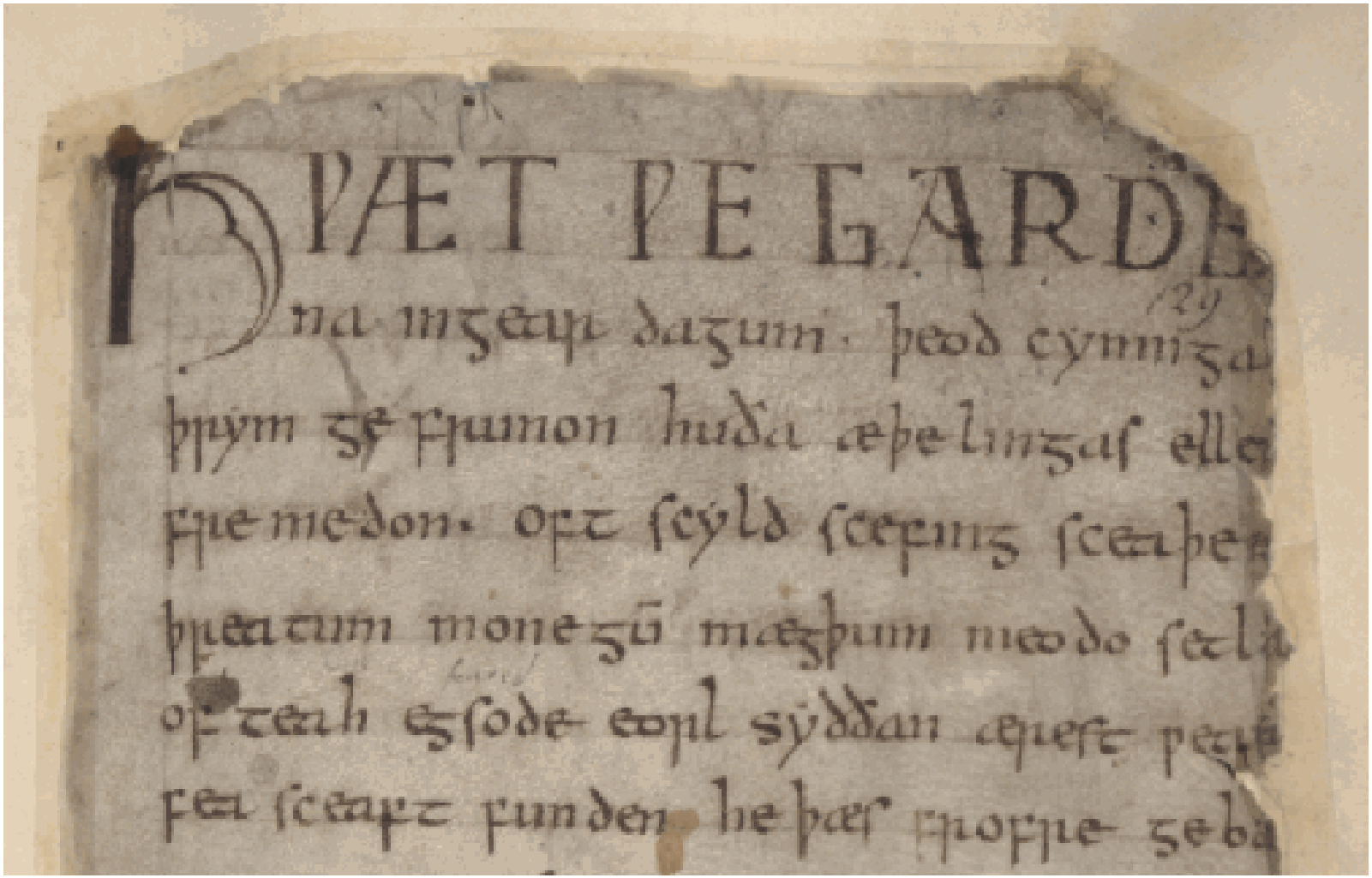
What is XML markup for?

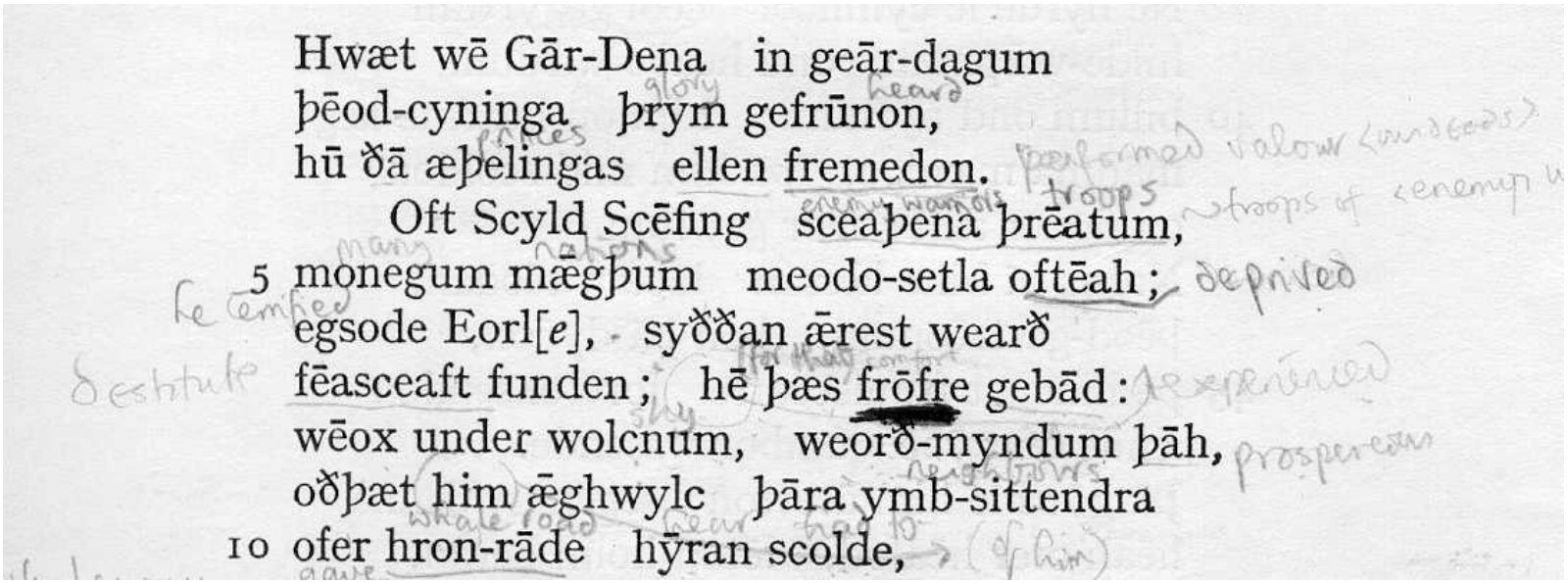
TEI @ Oxford

September 2008

What's in a text ?



Is this the same text?



Hwæt wē Gār-Dena in geār-dagum
þēod-cyninga þrym gefrūnon,
hū ðā æþelingas ellen fremedon.
Oft Scyld Scēfing sceaþena þrēatum,
monegum mægþum meodo-setla oftēah;
egsode Eorl[e], syððan ærest wearð
fēasceaft funden; hē þæs frōfre gebād:
wēox under wolcnum, weorð-myndum þāh,
oðþæt him æghwylc þāra ymb-sittendra
ofer hron-rāde hȳran 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

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

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

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

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

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

It's (usually) more useful to markup what we think things *are* than what they *look like*

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

```
<lb/>
<hi rend="dropcap">H</hi>
<g ref="#WYNN"/>ÆT WE GARDE
<lb/>na in gear-dagum þeod-cyninga
<lb/>þrym gefrunon, hu ða æþelingas
<lb/>ellen fremedon. oft scyld scefing sceaþe
<supplied>na</supplied>
<lb/>þreatum, moneg<ex>um</ex> mægþum meodo-setl
<supplied>a</supplied>
<lb/>of<damage>
  <desc>blotted</desc>
</damage>teah ...
```

and

```
<lg>
  <l>Hwæt! we Gar-dena in gear-dagum</l>
  <l>þeod-cyninga þrym gefrunon,</l>
  <l>hu ða æþelingas ellen fremedon,</l>
</lg>
<lg>
  <l>Oft Scyld Scefing sceaþena þreatum,</l>
  <l>monegum mægþum meodo-setla ofteah;</l>
  <l>egsode Eorle, syððan ærest wearþ</l>
  <l>feasceaft funden...</l>
</lg>
```

Some alphabet soup

SGML	Standard Generalized Markup Language
HTML	Hypertext Markup Language
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
DTD	Document Type Definition (or Declaration)
CSS	Cascading Style Sheet
Xpath	XML Path Language
XSLT	eXtensible Stylesheet Language - Transformations
XQuery	XML Querying
RELAXNG	Regular Expression Language for XML (New Generation)

Oh, and then there's also TEI, the *Text Encoding Initiative*

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: 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: 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: 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: 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: 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

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

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

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

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

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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
- Attribute names are predefined for a given element; values can also be constrained

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 ;

A complete XML document


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

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

A complete XML document


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

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

A complete XML document


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

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

A complete XML document


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

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

A complete XML document


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

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

The XML declaration


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

An XML document must begin with an XML declaration which does two things:

- specifies that this *is* an XML document, and which version of the XML standard it follows
- may specify a different character encoding for the document — if the default, and recommended, encoding UTF-8 is not being used

Namespace declarations

An XML document may include elements declared in different name spaces.

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

- 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
- All TEI documents are declared within the TEI namespace
- The xml namespace is available in all XML documents; TEI uses it for global attributes *@xml:id* and *@xml:lang*

The Doctype Declaration

You may sometimes find an optional "Document Type" declaration:

```
<?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 RELAXNG 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 — RELAXNG or W3C schemas are preferred.

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<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<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<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<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<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<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<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<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<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<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<gap/></seg>`
  - `<seg type="text">some text< /seg>`
  - `<seg type="text">some text</Seg>`