Talk 1: Introduction to the Workshop, Markup, and XML

Dot Porter

July 2009
Introduction to the Workshop

Aims of Course

1. Examine the concept of markup and XML encoding
2. Provide hands-on experience in using TEI XML markup
3. Introduce the TEI scheme, its assumptions, and its organization
4. Survey the whole landscape of the TEI recommendations
5. Demonstrate how the TEI scheme may be customized to particular needs
6. Demonstrate some real world applications of the TEI scheme
7. Provide routes into more detailed information for exploration at your leisure
8. Provide opportunities for questions and discussions relating to your own encoding needs and priorities
Day 1: Monday 13 July, 2009 -- Introductions

Lunch

Talk 1: Introduction to Course, Markup, XML, and the oXygen XML Editor

Talk 2: Introduction to oXygen

Practical: Exercise 1: Editing XML in oXygen

Tea/Coffee

Talk 3: The TEI and TEI Document Structures

Talk 4: Core TEI Elements

Practical: Exercise 2: Making Your Document Valid TEI
Day 2: Tuesday 14 July, 2009 -- Metadata and Manuscripts

Talk 5: The `<teiHeader>`

Practical: Exercise 3: Improve Your Header

Talk 6: Manuscript Description

Tea/Coffee

Practical: Exercise 4: Describe a Manuscript

Talk 7: Facsimile and Image Markup

Lunch

Extra Time Something Different!

Talk 8: Transcription and Primary Sources
Practical: Exercise 5: Add some <choice>
Talk 9: Critical Apparatus
Tea/Coffee Tea/Coffee
Talk 10: Names, Dates, People and Places
Lunch Lunch
Practical: Exercise 6: Marking Up People and Places
Talk 11: Pointing, Linking, and Stand-Off Markup
Day 4: Thursday 16 July, 2009 -- More TEI Modules

Talk 12: Analysis, Speech, and Linguistics
Exercise 7: Even More TEI Editing
Talk 13: TEI, Unicode, and Non-Standard Characters
Tea/Coffee Tea/Coffee
Practical: Exercise: Even More TEI Editing
Talk 14: Verse, Drama, and Dictionaries
Lunch Lunch
Day 5: Friday 17 July, 2009 -- Customisation

Talk 15: Documenting TEI Customisations
Practical: Exercise 8: Customising the TEI
Tea/Coffee
Talk 16: Exploring the TEI Community
Talk 17: Conclusions and Group Discussion
Lunch
Course Materials

- **All** course materials including:
  - **All** slides from lectures (in TEI XML, HTML, and PDF)
  - **All** exercises
  - **All** materials for the exercises
  - A PDF booklet combining all these with 'TEI Lite'

are available on the TEI @ Oxford website.

- The url is: http://tei.oucs.ox.ac.uk/Oxford/2009-07-dublin/
- All these materials are licensed with a Creative Commons Attribution license, which means they are freely available for re-use (though do let us know!)
- To save you downloading a huge zip with all the workshop materials we've copied them to a folder on your Summer School USB.
After the workshop...

• After the workshop, if you have questions about:
  - The workshop materials or teaching other workshops: James.Cummings@oucs.ox.ac.uk or d.porter@dho.ie
  - The TEI generally: TEI-L@listserv.brown.edu

If you mail the TEI-L mailing list it is better because:
• we'll still try to answer as well as we would privately
• you get answers not only from us, but TEI experts around the world
• questions from those of all levels of ability stop the list becoming too technical
• everyone benefits from having the answers be public -- and you benefit by reading (and sometimes answering!) others' problems
In order to talk about texts, markup and encoding of texts, we need to understand what we mean by these basic concepts. When we talk about text encoding, what do we mean by a text? What is in a text and what assumptions do we make in reading them?
What's in a text?

The Scene: A ship at sea; afterwards an uninhabited island.

Act One

Scene I. On a ship at sea; a tempestuous noise of thunder and lightning heard.

Enter a Shipmaster and a Boatswain.

Master. Boatswain!

Boats. Here, master; what cheer?

Master. Good! Speak to th’ mariners; fall to ’t yarely, or we run ourselves aground; bestir, bestir.

[Exit.

Enter Mariners.

Boats. Heigh, my hearts! cheerly, cheerly, my hearts! yare, yare! Take in the topsail. Tend to th’ master’s whistle. Blow till thou burst thy wind, if room enough.

Enter Alonso, Sebastian, Antonio, Ferdinand, Gonzalo, and Others.

Alon. Good boatswain, have care. Where’s the master? Play the men.

Boats. I pray now, keep below.

Ant. Where is the master, bosun?

Boats. Do you not hear him? You mar our labour; keep your cabins; you do

Cheerly, good hearts!—Out of our way, I say. [Exit.

Gon. I have great comfort from this fellow. Methinks he hath no drowning mark upon him; his complexion is perfect gallows. Stand fast, good Fate, to his hanging; make the rope of his destiny our cable, for our own doth little advantage. If he be not born to be hang’d, our case is miserable. [Exeunt.

Re-enter Boatswain.

Boats. Down with the topmast. Yare, lower, lower! Bring her to try wi’ th’ main-course. [A cry within] A plague upon this howling! They are louder than the weather or our office.

Re-enter Sebastian, Antonio, and Gonzalo.

Yet again! What do you here? Shall we give o’er, and drown? Have you a mind to sink?

Seb. A pox o’ your throat, you bawling, blasphemous, incharitable dog!

Boats. Work you, then.

Ant. Hang, cur; hang, you whoreson, infamous, accursed wretch! 

[Exit. [Exeunt.
Introduction to the Workshop

An Introduction to Textual Markup

An Introduction to XML

What's in a text (2)?
What's in a text (3)?

Hwæt wē Gār-Dēnā in gēār-dagum
bēod-cyninga þrym gefrūnōn,
hū ē āþelingas ellen fremenōn.

Oft Scyld Scēfing sceāpēna þrēatūm,
5 mōne gum mǣgbūm meodo-setla oftēah;
egaðode Eorl[e], syðōn ērest wearē
fēasceat funden; hē þæs frōfre gebād:
wēox under wulcnum, weorō-myndum þāh,
oðbæt him ēghwylc þāra ymb-sittendra
10 ofer hron-rāde ēyran scolde,
The ontology of text

Where is the text?

• in the shape of letters and their layout?
• in the original from which this copy derives?
• in the stories we read into it? or in its author's intentions?

A "text" is an abstraction, created by or for a community of readers. Markup encodes and makes concrete such abstractions.
The ontology of text

Where is the text?

- in the shape of letters and their layout?
- in the original from which this copy derives?
- in the stories we read into it? or in its author's intentions?

A "text" is an abstraction, created by or for a community of readers. Markup encodes and makes concrete such abstractions.
The ontology of text

Where is the text?

- in the shape of letters and their layout?
- in the original from which this copy derives?
- in the stories we read into it? or in its author's intentions?

A "text" is an abstraction, created by or for a community of readers. Markup encodes and makes concrete such abstractions.
Encoding of texts

• Texts are more than sequences of encoded glyphs
  • They have structure and content
  • They also have multiple readings

• Encoding, or markup, is a way of making these things explicit

Only that which is explicit can be reliably processed
Styles of markup

- In the beginning there was *procedural* markup
  RED INK ON; print balance; RED INK OFF
- which being generalised became *descriptive* markup `<balance type='overdrawn'>some numbers</balance>`
- also known as *encoding* or *annotation*

descriptive markup allows for easier re-use of data
Some more definitions

• Markup makes explicit the distinctions we want to make when processing a string of bytes
• Markup is a way of naming and characterizing the parts of a text in a formalized way
• It's (usually) more useful to markup what we think things *are* than what they *look like*
What's the point of markup?

- To make explicit (to a machine) what is implicit (to a person)
- To add value by supplying multiple annotations
- To facilitate re-use of the same material
  - in different formats
  - in different contexts
  - by different users
What's the point of markup?

• To make explicit (to a machine) what is implicit (to a person)
• To add value by supplying multiple annotations
• To facilitate re-use of the same material
  • in different formats
  • in different contexts
  • by different users
What's the point of markup?

- To make explicit (to a machine) what is implicit (to a person)
- To add value by supplying multiple annotations
- To facilitate re-use of the same material
  - in different formats
  - in different contexts
  - by different users
Separation of form and content

- Presentational markup cares more about fonts and layout than meaning
- Descriptive markup says what things are, and leaves the rendition of them for a separate step
- Separating the form of something from its content makes its re-use more flexible
- It also allows easy changes of presentation across a large number of documents
Separation of form and content

• Presentational markup cares more about fonts and layout than meaning
• Descriptive markup says what things are, and leaves the rendition of them for a separate step
• Separating the form of something from its content makes its re-use more flexible
• It also allows easy changes of presentation across a large number of documents
Separation of form and content

- Presentational markup cares more about fonts and layout than meaning
- Descriptive markup says what things are, and leaves the rendition of them for a separate step
- Separating the form of something from its content makes its re-use more flexible
- It also allows easy changes of presentation across a large number of documents
Separation of form and content

• Presentational markup cares more about fonts and layout than meaning
• Descriptive markup says what things are, and leaves the rendition of them for a separate step
• Separating the form of something from its content makes its re-use more flexible
• It also allows easy changes of presentation across a large number of documents
Markup as a scholarly activity

- The application of markup to a document can be an intellectual activity
- In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor
- There is (almost) no such thing as neutral markup -- all of it involves interpretation
- Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself
- Good textual encoding is never as easy or quick as people would believe
- Detailed document analysis is needed before encoding for the resulting markup to be useful
Markup as a scholarly activity

- The application of markup to a document can be an intellectual activity
- In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor
- There is (almost) no such thing as neutral markup -- all of it involves interpretation
- Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself
- Good textual encoding is never as easy or quick as people would believe
- Detailed document analysis is needed before encoding for the resulting markup to be useful
Markup as a scholarly activity

• The application of markup to a document can be an intellectual activity

• In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor

• There is (almost) no such thing as neutral markup -- all of it involves interpretation

• Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself

• Good textual encoding is never as easy or quick as people would believe

• Detailed document analysis is needed before encoding for the resulting markup to be useful
Markup as a scholarly activity

• The application of markup to a document can be an intellectual activity

• In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor

• There is (almost) no such thing as neutral markup -- all of it involves interpretation

• Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself

• Good textual encoding is never as easy or quick as people would believe

• Detailed document analysis is needed before encoding for the resulting markup to be useful
Markup as a scholarly activity

• The application of markup to a document can be an intellectual activity
• In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor
• There is (almost) no such thing as neutral markup -- all of it involves interpretation
• Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself
• Good textual encoding is never as easy or quick as people would believe
• Detailed document analysis is needed before encoding for the resulting markup to be useful
Markup as a scholarly activity

• The application of markup to a document can be an intellectual activity
• In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor
• There is (almost) no such thing as neutral markup -- all of it involves interpretation
• Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself
• Good textual encoding is never as easy or quick as people would believe
• Detailed document analysis is needed before encoding for the resulting markup to be useful
What does markup capture?

Compare

<hi rend="dropcap">H</hi>&WYN; ÆT WE GARDE <lb/>na in gear-dagum beod-cyninga <lb/>þrym gefrunon, hu ða æþelingas <lb/>ellen fremedon. oft scyld scefing sceapē <add>na</add> <lb/>þreatum, moneg<expan>um</expan> mægbum meodo-setl <add>a</add> <lb/>of<damage> <desc>blot</desc> </damage>teah ... and

<lg> <l>Hwæt! we Gar-dena in gear-dagum</l> <l>beod-cyninga þrym gefrunon, </l> <l>hu ða æþelingas ellen fremedon, </l> </lg> <lg> <l>Oft Scyld Scefing sceapēna þreatum, </l> <l>monegum mægbum meodo-setla ofteah; </l> <l>egsode Eorlē, syððan ærest wearþ </l> <l>feasceafth funden...</l> </lg>
A useful mental exercise

Imagine you are going to markup several thousand pages of complex material....

• Which features are you going to markup?
• Why are you choosing to markup this feature?
• How reliably and consistently can you do this?

Now, imagine your budget has been halved. Repeat the exercise!
Some alphabet soup

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGML</td>
<td>Standard Generalized Markup Language</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition (or Declaration)</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet</td>
</tr>
<tr>
<td>Xpath</td>
<td>XML Path Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>eXtensible Stylesheet Language - Transformations</td>
</tr>
<tr>
<td>XQuery</td>
<td>XML Querying</td>
</tr>
<tr>
<td>RELAXNG</td>
<td>Regular Expression Language for XML (New Generation)</td>
</tr>
</tbody>
</table>

Oh, and then there's also **TEI**, the *Text Encoding Initiative*
Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML also now plays an indispensable role in the exchange of a wide variety of data on the Web and elsewhere.
XML: what it is and why you should care

• XML is **structured data** represented as strings of text
• XML looks like HTML, except that:-
  • XML is **extensible**
  • XML must be **well-formed**
  • XML can be **validated**
• XML is application-, platform-, and vendor- independent
• XML empowers the **content provider** and facilitates data integration
XML terminology

An XML document may contain:-

- elements, possibly bearing attributes
- processing instructions
- comments
- entity references
- marked sections (CDATA, IGNORE, INCLUDE)

An XML document must be well-formed and may be valid
XML terminology Example

```xml
<?xml version="1.0" ?>
<root>
  <element attribute="value"> content </element>
<!-- comment -->
</root>-->
The rules of the XML Game

- An XML document represents a (kind of) tree
  - It has a single root and many nodes
  - Each node can be
    - a subtree
    - a single element (possibly bearing some attributes)
    - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

• An XML document represents a (kind of) tree
• It has a single root and many nodes
• Each node can be
  • a subtree
  • a single element (possibly bearing some attributes)
  • a string of character data
• Each element has a name or generic identifier
The rules of the XML Game

• An XML document represents a (kind of) tree
• It has a single root and many nodes
• Each node can be
  • a subtree
  • a single element (possibly bearing some attributes)
  • a string of character data
• Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
Representing an XML tree

• An XML document is encoded as a linear string of characters
• It begins with a special **processing instruction**
• Element occurrences are marked by **start- and end-tags**
• The characters `<` and `&` are *Magic* and must always be "escaped" if you want to use them as themselves
• **Comments** are delimited by `<!- - and - -->`
• **CDATA sections** are delimited by `<![CDATA[ and ]]>`
• Attribute name/value pairs are supplied on the start-tag and may be given in any order
• Entity references are delimited by `&` and `;`
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
<xml version="1.0" endian="little">
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>

• The XML declaration
• Namespace declarations
• The root element of the document itself
• Other elements and content
• Attribute and value
Parts of an XML document

<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>

• The XML declaration
• Namespace declarations
• The root element of the document itself
• Other elements and content
• Attribute and value
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
Parts of an XML document

<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
An XML document must begin with an **XML declaration** which does three things:

- specifies that this *is* an XML document
- specifies which version of the XML standard it follows
- specifies which character encoding the document uses

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

The default, and recommended, encoding is ‘UTF-8’ (Unicode)
Namespace declarations

All TEI documents are declared within the TEI namespace: `<TEI xmlns="http://www.tei-c.org/ns/1.0"> ... </TEI>`

XML documents can include elements declared in different namespacaes.

- a namespace declaration associates a namespace prefix with an external URI-like identifier
- the default namespace may be declared using a xmlns
- other name spaces must all use a specially declared prefix

```xml
<TEI xmlns="http://www.tei-c.org/ns/1.0" xmlns:math="http://www.mathml.org">  
<p>... <math:expr>...</math:expr>...</p>... </TEI>
```

The xml namespace is used by the TEI for global attributes @xml:id and @xml:lang
The Doctype Declaration

You may sometimes find an optional "Document Type" declaration at the start of a document:

```xml
<?xml version="1.0" ?>
<!DOCTYPE greeting SYSTEM "greeting.dtd []">
```

- The DTD is one way of associating the document with its schema (but is not used by W3C or RELAX NG for this purpose)
- The DTD subset is used to provide declarations additional to those in the schema, for example for external files
- The DTD subset may be internal, external, or both

DTDs are now considered old-fashioned -- RELAX NG schemas are preferred.
<head>SCENE I. On a ship at sea: a tempestuous noise of thunder and lightning heard. </head>

<stage>Enter a Master and a Boatswain</stage>

<sp>
  <speaker>Master</speaker>
  <ab>Boatswain!</ab>
</sp>

<sp>
  <speaker>Boatswain</speaker>
  <ab>Here, master: what cheer?</ab>
</sp>

<sp>
  <speaker>Master</speaker>
  <ab>Good, speak to the mariners: fall to't, yarely,</ab>
  <ab>or we run ourselves aground: bestir, bestir.</ab>
</sp>

<stage>Exit</stage>
Example deconstructed: root node

<?xml version="1.0" encoding="utf-8" ?>
<div n="1">
<!-- .... -->
</div>
Example deconstructed: head

<head>SCENE I. On a ship at sea: a tempestuous noise of thunder and lightning heard. </head>
Example deconstructed: stage direction and speech

<stage>Enter a Master and a Boatswain</stage>
<sp>
  <speaker>Master</speaker>
  <ab>Boatswain!</ab>
</sp>
An XML Tree For The Tempest
XML syntax: the small print

What does it mean to be well-formed?

1. there is a single root node containing the whole of an XML document
2. each subtree is properly nested within the root node
3. names are always case sensitive
4. start-tags and end-tags are always mandatory (except that a combined start-and-end tag may be used for empty nodes)
5. attribute values are always quoted

Note: You can be valid in addition to being well-formed. This means you obey the rules of a specified schema, such as the TEI.
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text< /seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • <seg>some text</seg>
  • <seg><foo>some</foo> <bar>text</bar></seg>
  • <seg><foo>some <bar></foo> text</bar></seg>
  • <seg type="text">some text</seg>
  • <seg type='text'>some text</seg>
  • <seg type=text>some text</seg>
  • <seg type="text">some text<seg/>
  • <seg type="text">some text<gap/></seg>
  • <seg type="text">some text< /seg>
  • <seg type="text">some text</Seg>
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/></seg>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text< /seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg>`<foo>some</foo> <bar>text</bar>`</seg>`
  • `<seg>`<foo>some <bar></foo> text</bar>`</seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type="text">some text</Seg>`
Test your XML knowledge

• Which are correct?
  • `<seg>some text</seg>`
  • `<seg><foo>some</foo> <bar>text</bar></seg>`
  • `<seg><foo>some <bar></foo> text</bar></seg>`
  • `<seg type="text">some text</seg>`
  • `<seg type='text'>some text</seg>`
  • `<seg type=text>some text</seg>`
  • `<seg type="text">some text<seg/>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text< /seg>`
  • `<seg type="text">some text</Seg>`
XML is an international standard

- XML requires use of ISO 10646 (also known as Unicode)
  - a 31 bit character repertoire including most human writing systems
  - encoded as UTF8 or UTF16
- other encodings may be specified at the document level
- language may be specified at the element level using `@xml:lang`

The `@xml:id` attribute is another W3C-defined attribute.