Python a XML

Obsah

- XML
- Validace – DTD a XSD
- Práce s XML - SAX a DOM
- Python a XML
- Tvorba XML bez použití knihoven
- Knihovna PyXML – SAX
- Knihovna PyXML – DOM
- Knihovna LXML – validace DTD a XSD

XML

- eXtensible Markup Language („rozšiřitelný značkovací jazyk“)
- Vyvinut a standardizován W3C
- Výměna dat mezi aplikacemi
- Publikování dokumentů – popisuje obsah ne vzhled (styly)
- Styly – vzhled CSS, transformace XSL
- DTD, XSD – definují jaké značky, atributy, typy bude XML obsahovat
- Parser zkontroluje, zda XML odpovídá definici

Syntaxe XML

- XML dokument je text, Unicode – zpravidla UTF-8
- „well-formed“ = správně strukturovaný
  - Jeden kořenový element
  - Neprázdné elementy musí být ohraničeny startovací a ukončovací značkou (<ovoce>Jablko</ovoce>)
  - Prázdné elementy mohou být označeny tagem „prázdný element“ (<ovoce/>)
  - Všechny hodnoty atributů musí být uzavřeny v uvozovkách – jednoduchých (') nebo dvojitých ("), ale jednoduchá uvozovka musí být uzavřena jednoduchou a dvojitá dvojitou. Opačný pár uvozovek může být použit uvnitř hodnot
  - Elementy mohou být vnořeny, ale nemohou se překrývat; to znamená, že každý (ne kořenový) element musí být celý obsažen v jiném elementu
- Příklad jidlo.xml

DTD

- Document Type Definition
- jazyk pro popis struktury XML případně SGML dokumentu
- Omezuje množinu přípustných dokumentů spadajících do daného typu nebo třídy
- DTD tak například vymezuje jazyky HTML a XHTML.
- Struktura třídy nebo typu dokumentu je v DTD popsána pomocí popisu jednotlivých elementů a atributů. Popisuje jak mohou být značky navzájem uspořádány a vnořeny. Vymezuje atributy pro každou značku a typ těchto atributů.
- Připojení ke XML: `<!DOCTYPE kořen SYSTEM "soubor.dtd">`
- Příklad DTD
- DTD je poměrně starý a málo expresivní jazyk. Jeho další nevýhoda je, že DTD samotný není XML soubor.

XSD

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

Aplikace XML

- XHTML – Nástupce jazyka HTML.
- RDF – Resource Description Framework, specifikace, která umožňuje popsat metadata, např. obsah a anotace HTML stránky.
- RSS – je rodina XML formátů, sloužící pro čtení novinek na webových stránkách
- SMIL – Synchronized Multimedia Integration Language, popisuje multimedia pomocí XML.
- MathML – Mathematical Markup Language je značkovací jazyk pro popis matematických vzorců a symbolů pro použití na webu.
- SVG – Scalable Vector Graphics je jazyk pro popis dvourozměrné vektorové grafiky, statické i dynamické (animace).
- DocBook – Sada definic dokumentů a stylů pro publikační činnost
- Jabber – Protokol pro Instant messaging
- SOAP – Protokol pro komunikaci mezi Webovými službami
- Office Open XML, OpenDocument – Souborový formát určený pro ukládání a výměnu dokumentů vytvořených kancelářskými aplikacemi

Verze XML

- Aktuální verze XML je 1.1 (od 16. srpna 2006)
- První verze XML 1.0
- Obě verze se liší v požadavcích na použité znaky v názvech elementů, atributů atd.
  - Verze 1.0 dovolovala pouze užívání znaků platných ve verzi Unicode 2.0, která obsahuje většinu světových písem, ale neobsahuje později přidané sady jako je Mongolština a podobně.
  - Verze XML 1.1 zakazuje pouze řídící znaky, což znamená, že mohou být použity jakékoli jiné znaky.
- Je třeba poznamenat, že omezení ve verzi 1.0 se vztahuje pouze na názvy elementů a atributů. Jinak obě verze dovolují v obsahu dokumentu jakékoli znaky. Verze 1.1 je tedy nutná, pokud potřebujeme psát názvy elementů v jazyku, který byl přidán do Unicode později.

Související technologie

- Jmenné prostory v XML - Umožňují kombinovat značkování podle různých standardů v jednom dokumentu (příklad `tabulky.xml`)
- XSLT - Transformace dokumentu v XML na jiný, odvozený dokument - v XML, HTML nebo textový.
- XQuery - Dotazy nad daty v XML.

Práce s XML - SAX

- SAX – Simple API for XML
  - Sériový přístup ke XML
  - Proudové zpracování, při kterém se dokument rozdělí na jednotlivé jeho části
  - Pak se volají jednotlivé události, které ohlašují nalezení konkrétní části
  - Způsob jejich zpracování je na programátorovi
  - Vhodné pokud se čte celý obsah souboru
  - Nízké paměťové nároky, vysoká rychlost, nelze zapisovat

Práce s XML - DOM

- Document Object Model
- Objektově orientovaná reprezentace XML
- Umožňuje modifikaci obsahu, struktury
- DOM umožňuje přístup k dokumentu jako ke stromu
- Celý XML dokument v paměti (náročné na paměť)
- Vhodné tam, kde přistupujeme k elementům náhodně

Tvorba XML bez použití knihoven

- Příklad txt_to_xml.py

XML knihovny Pythonu

- PyXML
  - http.//sourceforge.net/projects/pyxml
- LXML
  - http://pypi.python.org/pypi/lxml

PyXML – SAX

- Parser čte XML dokument a pro každou dílčí část vyvolá událost
- My musíme napsat obsluhu události (handler)
- Import z knihovny:
  - `from xml.sax import handler, make_parser`
- Vytvoření parseru:
  - `parser = make_parser()`
- Parsování:
  - `parser.parse(infile)`
- Vytvoření třídy handleru:
  - `Class MySaxHandler(handler.ContentHandler)`
  - Uvnitř např. metody `startElement`
- Nastavení handleru:
  - `parser.setContentHandler(handler)`
- Zjištění well-formed: Příklad `sax_ver.py`
- Práce s elementy: Příklad `sax_elem.py`
- Práce s atributy: Příklad `sax_elem.py`

PyXML – DOM

- Všechny objekty v paměti, stromová struktura
- Uzly stromu – nody
- Typy nodů
  - Node – základní prvek a předek dalších druhů nodů
  - Document – počáteční uzel
  - Attr – atribut
  - Element – element
  - Text – obsah elementu
- Knihovny pro práci s DOM – **minidom**, ElementTree,...
- Parsování: `doc = minidom.parse(inFile)`
- Kořen stromu: `rootNode = doc.documentElement`
- Zjištění určitých potomků: `ovoce = rootNode.getElementsByTagName("ovoce")`
- Příklady: `dom_ver.py, dom_elem.py, dom_atr, dom_add.py`

LXML – validace DTD

- Import z knihovny:
  - `from lxml import etree`
- Postup validace:
  - `doc = etree.parse(xmlName)`
  - `dtd = etree.DTD(dtdfile)`
  - `if dtd.validate(doc) == True: ...`
- Příklad: `val_dtd.py`

LXML – validace XSD

- Postup validace:
  - `doc = etree.parse(xmlName)`
  - `xmlschema_doc = etree.parse(xsdfile)`
  - `xmlschema = etree.XMLSchema(xmlschema_doc)`
  - `if xmlschema.validate(doc) == True:`
    `...`
- Příklad `val_xsd.py`