Comp 336/436 - Markup Languages

Fall Semester 2019 - Week 4

Dr Nick Hayward

XML - test 1

Think about how you might encode the following information:

- a car

- a sports team

- 2 Musical CDs including each song per album per CD (e.g. each album has 5 songs)

XML - test 1 - example solution

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vehicle>
  <car id="1" type="compact" manufacturer="Ford">
    <model>Focus</model>
    <variant>Hatchback</variant>
    <doors>5</doors>
    <trim>Sport</trim>
    <colour>Black</colour>
    <year>2014</year>
    <fuel>Unleaded</fuel>
    <transmission>Manual</transmission>
    …
  </car>
  <boat id="1" type="motor" manufacturer="Princess">
    ...
  </boat>
</vehicle>
```

- demo

To the Lighthouse

(Here Mr. Carmichael, who was reading Virgil, blew out his candle.)

3

But what after all is one night? A short space, especially when the darkness dims so soon, and so soon a bird sings, a cock crows loudly, or a faint green quickens, like a turning leaf, in the hollow of the wave.

It seemed now as if, touched by human penitence and all its tool, divine goodness had parted the curtain and displayed behind it, single, distinct, **the hare erect**; the wave falling; the boat rocking, which, did we deserve them, should be

XML - test 2 - example solution

```
<?xml version="1.0" encoding="UTF-8"?>
<novel>
  <page no="198">
    <line no="1">
      <title align="centre" style="underline" colour="black">To the Lighthouse</ti
    </line>
    <line no="2" type="empty"></line>
    <para no="1">
      <line no="3">(Here Mr. Carmichael, who was reading</line>
      …
    </para>
    <line no="6">
      <section align="centre" style="heading" colour="blue">3</section>
    </line>
    ...
    <line no="11"><rend style="highlight" colour="green">green</rend>...</line>
    ...
  </page>
</novel>
```

- demo

XML - processing instructions

- an XML document can also contain processing instructions
- processing instructions encode application-specific data
  - *e.g. document declaration which opens every XML document*
- each instruction is enclosed in <? and ?> delimiters
- the target identifies the application
  - *an application should ignore unrecognised processing instructions*
- processing instructions allow a developer to enter directives
  - *not part of the XML document's content*
  - *instructions are passed up to different ad-hoc applications*

XML - entities

- an XML document can use also entities
  - *similar to macros for programmers*
  - *or aliases for more complex functions*
- a single entity name can replace lots of text
  - *e.g.* `<!ENTITY editor "yvaine">`
- an entity can be recalled in the content of the document
  - *e.g.* `&editor;`
- use in an XML document, e.g.

```
<person>The editor's name is &editor;</person>
```

XML - CDATA

- we may also use CDATA elements in an XML document
- a CDATA element tells the XML parser not to interpret or parse characters
- the content may now be parsed
  - *even though it contains a character as part of an instruction...*
  - *parser expects an entity*

```
<editors><![CDATA[tristan & yvaine]]></editors>
```

XML - namespaces - intro

- in XML, element names are defined by developers
  - *different organisations may use the same tag*
  - *and then markup content with different semantics*
- XML also designed to enable interoperability and data exchange
  - *method to combine XML sources without ambiguity*
- *namespaces* designed to provide uniquely named elements and attributes
- as defined by the W3C a namespace is
  - *a collection of XML elements and attributes*
  - *identified by an Internationalised Resource Identifier (IRI)*
  - *often referred to as an XML vocabulary*

XML - namespaces - usage

- namespaces are declared as an attribute of an element
  - *using the xmlns name attribute in an element's start tag*
- not mandatory to declare namespaces only at the root element
  - *may be declared at any element in the XML document*
- namespace scope:
  - *begins at the element where it has been declared*
  - *applies to the entire content of that element*
  - *unless overridden by another namespace declaration*

```
<editor xmlns:editor="http://www.mynamespace.com" />
```

- a namespace declaration has the following syntax, e.g.
  - `xmlns:prefix="URI"`

XML - namespaces - structure

- URI is a string of characters, which identifies an Internet Resource
  - *most common URI is the Uniform Resource Locator (URL)*
  - *a less common type of URI is the Uniform Resource Name (URN)*
  - *e.g.* `urn:isbn:0030818516`
- a specific namespace identifies a collection of names
- a collection of element names
  - *e.g. standard XHTML*
- a collection of attribute names
  - *e.g. XLink*
- a namespace can collect names of properties
  - *e.g. FOAF (Friend of a Friend)*
  - *dictionary of named properties and classes*
  - *uses W3C's RDF technology*
- a namespace may describe a set of functions
  - *e.g. the XPath Data Model*

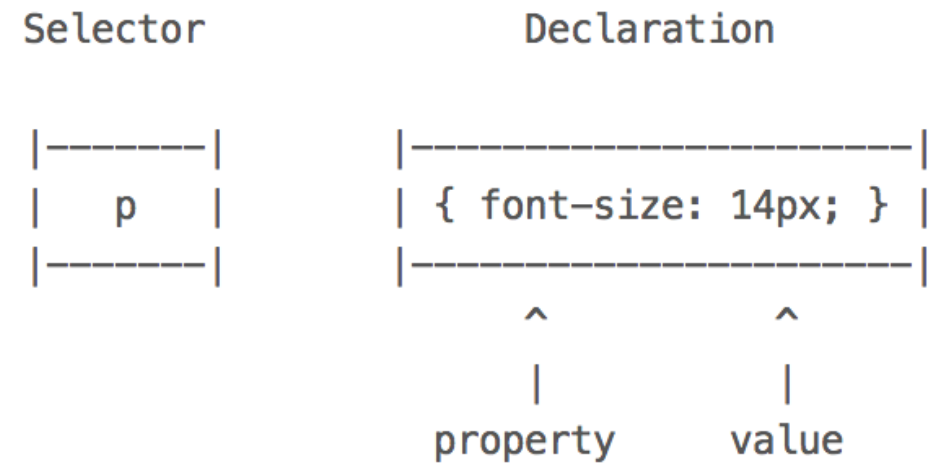
XML - rendering - css styling

- XML was created to structure content, not to display it for users &c.
- render XML with styles in a browser
  - *a CSS stylesheet can be applied similar to HTML files*
- CSS is a language used to describe the presentation of a document
  - *a document written in a markup language*
  - *e.g. SVG has its own CSS properties and values...*
- CSS document is basically constituted by a set of rules
- a ruleset consists of two parts: a selector and a declaration
  - *property and a value, e.g.*

```
selector {property1:value; property2:value;}
```

- selector is a reference to the element to be rendered
- declaration defines applied styles, effects...

Image - CSS Syntax



CSS Syntax

XML - rendering - css stylesheets

- with XML documents
  - *attach external stylesheets using `xml-stylesheet` processing instruction*
  - *add to the prolog of the XML document, e.g.*

```
<?xml-stylesheet href="style.css" type="text/css"?>
```

- there can be multiple XML stylesheet processing instructions
- possible to attach multiple stylesheets to an XML document
- possible attributes are
  - `type` *- specific type, e.g.* `text/css`
  - `medium` *- display medium type, e.g. print, screen...*
  - `title` *- specify local name, a title...*

XML - rendering - css stylesheets example

XML

```
<?xml version="1.0" standalone="yes" ?>
<library>
<title>Example with CSS</title>
<body>body in the library...</body>
</library>
```

XML & CSS stylesheet

```
<?xml version="1.0" standalone="yes" ?>
<?xml-stylesheet href="style1.css" type="text/css" ?>
<library>
<title>Example with CSS</title>
<body>body in the library...</body>
</library>
```

- XML document is still visible to the browser - i.e. data open and accessible
  - *tags rendered and styled with CSS*

XML - viewing examples

- view without styling in a web browser, editor...
  - *demo*
- render with CSS by associating a stylesheet with a given XML file
  - *demo*
- transform the document using XSLT
- access, query and manipulate XML using JavaScript
- parse XML using PHP, render HTML, manipulate with JavaScript, and style with CSS
  - *demo 1 - TEI Parser simple*
  - *demo 2 - TEI Parser full*
  - *Winchester XML*

XML - a few more demos and examples

- eHinman Collator
- regularise - soundex

XML - well-formed - intro

- HTML may often include errors and issues in the markup
  - *a browser helps resolve issues and errors in the markup*
- XML has strict rules
  - *XML document must be correctly structured*
  - *correct structure to enable machine parsing*
- XML specification prohibits XML parsers
  - *parsers may not try to fix and understand malformed documents*
- a parser may simply report the error

XML - well-formed - W3C definition

- W3C definition of a well-formed XML document
  - *unique opening and closing tag that enclose the whole document*
  - *all elements include a closing tag or correctly defined empty elements*
  - *all tags and attributes names adhere to case-sensitive rule*
    - e.g. the tag `<hello>` cannot be closed with `</hello>` &c.
  - *all elements properly nested*
    - i.e., an opening and closing tag
    - tags may not overlap &c.
  - *all attribute values always quoted correctly*
  - ...

n.b. these are the most important constraints - not an exhaustive list...

XML - validation

- *Well Formed* XML = correct syntax, e.g.
  - *root element*
  - *elements must have closing tags*
  - *tags are case sensitive*
  - *elements must be properly nested*
  - *attribute values must be quoted*
- *Valid* XML is *Well Formed* XML that conforms,
  - *e.g. to a DTD (document type definition)*

XML - validation - DTD

- DTD - Document Type Definition
- XML allows a developer to organise their own tags &c.
- XML documents often need to follow a specific grammar
- helps with sharing and reuse
  - *e.g. within a broader community and domain*
- the defined grammar will then become a known tool
  - *describes structure and context of all necessary topic specific XML documents*
- purpose of a Document Type Definition (DTD)
  - *to enable and permit such shared grammars*
- a DTD provides a framework for validating XML documents
  - *by defining legal building blocks of XML documents*
- a DTD outlines permitted elements in an XML document
  - *and any available attributes and sub-elements*
- DTD can be part of the XML document
  - *or it can be referred to by the XML document*
- external DTD is a simple text file with `.dtd` extension

XML - validation - DTD - element declarations

- element declarations describe a permitted set of elements within a document
  - *i.e. nature of declared elements, character data, &c.*
- elements in XML documents may
  - *enclose other elements*
  - *be empty*
  - *contain content*
  - *or be mixed (i.e. containing content and other elements...)*
- in a DTD, possible declarations for elements are as follows,
  - `<!ELEMENT element-name (child1,child2,...)>`
    - for an element containing other elements
  - `<!ELEMENT element-name EMPTY>`
    - for an empty element
  - `<!ELEMENT element-name (#PCDATA)>`
    - for an element containing content
  - `<!ELEMENT element-name (#PCDATA|child1|otherchild1)*>`
    - for a mixed element
  - `<!ELEMENT element-name ANY>`
    - for defining an element
    - no further specific detail provided
  - *a DTD must specify how the element may appear*
  - *i.e. in a given order, if they can be repeated...*

XML - validation - DTD - special characters

DTD special characters for element repetition and order.

character	definition
'	separate sequence items, indicate sequential order or items
|	choice separator, indicates selection of one item from list
( )	group elements
+	required occurrence with repetition
*	optional occurrence with repetition
?	optional occurrence

XML - validation - DTD - special characters

- if we create XML documents describing books
  - *with a title*
  - *multiple authors*
  - *different chapters*
- definition of the element book, e.g.

```
<!element book (title,author+,chapter+)>
```

- element book can contain
  - *only other elements*
  - *not content directly*
  - *a title (title)*
  - *one or more authors (author+)*
  - *successively one or more chapters (chapter+)*

XML - validation - DTD - ATTLIST declarations

- **ATTLIST** declaration
  - *XML element attributes are declared in the DTD*
- attribute-list declarations name
  - *a permitted set of attributes for each declared element*
  - *type of each attribute value,*
  - *explicit set of valid value(s) is not always necessary...*
- syntax for an attribute declaration, e.g.

```
<!ATTLIST elementName attributeName Type defaultValue>
```

XML - validation - DTD - ATTLIST declarations

value	definition
CDATA	character data value
ID	value is unique ID
IDREF	value is another element's ID
IDREFS	value is list of other IDs
NMTOKEN	value is valid XML name
NMTOKENS	value is list of valid XML names
ENTITY	value is entity
ENTITIES	value is list of entities
NOTATION	value is name of a notation
xml:	predefined value

Demos

- TEI Parser - simple
- TEI Parser - full
- XML Basic - cars
- XML Basic - literary text
- XML Basic - CD catalogue
- XML Basic - CD catalogue with CSS
- XML Sample - Winchester

Resources

- MDN - CSS
- W3C - CSS
- W3 Schools - CSS
- list of XML markup languages