XML
eXtensible Markup Language

What is XML?

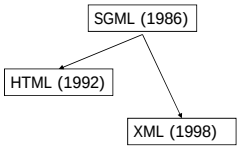
A meta language for creating new markup languages (the eXtensible Markup Language)

But *NOT* a programming language

Developed by the WWW Consortium & released 10 Feb 1998

Historical Context

Based on SGML



- HTML defines the layout and appearance of a document
- XML defines the structural elements of a document

XML Document Structure

A document's *logical structure* divides it into elements
- e.g. a book document has chapter elements with title, paragraph, table, and figure elements

The logical structure specifies constraints which must be met by a valid XML document

Tags are used to enclose identifiable parts of a document

An XML parser checks for conformance of an XML document to the logical structure given in a *Document Type Definition* (DTD)

Document Presentation

Style sheets are used for specifying the output format of XML elements

Style sheets can be changed & different style sheets created to suit the audience and publishing medium (WWW, printout, CD-ROM, video, etc.)

… but this is a separate issue from the structure of the data

Example XML Document

```
<?xml version="1.0"?>
<!-- a comment -->
 <item>
   <section type="poem">
     <title>The Purple Cow</title>
       <line>I never saw a purple cow,</line>
       <line>I never hope to see one;</line>
       <line>But I can tell you, anyhow,</line>
       <line>I'd rather see than be one.</line>
   </section >
 </item>
```

Well-formed Documents

XML documents must be *well-formed*

- Correct syntax (according to the XML spec)
- Tags match, nesting, all characters legal
- Parser must reject document if not well-formed

XML Specification
http://www.w3.org/TR/REC-xml/

Elements

The primary structures in an XML document

Example element with content:

```
<person>Konrad Zuse</person>
```

Empty element: `<br/>`

Element names and content can be any Unicode letter, digit, or '.', '-', '_', or ':'

```
<Straße>Plankalkül 1945</Straße>
```

Elements Have Structure

Title element is *inside* section element

```
<section type="poem">
   <title>The Purple Cow</title>
   …
</section>
```

Shows the lines of the poem, not the line breaks on the page

```
I never saw a purple cow<br/>      HTML
<line>I never saw a purple cow</line>   XML
```

Document Type Declaration - DTD

A grammar which specifies the rules for a properly formed XML document

Made up of markup declarations for

- Elements
- attribute lists
- and several others

Element Declarations

`<!ELEMENT NAME CONTENT>`

,	Separates members of a sequence list, which requires sequential use of all members
|	Separates members of a choice list, which require use of one and only one member
+	Indicates a required and repeatable occurrence
*	Indicates an optional and repeatable occurrence
?	Indicates an optional occurrence

DTD and Elements

Example DTD for a memo:

```
<!ELEMENT MEMO     (TO,FROM,SUBJECT,BODY,SIG)>
<!ELEMENT TO       (#PCDATA)>
<!ELEMENT FROM     (#PCDATA)>
<!ELEMENT SUBJECT  (#PCDATA)>
<!ELEMENT BODY     (PARA*)>
<!ELEMENT PARA     (#PCDATA)>
<!ELEMENT SIG      (#PCDATA)>
```

#PCDATA means Parseable Character Data, and represents a string of zero or more characters

An XML Document

```
<?xml version="1.0"?>
<!DOCTYPE MEMO SYSTEM "memo.dtd">
<MEMO>
  <TO>CSIS 4244 Students</TO>
  <FROM>Your Beloved Professor</FROM>
  <SUBJECT>This is XML</SUBJECT>
  <BODY>
   <PARA>This example should give you an idea of what XML is
     like, and how it fits in with our study of syntax and
     grammars.
   </PARA>
   <PARA>Note how elements are like EBNF grammar rules, and
     even regular expression notation is used.</PARA>
  </BODY>
  <SIG>Mike Olan</SIG>
</MEMO>
```

Attributes

Attributes are *name-value pairs* that occur in tags after the element name

Adding a priority attribute to a memo:

```
<!ATTLIST MEMO priority (HIGH|MEDIUM|LOW) "LOW">
```

Note that **LOW** will be used as the default priority value.

Revised Memo DTD

```
<!ELEMENT MEMO     (TO,FROM,SUBJECT,BODY,SIG)>
<!ATTLIST MEMO priority (HIGH|MEDIUM|LOW) "LOW">
<!ELEMENT TO       (#PCDATA)>
<!ELEMENT FROM     (#PCDATA)>
<!ELEMENT SUBJECT  (#PCDATA)>
<!ELEMENT BODY     (PARA*)>
<!ELEMENT PARA     (#PCDATA)>
<!ELEMENT SIG      (#PCDATA)>
```

XML Document with Attributes

```
<?xml version="1.0"?>
<!DOCTYPE MEMO SYSTEM "memo.dtd">
<MEMO priority="HIGH">
  <TO>CSIS 4244 Students</TO>
  <FROM>Your Beloved Professor</FROM>
  <SUBJECT>This is XML</SUBJECT>
  <BODY>
     <PARA>...</PARA>
  </BODY>
  <SIG>Mike Olan</SIG>
</MEMO>
```

Validity Checking

DTD's define rules for new languages

- A *valid* XML document satisfies these rules
- A validating parser will test for valid documents (Is this a *valid* memo?…)
- Browsers, generally don't do validity checking

Well-formed vs. Valid XML

A document is *well-formed* if it conforms to the syntax of XML.

A well-formed document is *valid* only if it has a document type declaration and the document meets the constraints of that declaration

(XML documents are not required to have a DTD)

DTD vs. Schema

DTD's are not common anymore

- DTD's have a different syntax than XML

XML Schema will eventually replace DTD's

- Schema use the same syntax as XML
- Schema provide more control over document content (type information, rather than just #PCDATA)

Constraints in XML Rules

Start-tag

```
[40] STag      ::= '<' Name (S Attribute)* S? '>'
                                    [WFC: Unique Att Spec]
[41] Attribute ::=   Name Eq AttValue
                                    [VC: Attribute Value Type]
```

Well-formedness constraint: Unique Att Spec

An attribute name MUST NOT appear more than once in the same start-tag or empty-element tag.

Validity constraint: Attribute Value Type

The attribute MUST have been declared; the value MUST be of the type declared for it.

XSL

XML Stylesheet Language

XSL describes how an XML document should be displayed

Has several parts, including

- XSLT - a scripting language for transforming XML documents
- Has control statements for loops, etc.
- Recognized by all modern browsers

http://www.w3schools.com/xsl/