



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost



TECHNICKÁ
UNIVERZITA
V LIBERCI
www.tul.cz

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Realizováno za finanční podpory ESF a státního rozpočtu ČR
v rámci v projektu *Zkvalitnění a rozšíření možností studia
na TUL pro studenty se SVP* reg. č. CZ.1.07/2.2.00/29.0011

Jazyky pro popis dat

Pavel Satrapa

pavel.satrapa@tul.cz

Literatura

- Aaron Skonnard, Martin Gudgin:
XML pohotová referenční příručka
Grada, 2005
- stránky Jiřího Koska
www.kosek.cz
- stránky WWW konsorcia
www.w3.org



Historie



Datové soubory

- **binární**

- Pascal: **file of** *Zamestnanec*
- rychlé zpracování (přímý přenos do/z paměti)
- srozumitelné jen pro zasvěcené programy

- **textové**

- zápis/čtení vyžaduje konverzi (i dost složitou)
- (víceméně) srozumitelné, snadno editovatelné
- ideální pro přenositelnost mezi programy

Tradiční textové soubory

- **pevná struktura**

- např. */etc/passwd* – pevný význam položek i oddělovač
- `satrapa:x:500:500:Pavel Satrapa:/home/satrapa:/bin/zsh`
- úsporné, ale nepřehledné

- **volná struktura**

- syntaktická pravidla umožňují jistou volnost v pořadí a/nebo formátu souboru
- např. CSS definice

Příklad: CSS definice

```
p {  
  margin: 0,5em 0;  
}
```

```
h1, h2, h3 {  
  font-family: sans-serif;  
  font-weight: bold;  
  color: blue;  
}
```

Formalizace – značkovací jazyky

- historie: **typografické jazyky**
 - cíl: popsat pomocí příkazů přímo v textu požadovaný vzhled, resp. charakterizovat části textu
 - *roff, T_EX
 - zkušenost: výhodný je strukturální popis
- nověji: **jazyky popisující strukturu dat**
 - cíl: jednotný mechanismus, obecné nástroje pro zacházení s nimi (kontroly, konverze apod.)

SGML (1)

- **1969: Generalized Markup Language (GML)**
 - vyvinulo IBM pro práci s dokumenty
 - editace, formátování, vyhledávání informací
- **1986: Standard Generalized Markup Language (SGML)**
 - přijato ISO jako mezinárodní standard
 - metajazyk: jazyk pro definici jazyků
 - popisuje struktury a významy, nikoli vzhled

SGML (2)

- **aplikace SGML:**
 - HTML
 - DocBook
- **vzhled:**
 - Document Style Semantics and Specification Language (DSSSL)
 - nověji lze i CSS
- příliš složité – málo implementací



XML



XML

- **eXtensible Markup Language**
- přijalo WWW konsorcium (1998)
- podmnožina SGML – jednodušší
 - míněno jako adaptace SGML pro web
- snazší implementace – rychlé rozšíření
- **metajazyk**, definuje
 - základní syntaktická pravidla dokumentů
 - nástroje pro popis prvků jazyka a jejich vztahů

Příklad XML dokumentu

The diagram shows an XML document structure with the following elements and annotations:

- `<?xml version="1.0" encoding="ISO-8859-2"?>` is annotated as *prolog (deklarace jazyka a kódování)*.
- `<diskoteka>` is annotated as *kořenový prvek*.
- `<disk id="disk1">` is annotated as *běžné prvky*.
- `<interpret>Divokej Bill</interpret>` is annotated as *běžné prvky*.
- `<nazev>Propustka do pekel</nazev>` is annotated as *běžné prvky*.
- `</disk>` is annotated as *běžné prvky*.
- `<disk id="disk2">` is annotated as *běžné prvky*.
- `<interpret>AC/DC</interpret>` is annotated as *běžné prvky*.
- `<nazev>Highway to Hell</nazev>` is annotated as *běžné prvky*.
- `</disk>` is annotated as *běžné prvky*.
- `</diskoteka>` is annotated as *běžné prvky*.

Arrows point from the annotations to the corresponding XML elements.

Možnosti a omezení XML

- XML definuje **strukturu dokumentu**
- obecné XML nástroje umožňují
 - zkontrolovat dokument
 - transformovat jej do jiného formátu (XML i ne-XML)
- XML **nic neříká o sémantice**, významy prvků musí znát aplikace
- „program X podporuje XML“ neznamená „program X rozumí jakémukoli XML dokumentu“

XML prolog

- nepovinný
- `<?xml version="1.0" ... ?>`
- nepovinné součásti:
 - `encoding="název"` – použité kódování, implicitně UTF-8
 - `standalone="yes|no"` – zda je dokument soběstačný (neodkazuje se na externí definice)

Prvky a značky

- dokument je strukturován rozdělením do prvků
- značka (tag) je zápisem prvku
- zápis: `<jméno>...</jméno>`
- ve jménech se rozlišují malá/velká písmena
- mezi značkami je obsah prvku (text, další prvky...)
- lze vnořovat: `<a>.........`
- nesmí se překřížit: `<a>.........` je chyba

Prázdné prvky

- nemá-li prvek obsah, je nutno ho ihned ukončit
`<prvek></prvek>`
- nebo explicitně vyjádřit, že je prázdný
`<prvek/>`
(v XHTML se doporučuje předřadit mezeru kvůli starým klientům `<prvek />`)
- obě varianty jsou ekvivalentní

Kořenový prvek

- celý dokument je vždy zabalen do jednoho prvku
- označován jako kořenový (též dokumentový) prvek
- pro daný jazyk je pevně definován
- např. pro XHTML se jedná o prvek `html`, u DocBooku `article` apod.

Atributy

- doplňkové informace o prvku
 - interní informace (identifikátor)
 - ovlivnění vzhledu (CSS styl, třída apod.)
- zapisují se do zahajující značky
 - `<div class="poznamka">...</div>`
- atribut musí mít hodnotu (checked="checked")
- hodnota musí být uzavřena do uvozovek
"..." nebo '...'

Atribut nebo vnořený prvek?

<cd id="disk1">

<nazev>Strach</nazev>

</cd>

<cd>

<id>disk1</id>

<nazev>Strach</nazev>

</cd>

- neexistuje striktní pravidlo, rozhoduje cit
- data, která jsou součástí popisovaných objektů, raději jako prvky
- data popisná/organizační lépe do atributů
- hierarchie je mezi prvky, nikoli atributy

Entity

- jak vyjádřit speciální znaky (např. <)?
- pomocí entit ve tvaru
&jméno;
- standardní entity pro speciální znaky:
 - *<* <
 - *>* >
 - *&* &
 - *"* “
 - *'* ’

Komentáře

- ignorovány při zpracování (oficiálně nejsou považovány za součást textu dokumentu)
- tvar:
`<!-- text komentáře -->`
- text komentáře nesmí obsahovat --, nelze vnořovat
- dají se použít k vypuštění části textu:
`<!--
<p>Ignorovaný text.</p>
-->`

Instrukce pro zpracování

- doplňkové informace pro obslužný program

- tvar

<?cíl data?>

cíl identifikuje, komu/k čemu jsou informace určeny,
data často sekvence *jméno*=“*hodnota*”; např.

<?xml-stylesheet href=“styl.css” type=“text/css”?>

- nejsou považovány za součást textu dokumentu, ale musí být předány zpracovávajícímu programu

CDATA

- někdy se hodí, aby určitou část textu XML procesor neanalyzoval, nehledal v ní prvky apod., ale opsal ji beze změny
- toto chování má obsah sekce CDATA
`<![CDATA[neinterpretovaný text]]>`
- vhodný např. pro ukázky XML kódu v textech o XML



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Realizováno za finanční podpory ESF
a státního rozpočtu ČR v rámci v projektu
*Zkvalitnění a rozšíření možností studia na TUL
pro studenty se SVP*
reg. č. CZ.1.07/2.2.00/29.0011