Python XML a Web

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`)
- XML Schema -Předpis struktury a datových typů pro třídu dokumentů v 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`

Obsah

- Webové frameworky
- XIST
- MVC architektura
- web2py
- Zope

Webové frameworky (1)

- XIST, HTMLTags, HTMLgen, HyperText
  - Tvorba statických stránek
  - Stránka je programově sestavena, lze ji uložit na disk
  - Lze snadno a rychle vygenerovat stránky
  - Použití spíše jako výstup programu (export sady stránek)
- Django
  - Oblíbená rozsáhlá knihovna, rychlý vývoj
  - Tvorba relačních webů – admin rozhraní, uživatelská rozhranní s různými právy

Webové frameworky (2)

- TurboGears
  - Velká knihovna
  - Tvorba webů založených na databázi (obsahuje např. knihovnu SQLObject – mapování objektů do relačních databází)
- Zope
  - Tvorba serverů se správou objemných systémů
  - O-o skriptovací skriptovací jazyk
  - Obsahuje ZODB – databází objektů
  - Nová verze Zope 3

Webové frameworky (3)

- Další
  - CherryPy
  - Pylons
  - web.py
  - QP, Gizmo
  - ...
- web2py (dříve Gluon)
  - Vývoj přes webové rozhranní
  - Využívá řízení psané v pythonu
  - Vícejazyčná podpora
  - Logování chyb

XIST

- Rozšířený HTML/XML generator
- http://www.livinglogic.de/Python/xist/
- Předchůdci: HTMLgen, HyperText
- Stromová tvorba HTML, parsing HTML transformace
- Příklad tvorba HTML

MVC architektura

- Často aplikovaný model při programování webu
  - MPO – Model, Pohled, Ovladač
  - MVC – Model, View, Controller
- Model – provádí obchodní logiku (práce s DB)
- Pohled – část zajišťující formátování výstupu systému
- Ovladač – zpracovává vstup a předává ho modelu, řídí aplikaci
- Oddělení aplikační logiky od zobrazování je vhodné
  - Aplikace je pružnější, snadno modifikovatelná
  - Kód vypadá přehlednější
  - Roste opětovná použitelnost částí systému

web2py

- Redakční systém založený na pythonu (řízení aplikace)
- Nemusí se instalovat, pouze se stáhnou knihovny:
  http://mdp.cti.depaul.edu (je zde i tutorial)
- Po stažení spustit web2py.exe (Windows)
- Nastavit adresu (localhost = 127.0.0.1), administrátorské heslo, port (8000)
- Otevře se admin prostředí
- Lze vytvořit novou aplikaci (MujWeb), upload existující

web2py – záložky

- errors – logované chyby
- cleanup – smaže všechna chybová hlášení v errors
- packall, compile – zabalení, kompilování
- design – tvorba/úprava webu

web2py – design

- Models – slouží k definování reprezentace dat, databázových tabulek, sestav
- Controllers – řízení, aplikační logika, každé url je mapováno na funkci v controllers (python)
- Views – prezentační vrstva, šablony, soubory, skrze které se provádí výstup z funkcí v controllers (html)
- Languages – překlad slov v aplikaci, pro vícejazyčné weby, překlad lze vkládat editačním polem
- Static file – vložení statických souborů (obrázky, javascript, css, text)

web2py – defaultní soubory

- Nová aplikace oddděděná od exmples, automaticky voženy některé soubory
- Controllers
  - appadmin.py – základní funkce pro práci s HTML stránkou
  - default.py – obsahuje základní zprávu zpbrazenou na stránce (index)
- Views
  - layout.html – šablona budoucí HTML stránky, základ našeho webu, obsahuje layout, css
  - default\index.html – vstupní stránka aplikace

Web2py – základní technologie

- Vytvoříme pythonovský soubor v Controllers – pokusy.py
- V něm funkci hello() vracející „ahoj světe“
- Výstup v prohlížeči:

  http://127.0.0.1:8000/[jméno_aplikace]/[controller]/[funkce]

  http://127.0.0.1:8000/MujWeb/pokusy/hello
- Pokud ve Views přidáme soubor pokusy.hello.html, pak výstup funkce hello() bude zobrazen skrze tento soubor (šablonu)

Web2py – další vlastnosti

- Výstup z funkce lze značkovat (return T(„ahoj světe“)), potom se řetězec vloží do slovníku pro překlad (v Languages)
- Výstup skrze šablonu: controller předá šabloně data:

  return dict(message=T(„ahoj světe“)), kde message je jméno proměnné kterou můžeme použít v šabloně
- Ve view vytvoříme soubor pokusy/hello.html:

  {{extend ‘layout.html’}}

  <h1>{{=message}}</h1>
  - Extend udává jméno šablony od které se bude dědit
  - {{python code}}
  - Na místo {{=message}} se vloží obsah proměnné message

Web2py – další vlastnosti

- Změna šablony v controlleru:
  - response.view='pokusy/hello'
- Vytvoření stránky nepodléhající šabloně:
  - Return HTML(BODY(H1(T('ahoj světe',_style="color: red;"),_bgcolor="black")),xml()
- Přesměrování:
  - Redirect(URL(r=request,f='ahoj'))
  - Přesměrovává na funkci ahoj ve stejném .py souboru
- Generování stránky s chybou:

```
def raisehttp():
    raise HTTP(400,"vnitrni chyba: spatny pozadavek")
```

Web2py – další vlastnosti

- Status:
  - return dict(request=request,session=session, response=response)
  - Použije se implicitní šablona generic.html
- Práce se session:

```
def pocitaniNacteni():
    if not session.counter: session.counter=0
    session.counter+=1
    return dict(nacteno=session.counter)
```

- Použití cyklu v šabloně:

```
<h3>{{for i in range(nacteno):}}{{=i}}...{{pass}}</h3>
( range(4) => [0,1,2,3] )
```

Web2py – další vlastnosti

- Layout – tvorba menu

```
def index():
    response.menu=[['index',True,URL(r=request,f='index')],['druhá
    strana',False,URL(r=request,f='strana2')]]
    response.flash='kliknul jsi na index'
    return dict(message="toto je index")
def strana2():
    response.menu=[['index',False,URL(r=request,f='index')],['druhá
    strana',True,URL(r=request,f='strana2')]]
    response.flash='kliknul jsi na druhou stranu'
    return dict(message="toto je druhá strana")
```

  - response.menu – vložení do menu nepozicovaném v css layout.html
  - Struktura položky menu: text, zvýraznění, url

Web2py – další vlastnosti

- Testy, cykly, proměnné uvnitř šablony (menu/strana2), výjimky, funkce
- Formuláře a jejich validace
- Práce s DB

Zope - technologie

- Akulální verze: 3
- Instalace serveru, jeho instance
- Publikování objektů
- Zope ORB – Zope Objekt Request Broker (zprostředkovatel požadavků na objekty)
- Vlastní konfigurační jazyk: ZCML

Zope - návody

- Instalace:

  http://zeapartners.org/scl/2006/04/25/z3-install/z3-install.html

- Hello World:

  http://zeapartners.org/scl/2006/04/25/z3-helloworld/z3-helloworld.html

- Book Marker:

  http://zissue.berlios.de/z3/Zope3In30Minutes.html