emmamorlock / teiHeader-demm

<> Code   ⊙ Issues 0   ⑂ Pull requests 0   ▥ Projects 0   ▦ Wiki   ⚡ Pulse   ▬ Graphs   ⚙ Settings

Branch: master ▾   teiHeader-demm / teiHeader-exo.md   Find file   Copy path

emmamorlock handout   28f5b4e 10 minutes ago

1 contributor

238 lines (199 sloc)   8.51 KB   Raw   Blame   History   🖵 ✎ 🗑

Exercice 1: A minimal teiHeader

Step One: A minimal complete TEI File

Once your instance of oXygen editor is open and ready:

- Go to file > "New Document" and select "Document XML"
- this should open a blank file with just an xml declaration looking like:

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

TEI files start either with a TEI element or a teiCorpus element. You must add the TEI namespace attribute to that root element to identify the vocabulary of xml elements your using. If you were to use some non TEI elements, you would indicate the vocabulary the belong to as well, so that every thing is clear. The attribute and its value are:

```
xmlns="http://www.tei-c.org/ns/1.0"
```

If you add a few blank lines between the start and the end tag, here's what your file should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">

</TEI>
```

Being pre-package with TEI knowledge and dedicated tool, oXygen can tell you if your file is valid or not. You'll know that every thing is OK looking at the color of a square at the right of the editing window. When it's red, it means you've got to fix something, like some element is missing in the hierarchy or some others can't be placed there. There's a message at the bottom of the editing pane that tells you what the problem is.

Now to get a green square, you will add the elements necessary to get a minimal tei valid document.

- clic inside the TEI root element and just put type an angle braquet '<'
- oXygen opens a box showing possible elements in that context: choose 'teiHeader'
- observe that the editor inserted the minimal elements of a teiHeader

The result should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title></title>
            </titleStmt>
            <publicationStmt></publicationStmt>
```

```
                <sourceDesc></sourceDesc>
            </fileDesc>
        </teiHeader>
    </TEI>
```

You don't have your green square yet, as other mandatory components are missing.

Start by adding a title and some information in the 'publicationStmt' and 'sourceDesc' element using the same method you used for inserting the teiHeader:

- insert an angle bracket
- choose a sub element looking at contextual information oXygen provides

Then add a text element and continue to add sub elements until the minimal TEI document is valid.

The result, depending on the sub elements you chose, should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>My first TEI file</title>
            </titleStmt>
            <publicationStmt>
                <p>
                    <!-- Publication information: -->
                    Unpublished demonstration file.
                </p>
            </publicationStmt>
            <sourceDesc>
                <p>
                <!--Information about the source-->
                    No source: this is an original work.
                </p>
            </sourceDesc>
        </fileDesc>
    </teiHeader>
```

Don't forget to save your work!

Step Two: Updating the teiHeader

There are many ways in which you can enhance the header and add useful information in it. Of course, your choices in that area will depend on the information already available to you, but it shall also be determined by the operations you will want to process on your TEI text (html display for a reading version vs structured search within the fields of a multifaceted search engine, The choices you have to make come under two main categories:

- the level of precision you need (cursory vs detailed)
- the level of formalization you need (prose or structured/standardized)

=> use the contextual help oXygen offers at every step. You can also use the TEI Guidelines and the examples it provides.

=> after every

Title Statement (`<titleStmt>`)

- Add a proper title: `<title>`
- Who was responsible for the digitization `<respStmt>`, `<funder>`, `<sponsor>` ...

Publication Statement (`<publicationStmt>`)

- Replace the `<p>` element with more structured information.
  - `<publisher>`
  - `<date>`

- ◦ `<availability>`
- ◦ `<licence>`

Source Description (`<sourceDesc>`)

- You can replace `<p>` by a `<bibl>` containing the bibliographic reference of your source or a `<biblStruct>` (the same but with an constrained order for the contained elements) or a `<msDesc>` if your source is a manuscript (or a text bearing object).

```
<sourceDesc>
  <p>No source: this is an original work</p>
</sourceDesc>
```

```
<sourceDesc>
<p>Transcribed directly from the first edition of <title>Harry Potter</title> </p>
</sourceDesc>
```

```
<sourceDesc>
 <scriptStmt xml:id="CNN12">
  <bibl>
   <author>CNN Network News</author>
   <title>News headlines</title>
   <date when="1991-06-12">12 Jun 91</date>
  </bibl>
 </scriptStmt>
</sourceDesc>
```

```
<sourceDesc>
   <biblStruct>
    <monogr>
     <editor>Foner, Philip S.</editor>
     <title>The collected writings of Thomas Paine</title>
     <imprint>
      <pubPlace>New York</pubPlace>
      <publisher>Citadel Press</publisher>
      <date>1945</date>
     </imprint>
    </monogr>
   </biblStruct>
  </sourceDesc>
```

Description of the Encoding (`<encodingDesc>`)

- after the fileDesc element, insert an `<encodingDesc>` element, then:
- a `<projectDesc>` and `<p>` to describe your project goal
- an `<editorialDecl>` with a `<normalization>` element indicating the extent of standardization or regularization of the source carried out in the transcription process.

```
<editorialDecl>
 <normalization>
  <p>All words converted to Modern American spelling using
     Websters 9th Collegiate dictionary
  </p>
 </normalization>
 <quotation marks="all" form="std">
  <p>All opening quotation marks converted to " all closing
     quotation marks converted to &amp;cdq;.</p>
 </quotation>
</editorialDecl>
```

Other descriptive information (`<profileDesc>`)

The profileDesc element is the place for several kinds of metadata (but non-bibliographic) that describe some useful characteristics of the text, like languages used, the production context, the participants and their settings.

- after the encodingDesc element, insert an `<profileDesc>` element, then:
- a `<creation>` element that details of a text's creation, e.g. the date and place it was composed.
- a `<langUsage>` element for every language used in the text
- a `<settingDesc>` element for the setting(s) within which a language interaction takes place.
- a subject indexation within `<textClass>`

non TEI metadata (`<xenoData>`)

Xenodata is an element recently introduced in the TEI schema to record non-TEI Metadata. It offers a convenient way of keeping in the TEI reference file other metadata formats that have to be dealt with about the text, like a bibliographic MARC record, a dublin-core record or a RDF one. It may contain anything except TEI elements.

Example combining RDF statements and Dublin Core properties:

```
<xenoData>
 <rdf:RDF>
  <cc:Work rdf:about="">
   <dc:title>Applied Software Project Management - review</dc:title>
   <dc:type rdf:resource="http://purl.org/dc/dcmitype/Text"/>
   <dc:license rdf:resource="http://creativecommons.org/licenses/by-sa/2.0/uk/"/>
  </cc:Work>
  <cc:License rdf:about="http://creativecommons.org/licenses/by-sa/2.0/uk/">
   <cc:permits rdf:resource="http://web.resource.org/cc/Reproduction"/>
   <cc:permits rdf:resource="http://web.resource.org/cc/Distribution"/>
   <cc:requires rdf:resource="http://web.resource.org/cc/Notice"/>
   <cc:requires rdf:resource="http://web.resource.org/cc/Attribution"/>
   <cc:permits rdf:resource="http://web.resource.org/cc/DerivativeWorks"/>
   <cc:requires rdf:resource="http://web.resource.org/cc/ShareAlike"/>
  </cc:License>
 </rdf:RDF>
</xenoData>
```

Step Three: Adding some information about the manuscript

In the `<sourceDesc>`, a manuscript description (or of any other text bearing object) can be provided with a `<msDesc>` element.

Inside `<msDesc>` only `<msIdentifier>` is mandatory. The description can be provided through prose in `<p>`, or using some structured elements:

- msContents
- physDesc
- history
- additional
- msPart
- msFrag

The manuscript

- Click on this link to an e-codice manuscript
- go to the page standard description
- Look at the 3 descriptions provided
- use them to create a new description of your own using the template 'template-ms-description.xml'
- you can also compare with the facsimile