

Aufbauseminar Python

Wintersemester 2016/17

Ahmed Albawabiji

Professur für Korpuslinguistik
Studiengang Linguistische Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg
ahmed.albawabiji@fau.de

Erlangen, 17.01.2017



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

PHILOSOPHISCHE FAKULTÄT
UND FACHBEREICH THEOLOGIE

Inhaltsverzeichnis

1 Einführung

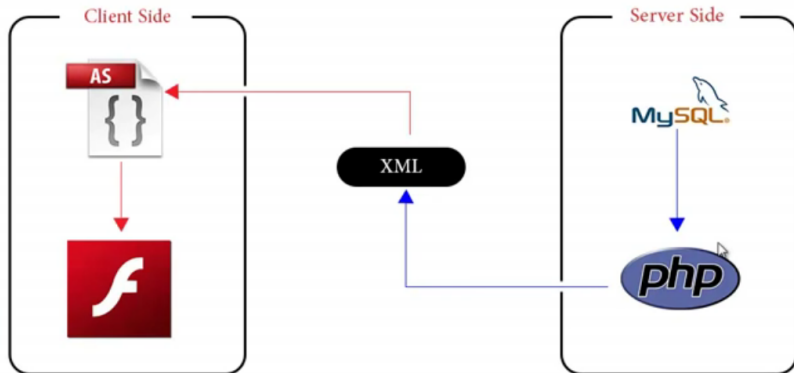
2 Hauptteil

- XML-Dokumente
- XML-Tree
- Aufgaben

3 Literaturverzeichnis

Abstakt:XML?

- XML ist eine Meta-Auszeichnungssprache zur Beschreibung strukturierter Daten.
- XML ist Text, aber nicht zum Lesen
- XML speichert die Daten in einer Verzeichnisstruktur
- Bei HTML geht es primär um die optische Darstellung der Daten. Mit XML alleine kann man nichts machen(außer Daten zu speichern).
- XML ist unterstützt von sehr vielen Softwarepaketen und Programmiersprachen , so dass er auf den unterschiedlichsten Medien, Browser, Handy, Drucker dargestellt werden kann
- Selbsterklärende Grundstruktur



Motivation

Daten speichern bzw. austauschen:

Otto, der in der Xstr.14 wohnt, hat von uns einen Laptop für 500 Euro erworben.

Früher wurde es so gemacht:

Laptop // Otto Xstr.14// 500 Euro

Es fehlt aber die Bedeutung der Daten, um sie interpretieren zu können.

Problem beim Weiterleiten!

XML als Lösung

Selbstbeschreibende Nachricht

```
<Transaktion art="kaufen">  
  <Ware>Laptop</Ware>  
  <Kundendaten>  
    <Kundenname>Otto</Kundenname>  
    <Adresse>Xstr.14</Adresse>  
  </Kundendaten>  
  <Betrag>500</Betrag>  
</Transaktion>
```

Einkunft über diese Nachricht zwischen Systemen (sendenden, empfangenden System)....deswegen...! XML

jetzt ist das Schema dieser Nachricht in der Nachricht mitrein.(XML)

anstatt Delimiters, benennt man die Felder

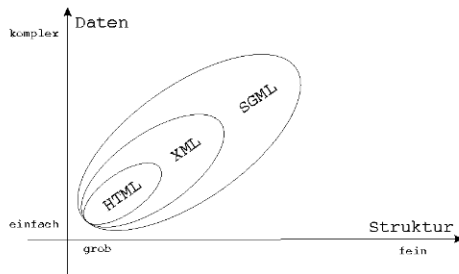
Ich beschreibe, was diese Felder bedeuten.(Meta-Daten) Jetzt hab ich Elemente drin

Am Ende hat man selbstbeschreibende Nachricht

Meta-Daten sollen klar sein, damit der Empfänger sie interpretieren kann.

Was ist XML?

- Untermenge der SGML. HTML ist auch ein Beispiel davon:
`< b >`
- eXtended Markup Language.
- 1996 vom W3C entwickelt(world Wide Web Consortium)
- Ziel Semantische Notation.
- Eine Meta-Sprache, nicht festgelegte Sprache(selber definiert).selbst beschreibendes Syntax.
- wird oft als Komplement zu HTML verwendet, um die Daten vom Präsentieren zu trennen.
- XML-Dokumente lassen sich mit JavaScript, Python oder anderen Programmiersprachen einlesen oder verändern.



Bestandteile einer XML-Datei:

- der Prolog- optional
- DTD-optional
- das Wurzelement,(das weitere Elemente enthalten kann) *ii*
Baumstruktur
- Kommentare und Verarbeitungsanweisungen(zusätzlich).

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

Prolog

```
<!DOCTYPE Personalakte [
```

DTD

```
    <!ELEMENT      Personalakte (Personalien)
```

```
    <!ATTLIST      Personalakte PersNr CDATA #REQUIRED>
```

```
    <!ELEMENT      Personalien (Nachname, Vorname+,GebDat)>
```

```
    <!ELEMENT      Nachname (#PCDATA)>
```

```
    <!ELEMENT      Vorname (#PCDATA)>
```

```
    <!ELEMENT      GebDat (#PCDATA)>
```

```
]>
```

```
<Personalakte PersNr="12345">
```

Wurzelement

```
    <Personalien>
```

Subelemente

```
        <Nachname> Meier </Nachname>
```

```
        <Vorname> Ambrosius </Vorname>
```

```
        <GebDat> 20. Juli 1974 </GebDat>
```

```
    </Personalien>
```

```
</Personalakte>
```

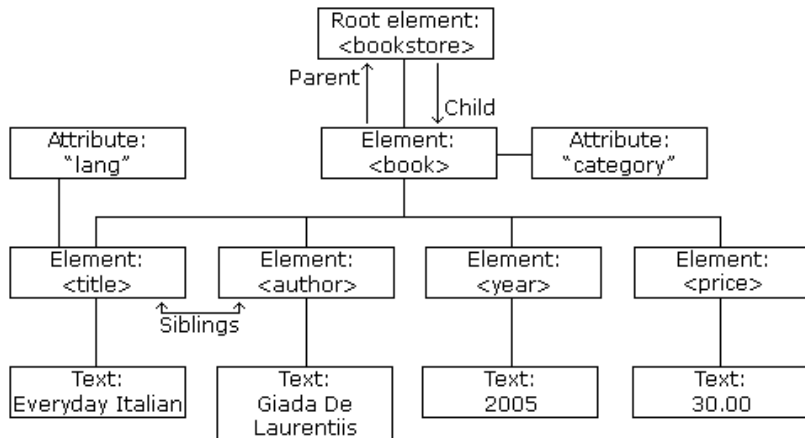
XML
Baum

Bestandteile Des XML-Baums

- Tags
- Attribute
- Werte
- Texte(Elementinhalt)

Ein XML-Tree beginnt mit einem Wurzelement, das sich in Subelemente verzweigt.

Der XML-Baum



Magie von xmlElementTree



Regeln

- Tags dürfen keine Zeichenfolge „xml“ oder Zahlen haben
- innerhalb des Namens darf es kein Leerzeichen geben. Es ist case Sensitive(groß und klein)
- Attribute im ‚ ‘ mehrere Attribute mit demselben Namen für dasselbe Element sind nicht erlaubt.
- Kommentare dürfen weder am Anfang der Datei noch innerhalb eines Tages vorkommen. <! – – Das könnte den Leser interessieren – – >
- Attribute können nicht mehrere Werte enthalten (nur Elemente)
- Baumstruktur gilt nicht für Attribute (nur Elemente)
- fast alle bekannten Browser haben einen enthaltenen XML-Parser
- „/“, „//“, „.“, „..“, „@“ werden für Xpath verwendet(darf man nicht verwenden)

Beispiel

```
<Weihnachtsgeschenk>  
  <für>mich</für>  
  <von>keinem</von>  
  <inhalt>hab kein Geschenk bekommen :( </inhalt>  
</Weihnachtsgeschenk>
```

So wird eine hierarchische Datenstruktur definiert.
für
von
kann man als Attribute definieren

```
<Weihnachtsgeschenk für="mich" von="keinem">  
  <inhalt>hab kein Geschenk bekommen :( </inhalt>  
</Weihnachtsgeschenk>
```

Hier ist die Struktur anders

Eigenschaften

Structure of Elements in XML Documents

< root > can be called Parent Element

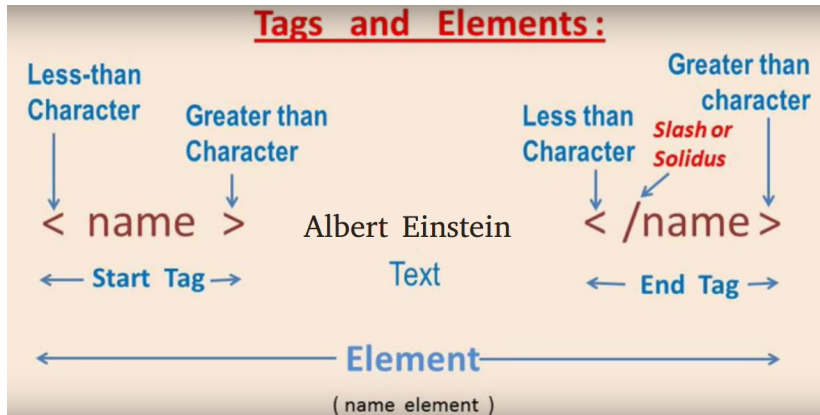
< child >

< subchild > < / subchild >

< / child >

< / root >

Eigenschaften




1. XML element **names** are **Case Sensitive** :

< **N**ame> Samuel Clinton < / **n**ame > **Wrong**
< **n**ame> Samuel Clinton < / **N**ame > **Wrong**
< name> Samuel Clinton < / name > **Correct**

2. XML document should have **One Root Element** :

< x > < / x > } **Wrong** because
< y > < / y > } No Root Element

 **Root Element**

< a > }
< x > < / x > }
< y > < / y > } **Correct**

3. Elements **should not overlap** :

< **book** > < author> Robert Lang < / **book** > < / author > **Wrong** (Not properly nested)
< book > < author> Robert Lang < / author> < / book > **Correct** (properly nested)

Vorteile

- Die Einfachheit und die zahlreichen Dinge, die uns erlauben, unsere XML-Datei öffnen und bearbeiten zu können.
- Bestimmte Elemente ausgeben, oder modifizieren ohne XSL haben zu müssen oder Xpath(Abfragesprache für XML-Dokumente) zu können.
- XSL(eine Sprache): speichert die Instruktionen von unserer XML-Datei
- verkleinert auch die Bedeutung der DTD(Data Type Definition).
- Neue Elemente oder sogar Tags erstellen oder ausgeben.
- Die Möglichkeit neue Daten von unserer XML Datei wie erwünscht herauszugeben.

xmlElementTree Python

Es muss nicht installiert werden
importieren:

```
import xml.etree.ElementTree as ET
```

Funktionen von xmlElementTree

- `xmltree.parse('nameDesDokus')`
- `Element.getroot()` / `.text` / `.tag` / `.attrib` (dictionary)
- mit normaler Schleife dann `Element.iter()` („gesuchtes Attribut“). Alles ausgeben
- `Element.getchildren()` gibt nur sub Elemente von der Wurzel
- `iterfind()` iterate alle Elemente, die mit Xpath Expression zusammenpassen.
- `findall()` gibt eine Liste betroffener Elemente(direkte Kinder) aus.
- `Find()` gibt lediglich das erste betroffene Element aus oder `finde().text` .
- `findtext()` gibt den Inhalt(Text) des Elements aus.
- `ET.SubElement(wurzel, 'neues element')`
- `parent.remove(subelement)`
- `ElementTree.write('output.xml')`

Aufgaben

Fehler finden!

```
<?xml version="1.0"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

Bild(2)

```
<note date=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Aufgaben

- 1 Klicke auf den Link `https://github.com/Colonel666/Codes/blob/xmlElementTree/sample.xml` und lade die XML-Datei herunter
- 2 Lies den XML-Inhalt der heruntergeladenen Datei mit Python ein.
- 3 Zeige nur 'author', 'title' und 'price' jedes Elements(book) an.
- 4 Speichere die Bücher, die weniger als 30 kosten in eine neue Datei 'günstig.xml'
- 5 veränder den Tag 'book' dessen id 'bk103' zu 'Magazine' stattdessen und speichere sie in eine neue Datei 'output.txt'

www.jeckle.de/xml/index.html

<https://docs.python.org/2/library/xml.etree.elementtree.html>

w3schools.com/xml/cd_catalog.xml

<https://www.youtube.com/watch?v=d0Wn9xcwmc0>

Eckstein/Echstein: XML und Datenmodellierung. Dpunkt.

Schöning: XML und Datenbanken. Hanser.