Working with well-formed documents

Simon Mahony
From an original document by Susan Hockey

This document is part of a collection of presentations and exercises on XML. For full details of this and the rest of the collection see the cover sheet at: http://humbox.ac.uk/3110/

OPEN EDUCATIONAL RESOURCES FOR THE DIGITAL HUMANITIES

Basics of XML Syntax

- Documents are composed of elements
- Start and end tags for every element

<name>Smith</name>

Element Names

- Must start with a letter or underscore
- Consist of letters, digits, underscore, hyphen or full stop (but avoid the latter). No spaces are permitted.
- Case sensitive
- Cannot start with 'xml' (we will see why later)

<PostCode> <postcode>

<author> <part.number>

Attributes

- Modify an element
- Attributes have a name and a value
- Name follows the rules for XML elements
- Value must be enclosed in matching quotes

Attributes

- An element may have several attributes

```
<name type="personal">Smith</name>
<name type="place">London</name>
<name type="personal"
norm="Smith">Smyth</name>
```

Elements vs Attributes

- Elements
  - permit nested (and repeating) substructures
  - order always present even if not significant for application
  - Can contain:
    - Plain text (PCDATA), other elements, combination of both (mixed content)
- Attributes
  - can have only simple values
  - no ordering implied by order of attributes
  - cannot repeat for a given element
  - tend to be used to provide additional info about elements, e.g., units

Attributes: Examples

```
<ingredient qty="450" unit="grams">chicken</ingredient>
<eacheader type="p" status="draft">
<head>Relationships</head>
   <eacrel type="parent">
      <persname>
        <part type="surname">Blair</part>
        <part type="forname">Richard Walmesley</part>✓
      </persname>
   </eacrel>
   <eacrel type="parent">
     <persname>
       <part type="surname">Blair</part>
       <part type="forname">Ida</part>
       <part type="forname">Mabel</part>X
      </persname>
    </eacrel>
```

Elements can repeat. Attributes cannot!

Empty Elements

- Elements that have no content
- Start and end tags merge with / before the closing >

```
<page.break n="14" />
<citation reference="oldtext" />
```

- Links can also be empty elements

Comments

- Additional information in the document to aid the human reader etc.
- Can be placed anywhere within the document, but starting on a new line (ie not within an element)

<!-- examples adapted by Simon Mahony from an original by Susan Hockey -->

Entities

- Entities are a way of referring to any character or piece of text
- Common uses of entities
  - Non-standard characters
  - Boiler-plate text (ie text to be unpacked eg: UCL)
- Entities begin with & and end with ; (cf XHTML)

Defining Entities

- Entities (and their expansions) are defined in the DTD (Document type Definition – next session)

<!ENTITY UCL "University College London">

&UCL; is in Central London.

Entities for Non-Standard Characters

- These must be declared with appropriate Unicode characters (examples to follow)
- See list for important ones and the code tables accessible via http://www.unicode.org/charts/ for others

Entities for Non-Standard Characters

Some common examples (predefined list to follow)

```
<!ENTITY agrave  "&#x00E0;" >
<!ENTITY eacute  "&#x00E9;" >
<!ENTITY egrave  "&#x00E8;" >
<!ENTITY pound "&#x00A3;" >
<!ENTITY euro "&#x20AC;" >
```

Unicode Entities in Oxygen

(Oxygen is an XML editor)

To add non-standard entities in Oxygen

Go to:

- Perspective
- Show Toolbar
- Unicode
- Look for icon on tool bar
- Opens character map (cf MS Word)
- Select Character entity (and it inserts the code)

Using Entities

```
<p>She is a student at &UCL;.</p>
<p>Moli&egrave;re was a French dramatist.</p>
```

Using Entities

&TM; is by the river.

needs an entity definition such as

<!ENTITY TM "Tate Modern">

Pre-Defined Entities

- The following are pre-defined – you do not need to define them

<	<
>	>
'	'
"	"
&	&

- You only need to use the entity where the markup would otherwise be ambiguous

Example with Pre-Defined Entities

```
<p>He is &lt; 20.</p>
```

He is < 20.

```
<p>Library &amp;amp; Archive Studies</p>
```

Library & Archive Studies

Entity Examples

Example 2.1

Character entity for é

What you type What you get

Example 2.2

Boiler plate or short-cut text

What you type What you get

Example 2.21

Mixing it

What you type What you get

Well-formed Documents

- Use XML syntax
- Cannot validate the structure
- Internet Explorer 5, and above and Mozilla Firefox can display them in nested format

(this will confirm for you that your document is 'well-formed')

Well-formed Document

- Has an outer (root) element
- Matching start and end tags
- All attribute values in quotes
- A nested structure

Document Structure

- An XML document is a nested structure (tree) of elements
- Elements can contain other elements
- Elements can contain only text (the leaves of the tree)
- Elements can have mixed content – text and/or other elements

Element Containing Another Element

<recipe>
<name>Fast Roast Chicken</name>
</recipe>

<recipe> contains <name>
<name> is directly inside <recipe>

Element Containing Only Text

<name>Fast Roast Chicken</name>

<name> contains only text and no other elements

Element with Mixed Content

 The novel <title>Pride and Prejudice</title> written by <author>Jane Austen</author>is a good read.

 contains a mixture of text (*The novel*, *written by*, *is a good read*) and other elements (<title> and <author>) which also contain text

Elements

- Elements can be repeated

  <body> can contain one or more <para>s
- Elements can be optional

  <body> can contain an optional <heading> and one or more <para>s