

# Y36XML – Technologie XML

---

Přednáší:

Irena Mlýnková ([mlynkova@ksi.mff.cuni.cz](mailto:mlynkova@ksi.mff.cuni.cz))

Martin Nečaský ([necasky@ksi.mff.cuni.cz](mailto:necasky@ksi.mff.cuni.cz))

ZS 2009

Stránka přednášky:

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

# Organizace přednášky, cvičení, zkoušky

---


<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
-

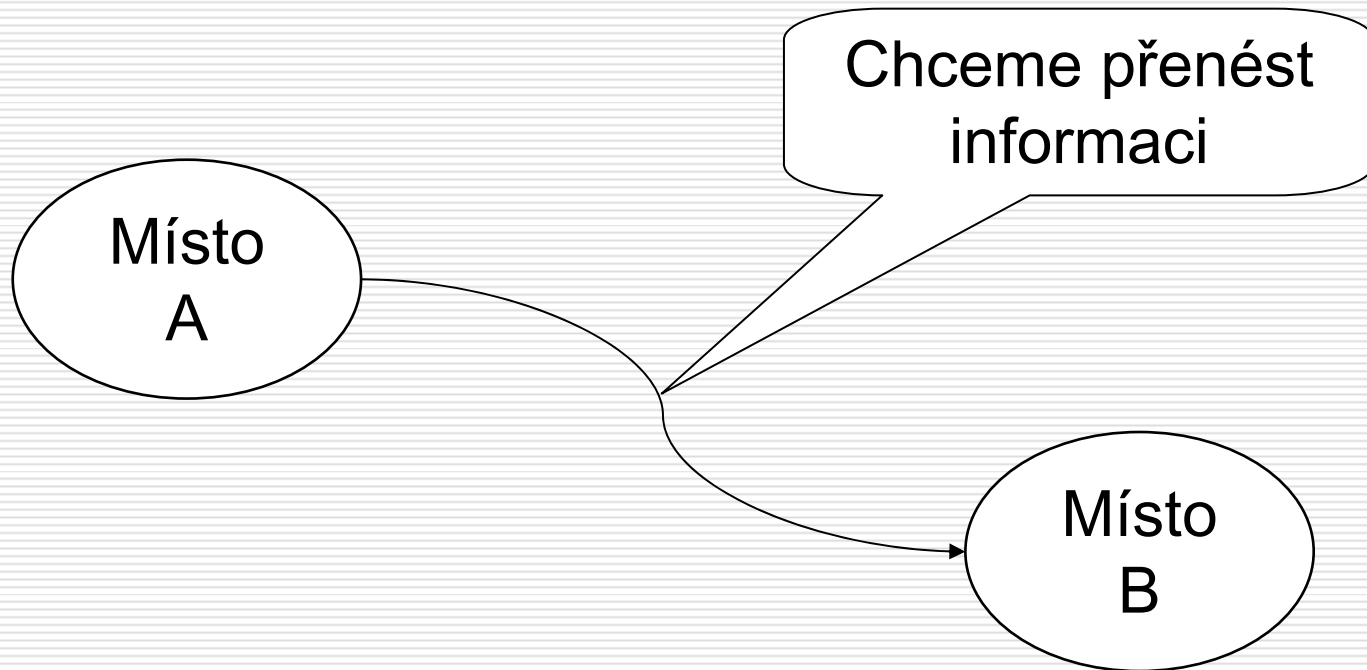


# Úvod do principů formátu XML



# Motivace

---



# Např.: chceme přenést zprávu

---

Jan Amos,  
Karel Hynek

Ahoj!

Pozdrav z říše divů!

Alenka

P.S. Napište mi!

---

# Jako „nestrukturovaný“ text?

---

Jan Amos, Karel Hynek Ahoj! Pozdrav z říše divů

---

# Jako „nestrukturovaný“ text?

---

Karel Hynek Ahoj! Pozdrav z říše divů! Alenka P

---



# Jako „nestrukturovaný“ text?

---

Ahoj! Pozdrav z říše divů! Alenka P.S. Napište

---

# Jako „nestrukturovaný“ text?

---

Pozdrav z říše divů! Alenka P.S. Napište mi!

---

# Jako „nestrukturovaný“ text?

---

**Alenka P.S. Napište mi!**

---

# Jako „nestrukturovaný“ text?

---

**Alenka P.S. Napište mi!**

Jak ale (automatizovaně) zjistit,  
kdo nám to vlastně píše?

---

# Zavedeme konstrukci „značka“ (tag)

---

Počáteční značka – otevírací  
závorka (start tag)

**<značka>tělo</značka>**

Koncová značka –  
zavírací závorka  
(end tag)

# Jednotlivé složky zprávy označíme

---

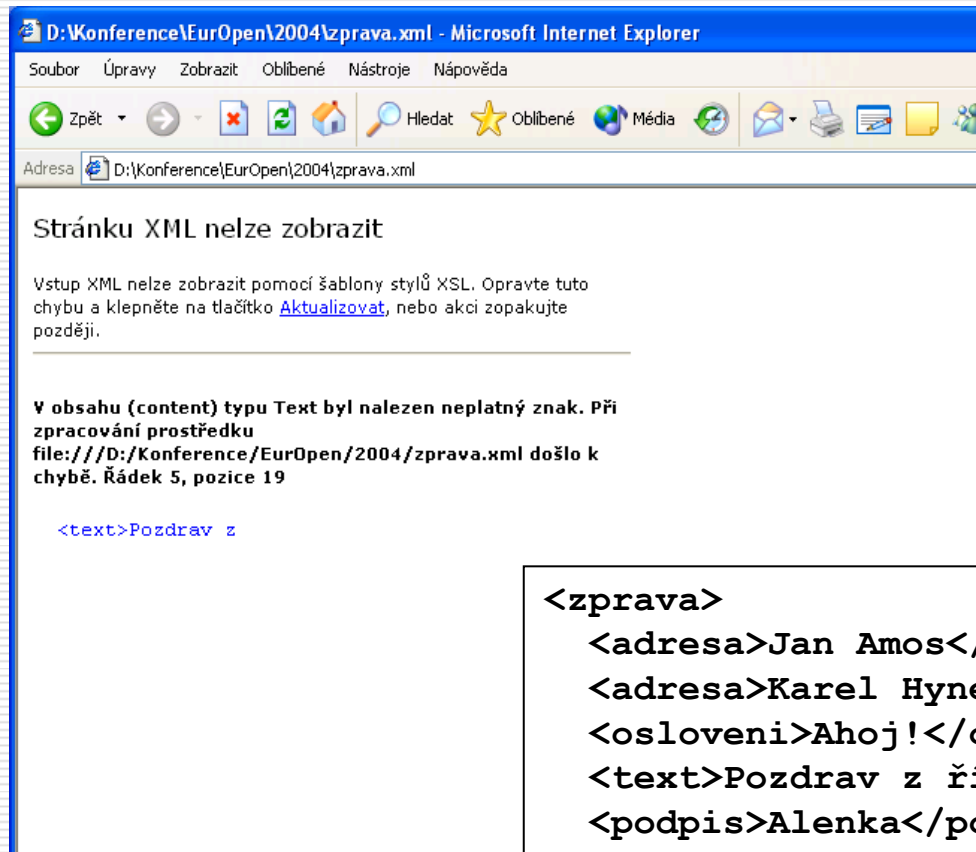
```
<adresa>Jan Amos</adresa>  
<adresa>Karel Hynek</adresa>  
<osloveni>Ahoj!</osloveni>  
<text>Pozdrav z říše divů!</text>  
<podpis>Alenka</podpis>  
<dodatek>Napište mi!</dodatek>
```

# A zabalíme do závorek

---

```
<zprava>  
  <adresa>Jan Amos</adresa>  
  <adresa>Karel Hynek</adresa>  
  <osloveni>Ahoj!</osloveni>  
  <text>Pozdrav z říše divů!</text>  
  <podpis>Alenka</podpis>  
  <dodatek>Napište mi!</dodatek>  
</zprava>
```

# Pro zobrazení správného textu prohlížečem to ještě nestačí



```
<zprava>
  <adresa>Jan Amos</adresa>
  <adresa>Karel Hynek</adresa>
  <osloveni>Ahoj!</osloveni>
  <text>Pozdrav z říše divů!</text>
  <podpis>Alenka</podpis>
  <dodatek>Napište mi!</dodatek>
</zprava>
```



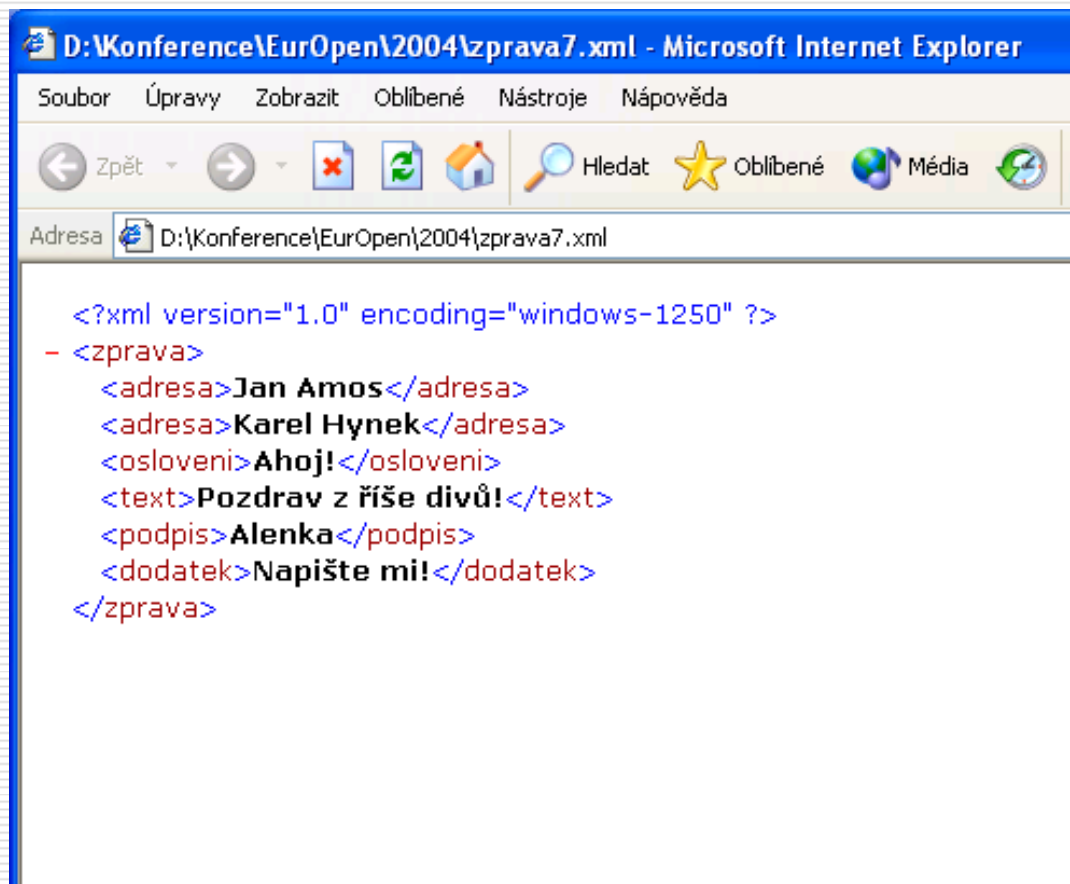
# Musí se přidat informace o kódování

---

- ☐ Implicitně je dokument v kódu ISO 10646 ([Unicode](#))
  - ☐ Pro komunikaci se světem se používá UTF-8
    - Kompatibilní s ASCII
    - Další znaky kódovány na 2 až 6-ti bytech
    - Obsahuje všechny znaky všech abeced
  - ☐ Pro češtinu lze použít ISO-8859-2 nebo Windows-1250
-

# Lepší, ale stále to ještě není ono – nepopsali jsme způsob zobrazení dokumentu

---



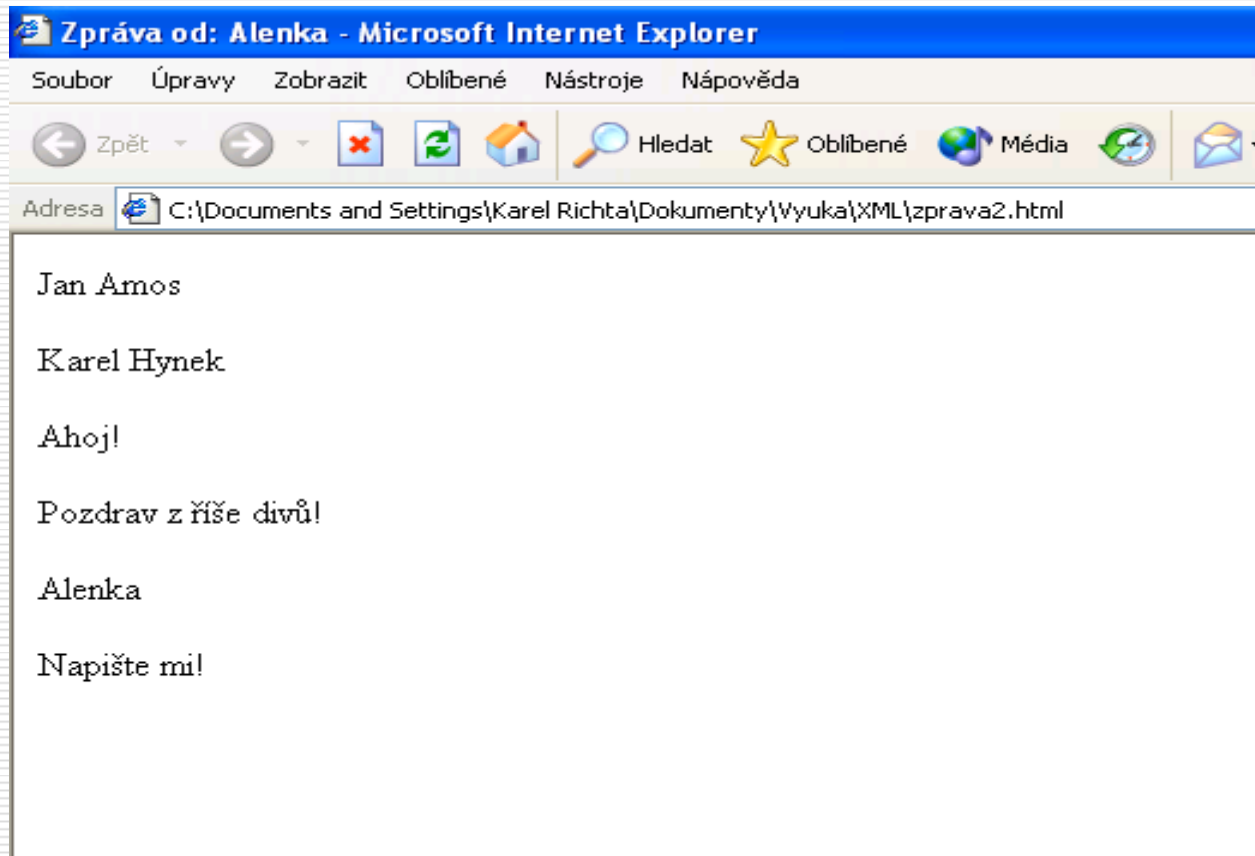
# Např. transformujeme do HTML

---

```
<html encoding="windows-1250">
  <head>
    <title>Zpráva od: Alenka</title>
  </head>
  <body>
    <p>Jan Amos</p>
    <p>Karel Hynek</p>
    <p>Ahoj!</p>
    <p>Pozdrav z říše divů!</p>
    <p>Alenka</p>
    <p>Napište mi!</p>
  </body>
</html>
```

# Prohlížeč teď „ví“ co s daty

---



# O co nám jde?

---

- ☐ Samotná data se těžko zpracovávají
  - ☐ Potřebujeme:
    - Zajistit aby příslušný program datům rozuměl = doplnit význam jednotlivých částí
  - ☐ Př. HTML – popis vizualizace dat pro HTML prohlížeč
    - Problém 1: Co když nás nezajímá jen vizualizace?
    - Problém 2: HTML má volnou strukturu
      - ☐ Komplikuje zpracování
  - ☐ Řešení: XML
-

# XML

---

- XML (eXtensible Markup Language) je formát pro přenos a výměnu obecných dokumentů
    - Extensible Markup Language (XML) 1.0 (Fifth Edition)
    - <http://www.w3.org/TR/xml/>
  - XML je podmnožina (aplikace) SGML (Standard Generalized Markup Language - ISO 8879)
    - Umožňuje přenos dokumentu spolu s popisem jeho struktury (syntaxe)
  - XML se nezabývá způsobem prezentace dokumentů – je to syntaktický nástroj
-

# XML dokument

---

- ❑ XML dokument je **správně formovaný** / dobře vytvořený (well-formed), když:
    - Má úvodní XML deklaraci (prolog)
    - Je dobře uzávorkován
      - ❑ Každý element obsahuje **počáteční** i **koncový tag/závorku**
      - ❑ Odpovídající závorky mají stejné jméno (case sensitivity)  
`<a></A>`
      - ❑ Dvojice závorek se nekříží  
`<a><b></a></b>`
      - ❑ Celý dokument je uzavřen v jediném **kořenovém elementu**
-

# Prolog

---

- ❑ Informace pro SW, že se jedná o XML dokument
    - Musí obsahovat deklaraci verze XML
      - ❑ Máme 1.0 a 1.1
    - Může obsahovat informací o kódování a samostatnosti dokumentu
  - ❑ Deklarace verze:  
`<?xml version="1.1"?>`
  - ❑ Pokud není v UTF-8:  
`<?xml version="1.1" encoding="iso-8859-2"?>`
  - ❑ Pokud je bez odkazů mimo dokument:  
`<?xml version="1.1" standalone="yes"?>`
-



# Elementy

```
<?xml version="1.1" encoding="iso-8859-2"?>
```

```
<zprava>
```

```
<adresa>
```

```
<jmeno>Jan Amos</jmeno>
```

```
<ulice>Severní 12</ulice>
```

```
</adresa>
```

```
<osloveni>Ahoj!</osloveni>
```

```
<text>Pozdrav z <it>říše divů</it>!</text>
```

```
<podpis>Alenka</podpis>
```

```
<priloha/>
```

```
</zprava>
```

Element s  
elementovým  
obsahem

Element s textovým  
obsahem

Element se smíšeným  
obsahem

Prázdný element

Kořenový  
element

```
<priloha></priloha>
```

# Atributy

---

```
<?xml version="1.1" encoding="iso-8859-2"?>
<zprava>
  <adresa>
    <jmeno>Jan Amos</jmeno>
    <ulice>Severní 12</ulice>
  </adresa>
  <osloveni>Ahoj!</osloveni>
  <text>Pozdrav z <it>říše divů</it>!</text>
  <podpis>Alenka</podpis>
  <priloha cesta="obr1.png"/>
</zprava>
```

Element s  
atributy

Název  
atributu

Hodnota  
atributu

# Další prvky XML dokumentu

```
<?xml version="1.1" encoding="iso-8859-2"?>
<zprava>
  <!-- komu zprávu doručit? -->
  <adresa>Jan Amos</adresa>
  <text>
    <![CDATA[
      for (i=0; i < 10; $++)
      {
        document.writeln("<p>Ahoj</p>");
      }
    ]]>
  </text>
  <podpis>Alenka</podpis>
  <datum><?php echo Date("d.m.Y") ?></datum>
</zprava>
```

Komentář

Sekce  
CDATA

Instrukce  
pro  
zpracování



Jazyk DTD



# DTD

---

- Problém: Správná strukturovanost nestačí
    - Potřebujeme omezit sadu značek a jejich obsah
  - Definice typu dokumentu (Document Type Definition – **DTD**) popisuje strukturu (gramatiku) dokumentu
    - Pomocí regulárních výrazů
  - **Validní** XML dokument = správně strukturovaný dokument odpovídající dané gramatice
    - Existují i další jazyky – XML Schema, Schematron, RELAX NG, ...
-

# Struktura validního dokumentu

---

```
<?XML version="1.0" ?>
```

```
<!DOCTYPE kořenový-element [
```

```
...
```

```
<kořenový-element> ... </kořenový-element>
```

Deklarace typu  
dokumentu

- Může být **interní** (gramatika je přímo uvnitř DOCTYPE) nebo **externí** (pouze odkaz na gramatiku uvedenou v externím souboru)
    - Interní nemá moc význam
    - Obojí současně je přípustné
      - Lokální deklarace mají přednost před externími
-

# Příklad: externí a interní DTD

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pozdrav [
    <!ELEMENT pozdrav (#PCDATA)>
]>
<pozdrav>Hello, world!</pozdrav>
```

```
<?xml version="1.0"?>
<!DOCTYPE pozdrav SYSTEM "pozdrav.dtd">
<pozdrav>Hello, world!</pozdrav>
```

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html> ... </html>
```

PUBLIC "veřejný identifikátor" "URI"

# Základní značky DTD

---

- ☐ Deklarace typu dokumentu  
`<!DOCTYPE ... >`
  - ☐ Deklarace typu elementu  
`<!ELEMENT ... >`
  - ☐ Deklarace seznamu atributů  
`<!ATTLIST ... >`
  - ☐ Deklarace entity  
`<!ENTITY ... >`
  - ☐ Deklarace notace  
`<!NOTATION ... >`
-



# Deklarace typu elementu

---

```
<!ELEMENT rodic (potomek*)>
```

```
<rodic>  
  <potomek> ... </potomek>  
  <potomek> ... </potomek>  
  ...  
</rodic>
```

- Název elementu + deklarace přípustného obsahu
    - Prázdný, libovolný, textový, smíšený, elementový
-

# Deklarace typu elementu

,	... sekvence
	... selekce
?	... iterace (0 nebo 1)
+	... iterace (1 a více)
*	... iterace (0 a více)

## ☐ Prázdný obsah

`<!ELEMENT priloha EMPTY>`

## ☐ Libovolný obsah

`<!ELEMENT kontejner ANY>`

## ☐ Textový obsah

`<!ELEMENT prijmeni (#PCDATA)>`

## ☐ Smíšený obsah

`<!ELEMENT text (#PCDATA|it)*>`

## ☐ Elementový obsah

`<!ELEMENT zprava (adresa,text)>`

`(nazev, (autor|editor)?, p*, (nadpis,p+)*)`

# Deklarace atributu

---

Pořadí v dokumentu  
je libovolné

```
<!ATTLIST osoba cislo ID #REQUIRED  
               zamestnan CDATA #FIXED "ano"  
               dovolena (ano|ne) "ne">
```

- ❑ Atributy elementu `osoba`
  - ❑ Atribut `cislo` je unikátní identifikace (ID) a je povinný (#REQUIRED)
  - ❑ Atribut `zamestnan` obsahuje text (CDATA), je konstantní (#FIXED) a má implicitní neměnnou hodnotu (`ano`)
  - ❑ Atribut `typ` je výčet (`ano` nebo `ne`), implicitní hodnota je `ne`
-

# Datové typy atributů

---

- ❑ CDATA – libovolný řetězec znaků
  - ❑ Výčtový typ
  - ❑ ID – jednoznačný identifikátor (**v rámci dokumentu**), musí to být řetězec písmen, cifer a znaků „-“, „\_“, „:“, „.“, nejlépe v ASCII, musí začínat písmenem, nebo znakem „\_“
  - ❑ IDREF – odkaz na ID jiného elementu v rámci dokumentu
  - ❑ IDREFS – seznam odkazů oddělených mezerami
  - ❑ NMTOKEN – hodnota, tj. řetězec podobný jako ID, který ale může začínat cifrou a není jednoznačný
  - ❑ NMTOKENS – hodnoty
  - ❑ ENTITY – odkaz na externí **entitu**
  - ❑ ENTITIES – seznam odkazů
-

# Požadavky na hodnoty atributů

---

- ❑ #REQUIRED – povinný atribut
  - ❑ #IMPLIED – nepovinný atribut
  - ❑ #FIXED – pevná hodnota atributu
-

# Deklarace entity

---

- ☐ Prakticky se využívají pouze triviální případy
  - ☐ Asociace názvu a hodnoty, kterou lze opakovaně využívat
  - ☐ Dělení 1:
    - Parsované = text, kterým je nahrazen odkaz na entitu a stává se součástí XML dokumentu
      - ☐ Odkazujeme referencemi
    - Neparsované = zdroj, jehož obsahem může být cokoli (např. binární data)
      - ☐ Odkazujeme atributem typu ENTITY/ENTITIES
      - ☐ Musí s ní asociována **notace**
  - ☐ Dělení 2:
    - Obecné – v XML dokumentech
    - Parametrické – v DTD
  - ☐ Dělení 3: Interní vs. externí
-

# Znakové entity

---

- Možnost vložení libovolného znaku s daným kódem
  - Hexadecimální nebo decimální

Vyřešte nerovnost  $3x \text{ \&\#x3C; } 5$

- Předdefinované entity pro speciální znaky

Vyřešte nerovnost  $3x \text{ \&lt; } 5$

&	... amp
<	... lt
>	... gt
'	... apos
"	... quot

# Obecné entity

---

## □ Interní entita

- Použití: Opakující se části XML dokumentů

```
<!ENTITY stav "pracovní verze">
```

```
<poznámka>Současný stav dokumentu je  
&stav;</poznámka>
```

## □ Externí parsovaná entita

- Použití: Modularizace XML dokumentů

```
<!ENTITY xml-serial SYSTEM "xml-serial.txt">
```

---



# Obecné entity

- Externí neparsovaná entita
  - Použití: Odkaz na ne-XML data

nebo PUBLIC

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE zprava [
  <!NOTATION avi SYSTEM
    "C:/Program Files/Video Player/Player.exe">
  <!ENTITY video SYSTEM "video.avi" NDATA avi>
  <!ELEMENT video-dovo (#PCDATA)>
  <!ATTLIST video-dovo src ENTITY>
]>
```

Deklarace notace

```
<zprava>Přikládám video z dovolené <video-dovo
src="video"/>.</zprava>
```

# Parametrické entity

---

## □ Interní entita

- Použití: Opakující se části DTD

```
<!ELEMENT clanek (automobil*)>
<!ENTITY % atributy
    "barva (modra|bila|cerna) #REQUIRED
    rychlost (velka|mala) #IMPLIED" >
<!ELEMENT automobil (#PCDATA)>
<!ATTLIST automobil %atributy; >
<!ELEMENT motocykl (#PCDATA)>
<!ATTLIST motocykl %atributy; >
<!ELEMENT kolo (#PCDATA)>
<!ATTLIST kolo %atributy; >
```

---

# Parametrické entity

---

## □ Externí entita

- Použití: Modularizace DTD

```
<!ENTITY % ISOLat2 SYSTEM "iso-pub.ent">  
...  
%ISOLat2;  
...
```

# Podmíněné sekce

---

```
<!ENTITY % draft 'INCLUDE' >
<!ENTITY % final 'IGNORE' >

<![%draft;[
<!ELEMENT book (comments*, title, body,
supplements?)]>
<![%final;[
<!ELEMENT book (title, body, supplements?)]>
```

# DTD – větší příklad

---

```
<?xml version="1.0" encoding="windows-1250"?>
<!ELEMENT zaměstnanci (osoba)+>
<!ELEMENT osoba (jméno, email*, vztahy?)>
    <!ATTLIST osoba id ID #REQUIRED>
    <!ATTLIST osoba poznámka CDATA #IMPLIED>
    <!ATTLIST osoba dovolená (ano|ne) "ne">
<!ELEMENT jméno ((křestní, příjmení)|(příjmení,
křestní))>
<!ELEMENT křestní (#PCDATA)>
<!ELEMENT příjmení (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT vztahy EMPTY>
    <!ATTLIST vztahy nadřízený IDREF #IMPLIED>
    <!ATTLIST vztahy podřízení IDREFS #IMPLIED>
```



# Přehled XML technologií



# Technologie XML

---

- XML = eXtensible Markup Language
  - Technologie XML = sada souvisejících technologií
    - Rozhraní pro práci s XML daty – DOM, SAX
    - Popis struktury XML dokumentů – DTD, XML Schema
    - Dotazování nad XML daty – XPath, XQuery
    - Aktualizace XML dat – XQuery Update
    - Transformace XML dat – XSLT
-

# Související problematika

---

## ☐ Standardní XML formáty

- XHTML, OpenOffice, MathML, SVG, ...

## ☐ Persistence XML dat

- Nativní XML databáze
  - Relační databáze s XML
  - SQL/XML
-





Konec

