PROGRAMOVÉ STRUKTURY: PYTHON A XML

Úvod do XML, parsery SAX a DOM, ukázkové příklady

XML

- eXtensible Markup Language
  - rozšířitelný značkovací jazyk
- Vyvinut a standardizován W3 konsorciem (W3C)
- Možnosti použití
  - výměna dat mezi aplikacemi
  - publikování dokumentů
    - popisuje obsah, ne vzhled
- Validace a "dobrá" forma
  - DTD a XSD
  - definují, jaká značky, atributy a typy bude XML dokument obsahovat
  - parser zkontroluje, zda XML odpovídá definici

Základní syntaxe

- XML dokument je textový dokument
  - zpravidla psán v kódování UTF-8
- Well-formed = dobře strukturovaný
  - jeden kořenový element
  - neprázdné elementy musí být ohraničeny startovací a ukončovací značkou
    - např. `<ovoce>Jablko</ovoce>`
  - všechny hodnoty atributů musí být uzavřeny v uvozovkách
    - jednoduchých (')
    - dvojitých (")
  - elementy mohou být vnořeny
    - nesmí se překrývat

Ukázkový XML dokument

- Dokument `jidlo.xml`
- Používají ho všechny následující programy

DTD

- Document Type Definition
- Poměrně starý a málo expresivní jazyk
  - nevýhodou je, že není XML dokument
- Jazyk pro popis struktury XML či SGML dokumentu
- Omezuje množinu přípustných dokumentů spadajících do daného typu či třídy
  - např. vymezuje jazyky HTML a XHTML
- Popisuje strukturu třídy nebo typu dokumentu
  - popisem jednotlivých elementů a atributů
- Popisuje uspořádání a vnořování značek
- Vymezuje atributy pro každou značku a typ těchto atributů
- Připojení ke XML:

```
<!DOCTYPE kořen SYSTEM "soubor.dtd">
```

Dokument jidlo.dtd


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

<!ELEMENT jidlo (ovoce, zelenina, pecivo) >

<!ELEMENT ovoce (polozka)* >

<!ELEMENT zelenina (polozka)* >

<!ELEMENT pecivo (polozka)* >

<!ELEMENT polozka (nazev, mnozstvi) >
<!ATTLIST polozka kod NMTOKEN #REQUIRED>

<!ELEMENT nazev (#PCDATA)>

<!ELEMENT mnozstvi (#PCDATA)>
<!ATTLIST mnozstvi cena NMTOKEN #REQUIRED jednotka NMTOKEN #REQUIRED>
```

XSD

- XML Schema Definition
- Popisuje strukturu XML dokumentu
- Definuje
  - místa v dokumentu, na kterých se mohou vyskytovat různé elementy
  - atributy elementů
  - které elementy jsou potomky jiných elementů
  - pořadí a počty elementů
  - zda element může být prázdný nebo musí obsahovat text
  - datové typy elementů a jejich atributů
  - standardní hodnoty elementů a atributů

Ukázkový XSD dokument

- Dokument jidlo.xsd
- Slouží k validaci XML dokumentu `jidlo.xml`

Aplikace XML

- XHTML - eXtensible HyperText Markup Language
  - nástupce HTML (do verze 4.01)
- RDF – Resource Description Framework
  - možný způsob pro popis metadat
  - např. obsah a anotace HTML stránky
- RSS – rodina XML formátů
  - vhodné pro čtení novinek na Webu
- SMIL – Synchronized Multimedia Integration Language
  - popisuje multimedia pomocí XML
- MathML – Mathematical Markup Language
  - popis matematických vzorců a symbolů pro použití na Webu

Aplikace XML (2)

- SVG – Scalable Vector Graphics
  - popis dvourozměrné vektorové grafiky
  - statické i dynamické (animace)
- DocBook
  - sada definic dokumentů a stylů pro publikační činnost
- Jabber
  - protokol pro IM (Instant Messaging)
- SOAP – Simple Object Access Protocol
  - protokol pro komunikaci mezi webovými službami
- Office Open XML, OpenDocument
  - souborový formát pro ukládání a výměnu dokumentů
  - balíky MS Office, LibreOffice, OpenOffice, …

Verze XML

- Aktuální verze je 1.1 (od 16. srpna 2006)
- První verze byla 1.0 (běžně používána)
- Verze se liší v požadavcích na použité znaky v názvech elementů, atributů, …
  - verze 1.0 dovolovala pouze užívání znaků platných ve verzi Unicode 2.0
    - obsahuje většinu světových písem
    - neobsahuje později přidané, např. Mongolštinu
  - verze 1.1 zakazuje pouze řídící znaky
    - mohou být použity jakékoliv znaky
- Obě verze dovolují v obsahu dokumentu jakékoliv znaky
  - názvy elementů v nově přidaném jazyku = použití verze 1.1

Související technologie

- Jmenné prostory v XML
  - umožňují kombinovat značkování podle různých standardů v jednom dokumentu
    - např. dokument `tabulky.xml`
- XSLT – eXtensible Stylesheet Language Transformations
  - transformace XML dokumentu na jiný, odvozený dokument
    - v XML, HTML nebo textový
- XPath, Xquery
  - dotazovací jazyky nad XML dokumenty

Dokument tabulky.xml


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

<hv>
        <v:tabulka xmlns:v="vynosy">
                <v:trzby>1582000</v:trzby>
                <v:uroky>1500</v:uroky>
                <v:ostatni>120000</v:ostatni>
        </v:tabulka>

        <n:tabulka xmlns:n="naklady">
                <n:spotrebovane_nakupy>950000</n:spotrebovane_nakupy>
                <n:uroky>89000</n:uroky>
                <n:sluzby>280000</n:sluzby>
                <n:ostatni>350000</n:ostatni>
        </n:tabulka>
</hv>
```

Parser SAX

- Simple API for XML
- Sériový přístup k XML
- Proudové zpracování
  - dokument se rozdělí na jednotlivé části
  - následně se volají jednotlivé události
    - které ohlašují nalezení konkrétní části
  - způsob zpracování událostí je na programátorovi
- Vhodné, pokud se čte obsah celého souboru
- Nízké paměťové nároky
- Vysoká rychlost
- Nelze zapisovat do stejného XML souboru

Parser DOM

- Document Object Model
- Objektově orientovaná reprezentace XML
- Umožňuje modifikaci obsahu i struktury XML dokumentu
- Umožňuje přístup k dokumentu jako ke stromu
- Celý XML dokument je načten do paměti
- u velkých dokumentů náročné na paměť
- Vhodné použít tam, kde přistupujeme k elementům náhodně

Vytvoření XML dokumentu

- Program `txt_to_xml.py`
- Nepoužívá žádné moduly pro práci s XML
- Na vstupu je textový soubor `databaze.txt`

Vstupní soubor databaze.txt


```
638512041589;Jan Novák;M;Jana Nováková;Petr Novák;;
529828296949;Petr Novák;M;Olga Nováková;Hynek Novák;Jana Nováková
325663431324;Hynek Novák;M;Pavla Nováková;Zbyněk Novák;Olga Nováková
533366644665;Zbyněk Novák;M;;;Pavla Nováková
966644855202;Pavla Nováková;Z;;;Zbyněk Novák
111000200001;Jan Novák;M;Pavla Nováková;Zbyněk Novák;;
330000333000;Petra Velká;Z;;;Josef Velký
220000222000;Josef Velký;M;;;Petra Velká
354235335632;Olga Nováková;Z;Petra Velká;Josef Velký;Hynek Novák
558715826998;Jana Nováková;Z;Petra Slaná;Josef Slaný;Petr Novák
541174414468;Josef Slaný;M;Lucie Slaná;Lubor Slaný;Petra Slaná
663339993331;Lucie Slaná;Z;;;Lubor Slaný
221113339991;Lubor Slaný;M;;;Lucie Slaná
```

Modul xml.sax

- Import požadovaných objektů z knihovny

```
from xml.sax import handler, sax_parser
```

- Definice handleru našeho XML dokumentu

```
class MySaxDocumentHandler(handler.ContentHandler):
```

- Tato třída obsahuje metody, které obsluhují události parseru
  - my musíme napsat obsluhy těchto událostí
  - tj. přetížit metody v třídě `handler.ContentHandler`
    - `startElement(),endElement(),characters(),...`
- Vytvoření instance našeho handleru

```
handler = MySaxDocumentHandler()
```

- Vytvoření instance parseru

```
parser = make_parser()
```

- Nastavení parseru – předání handleru

```
parser.setContentHandler(handler)
```

- Parsování

```
parser.parse(xmlFile)
```

Práce s modulem xml.sax

- Jednoduché ukázkové programy
  - všechny vyžadují vstupní argument `jidlo.xml`
- Zjištění správné formy XML dokumentu
  - program `sax_verify.py`
- Práce s elementy
  - program `sax_elementy.xml`
  - vypíše seznam názvů ovoce
- Práce s atributy elementů
  - program `sax_atributy.xml`
  - vypíše seznam názvů ovoce, které je vážené na kg

Modul xml.dom

- Celý XML dokument v paměti
  - v podobě stromu
- Uzly stromu jsou nody, mohou být typů (vybráno)
  - `NODE` – základní prvek a předek všech dalších druhů nodů
  - `DOCUMENT` – počáteční (kořenový) uzel
  - `ELEMENT` - element
  - `TEXT` – textový obsah elementu

Modul xml.dom (2)

- Import požadovaných objektů z knihovny

```
from xml.dom import minidom, Node
```

- Parsování (načtení) dokumentu

```
doc = minidom.parse(xmlFile)
```

- Získání kořene stromu

```
rootNode = doc.documentElement
```

- Zjištění určitých potomků – vrací jejich seznam

```
rootNode.getElementsByName('ovoce')
```

Práce s modulem xml.dom

- Jednoduché ukázkové programy
  - všechny vyžadují vstupní argument `jidlo.xml`
- Zjištění správné formy XML dokumentu
  - program `dom_verify.py`
- Práce s elementy
  - program `dom_elementy.xml`
  - vypíše seznam názvů ovoce
- Práce s atributy elementů
  - program `dom_atributy.xml`
  - vypíše seznam názvů ovoce, které je vážené na kg
- Přidání nového složeného elementu
  - program `dom_add.py`
  - výsledek vypíše na obrazovku a uloží do nového souboru