

XML 2010 Erik Tjong Kim Sang erikt(at)xs4all.nl 6 september 2010	XML 2010 College 1 XML 2010 Docent Erik Tjong Kim Sang <erikt(at)xs4all.nl> Harmoniegebouw kamer 11.0425, tel. 050 363 5882 Literatuur geen verplichte literatuur suggestie voor studenten die op zoek zijn naar een boek: Erik T. Ray, <i>Learning XML</i>. O'Reilly, 2001 Website http://ifarm.nl/erikt/xml0/ Informatiekunde 1
XML 2010 College 1 Cursusdoel Overzicht van de structuur en en het gebruik van de opmaaktaal XML. Informatiekunde 2	XML 2010 College 1 Bijeenkomsten en evaluatie De cursus duurt acht weken (zeven sessies: 6 september - 22 oktober). Naast de colleges op maandagen zijn er ook practica op dinsdagen en woensdagen (twee verschillende groepen; registratie via Nestor). Evaluatie zal plaats vinden via drie Nestorquizen over de collegestof, vijf wekelijkse praktische opdrachten, en een afsluitende opdracht. Het gemiddelde voor de vijf wekelijkse opdrachten bepaalt 50% van het eindcijfer, het gemiddelde van de Nestorquizen 30% en de eindopdracht 20%. De twee gemiddelden moeten elk minstens 5.0 zijn om het vak te kunnen halen. Informatiekunde 3
XML 2010 College 1 Onderwerpen van de zeven hoorcolleges 1. (06/09) Basisbegrippen in XML 2. (13/09) Definities van documentstructuur: DTDs 3. (20/09) Stijlbestanden: CSS 4. (27/09) Transformaties in XML: XSLT 5. (04/10) Zoeken in XML-bestanden: XPath 6. (11/10) Databases en XML 7. (18/10) (nog te bepalen) Informatiekunde 4	XML 2010 College 1 Relatie college XML met andere vakken De opmaaktaal XML wordt gebruikt in diverse tekstverzamelingen, bijvoorbeeld in het syntactisch geannoteerde LASSY-corpus. Dit corpus wordt gebruikt in het bachelorvak Natuurlijke Taalverwerking 1 en de Mastervakken Natural Language Processing en Semantic Webtechnology. Opgelet: voor het volgen van het vak XML is de afronding van het vak Webdesign vereist Informatiekunde 5
START!	XML 2010 College 1 Wat is XML? XML (Extensible Markup Language) is een verzameling van opmaaktalen voor het toevoegen van interne structuur in documenten XML-bestanden bevatten vaak tekst maar kunnen ook andere typen informatie bevatten: tabellen, plaatjes, geluid, animaties, verwijzingen, ... Informatiekunde 7

Een voorbeeldtekst

CDA in gespannen sfeer bijeen

de CDA-fractie is al urenlang bijeen voor overleg over de formatie . de sfeer in de fractie is volgens ingewijden gespannen .
...

Van de tekst is verder bekend dat de bron NOS Teletekst is en de datum 31-08-2010

We willen onderdelen van de tekst kunnen herkennen en bewerken

De voorbeeldtekst als XML-document

```
<tekst>
  <kop>CDA in gespannen sfeer bijeen</kop>
  <zin>
    de CDA-fractie is al urenlang bijeen voor overleg over de
    formatie .
  </zin>
  <zin>
    de sfeer in de fractie is volgens ingewijden gespannen .
  </zin>
  ...
  <bron>NOS Teletekst</bron>
  <datum>31-08-2010</datum>
</tekst>
```

XML-elementen

Naast informatie zoals tekst, bevat een XML-document ook extra codes, de zogenaamde tags

Tags zijn herkenbaar aan hun vaste vorm: ze beginnen met een kleiner-dan-teken (<) en worden afgesloten met een groter-dan-teken (>), ook wel vishaken genoemd)

Stukjes tekst kunnen worden gemarkeerd door ze te plaatsen tussen een opentag (<abcd>) en een sluittag (</abcd>)

Een tag kan ook zichzelf afsluiten: <abcd/>

XML-syntax

Binnen een gemarkeerde tekst kunnen ook weer XML-tags staan: het zogenaamde *nesten* van tags:

```
<tekst>
  <kop>CDA in gespannen sfeer bijeen</kop>
  ...
</tekst>
```

Het gedeelte tekst dat gemarkeerd is als *kop* valt ook onder het gedeelte gemarkeerd als *tekst*

Een documentdeel ingesloten tussen een open- en een sluittag heet een *element*

Het voorbeelddocument als boom

De structuur van een XML-document kan worden gerepresenteerd in een boomstructuur, met de meest algemene tag (hier <tekst>) boven en de specifieke tags onderaan:

```
      <tekst>
    <zin> <zin> ... <bron> <datum>
  <woord> ...
```

Correct gebruik van XML-tags

In een XML-document moeten geopende tags altijd worden afgesloten voordat de moedertags worden afgesloten:

Correct: <a>

Fout: <a>

Elementen mogen elkaar dus niet gedeeltelijk overlappen

Daarnaast kent XML niet iets als het automatisch afsluiten van tags, zoals bijvoorbeeld bij de <p>-tags in HTML

Alle geopende tags in XML moeten expliciet worden afgesloten

Mogelijke namen van XML-tags

XML is een metataal en heeft zelf geen voorschriften over welke tagnamen kunnen of moeten worden gebruikt

Iedereen kan zelf zijn eigen XML-tagnamen gebruiken

Op deze slides worden vooral Nederlandse tagnamen gebruikt

Hier is een voorbeeld van een tag in een andere taal:

<鱼>Пираниевые</鱼>

Over presentatie en beschrijving

De functie van XML-tagnamen kan erg verschillend zijn

Bij het maken van tagnamen voor teksten is er vaak de keuze tussen namen die de tekstpresentatie aangeven (bijvoorbeeld *vetgedrukt*) of de tekstinhoud beschrijven (bijvoorbeeld *belangrijk*)

De filosofie van XML is om beschrijvende tagnamen te gebruiken zodat een document op meerdere manieren gepresenteerd kan worden

Een voorbeeld van deze filosofie in HTML is
<http://www.csszengarden.com/>

XML-attributen

Extra informatie kan worden toegevoegd aan een tag, in zogenaamde attributen:

```
<zin nummer="23" functie="hoofdzin">
```

Attributen worden naast de naam van de tag geplaatst

Zij bestaan zelf uit een naam (hier `nummer` en `functie`), gevolgd door een `=`-teken en een waarde tussen quotes

De attributen van een tag moeten unieke namen hebben

In tags zijn spaties en newlines zijn toegestaan in achter de tagnaam

De voorbeeldtekst met attributen

Informatie over een document past soms beter in attributen:

```
<tekst bron="NOS Teletekst" datum="31-08-2010">
  <kop>CDA in gespannen sfeer bijeen</kop>
  <zin>
    de CDA-fractie is al urenlang bijeen voor overleg over de
    formatie .
  </zin>
  <zin>
    de sfeer in de fractie is volgens ingewijden gespannen .
  </zin>
  ...
</tekst>
```

De start van een XML-document

Het is verstandig om een XML-document te laten beginnen met een *prolog*: tags met informatie over het document, bijvoorbeeld:

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

(tags die beginnen met `<?` en eindigen met `?>` worden *processing instructions* genoemd)

Deze tag geeft aan dat het bestand een XML-document bevat dat XML-versie 1.0 gebruikt en dat de tekens in de tekst uit de tekenverzameling UTF-8 komen

Wat is een tekenverzameling?

Alle letters, cijfers en andere tekens (Z, 1, ±) worden in computers intern gerepresenteerd met reeksen van bits, enen en nullen

Voor de omzetting van bits naar tekens worden verschillende tabellen gebruikt:

- ASCII (1963): 95 tekens gerepresenteerd met 7 bits (VS-Engels)
- ISO 8859-1 (1983): 191 tekens gerepresenteerd met 8 bits (25 talen)
- UTF-8 (1993): ongeveer 100.000 tekens gerepresenteerd met een variabel aantal bits

Entities

Een entity is naam in een XML-document die verwijst naar een object. Het kan worden vergeleken met een variabele (constante) in programmeertalen

XML-entities beginnen met een ampersand (&), gevolgd door een naam en een punt-komma (;), bijvoorbeeld: `&test;`

Uit HTML kennen we de tekenentities als `´`; die ook in XML worden gebruikt

XML-entities kunnen ook naar andere objecten verwijzen dan tekens, bijvoorbeeld naar tekst, naar plaatjes of zelfs naar hele documenten

Entitydefinities

Alle entities behalve `amp`, `apos`, `gt`, `lt` en `quot` moeten gedefinieerd worden, bijvoorbeeld als volgt:

```
<!DOCTYPE rootelement
[
  <!ENTITY test1 "de tekst voor entity test1">
  <!ENTITY test2 "de tekst voor entity test2">
]
>
```

waarbij `rootelement` de naam is voor de roottag van het document

Verschillen tussen XML en HTML

In HTML maakt het niet uit of tagnamen worden geschreven met hoofdletters of kleine letters maar XML maakt wel verschil tussen bijvoorbeeld `<A>` en `<a>`

In HTML worden spaties en newlines in documenten overgeslagen maar XML houdt wel rekening met spaties en newlines: de inhoud van het element `<a> A </a>` bestaat uit 3 tekens

HTML kent opentags die automatisch worden afgesloten (`<p>`) en opentags zonder (`<br>`)

Zulke tags bestaan niet in XML: daar sluit je elke opentag expliciet af

Commentaar in XML-documenten

In XML-documenten kan commentaar worden geplaatst tussen de tekenscombinaties `<!--` en `-->`

Tussen commentaartekens mogen ook tags en elementen staan:

```
<tekst>
  <!-- <kop>CDA in gespannen sfeer bijeen</kop> -->
  ...
</tekst>
```

Alleen de combinatie `--` is verboden in XML-commentaar

CDATA

De tekens `<`, `>` en `&` zijn niet toegestaan in XML-documenten. Gebruik hiervoor de karakterentiteiten `<`, `>` en `&`;

Als je tekst hebt met veel van dit soort tekens dan kan je overwegen om het element CDATA te gebruiken:

```
<![ CDATA [
  In XML plaats je tags tussen < en >, en gebruik je & voor
  entities, bijvoorbeeld: <accentteken>&ntilde;</accentteken>
]]>
```

In een CDATA-element mogen alle tekens worden gebruikt behalve de combinatie `]]>`

Voorbeelden van goede en slechte XML-opmaak (1)

<lijst> (FOUT)

```
<element> eerste element
<element> tweede element
</lijst>
```

<lijst> (GOED)

```
<element> eerste element </element>
<element> tweede element </element>
</lijst>
```

Alle geopende elementen in XML moeten op een correcte manier worden afgesloten

Voorbeelden van goede en slechte XML-opmaak (2)

<plaatje bron="plaatje.jpg"> (FOUT)

<plaatje bron="plaatje.jpg" /> (GOED)

Elementen zonder inhoud moet zichzelf afsluiten

<plaatje bron=plaatje.jpg /> (FOUT)

<plaatje bron="plaatje.jpg" /> (GOED)

Attribuutwaarden moeten tussen quotes staan

Voorbeelden van goede en slechte XML-opmaak (3)

<z> een <q> twee </z> drie </q> (FOUT)

<z> een <q> twee </q> drie </z> (GOED)

Elementen mogen elkaar niet gedeeltelijk overlappen

<test> als 4/3 > 3/4 dan ... </test> (FOUT)

<test> als 4/3 > 3/4 dan ... </test> (GOED)

Gebruik de tekens `<`, `>` en `&` niet in elementinhoud

Zelf XML-documenten maken

Er bestaan speciale editors voor de bewerking van XML-documenten

Maar de meeste standaardeditors zijn ook geschikt om XML-documenten te bewerken, bijvoorbeeld kwrite, emacs en notepad

Editors kunnen door het gebruik van nadruk en kleur het verschil tussen tags en tekst duidelijker maken

Daarnaast wordt vaak de boomstructuur van de elementen in beeld worden gebracht

Verwerking van XML-documenten

XML-documenten worden gebruikt als invoer voor diverse programma's: editors, webbrowsers, databases, ...

Al deze programma's bevatten een XML-parser: een software routine die XML-tags herkent en XML-documenten kan omzetten in een boom van tags en de bijbehorende informatie

XML-parsers zijn erg streng: als de XML-opmaak niet correct is dan wordt de verwerking van het document gestopt

Controle van XML-syntax

Er bestaat diverse software voor het controleren van de syntactische structuur van XML-documenten

Op onze Linux-machines is het programma `xmllint` beschikbaar:

```
$ xmllint test.xml
```

Documenten kunnen ook met online tools worden gecontroleerd, bijvoorbeeld: <http://validator.w3.org/>

Het bekijken van XML-documenten

Zonder een stijlbestand is het niet duidelijk hoe XML-elementen het best zichtbaar moeten worden gemaakt

De structuur van een XML-document kan worden bekeken in een editor of in een webbrowser

Eventueel kan het document eerst worden geconverteerd naar HTML en daarna worden bekeken

Geschiedenis van XML (1)

In de eerste elektronische documenten ontbrak structuurinformatie en was naast de tekst alleen informatie over de gewenste presentatie aanwezig

Zoeken of selecteren van informatie in zulke teksten was lastig

In de jaren 60 ontwikkelde IBM een beschrijvende documenttaal: GML (Generalized Markup Language)

In de jaren 70 en 80 ontwikkelde de ANSI de taal SGML, met daarin de in XML gebruikte vishaaktags

Geschiedenis van XML (2)

Aan het begin van de jaren 90 ontwikkelde Tim Berners-Lee op het CERN een op SGML gebaseerde opmaaktaal voor hypertextdocumenten: HTML

Maar HTML bevatte diverse tags die op formattering waren gericht in plaats van informatiebeschrijving: `<b>` versus `<strong>`

Halverwege de jaren 90 begon het World Wide Web Consortium daarom aan een verzameling opmaaktalen die de voordelen van SGML moest combineren met die van HTML: XML

XML is inmiddels een standaard die in veel software en documenten wordt gebruikt

Wat zijn de voordelen van XML?

- standaardmethode voor het brengen van structuur in documenten
- documenten zijn gemakkelijk uitwisselbaar tussen gebruikers
- strenge definities maken controle van kwaliteit van document eenvoudig
- extra structuurelementen maken gevarieerde presentatie en gericht zoeken mogelijk

Nadelen van XML

- XML-annotaties nemen veel ruimte in beslag en XML-documenten zijn daarom vaak een stuk groter dan de informatie die zij bevatten
- Data die wordt aangeleverd in een verkeerde XML-structuur, is onbruikbaar
- Snel XML-documenten doorzoeken is een enorme uitdaging

Weekopdracht

In de opdracht van deze week moeten jullie een tekstdocument voorzien van XML-tags en deze valideren

Iedereen krijgt een eigen document toegewezen

Een beschrijving van de opdracht is te vinden op <http://ifarm.nl/erikt/xml0/>

De deadline voor het inleveren van deze opdracht is **maandag 13 september 2010, 09:00**

Achtergrondliteratuur

Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler and François Yergeau, *Extensible Markup Language (XML) 1.0* (Fifth Edition). W3C, 26 November 2008.

<http://www.w3.org/TR/2008/REC-xml-20081126/>

David Hunter, Jeff Rafter, Joe Fawcett and Eric van der Vlist, *Beginning XML (Programmer to Programmer)*. John Wiley & Sons, 2004.

Erik T. Ray, *Learning XML*, O'Reilly, 2001.

W3schools, *XML Tutorial*, <http://www.w3schools.com/xml/>, 2010.

THE END