

# ROZŠIŘITELNÝ ZNAČKOVACÍ JAZYK XML

Jaroslav Šmarda



MASARYKOVA UNIVERZITA

# Rozšiřitelný značkovací jazyk XML

Jaroslav Šmarda

# Rozšiřitelný značkovací jazyk XML

- XML jako standardní textový formát pro výměnu informací

# Značkovací jazyky

- Standard Generalized Markup Language (SGML)
  - HTML (HyperText Markup Language) – jazyk webových stránek
  - XML (eXtensible Markup Language) – rozšiřitelný značkovací jazyk
    - Skupina jazyků pro komunikaci mezi aplikacemi čitelná pro člověka

# XML jako standardní textový formát pro výměnu informací

- XML (eXtensible Markup Language):
  - výměna a sdílení informací
  - pro dokumenty obsahující strukturované nebo semistrukturované informace
  - internetové aplikace (XSL definice vzhledu XML dokumentu)
  - databázové technologie
  - standardizace datových struktur
- XML je metajazyk, tedy souhrn pravidel pro tvorbu jednotlivých XML jazyků

# 1. příklad souboru v XML formátu

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<!-- Students grades are updated bi-monthly -->
<class_list>
  <student>
    <name>Robert</name>
    <grade>A+</grade>
  </student>
  <student>
    <name>Lenard</name>
    <grade>A-</grade>
  </student>
</class_list>
```

## 2. příklad souboru ve formátu XML

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:media="http://search.yahoo.com/mrss" >
<channel>
  <item>
    <title>VIDEO: Tři muži přepadli na pumpě cizince</title>
    <link>http://www.denik.cz/z_domova/video-tri-muzi-prepadli-na-
pumpe-
      cizince20110212.html</link>
    <description>Karlovarsko /FOTOGALERIE/ - K loupežnému
přepadení dvou cizinců
      ve věku dvacet a dvaadvacet let došlo v pátek ve večerních
      hodinách.
    </description>
    <pubDate>Sat, 12 Feb 2011 17:40:00 GMT</pubDate>
    <category domain="http://www.denik.cz/z_domova/">Z
domova</category>
  </item>
```

# XML

## ➤ Značky (tags):

➤ počáteční: `<channel>` `<item>` `<title>`

➤ koncové: `</item>`

➤ počáteční i koncová: `<item/>`

## ➤ Data

➤ VIDEO: Tři muži přepadli na pumpě cizince



# Je XML značkovací jazyk ?

- Jazyk má:
  - slovník symbolů
  - gramatiku
  
- XML není značkovací jazyk (markup language)
- XML definuje syntaxi pro **celou třídu jazyků**

# Správně strukturovaný XML dokument

- Dokument splňující syntaxi XML = správně strukturovaný (well-formed) XML dokument
- Je následující dokument správně strukturovaný XML?
  - `<?xml version="1.0" encoding="ISO-8859-15"?>`
  - `<!-- Students grades are updated bi-monthly -->`
  - `<class_list>`
  - `<student>`
  - `<name>Robert</name>`
  - `<grade>A+</grade>`
  - `<student>`
  - `<student>`
  - `<name>Lenard</name>`
  - `<grade>A-</grade>`
  - `</student>`
  - `</class_list>`

# Platnost XML dokumentu

- Jaké značky v jakém pořadí (struktura)
- 2 možnosti ověření platnosti:
  - DTD – Document Type Definition
    - Pravidla definující elementy a další značkovací objekty a jejich pořadí
    - Nedefinuje datový obsah
    - Značkovací objekt, který není v definici, je chybný
  - Schémata
    - Volnější definice, např. definuje, že dokument může obsahovat element *datum* a data jsou typu YYYY-MM-DD

# XML editory

- NotePad
- Internet Explorer
- Speciální editory:
  - Morphon XMLEditor ([www.morphon.com](http://www.morphon.com))
  - Epic
  - XMLFox

# Značkovací objekty ...

Značkovací objekt	Účel	Příklad
prázdný element	Reprezentuje informaci v určitém místě dokumentu	<code>&lt;xref linkend="abc"/&gt;</code>
element-kontejner	Spojuje elementy a znakové údaje	<code>&lt;p&gt;This is a paragraph.&lt;/p&gt;</code>
deklarace	Přidává nový parametr, entitu nebo gramatickou definici	<code>&lt;!ENTITY author "Erik Ray"&gt;</code>
Instrukce pro zpracování	Poskytuje speciální instrukce softwaru	<code>&lt;?print-formatter force-linebreak?&gt;</code>

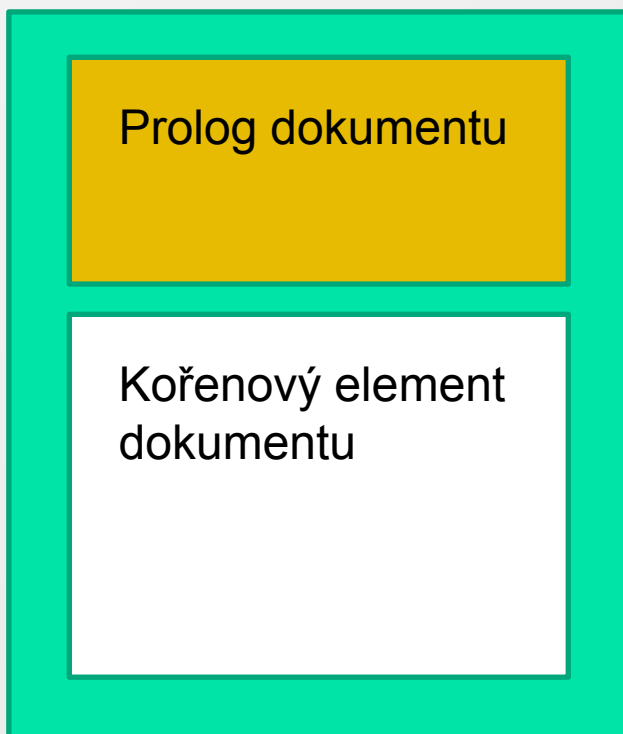
## ... Značkovací objekty

Značkovací objekt	Účel	Příklad
Poznámka	Vložení poznámky, XML procesor ji ignoruje	<code>&lt;!-- here's where I left off --&gt;</code>
Sekce CDATA	Sekce textových údajů, která nebude zpracovávána procesorem XML - může obsahovat speciální znaky, které mají význam v syntaxi XML	<code>&lt;![CDATA[Ampersands galore! &amp;&amp;&amp;&amp;&amp;]]&gt;</code>
Odkaz na entitu	Příkaz pro procesor XML, aby vložil text uložený jinde	<code>&amp;company-name;</code>

# Struktura XML dokumentu

- ☒ Logická entita spíše než fyzická
  - ☒ v jednom souboru může být více XML dokume

Například:



```
<?xml version="1.0" encoding="utf-8" ?>  
<rss version="2.0"  
xmlns:media="http://search.yahoo.com/mr  
ss">
```

```
<channel>
```

```
</channel>
```

# Prolog XML dokumentu

XML	Popis
<code>&lt;?xml version="1.0" standalone="no"?&gt;</code>	Deklarace XML verze
<code>&lt;!DOCTYPE reminder  SYSTEM "/home/eray/reminder.dtd" [   &lt;!ENTITY smile "&lt;graphic smile="smile.eps"/&gt;"&gt; ]&gt;</code>	Deklarace typu dokumentu Jméno kořenového elementu DTD identifikátor Oddělovač podmnožiny Deklarace entit



# Elementy XML

## ➤ Element-kontejner

```
<jméno atribut1 atribut2 ....>  
... obsah...  
</jméno>
```

## ➤ Prázdný element

```
<jméno atribut1 atribut2 ..../>
```

## ➤ Atribut elementu

```
jméno = "hodnota"
```

# Elementy XML

- Jméno elementu (stejně jako jméno atributu) = XML jméno
- XML jméno může obsahovat:
  - a-z, A-Z, a 0-9, znaky nelatinských abeced,
  - pomlčku (-), podtržítko (\_), tečku (.) a dvojtečku pro zvláštní účel (:)
- v XML jménech se rozlišují malá a velká písmena
- XML jména **začínají písmeny nebo podtržítkem**

## Správně strukturovaný XML?

<to-do>Clean fish tank</to-do>

<street\_address>1420 Sesame  
Street</street\_address>

<notes+comments>Huh?</notes+commments>

<2nd-phone-number>785-555-1001</2nd-phone-  
number>

<MP3.name>Where my doggies at?</MP3.name>

# Element-kontejner

- Základním prvkem XML dokumentu je element:
  - počáteční značka
  - obsah
  - ukončovací značka

**<význam\_elementu > obsah\_elementu  
</význam\_elementu >**

# Prázdný element

➤ Prázdný element:

**<význam\_elementu ></význam\_elementu >**

nebo zkráceně **<význam\_elementu />**

## Hierarchie XML elementů

- ❏ Elementy lze vnořovat do jiných elementů, ale ne takto:

```
<a>Don't <b>do</a> this!</b>
```

- ❏ Ale toto je v pořádku:

```
<a>No problem</a><b>here</b>
```

- ❏ Jméno značky na začátku a na konci musí být stejné:

```
<para>I like to ride my motorcycle  
<emphasis>really</emphasis> fast.  
</para>
```

# Hierarchie XML elementů

- Vkládáním vnořených elementů do obsahu kořenového elementu

```
<význam_elementuA> obsah_elementuA  
  <význam_elementuAA> obsah_elementuAA  
  </význam_elementuAA>  
</význam_elementuA>
```

# Entity v XML

```
<?xml version="1.0"?>
<!DOCTYPE message SYSTEM "/xmlstuff/dtds/message.dtd"
[
  <!ENTITY client "Mr. Rufus Xavier Sasperilla">
  <!ENTITY agent "Ms. Sally Tashuns">
  <!ENTITY phone "<number>617-555-1299</number>">
]>
<message>
  <opening>Dear &client;</opening>
  <body>We have an exciting opportunity for you! A set of ocean-front
  cliff dwellings in Mexico, have been renovated as time-share vacation
  homes. They're going fast! To reserve a place for your holiday, call
  &agent; at &phone;. Hurry, &client;. Time is running out!</body>
</message>
```



## Entity v XML

### ➤ Definice entit:

```
<!ENTITY client "Mr. Rufus Xavier Sasperilla">
```

```
<!ENTITY agent "Ms. Sally Tashuns">
```

```
<!ENTITY phone "<number>617-555-1299</number>">
```

### ➤ Použití entit:

call &agent; at &phone;. Hurry, &client;. Time is

### ➤ Textová náhrada &agent za **Ms. Sally Tashuns**

# XML schéma

- Schéma:
  - Generická reprezentace třídy objektů
  - Předpis, jak mají vypadat objekty stejné třídy
- XML schéma
  - Test, zda dokument XML odpovídá předpisu
- XML schémata
  - DTD (Document Type Definition)
  - XSD (XML Schema Definition)

## DTD – příklad XML souboru

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE clovek SYSTEM "příklad.dtd">
<clovek oscis="174762">
  <jmeno>Tomáš</jmeno>
  <adresa>
    <ulice>Nesmrtelná</ulice>
    <cislo>99</cislo>
    <mesto>Brno</mesto>
  </adresa>
</clovek>
```

## Příklad.dtd

```
<!ELEMENT clovek (jmeno, adresa*)>
<!ELEMENT jmeno (#PCDATA)>
<!ELEMENT adresa (ulice?, cislo?, mesto)>
<!ELEMENT ulice (#PCDATA)>
<!ELEMENT cislo (#PCDATA)>
<!ELEMENT mesto (#PCDATA)>
<!ATTLIST clovek
  oscis #REQUIRED
>
```

# DTD syntaxe

- #PCDATA
  - Znaková data, ale ne elementy
- ( e1, e2, e3)
  - Posloupnost elementů
- (e1| e2)
  - e1 nebo e2
- (...)\*
  - 0 až N opakování
- (...)+
  - 1 až N opakování
- (...)?
  - 0 až 1 krát

# DTD

- DTD deklaruje:
  - Slovník (elementy, které se mohou v XML dokumentu vyskytovat)
  - obsahový (content) model – jaké elementy může element-kontejner obsahovat
  - množinu atributů elementu

# Obsah jako element nebo atribut?

- Textový obsah v XML může být umístěn uvnitř elementu-kontejneru nebo jako atribut, o umístění rozhoduje autor
- Element-kontejner
  - Komplexnější text, který může být v budoucnosti dále strukturován
  - Nezáleží na přesném počtu mezer mezi slovy
- Atribut
  - Textová položka, která nebude dále strukturována (např. Číslo zaměstnance)
  - Potřebujeme přenést přesný obsah včetně mezer

## XSD – příklad XML souboru

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

```
<Address
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:noNamespaceSchemaLocation="SimpleAddress.xsd">
```

```
  <Recipient>Mr. Walter C. Brown</Recipient>
```

```
  <House>49</House>
```

```
  <Street>Featherstone Street</Street>
```

```
  <Town>LONDON</Town>
```

```
  <PostCode>EC1Y 8SY</PostCode>
```

```
  <Country>UK</Country>
```

```
</Address>
```



# XSD – SimpleAddress.xsd 1. část

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Recipient" type="xs:string" />
        <xs:element name="House" type="xs:string" />
        <xs:element name="Street" type="xs:string" />
        <xs:element name="Town" type="xs:string" />
        <xs:element name="County" type="xs:string" minOccurs="0"
        />
        <xs:element name="PostCode" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# XSD – SimpleAddress.xsd 2.část

```
<xs:element name="Country">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="FR" />
      <xs:enumeration value="DE" />
      <xs:enumeration value="ES" />
      <xs:enumeration value="UK" />
      <xs:enumeration value="US" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

# XSD – element-kontejner...

```
<xs:element name="address">
  <xs:complexType>
    <xs:all>
      <xs:element ref="street"/>
      <xs:element ref="city"/>
      <xs:element ref="county"/>
      <xs:element ref="country"/>
      <xs:element ref="postalcode"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

## ... XSD – element-kontejner

```
<xs:element name="street" type="xs:string"/>
```

```
<xs:element name="city" type="xs:string"/>
```

```
<xs:element name="county" type="xs:string"/>
```

```
<xs:element name="country" type="xs:string"/>
```

# XSD – element-kontejner

```
<xs:element name="address">
  <xs:complexType>
    <xs:all>
      <xs:element ref="street"/>
      <xs:element ref="city"/>
      <xs:element ref="county"/>
      <xs:element ref="country"/>
      <xs:element ref="postalcode"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

## XSD - atribut

```
<xs:attribute name="taker">  
  <xs:simpleType>  
    <xs:restriction base="xs:integer">  
      <xs:minInclusive value="1"/>  
      <xs:maxInclusive value="9999"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:attribute>
```

# XSD – opakování elementů

```
<xs:element name="census-record">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="date"/>  
      <xs:element ref="address"/>  
      <xs:element ref="person" maxOccurs="unbounded"/>  
    </xs:sequence>  
    <xs:attribute ref="taker"/>  
  </xs:complexType>  
</xs:element>
```

## XSD – povinný výskyt elementu

```
<xs:element name="name">  
  <xs:complexType>  
    <xs:all>  
      <xs:element ref="first"/>  
      <xs:element ref="last"/>  
    </xs:all>  
    <xs:choice minOccurs="0">  
      <xs:element ref="junior"/>  
      <xs:element ref="senior"/>  
    </xs:choice>  
  </xs:complexType>  
</xs:element>
```



# Definice typů v XSD

Typ	Použití
xs:string	textová hodnota
xs:token	slova oddělená mezerami
xs:decimal	čísla např. 1.356 ; -20
xs:integer	celá čísla 871; -13; +72
xs:float	32 bitové číslo s pohyblivou desetinnou tečkou Např. 123.456; +1234.456; -1.2344e56 ; -.45E-6; INF; - INF; NaN
xs:boolean	true;false; 0; 1
xs:time	čas HH:MM:SS-Timezone jako např. 21:55:00-06:00
xs:date	datum CCYY-MM-DD jako 2011-03-25
xs:dateTime	datum a čas CCYY-MM-DDTHH:MM:SS-Timezone jako 2011- 03-25T21:55:00-06:00