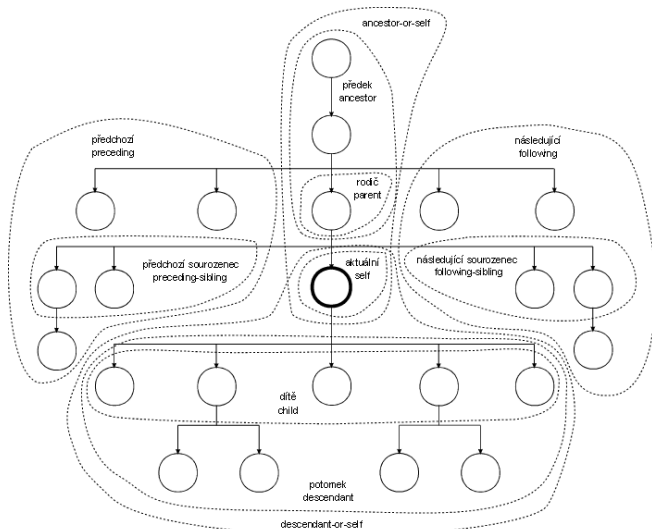
- ▶ Abstraktní model XML dokumentu, se kterým pracuje XSLT procesor
- ▶ Stromová struktura
- ▶ Zahrnuje kořenový uzel, element, atribut, textový uzel, instrukce pro zpracování, komentář, jmenný prostor

XML Infoset – příklad

```
<Atlas>  
  <Mapa>Klaudyjánova mapa</Mapa>  
  <Mapa>Fabriciova mapa</Mapa>  
  <Autor>Jan Novák</Autor>  
</Atlas>
```



XSLT / XPath – vztahy mezi uzly



Podle <http://www.kosek.cz/xml/xslt/xpath.html#model>

XPath – dotaz

- ▶ Identifikátor osy – ten určuje, ve kterém směru se budeme od aktuálního uzlu pohybovat (viz vztahy mezi uzly)
- ▶ Test uzlu – umožňuje vybírat jen některé uzly na základě jejich typu (`node()`, `text()`, `*. . .`) a názvu
- ▶ Predikát – vybrané uzly můžeme dále filtrovat pomocí podmínek, které jsou testována pro každý uzel

```
child::Mapa[1]
```

Zkrácené odkazy na uzly

- ▶ `child::Mapa` → `Mapa`
- ▶ `attribute::id` → `@id`
- ▶ `self::node()` → `.`
- ▶ `parent::node()` → `..`
- ▶ `/descendant-or-self::node()/child::Mapa` → `//Mapa`

XPath – operátory a funkce

- ▶ Sjednocení výrazů: |
- ▶ Logické spojky: and, or, not
- ▶ Relační operátory: =, !=, <=, <, >=, >
- ▶ Matematické operátory: +, -, *, div, mod
- ▶ Práce s uzly: last(), position(), count()...
- ▶ Řetězcové funkce: string(), concat(), substring()...
- ▶ Logické funkce: boolean(), not(), true(), false()
- ▶ Funkce pro práci s čísly: sum(), round()...
- ▶ Rozšiřující funkce definované v XSLT: document(), generate-id(), format-number()...

XSLT styl – základní rámec

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="2.0">
<xsl:output method="xml" encoding="utf-8"/>

<xsl:template match="/">
...
</xsl:template>

</xsl:stylesheet>
```

Co všechno XSLT styl umí

- ▶ Zpracovávat šablonu – `xsl:apply-templates`
- ▶ Vypsát hodnotu – `xsl:value-of`
- ▶ Cyklus – `xsl:for-each`
- ▶ Rozhodování / podmínky – `xsl:if`, `xsl:choose`
- ▶ Vytvořit obsah (element, atribut. . .)
- ▶ Základní matematické funkce
- ▶ Pracovat s proměnnými a sekvencemi
- ▶ Řazení a seskupování
- ▶ Používání klíčů
- ▶ Regulární výrazy
- ▶ Kombinování stylů
- ▶ Výstup do více souborů
- ▶ Více souborů na vstupu

Komunikace

- ▶ Vazby – XLink, XPointer, XInclude
- ▶ Dotazy – XPath, XQuery, Quilt
- ▶ API – DOM, SAX
- ▶ Bezpečnost a autentifikace – SAML

Ostatní XML jazyky a formáty...

... použitelné v geomatice

- ▶ OpenDocument, Office Open XML (OOXML), DocBook, HTML
- ▶ Metadata, ontologie, slovníky (RDF, RDFS, DCMI, SKOS, OWL...)
- ▶ ArcXML
- ▶ ...

XML Zdroje

- ▶ Stránky na W3C a OGC – specifikace a jiné informace
- ▶ W3C Schools – tutoriály
- ▶ Zvon.org & Kosek.cz – tradiční, léty prověřené české zdroje
- ▶ XML Prague

Děkuji za pozornost

<https://www.linkedin.com/in/otakarcerba/>

Prezentace byla vytvořena pro výuku předmětu GeoWeb na Katedře geomatiky Západočeské univerzity v Plzni.
Prezentace nebo její části jsou k dispozici k dalšímu využití za podmínky regulární citace.

Cvičení – Transformace lokálních SPOI

- ▶ Vstupní data: Tynec.rdf (vstupní data pro SPOI ve formátu RDF)
- ▶ Transformace (úkoly):
 - ▶ Výpis bodů do textového souboru
 - ▶ Výpis bodů a souřadnic do tabulky (CSV) – samostatná práce
 - ▶ Počet bodů s adresou a jejich procentuální vyjádření – samostatná práce
 - ▶ Transformace do GeoJSON – samostatná práce
- ▶ Transformační procesor: XSL Transformer nebo jiné online řešení

Cvičení – Vstupní data

```
<rdf:Description rdf:about="http://www.sdi4apps.eu/poi/#Týnec" >
  <rdfs:label xml:lang="cs">Týnec č.p. 9</rdfs:label>
  <rdfs:comment xml:lang="cs">Dům s popisným číslem 9 v
  <geos:asWKT rdf:datatype="http://www.openlinksw.com/s
  <rdfs:seeAlso rdf:resource="https://www.what3words.co
    <rdfs:comment xml:lang="en">what3words.com</rdfs:c
  </rdfs:seeAlso>
  <poi:class rdf:resource="http://gis.zcu.cz/SPOI/Onto
  <poi:class rdf:resource="http://gis.zcu.cz/SPOI/Onto
  <locn:fullAddress>Týnec 9, 340 21 Janovice na Úhlavou
  <rdfs:seeAlso>
    <rdf:Description rdf:about="http://klatoviny.blogspot
      <rdfs:label xml:lang="cs">Zákeřná vražda v Týnc
      <dc:type rdf:resource="http://purl.org/dc/dcmity
      <dc:language>cs</dc:language>
      <rdfs:comment xml:lang="cs">článek na webu</rdfs:
      <rdfs:comment xml:lang="cs">Článek z blogu Klat
    </rdf:Description>
```

Cvičení – na co si dát pozor

- ▶ Kořenový element `rdf:RDF`
- ▶ Jmenné prostory
- ▶ Výskyt elementu na různých úrovních (například `rdf:Description`)
- ▶ Navržené řešení není jediné správné

Cvičení – výpis bodů do textového souboru

- ▶ Hlavička XML dokumentu

```
<?xml version="1.0" encoding="utf-8"?>
```

- ▶ Kořenový element XSLT stylu + jmenné prostory

```
<xsl:stylesheet  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  
version="2.0">
```

Cvičení – výpis bodů do textového souboru

- ▶ Výstupní formát (hned za kořenový element)

```
<xsl:output method="text" encoding="utf-8" indent="yes"/>
```

- ▶ Šablona

```
<xsl:template match="/">
```

Cvičení – výpis bodů do textového souboru

- ▶ Dotaz + transformace (do šablony)

```
<xsl:for-each
  select="//rdf:RDF/rdf:Description/rdfs:label[1]">
  <xsl:value-of select="."/><xsl:text>&#xa;</xsl:text>
</xsl:for-each>
```

Cvičení – výpis bodů do textového souboru – XSLT soubor

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
version="2.0">
<xsl:output method="text" encoding="utf-8" indent="yes"/>

<xsl:template match="/">
<xsl:for-each
  select="//rdf:RDF/rdf:Description/rdfs:label[1]">
<xsl:value-of select="."/><xsl:text>&#xa;</xsl:text>
</xsl:for-each>
</xsl:template>

</xsl:stylesheet>
```

Cvičení – výpis bodů do textového souboru – výsledek

TRANSFORM XML

TRANSFORM XML IN NEW WINDOW

Transformed XML:

```
Týnec ž.p. 9
Boží muka na Rozpáralce
Týnec
Vacovy
Rozpáralka
Horní Lhota
Loreta
Vacovský rybník
Rybník Bašta
Rybník Mlýnek
Rybník Gomora
Autobusová zastávka Týnec, Loreta, rozc.0.3 (směr Týnec, Janovice, Javor)
Autobusová zastávka Týnec, Loreta, rozc.0.3 (směr Klatovy)
Autobusová zastávka Klatovy, Luby, u drůbežárny II (směr Týnec, Janovice, Javor)
Autobusová zastávka Klatovy, Luby, u drůbežárny II (směr Klatovy)
Autobusová zastávka Týnec, náves (směr Janovice, Javor)
Autobusová zastávka Týnec, náves - 2 (směr Klatovy)
Autobusová zastávka Týnec, točna (bus)
Autobusová zastávka Týnec, Rozpáralka (směr Javor)
```

COPY TO CLIPBOARD

SELECT ALL

Cvičení – samostatné práce

- ▶ Výpis názvů bodů a souřadnic do tabulky (CSV)
- ▶ Počet bodů s adresou a jejich procentuální vyjádření (TXT, HTML)
- ▶ Transformace dat (název a souřadnice) z RDF do formátu GeoJSON
 - ▶ validace například v GeoJSONLint – včetně vizualizace