

Lecture #8

NEWM N510: Web-Database Concepts

XML

kharrazi@iupui.edu
<http://www.info510.com>

Review Last Lecture

- HTML: Head
- HTML: DTD (Meta)
- HTML: Styles (CSS)
- HTML: Scripts (JS)
- HTML: Forms (Get/Post)

XML in a Nutshell

1. XML
2. XSL
3. XSLT
4. XSL-FO
5. XPath
6. XQuery
7. XLink/XPointer
8. DTD
9. Schema (XSD)
10. XML DOM
11. XForms
12. SOAP
13. WSDL
14. RDF
15. RSS
16. WAP

1. XML

- **XML and HTML Differences:**
 - XML was designed to **carry data**.
 - XML is **not a replacement** for HTML.
 - XML and HTML were designed with different goals
 - XML was designed to describe data and to focus on **what** data is.
 - HTML was designed to display data and to focus on **how** data looks.
 - HTML is about displaying information, while XML is about describing information.

XML (cont.)

- **What is XML?**
 - XML stands for **EXtensible Markup Language**
 - XML is a markup language much like HTML
 - XML was designed to describe data
 - XML **tags are not predefined**. You must define your own tags
 - XML uses a Document Type Definition (**DTD**) or an **XML Schema (XSD)** to describe the data
 - XML with a DTD or XML Schema is designed to be self-descriptive
 - XML is a W3C Recommendation

XML (cont.)

- XML is free and extensible
- XML tags are **NOT** predefined. You must "invent" your own tags.
- The tags used to mark up HTML documents and the structure of HTML documents are predefined. The author of HTML documents can only use tags that are defined in the HTML standard (like `<p>`, `<h1>`, etc.).
- XML allows the author to define his own tags and his own document structure. The tags in the example (like `<patient>` and `<lab>`) are not defined in any XML standard. These tags are "invented" by the author of the XML document.

XML (cont.)

- XML is a complement to HTML
- XML is **not a replacement for HTML**.
- It is important to understand that XML is not a replacement for HTML. In future Web development it is most likely that XML will be used to describe the data, while HTML will be used to format and display the same data.
- The best description of XML is this: *XML is a cross-platform, software and hardware independent tool for transmitting information.*
- XML in future Web development

XML (cont.)

- **XML usage:**
 - XML can Separate Data from HTML
 - XML is used to **Exchange Data** (such as HL7)
 - XML and B2B
 - XML can be used to Share Data
 - XML can be used to Store Data
 - XML can make your Data more Useful
 - XML can be used to Create new Languages

XML (cont.)

- XML **does not do anything**
- XML was not designed to do anything.
- Maybe it is a little hard to understand, but XML does not do anything. XML was created to structure, store and to send information.

```
<root>  
    <node1>item_1</node1>  
    <node2>item_2</node2>  
    <node3>item_3</node3>  
    <node4>item_4</node4>  
    <node5>item_5</node5>  
</root>
```

XML (cont.)

- All XML elements must have a **closing tag**
- XML tags are **case sensitive**
- All XML elements must be **properly nested**
- All XML documents must have a **root element**
- With XML, **white space** is preserved
- XML is just **plain text** with the addition of some XML tags enclosed in angle brackets.
- Attribute values must always be **quoted**

```
<root>  
  <node1 id='item'>item_1</node1>  
  <node2>item_2</node2>  
  <node3>item_3</node3>  
</root>
```

XML (cont.)

- **XML Elements:**

- Names can contain letters, numbers, and other characters
- Names must **not start with a number or punctuation** character such as 3, ", : or '
- Names **must not start with the letters xml** (or XML or Xml ..)
- Names **cannot contain spaces**
- Non-English letters like éòá are perfectly legal in XML element names

```
<root>  
    <node1>item_1</node1>  
    <2node>item_2</2node>  
    <:node3>item_3</:node3>  
</root>
```

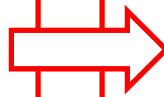
XML (cont.)

- **XML Attributes:**
 - XML elements can have attributes. From HTML you will remember this: ``. The SRC attribute provides additional information about the IMG element.
 - Attributes often provide information that is **not a part of the data**. There are no rules about when to use attributes, and when to use child elements.

```

<root>
  <node name='file'>
    <child1>data1</child1>
    <child1>data2</child1>
  </node>
</root>

```



```

<root>
  <node>
    <name>file</name>
    <child1>data1</child1>
    <child1>data2</child1>
  </node>
</root>

```

XML (cont.)

- **XML Attributes Problems:**
 - Attributes **cannot contain multiple values** (child elements can)
 - Attributes are **not easily expandable** (for future changes)
 - Attributes **cannot describe structures** (child elements can)
 - Attributes are more difficult to manipulate by program code
 - Attribute values are **not easy to test against a Document Type Definition (DTD)** - which is used to define the legal elements of an XML document
- But attributes are perfect for defining ID values (unique identifiers) of different nodes.

XML (cont.)

```
<patient>
```

```
  <first_name>Peter</first_name>
```

```
  <last_name>McDonald</last_name>
```

```
  <lab>
```

```
    <rbc>5.5</rbc>
```

```
    <wbc>6.1</wbc>
```

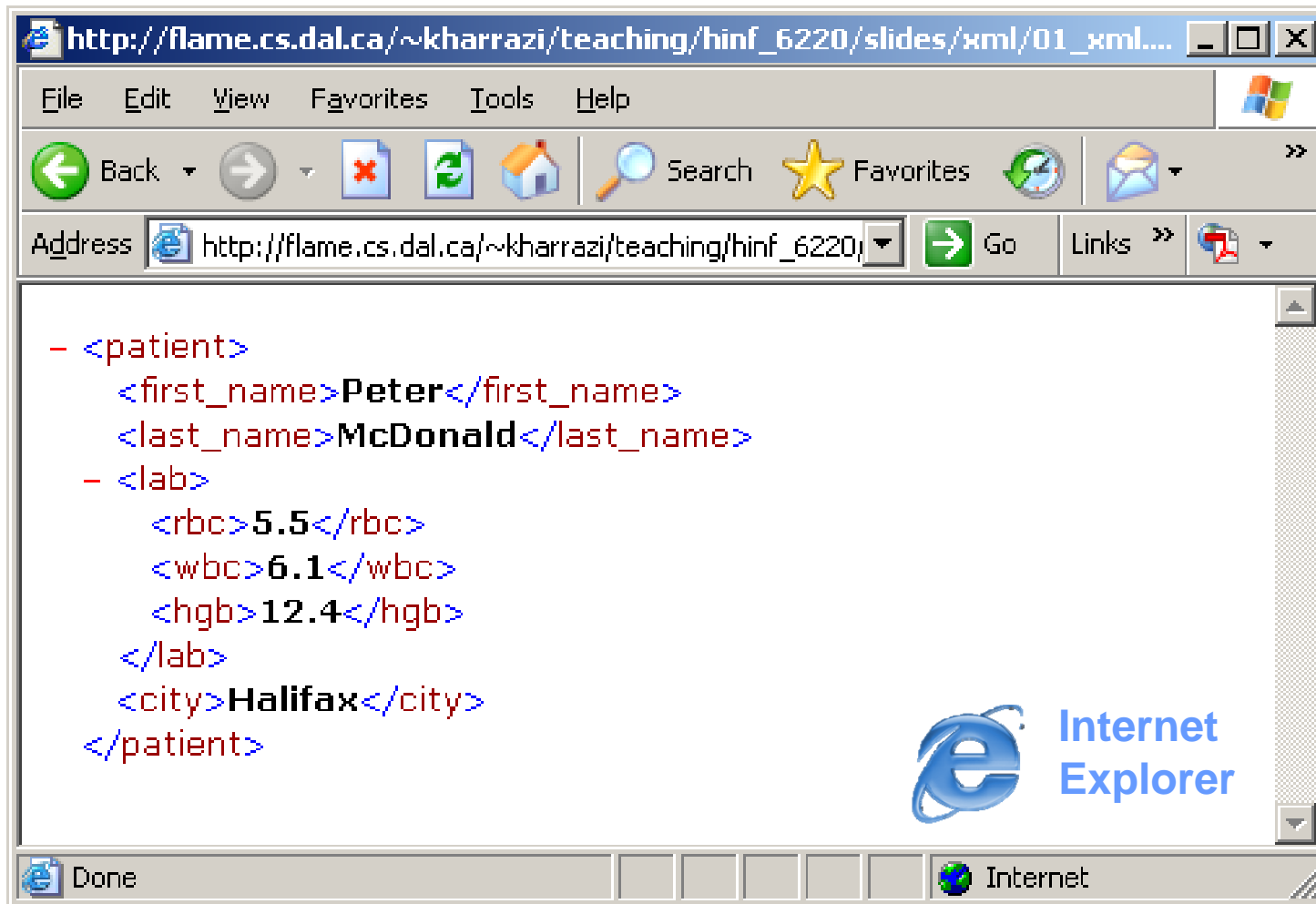
```
    <hgb>12.4</hgb>
```

```
  </lab>
```

```
  <city>Halifax</city>
```

```
</patient>
```

XML (cont.)



XML (cont.)



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

- <patient>
  <first_name>Peter</first_name>
  <last_name>McDonald</last_name>
  - <lab>
    <rbc>5.5</rbc>
    <wbc>6.1</wbc>
    <hgb>12.4</hgb>
  </lab>
  <city>Halifax</city>
</patient>

```

Done

XML (cont.)

- **XML Browser Support**

- **Internet Explorer 6:** Internet Explorer 6 has full XML support, including Namespaces, Style sheets in CSS, and XSLT 1.0. The built-in XML Parser 3.0 in Internet Explorer 6.0 and Windows XP is based on both the W3C XSLT 1.0 and the W3C XPath 1.0 Recommendations.
- **Firefox 1.0.2:** The Firefox browser has support for XML and XSLT (and XML + CSS).
- **Mozilla 1.7.8:** Mozilla includes Expat for XML parsing and has support to display XML + CSS.
- **Opera 8:** Opera supports XML and XML + CSS.
- **Netscape 6:** Netscape supports XML.

XML (cont.)

- **Further Readings:**
 - XML Data Island
 - XML Parser
 - XML Namespaces
 - XML CDATA
 - XML Encoding
 - XML Server
 - XML Application
 - XML HTTP Request
 - XML Save Data
 - XML Behaviors
 - XML Technologies
 - XML Editors

2. XSL

- XSL stands for **EXtensible Stylesheet Language**.
- The World Wide Web Consortium (W3C) started to develop XSL because there was a need for an XML-based Stylesheet Language.
- **CSS = HTML Style Sheets**: HTML uses predefined tags and the meaning of the tags are well understood.
- **XSL = XML Style Sheets**: XML does not use predefined tags (we can use any tag-names we like), and the meaning of these tags are not well understood.

XSL (cont.)

- XSL consists of three parts:
 - **XSLT** - a language for transforming XML documents
 - **XPath** - a language for navigating in XML documents
 - **XSL-FO** - a language for formatting XML documents

3. XSLT

- **What is XSLT?**
 - XSLT stands for **XSL Transformations**
 - XSLT is the most important part of XSL
 - XSLT transforms an XML document into another XML document
 - XSLT uses XPath to navigate in XML documents
 - XSLT is a W3C Recommendation

XSLT (cont.)

- **Correct Style Sheet Declaration:**
- The root element that declares the document to be an XSL style sheet is `<xsl:stylesheet>` or `<xsl:transform>`.
- The correct way to declare an XSL style sheet according to the W3C XSLT Recommendation is:

```
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

or

```
<xsl:transform version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

XSLT (cont.)

XML File

```
<?xml-stylesheet type="text/xsl" href="02_patient.xsl"?>
<root>
  <patient id='1'>
    <first_name>Peter</first_name>
    <last_name>McDonald</last_name>
    <lab>
      <rbc>5.5</rbc>
      <wbc>6.1</wbc>
      <hgb>12.4</hgb>
    </lab>
    <city>Halifax</city>
  </patient>
  <patient id='2'>
    <first_name>Mark</first_name>
    <last_name>Fraser</last_name>
    <lab>
      <rbc>8.3</rbc>
      <wbc>9.2</wbc>
      <hgb>16.7</hgb>
    </lab>
    <city>Montreal</city>
  </patient>
</root>
```

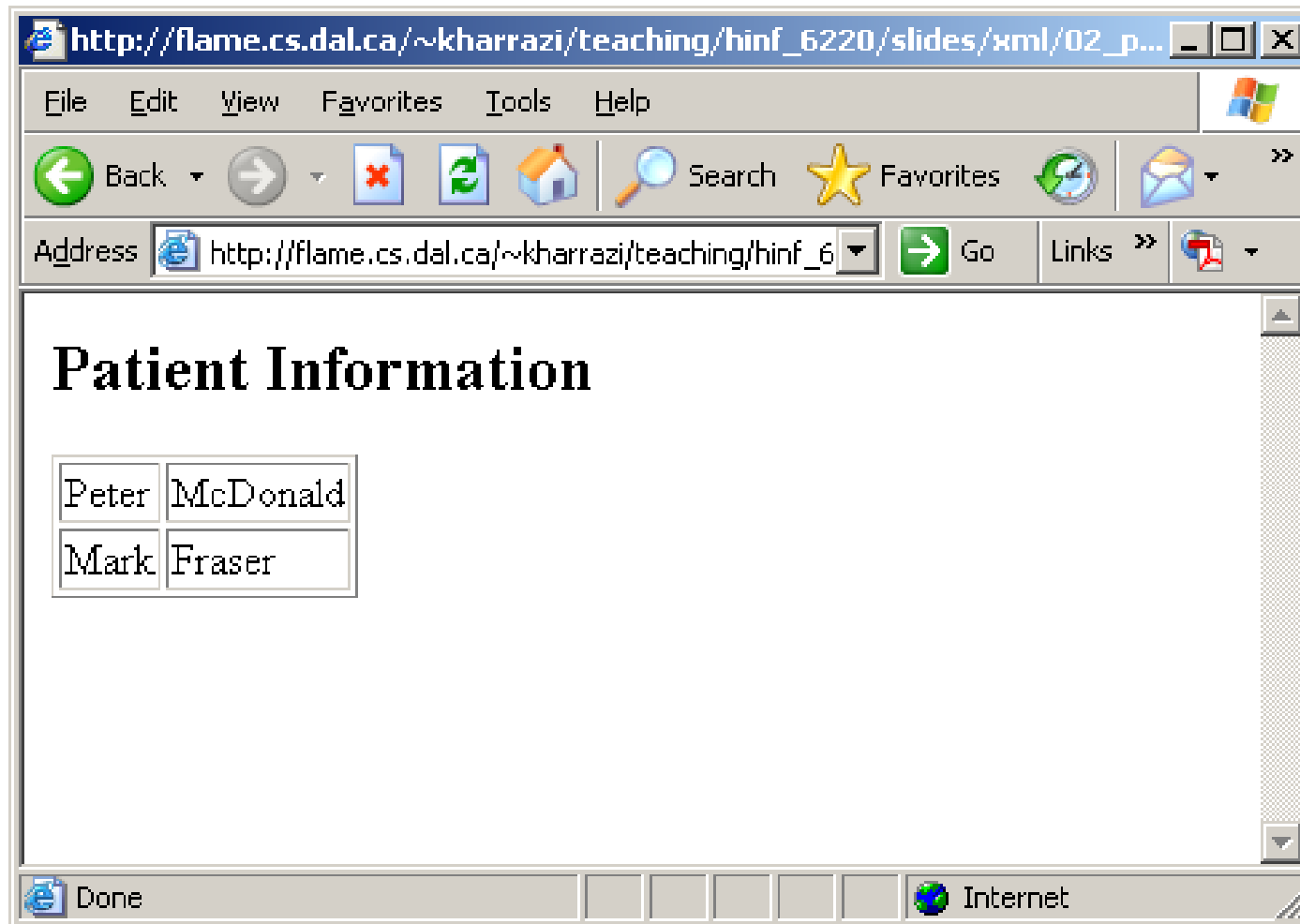
XSLT (cont.)

XSLT File

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Patient Information</h2>
    <table border="1">
      <xsl:for-each select="root/patient">
        <tr>
          <td><xsl:value-of select="first_name"/></td>
          <td><xsl:value-of select="last_name"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>

```


XSLT (cont.)**XML File Output**

XSLT (cont.)

- **Further Readings:**
 - XSLT Templates
 - XSLT <value-of>
 - XSLT <for-each>
 - XSLT <sort>
 - XSLT <if>
 - XSLT <choose>
 - XSLT Elements
 - XSLT Functions

4. XSL-FO

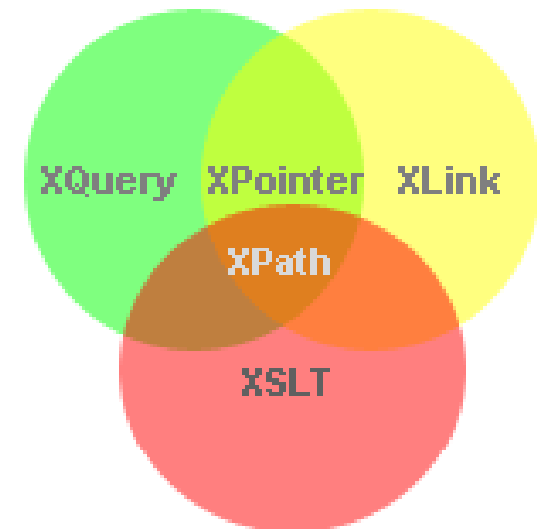
- What is XSL-FO?
 - XSL-FO is a language for formatting XML data
 - XSL-FO stands for Extensible Stylesheet Language Formatting Objects
 - XSL-FO is a W3C Recommendation
 - **XSL-FO is now formally named XSL (formatting)**

XSL-FO (cont.)

- **Further Readings:**
 - XSLFO Documents
 - XSLFO Areas
 - XSLFO Output
 - XSLFO Flow
 - XSLFO Pages
 - XSLFO Blocks
 - XSLFO Lists
 - XSLFO Tables
 - XSLFO & XSLT
 - XSLFO Software

5. XPATH

- **What is XPATH?**
 - XPath is a syntax for defining parts of an XML document
 - XPath uses path expressions to **navigate in XML documents**
 - XPath contains a library of standard functions
 - XPath is a major element in XSLT
 - XPath is a W3C Standard
 - In XPath, there are seven kinds of nodes: *element, attribute, text, namespace, processing-instruction, comment, and document (root) nodes.*



XPATH (cont.)

- XPath is a language for **finding information in an XML document**. XPath is used to navigate through elements and attributes in an XML document.
- XPath is a major element in the W3C's XSLT standard - and **XQuery and XPointer are both built on XPath expressions**.
- So an understanding of XPath is fundamental to a lot of advanced XML usage.
- XPath uses path expressions to select nodes or node-sets in an XML document.
- XPath includes over 100 built-in functions.

XPATH (cont.)

XPath Terminology:

Node: In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and document (root) nodes. XML documents are treated as trees of nodes. The root of the tree is called the document node (or root node). In the next slide:

<code><patient>...</patient></code>	<i>root node</i>
<code><first_name>...<first_name></code>	<i>element node</i>
<code>id = 'CBC'</code>	<i>attribute node</i>

Atomic values: Atomic values are nodes with no children or parent. In the next slide *'CBC'* and *Peter* are atomic values.

XPATH (cont.)

XML File

```
<patient>  
  
  <first_name>Peter</first_name>  
  <last_name>McDonald</last_name>  
  
  <lab id='CBC'>  
    <rbc>5.5</rbc>  
    <wbc>6.1</wbc>  
    <hgb>12.4</hgb>  
  </lab>  
  
  <city>Halifax</city>  
  
</patient>
```


XPATH (cont.)

Relationship of Nodes:

Parent: Each element and attribute has one parent. In the previous slide `<lab>` is the parent of `<rbc>`

Children: Element nodes may have zero, one or more children. In the previous slide `<rbc>`, `<wbc>` and `<hgb>` are the children of `<lab>`

Siblings: Nodes that have the same parent. In the previous slide `<rbc>`, `<wbc>` and `<hgb>` are siblings.

Ancestors: A node's parent, parent's parent, etc.

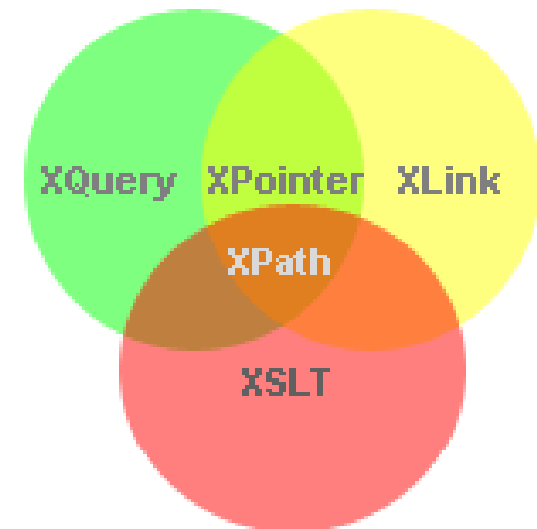
Descendants: A node's children, children's children, etc.

XPATH (cont.)

- **Further Readings:**
 - XPath Syntax
 - XPath Axes
 - XPath Operators
 - XPath Examples
 - XPath Functions

6. XQUERY

- What is XQUERY?
 - XQuery is the language for **querying** XML data
 - XQuery for XML is **like SQL for databases**
 - XQuery is built on XPath expressions
 - XQuery is defined by the W3C
 - XQuery is supported by all the major database engines (IBM, Oracle, Microsoft, etc.)
 - XQuery will become a W3C standard - and developers can be sure that the code will work among different products



XQUERY (cont.)

- The best way to explain XQuery is to say that XQuery is to XML what **SQL is to database tables**.
- XQuery is designed to query XML data - not just XML files, but **anything that can appear as XML**, including databases.
- XQuery can be used to:
 - Extract information to use in a Web Service
 - Generate summary reports
 - Transform XML data to XHTML
 - Search Web documents for relevant information

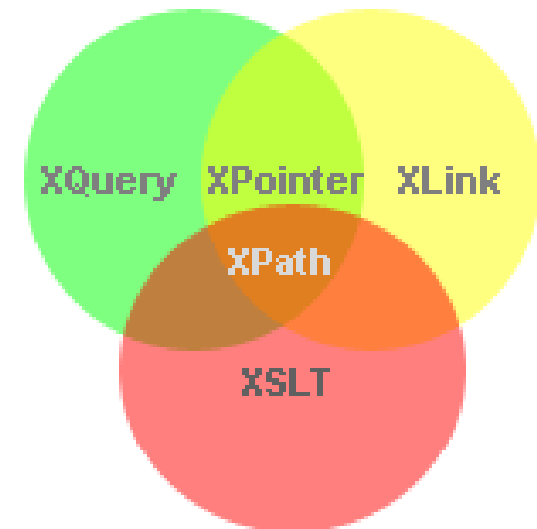
XQUERY (cont.)

- **Further Readings:**

- XQuery FLWOR (for loop, where, order, return)
- XQuery HTML
- XQuery Terms
- XQuery Syntax
- XQuery Add
- XQuery Select
- XQuery Functions

7. XLINK & XPOINTER

- Linking in XML is divided into two parts: XLink and XPointer.
- **XLink** defines a standard way of creating hyperlinks in XML documents.
- **XPointer** allows the hyperlinks to point to more specific parts (fragments) in the XML document.



XLINK & XPOINTER (cont.)

- **What is XLINK?**
 - XLink is short for the **XML Linking Language**
 - XLink is a language for **creating hyperlinks in XML** documents
 - XLink is similar to HTML links - but it is a lot more powerful
 - ANY element in an XML document can behave as an XLink
 - XLink supports simple links (like HTML) and extended links (for linking multiple resources together)
 - With XLink, the links can be defined outside of the linked files
 - XLink is a W3C Recommendation

```
<homepage xlink:type="simple"  
  xlink:href="http://informatics.iupui.edu/">  
  Visit School of Informatics  
</homepage>
```

XLINK & XPOINTER (cont.)

- **What is XPOINTER?**

- XPointer is short for the XML Pointer Language
- XPointer allows the **hyperlinks to point to specific parts** of the XML document (like a # bookmark in HTML)
- XPointer uses XPath expressions to navigate in the XML document
- XPointer is a W3C Recommendation

```
<homepage xlink:type="simple"
  xlink:href="http://iupui.edu#id(faculty').child(5,item)">
  Visit School of Informatics Faculty Page
</homepage>
```


8. DTD

- The purpose of a **Document Type Definition (DTD)** is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared **inline** in your XML document, or as an **external** reference.
- With DTD, each of your XML files can carry a **description** of its own format with it.
- With a DTD, independent groups of people can agree to use a common DTD for **interchanging data**.
- Your application can use a standard DTD to verify that the data you receive from the outside world is **valid**. You can also use a DTD to **verify** your own data.

DTD (cont.)

- **Internal DOCTYPE declaration**

- If the DTD is included in your XML source file, it should be wrapped in a DOCTYPE definition with the following syntax:

```
<!DOCTYPE root-element [element-declarations]>
```

- **External DOCTYPE declaration**

- If the DTD is external to your XML source file, it should be wrapped in a DOCTYPE definition with the following syntax:

```
<!DOCTYPE root-element SYSTEM "filename">
```

DTD (cont.)

XML File + Embedded DTD

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT patient (first_name, last_name, city)>
  <!ELEMENT first_name      (#PCDATA)>
  <!ELEMENT last_name       (#PCDATA)>
  <!ELEMENT city            (#PCDATA)>
]>
<patient>

  <first_name>Peter</first_name>
  <last_name>McDonald</last_name>
  <city>Halifax</city>

</patient>
```

DTD (cont.)

XML File

```

<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "hospital.dtd">
<patient>
  <first_name>Peter</first_name>
  <last_name>McDonald</last_name>
  <city>Halifax</city>
</patient>

```

External DTD File (hospital.dtd)

```

<!ELEMENT patient (first_name, last_name, city)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT city (#PCDATA)>

```

DTD (cont.)

- **The building blocks of XML documents:** Seen from a **DTD point of view**, all XML documents (and HTML documents) are made up by the following building blocks:
 - **Elements** are the main building blocks of both XML and HTML documents such as `<patient>...</patient>`
 - **Tags** are used to markup elements such as `<patient>`
 - **Attributes** provide extra information about elements such as `<patient id='45'>`
 - **Entities** are variables used to define common text such as `&` for `&` or `<` for `<`
 - **PCDATA** means parsed character data and it will be parsed by a parser.
 - **CDATA** also means character data but it will not be parsed by a parser.

DTD (cont.)

- **Further Readings:**
 - DTD Elements
 - DTD Attributes
 - DTD Entities
 - DTD Validation

9. XML Schema

- XML Schema is an XML based **alternative to DTD**.
- An XML schema describes the structure of an XML document.
- The XML Schema language is also referred to as XML Schema Definition (**XSD**).

XML Schema (cont.)

- What is an XML Schema?
 - defines **elements** that can appear in a document
 - defines **attributes** that can appear in a document
 - defines which elements are **child** elements
 - defines the **order** of child elements
 - defines the **number** of child elements
 - defines whether an element is **empty** or can include **text**
 - defines **data types** for elements and attributes
 - defines **default and fixed values** for elements and attributes
 - *XML Schemas usually are linked as an external file.*

Very soon XML Schemas will be used in most Web applications as a replacement for DTDs

XML Schema (cont.)

XML File

```
<?xml version="1.0"?>
<patient xmlns="http://www.yoursite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.yoursite.com/hospital.xsd">
    <first_name>Peter</first_name>
    <last_name>McDonald</last_name>
    <city>Halifax</city>
</patient>
```

XML Schema (cont.)

External XSD File (hospital.xsd)

```
<?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="patient">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="first_name" type="xs:string"/>
        <xs:element name="last_name" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML Schema (cont.)

- **Further Readings:**

- XSD <schema>

- **Simple Types**

- XSD Elements
- XSD Attributes
- XSD Restrictions

- **Data Types**

- XSD String
- XSD Date
- XSD Numeric
- XSD Misc

- **Complex Types**

- XSD Elements
- XSD Empty
- XSD Elements Only
- XSD Text Only
- XSD Mixed
- XSD Indicators
- XSD <any>
- XSD <anyAttribute>
- XSD Substitution
- XSD Example

- **References**

- XSD Reference
- XSD Validator

10. XML DOM

- The XML **Document Object Model** (XML DOM) defines a standard way for accessing and manipulating XML documents.
- The DOM presents an XML document **as a tree structure**, and gives access to the structure through a set of objects.

11. XFORMS

- What is XFORMS?
 - XForms is the **next generation of HTML forms**
 - XForms is richer and **more flexible than HTML forms**
 - XForms will be the forms standard in XHTML 2.0
 - XForms is platform and device independent
 - XForms **separates data and logic** from presentation
 - XForms uses XML to define form data
 - XForms stores and transports data in XML documents
 - XForms contains features like calculations and validations of forms
 - XForms **reduces or eliminates the need for scripting**
 - XForms is a W3C Recommendation

12. SOAP

- What is SOAP?
 - SOAP stands for **Simple Object Access Protocol**
 - SOAP is a communication protocol
 - SOAP is for **communication between applications**
 - SOAP is a format for sending messages
 - SOAP is designed to communicate via Internet
 - SOAP is platform independent
 - SOAP is language independent
 - SOAP is **based on XML**
 - SOAP is simple and extensible
 - SOAP allows you to get around firewalls
 - SOAP will be developed as a W3C standard

13. WSDL

- What is WSDL?
 - WSDL stands for **Web Services Description Language**
 - WSDL is **written in XML**
 - WSDL is an XML document
 - WSDL is used to **describe Web services**
 - WSDL is also used to locate Web services
 - WSDL is now a W3C standard

14. RDF

- What is RDF?
 - RDF stands for **Resource Description Framework**
 - RDF is a **framework for describing resources on the web**
 - RDF provides a model for data, and a syntax so that independent parties can exchange and use it
 - RDF is designed to be read and understood by computers
 - RDF is **not designed for being displayed to people**
 - RDF is **written in XML**
 - RDF is a part of the W3C's **Semantic Web** Activity
 - RDF is a W3C Recommendation

RDF (cont.)

- **Further Readings:**

- RDF Rules
- RDF Example
- RDF Elements
- RDF Containers
- RDF Collections
- RDF Schema
- RDF Dublin Core
- RDF Reference
- RDF OWL (Web Ontology Language)

15. RSS

- What is RSS?
 - RSS stands for **Really Simple Syndication**
 - RSS allows you to **syndicate your site content**
 - RSS is a selected list of defined items on a site
 - RSS defines an **easy way to share and view headlines and content**
 - RSS files can be automatically updated (RSS Feed)
 - RSS allows personalized views for different sites
 - RSS is **written in XML**

16. WAP

- What is WAP?
 - WAP stands for **Wireless Application Protocol**
 - WAP is an application communication protocol
 - WAP is used to access services and information
 - WAP is inherited from Internet standards
 - WAP is for **handheld devices such as PDA and mobile phones**
 - WAP is a protocol designed for micro browsers
 - WAP enables the creating of **web applications for mobile devices**
 - WAP uses the **mark-up language WML (not HTML)**
 - WML is defined as an XML 1.0 application

Summary

1. XML
2. XSL
3. XSLT
4. XSL-FO
5. XPath
6. XQuery
7. XLink/XPointer
8. DTD
9. Schema (XSD)
10. XML DOM
11. XForms
12. SOAP
13. WSDL
14. RDF
15. RSS
16. WAP

Next Session

- PHP Intro
- PHP Syntax
- PHP echo
- PHP Comment
- PHP Variables
- PHP Operators
- PHP Conditions
- PHP Loops
- PHP Functions
- PHP Cookies
- PHP SSI
- PHP Forms
- PHP/MySQL Integration

Exercise

- Please refer to the available text file in the slides section for this session on the course website:
- http://info510.com/core/public_page.php?page_name=slides