

# Informační systémy

## XML, Atom, RSS, Web Services

Martin Trnečka

Katedra informatiky  
Univerzita Palackého v Olomouci

# XML

- Extensible Markup Language (XML).
- Textový formát reprezentující strukturovaná data.
- Vznik ze standardu SGML (ISO 8879), přepracován pro pohodlnější použití na webu (platformová nezávislost).
- Určitá podobnost s HTML (není náhradou HTML, ale spíše doplněk).
- Primárně určen jako nosič dat (struktura pro uložení dat) nikoliv pro zobrazování dat (zásadní rozdíl s HTML).
- Není definována žádná sada elementů (uživatel si jí musí definovat sám).
- Striktní politika (na rozdíl od HTML). Dokument s chybou nebude zobrazen (je zobrazena chybová hláška).
- Pod záštitou W3C.
- Velice rozšířený omyl: XML je triviální záležitost.
- <http://www.w3.org/standards/xml/core>

# XML v kontextu webu

- Velká popularita XML.
- O XML jsme již mluvili na přednášce týkající se AJAXu a MySQL.
- Budeme mluvit o XML v kontextu www stránek, ale vše je přenositelné.
- Běžně používané i v klasických aplikacích.
- Multiplatformní, snadná výměna informací a sdílení dat.

## Příklad (Jednoduchý XML dokument)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <note>
3   <to>Me</to>
4   <from>My Self</from>
5   <heading>I</heading>
6   <body>My, My Self and I</body>
7 </note>
```

Analogicky s HTML je možné XML reprezentovat stromovou strukturou.

# Specifika XML

- Všechny elementy musejí mít uzavírací tag. Je možné použít /.
- Stejně jako v HTML je třeba elementy korektně uzavírat.
- Case sensitive. Uzavírací a otevírací tag musejí být stejné!

## Příklad

```
1 <Message>This is incorrect</message>
2 <message>This is correct</message>
```

- Hodnoty všech atributů musejí být uvozeny uvozovkami (rozdíl oproti HTML).

## Příklad

```
1 <note date="16. 04. 2014"></note>
```

- Je zde pouze 5 entit: &lt; &gt; &amp; &apos; &quot;
- Komentáře stejné jako v HTML. XML nemění bílé znaky.

# Pojmenovávání elementů

- Název může obsahovat písmena, čísla a jiné znaky.
- Nesmí začínat číslem a znakem s interpunkcí.
- Nesmí začínat `xml`, `XML`, `Xml`, ...
- Nesmí obsahovat mezeru.
- **Doporučené:** vyhnout se znakům `:`, `.`, `-` kvůli kompatibilitě.
- Obvykle camel nebo pascal styl.

**Poznámka:** Názvy elementů jsou obvykle srozumitelné pro člověka, stejně tak i jejich obsah (nic ale nebrání použít např. `m45781sfh1`).

**Poznámka:** Obsah je obvykle uložen pro člověka čitelným způsobem. Nic nebrání uložit do XML například výsledek hash funkce, ale v praxi se to používá velice zřídka.

# Atributy

- Slouží pro rozšíření informace.
- Lze chápat jako syntaktický cukr. Lze se jím zcela vyhnout.
- Občas chápány dosti rozporuplně (často kritizovány).
- Nemohou obsahovat více hodnot, netvoří stromovou strukturu.
- **Doporučené:** používat atributy pro informace ne zcela relevantní pro data. Z našeho pohledu poněkud ošemetné, ale lze na ně nahlížet jako prostředek pro definici metadat.
- Atribut `id` pro identifikaci elementů.

# Konflikt názvů

- Elementy definují uživatele.
- Možné kolize. Řešení pomocí namespace.

## Příklad (Namespace)

```
1 <h:table xmlns:h="http://www.w3.org/TR/html4/">
2   <h:tr>
3     <h:td>Apples</h:td>
4     <h:td>Bananas</h:td>
5   </h:tr>
6 </h:table>
7
8 <f:table xmlns:f="http://www.w3schools.com/furniture">
9   <f:name>African Coffee Table</f:name>
10  <f:width>80</f:width>
11  <f:length>120</f:length>
12 </f:table>
```

Namespace lze definovat i v kořenovém elementu.

# Validní XML

- Není totéž co dobře formátovaný XML dokument (viz. pravidla pro zápis XML).
- Dobře formátovaný = nezpůsobí chybu.
- Validní XML musí odpovídat typu dokumentu (schématu).
- Document Definition:
  - ① Document Type Definition (DTD).
  - ② Novější XML Schema.
- Prostředek pro definici korektní struktury dokumentu.
- Pro triviální práci s XML nemá smysl.



# Document Type Definition

## Příklad (XML dokument)

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE note SYSTEM "Note.dtd">
3 <note>
4   <to>Me</to><from>My Self</from><heading>Reminder</heading>
5   <body>Don't forget me this weekend!</body>
6 </note>
```

## Příklad (DTD soubor)

```
1 <!DOCTYPE note
2 [
3 <!ELEMENT note (to,from,heading,body)>
4 <!ELEMENT to (#PCDATA)> <!-- #PCDATA = parse-able text data. CDATA neinterpretuje -->
5 <!ELEMENT from (#PCDATA)>
6 <!ELEMENT heading (#PCDATA)>
7 <!ELEMENT body (#PCDATA)>
8 ]>
```

# Document Type Definition

- Lze definovat počty výskytů, zda je pole povinné a další různá omezení.
- Velice rozsáhlé možnosti.

## Příklad (DTD příklady)

```
1 <!ELEMENT wonder (name+, location?, height, history , mainImage, source*)>
2 <!ELEMENT location ANY>
3
4 <!ELEMENT square EMPTY>
5 <!-- definice atributu -->
6 <!ATTLIST square width CDATA "0">
```

Základní tutoriál na: <http://www.w3schools.com/dtd/default.asp>

# XML schema

## Příklad (XML schema)

```
1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4 <xs:element name="note">
5   <xs:complexType>
6     <xs:sequence>
7       <xs:element name="to" type="xs:string"/>
8       <xs:element name="from" type="xs:string"/>
9       <xs:element name="heading" type="xs:string"/>
10      <xs:element name="body" type="xs:string"/>
11    </xs:sequence>
12  </xs:complexType>
13 </xs:element>
14
15 </xs:schema>
```

Je zapsáno přímo v XML (musí být korektní). Celá řada výhod oproti DTD.

# XML schema

## Příklad (XML dokument)

```
1 <note
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3   xmlns:ex="http://www.example.org/contactExample"
4   xsi:schemaLocation="http://www.example.org/contactExample schema.xsd"
5   xsi:noNamespaceSchemaLocation="schema.xsd">
```

## Základní datové typy:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Přehled: <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>

# XML schema

## Příklad (Omezení počtu výskytů)

```
1 <xs:element name="addresses">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="address" minOccurs='1' maxOccurs='unbounded' />
5     </xs:sequence>
6   </xs:complexType>
7 </xs:element>
```

## Příklad (Omezení délky)

```
1 <xs:element name="story">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:length value="1024"/>
5     </xs:restriction>
6   </xs:simpleType>
7 </xs:element>
```

## Příklad (Omezení hodnoty)

```
1 <xs:element name="age">
2   <xs:simpleType>
3     <xs:restriction base="xs:integer">
4       <xs:minInclusive value="0"/>
5       <xs:maxInclusive value="120"/>
6     </xs:restriction>
7   </xs:simpleType>
8 </xs:element>
```

## Příklad (Výčtový typ)

```
1 <xs:element name="car">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:enumeration value="Audi"/>
5       <xs:enumeration value="Golf"/>
6       <xs:enumeration value="BMW"/>
7     </xs:restriction>
8   </xs:simpleType>
9 </xs:element>
```

## Příklad (Pattern)

```
1 <xs:element name="initials">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:pattern value="[A-Z] [A-Z] [A-Z]" />
5     </xs:restriction >
6   </xs:simpleType>
7 </xs:element>
```

## Příklad

```
1 <xs:element name="letter">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:pattern value="([a-z])*" />
5     </xs:restriction >
6   </xs:simpleType>
7 </xs:element>
```

Přehled: [http://www.w3schools.com/schema/schema\\_facets.asp](http://www.w3schools.com/schema/schema_facets.asp)

# Transformace XML

- XML nese pouze informaci.
- Jazyk XSLT umožňuje transformovat XML na jiný formát.

## Příklad

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="sablonu.xsl"?>
3
4 <ancient_wonders>
5   <wonder>
6     <name language="English">Colossus of Rhodes</name>
7     <name language="Greek">Hatmatilka</name>
8     <location>Rhodes, Greece</location>
9     <height units="feet">107</height>
10  </wonder>
11 </ancient_wonders>
```

Přehled možností XSLT:

[http://www.w3schools.com/xsl/xsl\\_w3celementref.asp](http://www.w3schools.com/xsl/xsl_w3celementref.asp)



# Transformace XML

## Příklad (XSL)

```
1 <?xml version="1.0"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3 <xsl:output method="html"/>
4
5 <xsl:template match="/">
6 <html>
7   <head>
8     <title>Wonders of the World</title>
9   </head>
10  <body>
11    <h1>Wonders of the World</h1>
12
13    The <xsl:value-of select="ancient_wonders/wonder/name"/> is located in
14    <xsl:value-of select="ancient_wonders/wonder/location"/>.
15  </body>
16 </html>
17
18 </xsl:template>
19 </xsl:stylesheet>
```

# Pokročilejší transformace XML

## Příklad (Cyklus v XSL)

```
1 <table>
2 <xsl:for-each select="ancient_wonders/wonder">
3
4 <tr>
5   <td><strong><xsl:value-of select="name[@language='English']"/></strong><br/>
6   (<em><xsl:value-of select="name[@language!='English']"/></em>)</td>
7   <td><xsl:value-of select="location"/></td>
8 </tr>
9
10 </xsl:for-each>
11 </table>
```

## Příklad (Podmínka v XSL)

```
1 <xsl:if test="name[@language!='English']">
2   <br/>(<em><xsl:value-of select="name[@language!='English']"/></em>)
3 </xsl:if>
```

# Pokročilejší transformace XML

## Příklad (Podmínky v XSL)

```
1 <xsl:choose>
2   <xsl:when test="height != 0">
3     <xsl:value-of select="height"/>
4   </xsl:when>
5   <xsl:otherwise>
6     unknown
7   </xsl:otherwise>
8 </xsl:choose>
```

## Příklad (Setřídění)

```
1 ...
2 <xsl:for-each select="ancient_wonders/wonder">
3 <xsl:sort select="height" order="descending" data-type="number" />
4 ...
```

# Jazyk XPath

- Slouží pro vyjádření části XML dokumentu. Analogie s cestou v systému.
- Typické použití pro vyhledání určitých dat v XML.
- Netriviální jazyk.
- Zavádí pojmy předchůdce, následník, rodič, potomek, sourozenec.
- Jádrem jazyka XSLT.
- <http://www.w3schools.com/XPath/default.asp>

## Příklad (XPath)

```
1 /bookstore/book[price>35]/price
2 /bookstore/book/@attribut
3 //title[@lang='eng']
```

Poznámka: založeno na dotazovacím jazyce XQuery (analogie SQL pro XML). Jazyk XPointer rozšiřuje XPath.

# Odbočka: PHP a XML

- Rozsáhlá podpora. Bouřlivý vývoj (bolestivý přechod mezi PHP 4 a PHP 5).
- Rozhraní SAX, SimpleXML, DOM, XPath dále podpora transformací.
- Založeno na populární knihovně libxml2 (Perl, Python, Ruby ...).
- Nejednoduší SimpleXML.

## Příklad (SimpleXML procházení XML)

```
1 $xml = simplexml_load_file ("dokument.xml");  
2 $xml->chanel->item[1]->title  
3 $xml->chanel->item[1]->title->attributes(); // pole atributu
```

- Vytváření XML, metody: addChild(), addAttribute(), asXML()
- Zobrazení XML: header("Content-type: text/xml");
- Přehled: <http://www.php.net/manual/en/class.simplexmlelement.php>

# Odbočka: PHP a XML

## Příklad (Vytvoření XML pomocí DOM)

```
1 $xml = new DOMDocument("1.0");
2
3 $root = $xml->createElement("data");
4 $xml->appendChild($root);
5 $id = $xml->createElement("id");
6 $idText = $xml->createTextNode('1');
7 $id->appendChild($idText);
8
9 $title = $xml->createElement("title");
10 $titleText = $xml->createTextNode(' "PHP Undercover" ');
11 $title->appendChild($titleText);
12
13 $book = $xml->createElement("book");
14 $book->appendChild($id);
15 $book->appendChild($title);
16 $root->appendChild($book);
17 $xml->formatOutput = true;
18 echo "<xmp>". $xml->saveXML() . "</xmp>";
19
20 $xml->save("mybooks.xml") or die("Error");
```

# Odbočka: JavaScript a XML

- `window.XMLHttpRequest` (viz. minulé přednášky).
- Lze parsovat jako podvržený AJAX request.
- Podpora v knihovně jQuery (XPath).

# RSS

- Formát založený na XML.
- Primárně určen pro syndikaci obsahu webu (informace o zveřejnění nového obsahu). Fenomenální počín, značně ovlivnil podobu internetu.
- V minulosti velice populární. Dnes ve značné míře na ústupu.
- V roce 2003 vydána poslední verze RSS 2.0.1 (vydávány upřesnění specifikace nikoliv nová funkcionalita).

## Příklad (RSS)

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <rss version="2.0">
3 <channel>
4   <title>W3Schools Home Page</title>
5   <link>http://www.w3schools.com</link>
6   <description>Free web building tutorials </description>
7   <item>
8     <title>RSS Tutorial</title>
9     <link>http://www.w3schools.com/rss</link>
10  </item>
```



# Kanál Atom

- Atom Syndication Format. Nástupce formátu RSS.
- Vznik jako reakce na nedostatky RSS a zmrazení vývoje RSS.
- <http://tools.ietf.org/html/rfc5023>

## Příklad (Atom, začátek)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <feed xml:lang="cs" xmlns="http://www.w3.org/2005/Atom" xml:base="http://www.inf.upol.cz">
3
4 <id>tag:www.inf.upol.cz,2014:/kalendar/</id>
5 <updated>2014-04-14T13:52:33Z</updated>
6 <title>Katedra informatiky, Univerzita Palackeho Kalendar</title>
7 <author>
8   <name>Katedra informatiky, Univerzita Palackeho</name>
9 </author>
10 <link rel="self" href="http://www.inf.upol.cz/atom/kalendar"/>
11 <link rel="alternate" href="http://www.inf.upol.cz/kalendar"/>
12 <category term="kalendar" label="kalendar"/>
13 <rights>Katedra informatiky, Univerzita Palackeho</rights>
```

## Příklad (Atom, položka feedu)

```
1 <entry>
2   <id>tag:www.inf.upol.cz,2014-04-17:/kalendar/415</id>
3   <updated>2014-04-17T19:00:00Z</updated>
4   <title>Cteci seminar(DAMOL)</title>
5   <published>2014-04-17T19:00:00Z</published>
6   <content type="xhtml">
7     <div xmlns="http://www.w3.org/1999/xhtml"><div>17.04.2014</div>
8     Radim Belohlavek (Palacky University , Czech Republic): Development of Many-Valued
9     and Fuzzy Logics 3. Seminar se kona v mistnosti 5.077.
10    </div>
11  </content>
12 </entry>
```

# Web Services

- SOAP (Simple Object Access Protocol) - framework pro platformově nezávislé zasílání informací formou XML.
- Základem Web Services (webová služba). Dále pak technologie WSDL (Web Services Description Language) pro popis funkcionality webové služby.
- Populární v rozsáhlých IS. Obvykle nabízejí API formou webové služby.
- Příklad: <https://stagservices.upol.cz/ws/help/>
- Další příklady: exporty zboží pro porovnávače cen (např. Heureka.cz, Zbozi.cz).
- Rozsáhlá funkcionalita v PHP.
- Tutorial: <http://www.w3schools.com/Webservices/default.asp>

## Příklad (Web Services v PHP, server side)

```
1 require_once ('lib/nusoap.php');
2
3 $server = new soap_server;
4
5 // register a function that works on server
6 $server->register('get_message');
7
8 // create the function
9 function get_message($your_name) {
10     if(!$your_name) {
11         return new soap_fault('Client','', 'Put Your Name!');
12     }
13
14     $result = "Welcome to ".$your_name.". Thanks for Your First Web Service Using PHP with SOAP";
15     return $result;
16 }
17
18 // create HTTP listener
19 if ( !isset( $HTTP_RAW_POST_DATA ) ) {
20     $HTTP_RAW_POST_DATA = file_get_contents( 'php://input' );
21     $server->service($HTTP_RAW_POST_DATA);
22     exit();
23 }
```

## Příklad (Web Services v PHP, client side)

```
1 require_once ('lib/nusoap.php');
2
3 //Give it value at parameter
4 $param = array( 'your_name' => 'Monotosh Roy');
5
6 //Create object that referer a web services
7 $client = new soapclient('http://localhost/ws/server.php');
8
9 //Call a function at server and send parameters too
10 $response = $client->call('get_message', $param);
11
12 //Process result
13 if( $client->fault) {
14     echo "FAULT: <p>Code: (". $client->faultcode. "</p>";
15     echo "String: ". $client->faultstring;
16 }
17
18 else {
19     echo $response;
20 }
21 ?>
```

# Scalable Vector Graphics (SVG)

- Jazyk pro popis 2D grafiky v XML.
- Přímá podpora v HTML přes vnořené prostředí.
- Základní formát pro vektorový editor Inkscape.

## Příklad (Příklad SVG)

```
1 <svg width="100" height="100">
2   <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
3 </svg>
```

## Příklad (SVG a CSS)

```
1 <svg width="400" height="180">
2   <rect x="50" y="20" width="150" height="150"
3     style="fill:blue;stroke:pink;stroke-width:5;fill-opacity:0.1;stroke-opacity:0.9" />
4 </svg>
```

# Scalable Vector Graphics (SVG)

## Příklad (Transformace)

```
1 <svg height="60" width="200">  
2   <text x="0" y="15" fill="red" transform="rotate(30 20,40)">I love SVG</text>  
3 </svg>
```

- Možnosti filtru.
- Animace.
- Gradient a další.

Více na: <http://www.w3schools.com/svg/default.asp>

# Další aplikace XML

- ODF (Open Document format) a OOXML (Office Open XML).
- KML (Keyhole Markup Language) slouží pro uložení kartografických dat.
- eBooks a ePub.
- AJAX.

## O čem jsme nemluvili:


- XSL-FO jazyk pro formátování XML pro různá zařízení.
- Stylování XML (velice okrajová záležitost).
- XLink slouží pro propojení XML dokumentů.
- Simple API for XML (SAX) proudové zpracování XML.




## Důležité pojmy:

- Technologie XML a její použití.
- Validita a schemata pro XML, XPath.
- RSS, Atom, Web Services.

## Čtení na doma:

 Goldberg K. H., Visual Quickstart guide XML, 2009, ISBN 978-0-321-55967-8. (kapitoly 1-4, 6, 9-15, 17).

 <http://www.w3schools.com/schema/default.asp>