Adventures in XML

A BASIC INTRODUCTION TO XML

What Is XML and Why Do I Care?

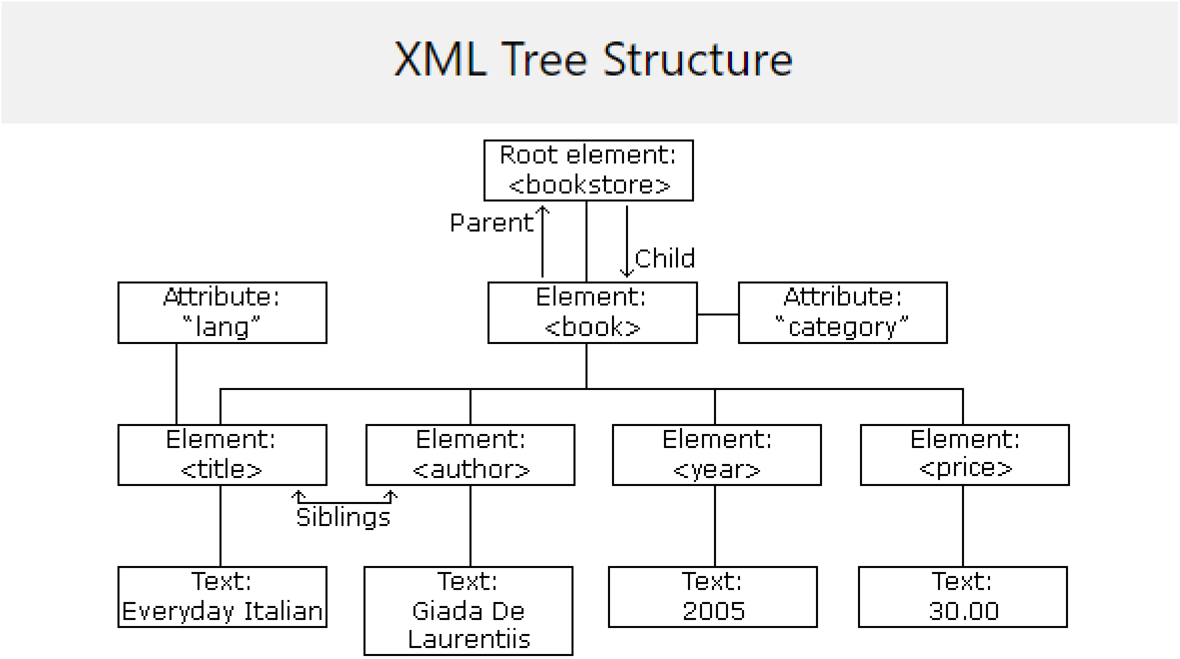
- XML = eXtensible Markup Language, which is:
  - Markup language that defines a set of rules for encoding documents
  - More interested in the meaning of data than its presentation
  - Composed of many different flavors
    - TEI (Text-Encoding Initiative)
    - MEI (Music-Encoding Initiative)
- Designed to store and transport data in a way that is:
  - Software- and hardware-independent
  - Human- and machine-readable

Well-Formed vs. Valid

- If a document is well-formed, it conforms to the basic rules of XML.
- If a document is valid, it conforms to the rules of a DTD or schema. **A document can be well-formed but not valid.**
- We will come back to this!

XML Trees from Root to Leaf

- Root element
  - Parent of all other elements in the document
- Relationships between elements
  - Parent, Child, Sibling
- Elements can be composed of
  - Text content
  - Attributes



XML Tree Structure

XML Syntax: Rules to Live By

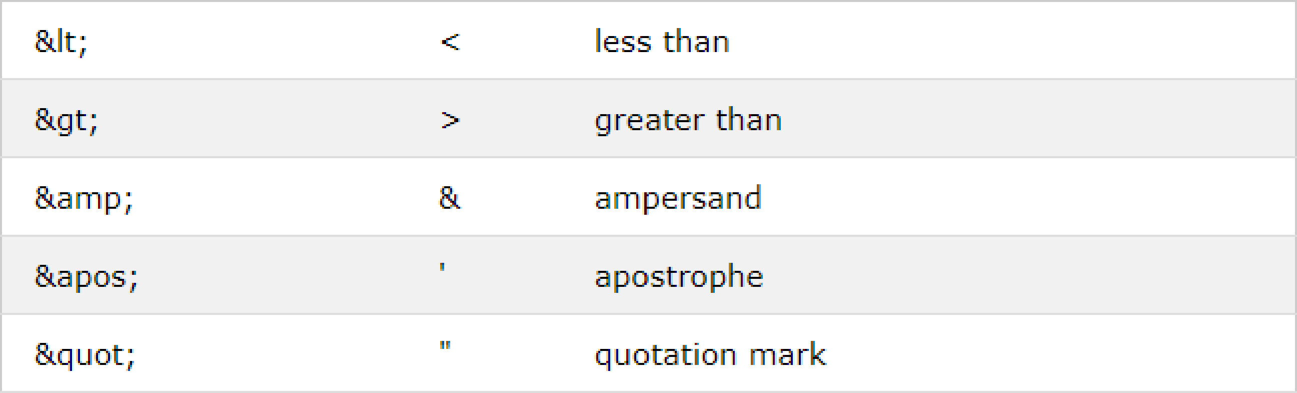
- XML documents must contain one root element
  - Root element for TEI = **<TEI></TEI>**
- XML prolog
  - Optional, but oXygen puts it in automatically
  - If it exists, then it must come *first* in the document
  - Contains information that applies to the document as a whole
    - Character encoding, document structure, style sheets…
    - <?xml version="1.0" encoding="UTF-8"?>
  - Immediately followed by the opening tag of the root element

XML Syntax: Rules to Live By cont.

- All elements must have an opening and closing tag
  - <TEI></TEI>
  - **Exception: elements in the prolog do not have a closing tag**
- XML tags are case sensitive
  - <TEI> ≠ <tei>
- XML elements must be properly nested
  - **Bad:** <b><i>This text is bold and italic</b></i>
  - **Good:** <b><i>This text is bold and italic</i></b>
- XML attribute values must always be quoted
  - <note date=“5/17/2018”></note>

XML Syntax: Rules to Live By cont.

- Entity References
  - Characters that have a special meaning in XML
  - Improper use will generate an XML error
  - Five pre-defined entity references in XML



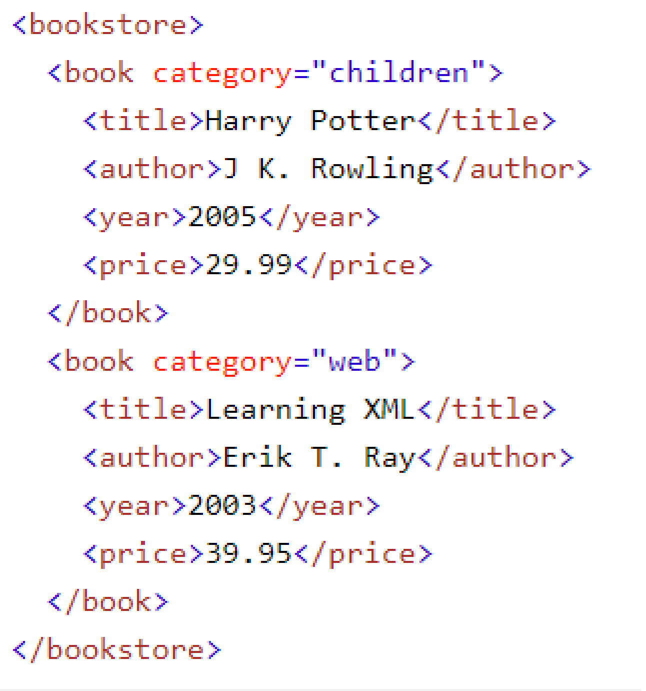
&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

XML Syntax: Rules to Live By cont.

- Syntax for comments in XML
    - <!-- This is a comment -->
    - Two dashes in the middle of the comment are not allowed
        - <!-- This is an invalid -- comment -->
- White-space is preserved in XML
    - XML does not truncate multiple white-spaces
- Follow all of the above rules for a “well-formed” XML Document!

Parts of an XML Document: Elements

- XML documents contain XML elements
- An XML element is everything from the element's opening tag to the closing tag
  - Ex: <name>Michelle</name>
  - Elements can contain
    - Text
    - Attributes
    - Other elements
    - Combination of the above



```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML Elements cont.

- Elements with no content are said to be empty
  - <name></name>
  - <name/>
- Naming rules for elements
  - Case-sensitive
  - Must start with a letter or underscore
  - Cannot start with the letters xml
  - Can contain letters, digits, hyphens, underscores or periods
  - Cannot contain spaces

Parts of an XML Document: Attributes

- XML elements can have attributes
  - Attributes are designed to contain data related to a specific element
- Attribute values must always be quoted
  - <person gender=“female”>
  - <gangster name=“George &quot;Shotgun&quot; Ziegler”>

XML Namespaces

- XML defines a set of rules for encoding documents
  - Element names are defined by the developer
    - Element <name>
      - Can mean different things depending on your flavor of XML: TEI vs MEI
      - XML namespaces are a method for avoiding element name conflicts
- Declaring an XML namespace
  - Defined by an xmlns attribute in the start tag of an element
    - xmlns="namespaceURI"
    - Can be declared in the root element of an XML document
      - <TEI xmlns="http://www.tei-c.org/ns/1.0">

XML DOM

- DOM = Document Object Model
  - Defines a standard for accessing and manipulating documents
  - XML DOM
    - How to get, change, add, and delete XML Elements
- XML DOM and Nodes
  - Everything in an XML document is a node
    - Document node
    - Element node
    - Text nodes
    - Attribute nodes
    - Comment nodes

XPath and XSLT

- XPath
  - Major element in the XSLT standard
  - Can be used to navigate through elements and attributes in an XML document
  - Uses path expressions to select nodes or node-sets in an XML document
- XSLT
  - eXtensible Stylesheet Language Transformations
    - Recommended stylesheet language for XML
    - More sophisticated than CSS
  - Transform an XML document into HTML for display

Validating Your XML Document

- A “valid” XML document must be
  - “Well-formed”
  - Conform to a document type definition
    - Defines the rules and legal element names and attributes for an XML document
    - Two different document type definitions can be used with XML:
      - DTD – The original Document Type Definition
      - XML Schema – An XML-based alternative to DTD

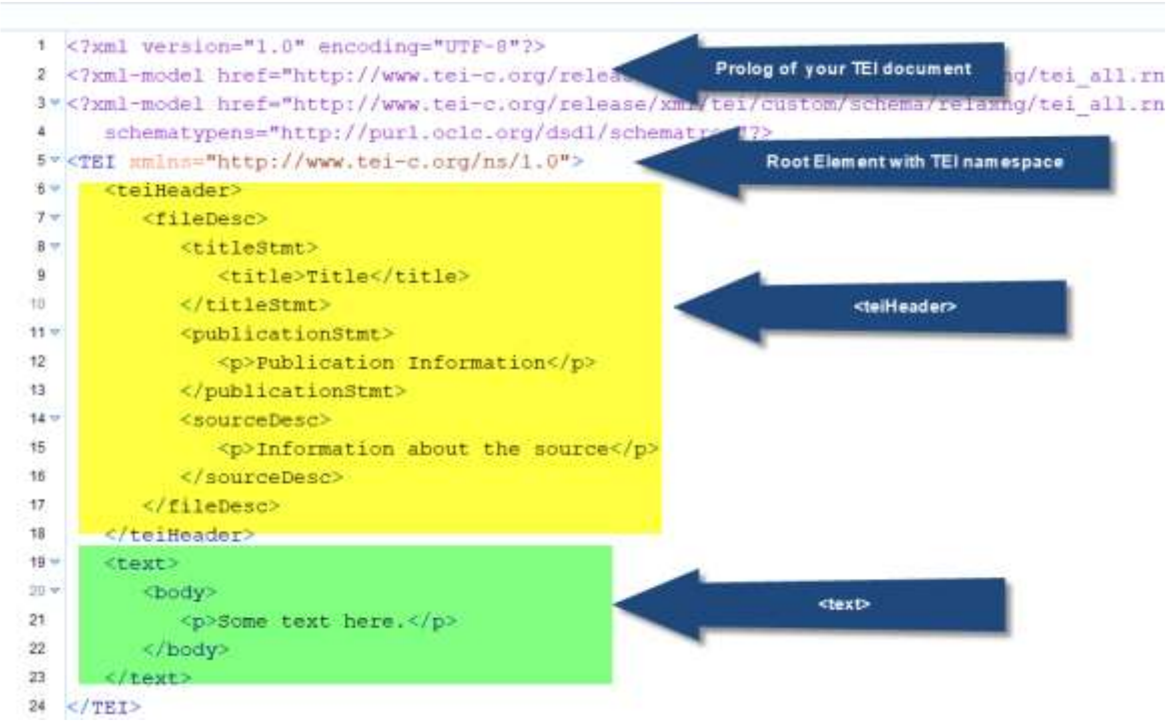
Validating with a Schema

- An XML Schema describes the structure of an XML document, just like a DTD
- XML Schemas are more powerful than DTDs
  - Written in XML
    - RELAX NG – one schema language for XML
  - Are extensible to additions
  - Support data types
  - Support namespaces

Validating Your TEI: ODD & Schemas

- TEI Customizations from the TEI Consortium
  - Examples: Lite, All, Corpus, MS
- Create your own customized version of TEI
  - ODD – One Document Does it all
    - Includes the schema fragments, prose documentation, and reference documentation for the TEI Guidelines in a single document
    - Used to generate a DTD, RELAX NG schema or W3C Schema for validation
    - Can be made using Roma, a tool available from the TEI Consortium (however, the tool is sometimes faulty)

Elements of a TEI Document



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://www.tei-c.org/relea                        g/tei_all.rn
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rn
   schematypens="http://purl.oclc.org/dsdl/schematr      "?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title>Title</title>
         </titleStmt>
         <publicationStmt>
            <p>Publication Information</p>
         </publicationStmt>
         <sourceDesc>
            <p>Information about the source</p>
         </sourceDesc>
      </fileDesc>
   </teiHeader>
   <text>
      <body>
         <p>Some text here.</p>
      </body>
   </text>
</TEI>
```

Prolog of your TEI document

Root Element with TEI namespace

<teiHeader>

<text>

<teiHeader>

- <teiHeader>
  - Information about the document that you are creating
  - Required elements
    - <fileDesc>
      - <titleStmt>
      - <publicationStmt>
      - <sourceDesc>

<text>

- <text>
  - The text you are encoding
  - Required element
    - <body> - the main body of the text
  - Optional elements
    - <front> - used for front matter of a text (contents, preface)
    - <back> - used for back matter of a text (index, appendix)