

Text Encoding Assignment

Spring 2022

Overview

Electronic editing and text encoding are important scholarly practices as well as marketable skills. Digital editing and preservation requires much skill and knowledge of the subject areas and material being encoded, just as it is for the Special Collections librarians who preserve our printed texts and the cultural memory they contain. We will now begin to explore the transformation of printed materials into viable digital ones (that are legible, searchable, and computable) through hands-on experience.

This two-part assignment requires you to complete (1) a basic structural encoding of the electronic transcript of one issue of a modernist periodical—*The New Freewoman* (September 15, 1913)—and (2) a metadata index (bibliography) of all its content items. Extensible Markup Language (XML) is the industry standard tagging system for describing textual data and metadata.

https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction
<https://en.wikipedia.org/wiki/XML>

The Text Encoding Initiative (TEI) is the community that governs standards and best XML encoding practices for scholarly electronic editing and digital humanities.

<https://tei-c.org/>
https://en.wikipedia.org/wiki/Text_Encoding_Initiative
https://en.wikipedia.org/wiki/Digital_humanities

We will be encoding one issue of *The New Freewoman* according to the basic TEI Guidelines, which you should consult as a reference when you have any questions about how to encode a text.

<https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html>

Procedure

Please complete the following two parts by editing the provided XML files and uploading them back to the Assignment link on Harvey. Refer to the PDF file of the magazine issue in order to edit both XML files.

Anatomy of an XML Tag

`<tag value="attribute">`

Some tags, like `<div>` (for division) identify regions of a text according to a value and attribute (or multiple of these). In these cases, a closing tag using the `/` indicates where a region ends: `</div>`. For instance, in an electronic transcript of a magazine containing a few pages of poems, the region would begin with `<div type="poetry">` placed immediately before the title of the first poem, while the closing `</div>` tag would go exactly after the last bit of text that could be associated with the poetic content. If you do not close the `<div>`, then the computer will think it continues perpetually and it will screw up other regions you are trying to encode.

Some tags might cover a region but can be indicated with a slash in the initial tag, indicating the next similar region starts with the next implementation of that tag `<tag value="attribute"/>`. The `<pb>` (page break) tag is one such tag you'll use to identify page numbers.

Tags must also be placed symmetrically, in order to keep textual features clear:

incorrect: `<i>Some text that is in bold and italics.</i>`

correct: `<i>Some text that is in bold and italics.</i>`

Part I. TEI Encoding *The New Freewoman* (September 15, 1913)

1. Refer to the magazine PDF file to perform basic structural encoding of *The New Freewoman* (September 15, 1913) in the file `new-freewoman-1913-09-15-tei.xml`, inserting TEI XML tags to encode page numbers and mark regions of the text by genre, as follows:

`<pb n="121"/>`

For returning page numbers in search engine results and for making them legible to readers of the XML document: **page break, number equals 121**). This tag indicates the page number in the original text, whatever it might be (page 121 in this example), and is *placed exactly before the first text appearing at the **top** of the page*. It might go right before a page number itself or before something like a title header at the top. E.g.:

`<div type="articles">Lots of article content here.</div>`

For providing advanced searches by genre, in the Advanced Search page at MJP (<https://modjourn.org/advanced-search/>). This tag marks a feature of the text across a particular region of it and is what allows the search engine to “see” the genres for

returning search results.

Genre Tags:

articles
poetry
fiction
images
essay
manifesto
letters
advertisements

Part II. Encoding Content in the Issue with Metadata Object Description Schema (MODS)

1. Perform basic metadata encoding, using the Metadata Object Description Schema (MODS) standard to create an index of all the content items from the magazine issue in the file [new-freewoman-1913-09-15-mods.xml](#). The first couple of items have been entered as examples that you can use as templates for the following ones, which you'll enter below the comment that reads <!-- INSERT YOUR ENTRIES BELOW -->. When completed, please upload the file, along with the TEI one, back to this Assignment on Harvey.

Metadata are data about data, or that describe other data. A bibliography is an example of descriptive metadata: at the back of your papers, you indicate your sources in an ordered way attributes such as author, editors, translators, title, publisher, publication place, publication date, and page numbers (among others). What you're doing here is likewise an ordered bibliographic description, just formatted in a different way than you're probably used to.

Go through the PDF file of the magazine issue to find the title, author, genre, and page number data for each piece, and enter it into the nest of <relatedItem type="constituent"> tags. Any item that does not have numbered or titled parts may omit the <partNumber> and <partName> tags seen in the below example. As you go through each item, think about which elements need to be encoded and how you'd adapt this format to do it.

```
<relatedItem type="constituent">
  <titleInfo>
    <title>On Interference with the Environment</title>
    <partNumber>I</partNumber>
    <partName>The Presuppositions I Start From</partName>
  </titleInfo>
  <name type="personal">
    <namePart>Byington, Steven T.</namePart>
    <role>
```

```
      <roleTerm authority="marcrelator" type="text">creator</roleTerm>
    </role>
  </name>
  <genre authority="aat">articles</genre>
  <part>
    <extent unit="pages">
      <start>121</start>
      <end>123</end>
    </extent>
  </part>
</relatedItem>
```