

Exercise 1: Creating an XML File

TEI @ Oxford

14 February 2013 (Valentine's Day)

1 Exercise 1: Creating an XML File

1.1 Learning Outcomes

When you successfully complete this exercise you should be able to:

- mark up an XML declaration
- insert a text file into an XML editor
- mark up basic features of a poem
- create a well-formed XML document

1.2 Summary

This exercise will walk you through creating an XML document in the oXygen editor and introduce a variety of ways to mark this document up. You will first start a new document, then insert some unmarked up text into the editor, and then mark up the stanzas or line-groups (lg) and lines (l). You will learn to check that your document is well-formed or not.

1.3 Starting A New XML File

Let's start a new XML file by following the following steps:

- Load up the oXygen XML Editor if it isn't already loaded by using the Windows Start Menu, or double-clicking the icon on the desktop.
- Once the editor has fully loaded from the 'File' menu select 'New' and under 'New Document' select 'XML Document'. This should open up a blank document with an XML Declaration added.
- An XML Declaration looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
```

and The XML declaration in the element tells anything processing your XML file, including the editor, that this is an XML file and what version of XML you are using through the @version attribute. It also conveys which characters the program may expect in attribute @encoding. XML version 1.0 is a W3C recommendation from 2008. UTF-8 (Universal Character Set Transformation Format - 8 bit) contains most characters from all human writing systems. The XML declaration needs no closing tag as it takes the form of a special processing-instruction that starts and ends with an angle-bracket and a question mark.

1.4 Creating a Division

Let's create a division of a text using the <div> element. This is a generic division or section element.

- On the line below the XML declaration type: <div>.

1 EXERCISE 1: CREATING AN XML FILE

- Notice what happens when you type the final '>'. oXygen is trying to help you and inserts in the closing `</div>` tag. This is because it knows the rules of XML, and knows that if you type an opening `<div>` you are required to have a closing `</div>` sooner or later.
- We haven't said what type of division this is, so let's categorise it as 'verse' by adding a `@type` attribute. Move the cursor back until you're just after the letter 'v' in the opening tag. Press space, and then type: `type="` and notice what happens when you type the quotation mark. oXygen is again trying to help you by putting the closing quotation mark, because it knows that attribute values must always be quoted.
- In between the quotation marks type `'verse'` to categorise our division as being verse.
- Move back until you are directly in between the opening `<div>` and closing `</div>`. Press 'enter' a couple of times to give yourself some space inside the element.

1.5 Inserting Some Text

We are going to use the Wilfred Owen poem *Strange Meeting* as an example for this exercise. But it would waste a lot of time if we asked you to type the whole poem in, so we've done that for you.

- Make sure your cursor is in-between the opening `<div>` and the closing `</div>` and go to the *Document* menu and select 'File' and from there then 'Insert File'. *Note: This is from the 'Document' menu on the menu bar, not the 'File' one.*
- Select 'strange-meeting.txt' as the file to insert.
- The start of your document should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<div type="verse">
STRANGE MEETING

It seemed that out of battle I escaped
Down some profound dull tunnel, long since scooped
Through granites which titanic wars had groined.

[...a lot more text...]
</div>
```

1.6 Encoding the Heading (using 'Surround with Tags')

The text 'STRANGE MEETING' at the top of the poem is obviously a heading. The TEI `<head>` element should be used to mark this. To mark this do the following:

- Highlight the text 'STRANGE MEETING' with the mouse.
- Either press control-e as a shortcut key, or right-click and under 'Refactoring' select 'Surround with Tags'. A box should pop up and type `head` into it. Notice how oXygen helps you again by putting the opening tag before what you had highlighted and the close tag afterwards.

1.7 Marking Stanzas (using both 'Surround with Tags' and 'Split Element')

Let's mark the stanzas that appear doing the following steps:

- Highlight the first stanza, from "It seemed" to "had groined".

- Using control-e, or the menus, as you did above, mark this stanza as an `<lg>` element.
- Add a `@type` attribute with a value of 'stanza' to the `<lg>` element so it looks like: `<lg type="stanza">`.
- The start of your document should now look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<div type="verse">
  <head>STRANGE MEETING</head>
  <lg type="stanza"> It seemed that out of battle I escaped
    Down some profound dull tunnel, long since scooped
    Through granites which titanic wars had groined.
  </lg>

  [...a lot more text...]

</div>
```

- But if we have lots of stanzas, marking each one of them seems a lot of work, but there is a (possibly) easier way.
- Highlight the entire rest of the poem, from "Yet also there" to "Let us sleep now...'", and then surround all of it in an `<lg>` element (by pressing control-e)
- Of course it is silly to have the entire rest of the poem marked as a single line-group, but go and add a `type="stanza"` attribute to the opening tag.
- If you move the cursor to just before the start of each stanza, e.g. just before where it says "With a thousand pains", and press alt-shift-d (or select Refactoring -> Split Element from the right-click menu), oXygen should split the `<lg>` element, ending it here and starting it just before where there cursor is located.
- Do this for other stanzas that are not marked yet.

1.8 Marking Lines

We've marked all the stanzas but we've not marked the lines.

- Highlight the first line in the first stanza, press control-e to surround with a tag, and type 'l' as the element name. (`<l>` is the line element, meaning a line of metrical verse).
- It might be a bit painful to mark up each and every line this way, you could try using the split-element technique above, but there is another shortcut to try as well. Highlight the second line and press control-/ and notice that oXygen has wrapped the line in a `<l>` element. The reason for this is that control-/ is the 'surround with the last element I surrounded something with' shortcut key.
- Using this technique, quickly mark all the remaining lines.

1.9 Format and Indent

Our poem is marked up, but some of the markup might be a bit messy.

- Make sure that your file is 'well-formed'. You'll be able to tell it is well-formed because oXygen will have a happy green square in the upper right-hand corner. If it is red, you better find the problem (where a red bar on the right-hand side is) and correct the mistake!

1 EXERCISE 1: CREATING AN XML FILE

- Now let's format and indent our file. This tidies up some of the whitespace and indents elements based on their place in the hierarchy. Either select the 'Format and Indent' icon from the toolbar (it looks like some indented lines), or go to the menus: 'Document' -> 'Source' -> 'Format and Indent'.
- Formatting and indenting your markup is not necessary, it could all be on one big long line, but it makes it much easier for other people to read.

1.10 Saving Your Work

Let's save our work:

- Is your work well-formed? Do you have a happy green square or an angry red one?
- From the 'File' menu select 'Save' or click on the Save icon (looks like an old-style 3.5" disk)
- Save the file using the name 'exercise01.xml' or another name of your choice.

1.11 Self-Assessment

Check if you understand some of the core principles of this exercise by answering the following questions to yourself:

- How do you start a new XML document in oXygen?
- What is an XML declaration?
- What is a well-formed document?
- How do I 'Surround with tag' and repeat that action quickly?
- Why might using the 'Split element' approach be useful?
- What is the function of each element and attribute in your current file?
- What is the advantage of formatting and indenting your markup?

1.12 Next?

Your XML file may be well-formed but it is not yet **valid** because it doesn't validate against a particular schema (such as those which are customisations of the TEI). Next we will have a short introduction to the structure of TEI documents. If you are finished early you may wish to browse through the TEI Guidelines online at <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index-toc.html>. In particular you might want to look at the Elements appendix of reference pages for individual elements. Consider looking up all the elements you've used in this file to see how they are defined.