

XML

<XML>

What is XML?

eXstensible
XML = Markup
Language

<XML>

XML is a Markup Language

- A **markup language** combines text and information about the text in the same file.
- Markup languages were originally used by publishers to present documents.

```
Document: Bungler OED      At: "entry"
<entry>
  (hwsec)
  (hwsp)
  (hulen)bungler</hulen>
  (pron)<179</ID>ngju</pron> </hwsp>
  <rf1>miss <rd></rd> <rf>bungler</rf>
  </rf1>
  (etym)f. as prec. + (xra)<xlem>-ER</xle
  <ren>One who bungles; a clumsy unskilful
  <quot>
  <qdat>1533 </qdat>
  <auth>MORE </auth>
  (sk)Jasw. Poynt. Jk. </sk>Wis. (1557
  <txt>He is euen but a very bungler.
```

SGML, a markup language.
From Wikipedia
<http://en.wikipedia.org/wiki/Image:OED-LEX-Bungler.jpg>

<XML>

XML Doesn't Do Anything

- XML is not a programming language like C++ or Java. An XML document doesn't do anything.
- XML was designed to structure and store information.
- XML files are plain text files consisting of the data to be marked up along with **tags** that describe the data.

<XML>

Why XML? cont'd

- XML is platform-independent, so it can be used on a wide range of different devices: PCs, Macintoshes, cell phones, mp3 players, etc.
- XML applications must adhere strictly to rules governing syntax and structure, ensuring standards compliance.

<XML>

XML is eXtensible

- Unlike HTML, XML requires you to define your own tags.
- We can create special markup languages using our own tags.
- These languages are XML **applications**.

Some XML Applications

- XHTML
- RSS
- MathML
- News Markup Language (NML)
- Open eBook
- Open Document
- Recipe ML (formerly known as DESSERT -- Document Encoding and Structuring Specification for Electronic Recipe Transfer
- And hundreds more...

<XML>

XML and XHTML

- XML was not intended to replace HTML, but to complement it.
- XHTML is a reformulation of HTML as an XML application.
- Because it's an XML application, XHTML is more extensible than HTML.

<XML>

DTDs and Schemas

```
$Date: 2002/05/01 18:37:55 $
-->
<!--==== Character mnemonic ent
<!ENTITY % HTMLa1 PUBLIC
"-//W3C//ENTITIES Latin 1 for XHTML//EN"
"html-lat1.ent">
%HTMLa1;
<!ENTITY % HTMLsymbo PUBLIC
"-//W3C//ENTITIES Symbols for XHTML//EN"
"html-symbol.ent">
%HTMLsymbo;

<!--==== Character mnemonic entities -----
XML entity sets are identified by the PUBLIC and SY
PUBLIC "-//W3C//ENTITIES Latin 1 for XHTML//EN"
SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1-en
PUBLIC "-//W3C//ENTITIES Special for XHTML//EN"
SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml-special
PUBLIC "-//W3C//ENTITIES Symbols for XHTML//EN"
SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml-symbo
</!doctypeatt
</!doctypeatt
<!--==== Imported Names -----
</!doctypeatt
</!doctypeatt
```

An excerpt from the DTD for XHTML

An excerpt from the Schema for XHTML

- The rules that specify the structure and syntax of an XML application are provided in an XML **DTD** (Document Type Declaration) or **Schema**.

<XML>

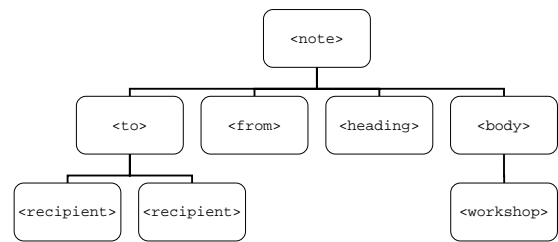
DTDs and Schemas cont'd

- XML with a DTD or Schema is designed to be self-documenting.
- Programs that understand XML can use DTDs or Schemas to **validate** an XML document.
- Validating an XML document ensures that it adheres to standards of structure and syntax.

<XML>

XML Structure

- XML documents are structured like an upside-down tree.



<XML>

XML and Display Information

- Unlike HTML, XML separates content from display information.
- Keeping content separate from display information makes it easier for documents to be processed in different ways (ex: on a PC vs. on a cell phone, or as a web page vs. as a record in an OPAC).

Some HTML tags and attributes that contain display information:

- <center>
-
- <body bgcolor="blue">
-

<XML>

XML Syntax

- The syntax rules for XML are very simple and very strict:

- All XML elements must have opening and closing tags.
- XML tags are case-sensitive
- XML tags must be properly nested
- All XML documents must have a root element
- All XML attributes must be quoted

<XML>

All XML Elements Must Have Opening and Closing Tags

- In XML, all tags must have a matching closing tag.
- In other words, for every `<element>` there must be an `</element>`.
- An exception is made for solo tags: `<element />`

Valid XML

```
<book>
<author>Irma Rombauer</author>
<title>The Joy of Cooking</title>
<pub_date year="1936"/>
</book>
```

Invalid XML

```
<book>
<author>Irma Rombauer
<title>The Joy of Cooking
<pub_date>1936
</book>
```

<XML>

XML Tags are Case-Sensitive

- In XML, tags are case sensitive.
- In other words, `<element>` does not match with `</Element>`.

Valid XML

```
<recipe>
<name>Toast</name>
<Ingredient>bread</Ingredient>
<ingredient>butter</ingredient>
</recipe>
```

Invalid XML

```
<recipe>
<Name>Toast</name>
<ingredient>bread</Ingredient>
<INGREDIENT>butter</ingredient>
</Recipe>
```

<XML>

XML Must Be Properly Nested

- Each XML tag can have only one parent node, and both the opening and closing tags must be within that node.
- In other words, tags should *nest* like:


```
<a><b></b></a>
```

 not


```
<a><b></a></b>
```

Valid XML

```
<joke>
<char>String</char>
<loc>Bar</loc>
<punch_line>
  I'm a frayed knot!
</punch_line>
</joke>
```

Invalid XML

```
<joke>
<char>String</char>
<loc>Bar</loc>
<punch_line>
  I'm a frayed knot!
</joke>
</punch_line>
```

<XML>

All XML Documents Must Have a Root Element

- Every XML document needs to have a unique root element.
- In other words, there should be one tag in every document that all the other tags nest inside of.

Valid XML

```
<?xml version="1.0"
encoding="UTF-8"?>
<player>
<name>Williams</name>
<number>9</number>
</player>
```

Invalid XML

```
<?xml version="1.0"
encoding="UTF-8"?>
<name>Williams</name>
<number>9</number>
```

<XML>

All XML Attributes Must Be Quoted

- The data portion of an attribute in an XML tag must always be inside quotes.

Valid XML

```
<animal type="Otter">
<name>Marshall</name>
<friend>Bucky</friend>
</animal>
```

- In other words, a tag with an attribute should look like:

```
<element attribute="data">.
```

Invalid XML

```
<animal type=Otter>
<name>Marshall</name>
<friend>Bucky</friend>
</animal>
```

<XML>

XML in Real Life: RSS



- RSS (Really Simple Syndication) is an XML application that allows users to "subscribe" to websites.

- Some uses of RSS: Podcasts, GSLIS events calendar, Simmons Library new arrivals announcements.

- After XHTML, RSS is probably the XML application that web users see most often.

<XML>

Sample RSS Document

- From the GSLIS events calendar RSS feed.

```
<rss version="2.0">
<channel>
<title>GIS Events</title>
<description>
Events of interest to GIS students, staff, faculty and alumni.
</description>
<link>
http://my.simmons.edu/gis/resources/calendars/events.shtml
</link>
<lastBuildDate>Tue, 30 May 2006</lastBuildDate>
<pubDate>Oct. 17, 2006</pubDate>
<item>
<title>Cool Tools: Digital Tools Used in the Library Setting</title>
<description><p>When: October 17, 1:00 p.m.<br />
Where: Katzen Room, Simmons College Library, Boston Campus</p>
</description>
<link>
http://my.simmons.edu/gis/resources/calendars/events.shtml#cool_tools
</link>
<pubDate>Tue, Oct. 10, 2006</pubDate>
</item>
```

<XML>

XML in Real Life: Google Earth

- KML (Keyhole Markup Language) is an XML application used by Google Earth and Google Maps for modeling and storing geographic information.



A map of nature reserves in Cornwall generated by Google Earth
From the Cornwall Wildlife Trust
http://www.cornwallwildlifetrust.org.uk/reserves/google_earth.htm

<XML>

Sample KML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Placemark>
<description>Tethered to the ground by a
customizable tail</description>
<name>Tethethed placemark</name>
<LookAt>
<longitude>-122.0856375356631</longitude>
<latitude>37.42240551227282</latitude>
<range>305.8880792294568</range>
<tilt>46.72425699662645</tilt>
<heading>49.06133439171233</heading>
</LookAt>
```

<XML>

XML On Your Desktop

- Lots of common applications can save their data in an XML format, including Excel, Access, and iTunes.
- Example uses: importing an XML version of an Excel spreadsheet to a web-based application or sharing your iTunes library information with another mp3 player.

<XML>

XML in Libraries: METS

- Metadata Encoding and Transmission Standards



- Encodes descriptive, administrative, and structural metadata regarding objects within a digital library.
- The METS Schema is maintained by the Library of Congress:
<http://www.loc.gov/standards/mets/>

<XML>

Sample METS Document

- Part of a sample METS citation document from
<http://www.loc.gov/standards/mets/mets-examples.html>

```
<METS:dmdSec ID="DM129277">
<METS:mdWrap MDTYPE="OTHER" OTHERMDTYPE="GDM">
<METS:xmlData>
<gdm:gdm>
<gdm:core>
<gdm:coreDate BEGINDATE="1912" ENDDATE="1914" DATE="1912-1914"/>
<gdm:localID LOCALIDTYPE="(MH)HOLLIS">BN13165</gdm:localID>
<gdm:origin>Cambridge [Mass.]</gdm:origin>
<gdm:title>Reports of the president and treasurer for...</gdm:title>
</gdm:core>
<gdm:creator NAMEATYPE="cn" ROLE="author">Radcliffe College</gdm:creator>
<gdm:creator NAMEATYPE="cn" ROLE="publisher">University Press</gdm:creator>
</gdm:gdm>
</METS:xmlData>
</METS:mdWrap>
</METS:dmdSec>
```

<XML>

XML in Libraries: MODS

- Metadata Object Description Schema



- Intended to be able to carry selected data from existing MARC 21 records as well as to enable the creation of original resource description records
- The MODS Schema is also maintained by the Library of Congress:
<http://www.loc.gov/standards/mods/>

<XML>

Sample MODS Document

- Part of a sample MODS document describing a journal article.

```
<typeOfResource>text</typeOfResource>
<genre>journal article</genre>
<originInfo>
  <place>
    <placeTerm type="text">Baltimore, Md.</placeTerm>
  </place>
  <publisher>Johns Hopkins University Press</publisher>
  <dateIssued>2003</dateIssued>
</originInfo>
```

<XML>

XML in Libraries: MARCXML



- A framework for working with MARC in an XML environment.
- Built for flexibility.
- Also LOC:
<http://www.loc.gov/standards/marcxml/>

<XML>

Sample MARCXML Document

- Part of a sample MARCXML document for Carl Sandburg's *Arithmetic*.

```
<datafield tag="050" ind1="0" ind2="0">
  <subfield code="a">PS3537.A618</subfield>
  <subfield code="b">A88 1993</subfield>
</datafield>
<datafield tag="082" ind1="0" ind2="0">
  <subfield code="a">811/.52</subfield>
  <subfield code="2">20</subfield>
</datafield>
<datafield tag="100" ind1="1" ind2=" " >
  <subfield code="a">Sandburg, Carl,</subfield>
  <subfield code="d">1878-1967.</subfield>
</datafield>
```

<XML>

XML in Libraries: EAD

- Encoded Archival Description

<ead>

- A standard for archival finding aids
- Also maintained by the LOC:
<http://www.loc.gov/standards/mods/>

<XML>

Sample EAD Document

- Part of a sample EAD document describing a set of personal papers at the collection and file levels.

```
<archdesc level="collection">
  <did>
    <head>Descriptive Summary</head>
    <unittitle label="Title">
      Mildred Davenport dance programs and dance school materials
    </unittitle>
    <unitdate type="inclusive" normal="1934/1942">
      1934-1942
    </unitdate>
    <unitid countrycode="us" repositorycode="cu-i"
      label="Collection number">
      MS-P29
    </unitid>
  </did>
</archdesc>
```

<XML>

Resources

- XML basics
 - W3 Schools
<http://w3schools.com/xml/>
 - XML Files
<http://www.xmlfiles.com/xml/>
- The XML 1.1 specification
<http://www.w3.org/TR/xml11>

Resources

- XML utilities
 - HiT Software XML Tools
Generates DTDs from XML documents,
converts between DTDs and Schemas
http://www.hitsw.com/xml_utilites/
 - Online XML validator
<http://www.ltg.ed.ac.uk/~richard/xml-check.html>

Resources

- Others

- XML4Lib

- A mailing list dedicated to applications of XML within the library

- <http://lists.webjunction.org/xml4lib/>

- *XML in Libraries*. Tennant (Ed.) 2003

- A collection of case studies of XML applications in the library.

- 3 copies at Beatley: Z678.93.X54 X54 2002