

Extensible Markup Language (XML)

Pat Morin
COMP 2405

Outline

- What is XML?
- XML versus HTML
- Simple XML Documents
- XML Document Type Definitions
- Example DTDs
- XML Schema
- XML and CSS
- XHTML

Examples taken from <http://www.w3schools.com/>

What is XML?

- Stands for Extensible Markup Language
- Similar to HTML
- Used to describe data
- Has no predefined tags
- Uses a Document Type Definition (DTD) or XML Schema to describe data
 - XML Data is self-describing

XML Versus HTML

- XML and HTML are both markup languages
- HTML is for displaying data
- XML is for describing data
- XHTML is a version HTML in XML
- XML does not DO anything
 - Designed to structure and store information only
 - Applications that use XML are what do things

XML Tags

- XML tags are similar to HTML tags but
 - They are case-sensitive
 - All tags must be closed
- Like HTML tags they must be properly nested
- All XML documents must have a single root element that contains all other elements
 - This root element can have any name
- All XML attribute values must be quoted

Simple Example

- Note: The first line is not an XML tag

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

<!-- This is my first picture -->
<picture title="My First Picture">
  <polygon boundary="ed" interiorr="# fefefe">
    <point x="0" y="0"/>
    <point x="0" y="1"/>
    <point x="1" y="1"/>
    <point x="1" y="0"/>
  </polygon>
</picture>
```

XML Content

- XML elements have different kinds of content
 - Element content - tags
 - Simple/Text content – plain text (no tags)
 - Mixed content – simple and element content
 - Empty content – empty content
- XML elements can also have attributes

Book Example

```
<book>
<title>My First XML</title>
<prod id="33-657" media="paper"></prod>
<chapter>Introduction to XML
<para>What is HTML</para>
<para>What is XML</para>
</chapter>
<chapter>XML Syntax
<para>Elements must have a closing tag</para>
<para>Elements must be properly nested</para>
</chapter>
</book>
```


XML Attributes

- XML elements can have attributes
- Attribute values must be quoted with either single or double quotes

```
<book title="Pat's first book">
```

```
<player name='Pat "The Pwnerer" Morin' />
```

- Use with care (attributes have limitations)
 - It's possible to use child elements instead, and these are more flexible

```
<player>  
  <name>Pat Morin</name>  
  <alias>The Pwnerer</alias>  
</player>
```

Well-Formed and Valid XML Documents

- A *well-formed* XML document conforms to the XML syntax rules
 - Has a root element
 - Every element has a closing tag
 - Elements are properly nested
 - Has all attribute values quoted
- A *valid* XML document is well-formed and conforms to a document type definition (DTD)

XML Document Type Definitions

- Most applications will not be able to deal with general XML documents
- Instead, they expect documents that have a specific structure
- This structure can be defined with an XML *Document Type Definition* (DTD)
- A DTD specifies the root node's tag name and what it contains

XML DTD Example

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from    (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body    (#PCDATA)>
]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

XML DTD's

- Since a DTD is used for many documents, they can be included as separate files

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

note.dtd

- The note.dtd file would contain

```
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

DTD Building Blocks

- In a DTD we can specify
 - Elements – tags and the stuff text between them
 - Attributes – information about elements
 - Entities – special character <, > & " '
 - PCDATA – parsed character data
 - Parsed by the XML parser and examined for markup
 - CDATA – (unparsed) character data

Elements

- There are different ways to declare an element
 - Empty
 - Parsed character data
 - Anything
 - With a specific sequence of children

```
<!ELEMENT br EMPTY>  
<!ELEMENT p (#PCDATA)>  
<!ELEMENT x ANY>  
<!ELEMENT note (to,from,heading,body)>
```


Elements with Children

- Child sequences can be specified using a syntax similar to regular expressions
 - `<!ELEMENT picture (polygon+)>`
 - `<!ELEMENT picture (polygon+)>`
 - `<!ELEMENT picture (polygon?)>`
 - `<!ELEMENT polygon (point,point,point+)>`
 - `<!ELEMENT picture (polygon|image)>`
 - `<!ELEMENT picture (polygon|image)*>`

Element Attributes

- We can also specify which attributes an element has
 - `<!ATTLIST element-name attribute-name attribute-type default-value>`

```
<!ATTLIST polygon boundary CDATA "black">
<!ATTLIST polygon interior CDATA "white">
<!ATTLIST polygon fill (true|false) "true">
<!ATTLIST point x CDATA "0">
```

Attribute Value Types

- Attribute values types can be
 - CDATA - The value is character data
 - (en1|en2|..) - The value must be one from an enumerated list
 - ID - The value is a unique id
 - IDREF - The value is the id of another element
 - IDREFS - The value is a list of other ids
 - NMTOKEN - The value is a valid XML name
 - NMTOKENS - The value is a list of valid XML names
 - ENTITY - The value is an entity
 - ENTITIES - The value is a list of entities
 - NOTATION - The value is a name of a notation
 - xml: - The value is a predefined xml value

Default Attribute Values

- Default attribute values can be
- Value - The default value of the attribute
- #REQUIRED - The attribute value must be included in the element (no default)
- #IMPLIED - The attribute does not have to be included
- #FIXED value - The attribute value is fixed

Entities

- Entities are variables used to define common text
 - `<!ENTITY entity-name "entity-value">`

```
<!ENTITY copyright "Copyright 2007 Pat Morin">
```

```
...  
(in XML file:)  
&copyright;
```

Example – TV Schedule

```
<!DOCTYPE TVSCHEDULE [  
<!ELEMENT TVSCHEDULE (CHANNEL+)>  
<!ELEMENT CHANNEL (BANNER, DAY+)>  
<!ELEMENT BANNER (#PCDATA)>  
<!ELEMENT DAY (DATE, (HOLIDAY|PROGRAMSLOT+)+)>  
<!ELEMENT HOLIDAY (#PCDATA)>  
<!ELEMENT DATE (#PCDATA)>  
<!ELEMENT PROGRAMSLOT (TIME, TITLE, DESCRIPTION?)>  
<!ELEMENT TIME (#PCDATA)>  
<!ELEMENT TITLE (#PCDATA)>  
<!ELEMENT DESCRIPTION (#PCDATA)>  
<!ATTLIST TVSCHEDULE NAME CDATA #REQUIRED>  
<!ATTLIST CHANNEL CHAN CDATA #REQUIRED>  
<!ATTLIST PROGRAMSLOT VTR CDATA #IMPLIED>  
<!ATTLIST TITLE RATING CDATA #IMPLIED>  
<!ATTLIST TITLE LANGUAGE CDATA #IMPLIED>  
>]
```

Example – Newspaper

```
<!DOCTYPE NEWSPAPER [  
<!ELEMENT NEWSPAPER (ARTICLE+)>  
<!ELEMENT ARTICLE (HEADLINE, BYLINE, LEAD, BODY, NOTES)>  
<!ELEMENT HEADLINE (#PCDATA)>  
<!ELEMENT BYLINE (#PCDATA)>  
<!ELEMENT LEAD (#PCDATA)>  
<!ELEMENT BODY (#PCDATA)>  
<!ELEMENT NOTES (#PCDATA)> <!ATTLIST ARTICLE AUTHOR  
CDATA #REQUIRED>  
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>  
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>  
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED> <!ENTITY  
NEWSPAPER "Vervet Logic Times">  
<!ENTITY PUBLISHER "Vervet Logic Press">  
<!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic  
Press"> ]>
```

XML Schema

- XML Schema is an XML-based alternative to DTDs
- Differences between Schema and DTDs
 - XML schemas use XML syntax
 - XML schemas support data types
 - XML schemas are extensible

XML Schema – An Example

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com"
  elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML and CSS

- Formatting information can be added to XML documents using CSS
- This works by adding a reference to a CSS stylesheet in the XML document header

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/css" href="note.css"?>  
  
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend</body>  
</note>
```

XML and CSS Continued

- The :before and :after CSS pseudo-elements can be very useful for this

```
to {  
  display: block;  
}  
  
to:before {  
  content: "To: ";  
  font-weight: bold;  
}
```

XML and CSS - Warning

- "CSS is *not* the future of formatting XML"
 - Anonymous web source
- XSL Transformations (XSLT) is a language for transforming XML documents into other XML documents
- To display XML on the web, we could use XSLT to convert our XML document into an XHTML document
- <http://www.w3.org/TR/xslt>

XHTML, an XML Version of HTML

- XHTML is a version of HTML that is proper XML
- Actually, there are several versions
 - XHTML 1.0 Frameset
 - XHTML 1.0 Transitional
 - XHTML 1.0 Strict
- XHTML 1.0 became a W3C Recommendation on January 26, 2000
- Because it is XML, it is defined formally using a DTD
 - E.g., XHTML 1.0 Strict is defined in the DTD <http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

XHTML versus HTML

- XHTML and HTML have mostly the same tags
- Main differences have to do with XML syntax
 - All tags must be closed
 - Empty tags must also be closed
 - Elements must be properly nested
 - Tag names *must be* lowercase
 - Attribute values must be quoted
 - Attributes must have values
 - `<input type="checkbox" checked="checked" />`
 - `<input type="text" readonly="readonly" />`
 - The `id` attribute replaces the `name` attribute

XHTML versus HTML

- All XHTML documents must have a DOCTYPE
- The `html` tag must have an `xmlns` attribute

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title goes here</title>
  </head>
  <body>
  </body>
</html>
```

Converting HTML to XHTML

- w3schools.com was converted to XHTML in two days
 - DOCTYPEs were added
 - Tag and attribute names were converted to lowercase
 - All attributes were quoted
 - Empty tags `<hr>` , `
`, `<input>` and `` were closed
 - The web site was validated
- Could also have used the HTML Tidy utility
 - <http://tidy.sourceforge.net/>

Other Features of XHTML

- The XHTML DTD has been modularized into several different parts
 - Text module
 - Hypertext Module
 - List Module
 - Forms Module
 - Tables Module
 - ...
- Application developers can choose a subset of XHTML tags to support

Summary

- XML is an extensible markup language
- XML is used to describe (store and transmit) data
- DTDs and Schemas can be used to define valid documents
- XML can be formatted with CSS and XSLT
- XHTML is (mostly) just like HTML