Part 1: Markup Languages

Markup languages are formats for electronic documents, in which certain words (or characters) in a text are marked as having a special function. The most well-known markup language is probably HTML (HyperText Markup Language), the standard format for any document on the Internet, including this one.

There have been many different markup languages over the yeard, including LaTex and SGML, but these days, almost all markup is done in XML. XML has a lot of rules, but the only ones we will be concerned with are the simple rules below.

1. To mark out a word, you put a tag at the beginning, and the same one at the end.
2. Tags in XML is a tag name between angular brackets, so `<name>` is a tag (an opening tag)
3. Closing tags are the same as opening tag, but with a slash at the beginning: `</name>`

So if we want to say that Maarten Janssen is a name, we do that by putting a `<name>` tag around it: `<name>Maarten Janssen</name>`. We call that whole element with the opening and closing tags, and the content a *node*.

What you can use as a tag name is also defined:

1. Tag names can contain letters, numbers, and underscores, so `<my_tag1>` for instance
2. Tag names are case sensitive, so (in principle) `<myTag>` is not the same as `<mytag>`

XML itself does not tell you which tags you can use, or what they mean. For that, you need an XML schema (traditionally a DTD). For instance, XHTML is an XML schema which tell you that `<b>` is a valid tag, and that it is used to indicate that the content is to be displayed in bold face. In this course we will be working with a schema called the Text Encoding Initiative (TEI).

We will use two more rules on XML tags:

1. Nodes without any content can be written as *self-closing* tags, in which case we write the slash at the end. So instead of `<name></name>` we can also write `<name/>`
2. Nodes can have *attributes*, with values. Attributes are only added to the opening tag, not to the closing tags. Attributes are added after the name of the tag, separated by a space, with the value in between quotes behind a = sign.

So if we want to specify that Maarten Janssen is not just any type of name, but the name of a person, we can add an attribute *type* to it, and set it to *person*. So the full node then looks like this: `<name type="person">Maarten Janssen</name>`

If we want to refer to an attribute, the convention is to put an ampersand before it. So `@person` is used to refer to an attribute name *person* on some node in the XML - the actual XML itself does not contain the @ character, it is just a naming convention to refer to attributes.

In this course, we will not be concerned with any other options in XML - with these simple rules you can understand all nodes we will encounter in this course.

The only rules that are still missing is how nodes can be combined. And there are three important rules in XML, which we call syntactic rules:

1. Everything has to be inside a node in XML - so the very first element of an XML document is always a tag - called the *root node*. And typically the root node indicates what type of document we are dealing with, in the case of TEI the root tag is `<TEI>`
2. Nodes can be *nested*, so we can use one tag inside another. So if we want to indicate that Maarten is the first name, you can use: `<name><first_name>Maarten</first_name> Janssen</name>`
3. Nodes can never *cross:* you have to close all tags inside another one. So this is not correct: `<a>This <b>cannot</a> happen</b>`

If these rules are broken, a document is called (syntactically) *invalid*, and programs will typically refuse to open it. HTML is one of the few markup languages where many documents are invalid but can still be opened.

The schema not only defines which tags can be used, but also where it can be used. So for instance, a schema would define that *first_name* is a valid tag, but that it can only be used with a *name* node. If the rules of the schema are broken, a document is called *unwellformed* or sometimes *semantically invalid*.

Part 2: Text Encoding Initiative

The Text Encoding Initiative (TEI) is a standard for the representation of texts in digital form. In this course, we will always use TEI as a *transcription format*, that is to say, a TEI document in our case is never something we write in TEI, but rather there is some linguistic entity in the world - a book, a manuscript, a spoken document, etc. which is transcribe in our TEI document.

TEI can describe a lot of things, but we will only deal with two parts of it: the `<teiHeader>` and the `<text>`. And both are directly under the root. So in our case, a TEI document always looks like this:

```
<TEI>
<teiHeader>
...
</teiHeader>
<text>
...
</text>
</TEI>
```

The teiHeader describes the entity itself, and anything we want to say about the transcription as a whole: the title, who transcribed it, the recording device used to make the recording, the physical state of the manuscript, the address of the institute responsible for the corpus that the document is part of, and almost anything else you could think of.

The text contains the actual transcription itself. In this first part, we will only be concerned with the <text> node. In principle, a TEI document can have more than one `<text>` element, but not in the way we are going to use it.

Text

The text element of TEI is there to contain the transcription of the original document, whether it be a transcribed manuscript, monograph, spoken dialogue, or anything else. Depending on the kind of source material, there are a lot of different elements that can be marked up.

Text Structure

`<p>`	Paragraph	A paragraph
`<s>`	Sentence	A sentence
`<head>`	Heading	Any type of heading - section titles, headings of a list, etc.

Manuscripts

`<add>`	Added	A bit of text that was inserted later by the scribe
`<del>`	Deleted	A bit of text that was deleted by the scribe

`<gap/>`	Gap	A bit of text missing in the transcription - either because it was not transcribed, or because it was missing in the original, or because it was illegible.
`<unclear>`	Unclear	A bit of text that seems readable, but does does make sense, so is probably something else.
`<supplied>`	Supplied	A bit of text that is provided from another source in the place of a missing bit of the manuscript.

Spoken Data

`<u>`	Utterance	An utterance (roughly a spoken sentence)
`<pause/>`	Pause	A pause in the speech
`<del>`	"Deleted"	In spoken TEI, del is used to mark any text that was "retracted", which is subdivided by `@type` into repititions, reformulations, and truncations.

More Fields

`<ab>`	anonymous block	contains any arbitrary component-level unit of text, acting as an anonymous container for phrase or inter level elements analogous to, but without the semantic baggage of, a paragraph.
`<add>`	added later	contains letters, words, or phrases inserted in the source text by an author, scribe, or a previous annotator or corrector.
`<cb>`	column beginning	marks the beginning of a new column of a text on a multi-column page.
`<date>`	date	contains a date in any format.
`<del>`	deleted by author	contains a letter, word, or passage deleted, marked as deleted, or otherwise indicated as superfluous or spurious in the copy text by an author, scribe, or a previous annotator or corrector.
`<desc>`	description of parent elm	contains a short description of the purpose, function, or use of its parent element, or when the parent is a documentation element, describes or defines the object being documented.

	text division	contains a subdivision of the front, body, or back of a text.
<foreign>	in another language	identifies a word or phrase as belonging to some language other than that of the surrounding text.
<fw>	running head	contains a running head (e.g. a header, footer), catchword, or similar material appearing on the current page.
<gap>	untranscribed text	indicates a point where material has been omitted in a transcription, whether for editorial reasons described in the TEI header, as part of sampling practice, or because the material is illegible, invisible, or inaudible.
<hi>	highlighted text	marks a word or phrase as graphically distinct from the surrounding text, for reasons concerning which no claim is made.
<head>	heading	contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.
<kinesic>	non-vocalic	marks any communicative phenomenon, not necessarily vocalized, for example a gesture, frown, etc.
<l>	verse line	contains a single, possibly incomplete, line of verse.
<lb>	line beginning	marks the beginning of a new (typographic) line in some edition or version of a text.
<lg>	line group (in a verse)	contains one or more verse lines functioning as a formal unit, e.g. a stanza, refrain, verse paragraph, etc.
<m>	morpheme	represents a grammatical morpheme.
<milestone>	marker in text	marks a boundary point separating any kind of section of a text, typically but not necessarily indicating a point at which some part of a standard reference system changes, where the change is not represented by a structural element.
<name>	proper name	contains a proper noun or noun phrase.
<note>	explanatory note	contains a note or annotation.
<orgName>	organization name	contains an organizational name.

<p>	paragraph	marks paragraphs in prose.
<pause>	pause	marks a pause either between or within utterances.
<persName>	person name	contains a proper noun or proper-noun phrase referring to a person, possibly including one or more of the person's forenames, surnames, honorifics, added names, etc.
<placeName>	place name	contains an absolute or relative place name.
<pb>	page beginning	marks the beginning of a new page in a paginated document.
<q>	quoted	contains material which is distinguished from the surrounding text using quotation marks or a similar method, for any one of a variety of reasons including, but not limited to: direct speech or thought, technical terms or jargon, authorial distance, quotations from elsewhere, and passages that are mentioned but not used.
<quote>	quotation	contains a phrase or passage attributed by the narrator or author to some agency external to the text.
<ref>	reference	defines a reference to another location, possibly modified by additional text or comment.
<s>	sentence	contains a sentence-like division of a text.
<sic>	apparently incorrect	contains text reproduced although apparently incorrect or inaccurate.
<seg>	arbitrary segment	represents any segmentation of text below the ‘chunk’ level.
<space>	significant space	indicates the location of a significant space in the text.
	annotation span	associates an interpretative annotation directly with a span of text.
<stage>	stage direction	contains any kind of stage direction within a dramatic text or fragment.
<supplied>	estimated content	signifies text supplied by the transcriber or editor for any reason; for example because the original cannot be read due to physical damage, or because of an obvious omission by the author or scribe.

<surplus>	redundant text	marks text present in the source which the editor believes to be superfluous or redundant.
<term>	technical term	contains a single-word, multi-word, or symbolic designation which is regarded as a technical term.
<time>	time	contains a phrase defining a time of day in any format.
<tok>	token	contains a word or punctuation mark [TEITOK specific]
<u>	utterance	contains a stretch of speech usually preceded and followed by silence or by a change of speaker.
<unclear>	seemingly readable text	contains a word, phrase, or passage which cannot be transcribed with certainty because it is illegible or inaudible in the source.
<unit>	unit	contains a symbol, a word or a phrase referring to a unit of measurement in any kind of formal or informal system.
<vocal>	non-word	marks any vocalized but not necessarily lexical phenomenon, for example voiced pauses, non-lexical backchannels, etc.

Part 3 - TEITOK

TEITOK is an online environment for corpora consisting of tokenized TEI/XML document. It provides three main functions:

- view individual TEI/XML documents
- edit TEI/XML documents (for corpus administratos)
- search through all documents

To create a new file in TEITOK:

- we recommend that you first create a folder with your name. You do this by going to "XML Files" in the menu, and there select "Create new folder".
- Once the folder is created, click on it to enter
- Inside the folder, select "Create new XML file" from the bottom
- Provide your file with a filename (otherwise you will get an error)
- For testing purposes, the best is to leave the metadata empty, and create the content "from plain text" (just type in some text)
- Select "create XML"
- You should now see your new XML file (if not, select it from the XML files)
- To play around with the raw XML, select the "to edit, click here" above the text

Tokenization

As the name says, TEITOK works with tokenized TEI/XML files, that is to say, TEI documents in which the text is split into words. In TEI P5, this is done by marking words as , and punctuation marks as . But TEITOK rather uses for tokens, which are roughly either or . The tokenization in TEITOK is done automatically by clicking a button. The tokenization process will insert token nodes around all words. So an input like

```
<p>This is a smal paragraph.</p>
```

Will be converted into

```
<p><tok>This</tok> <tok>is</tok> <tok>a</tok> <tok>smal</tok>
<tok>paragraph</tok><tok>.</tok></p>
```

Editing annotations

Once the text is tokenized, annotations can be added to each token - this can be linguistic annotation like part-of-speech tags, lemmas, and dependency relations as we will see in the next lesson. But it can also be a normalized form of the word - in the case of a typographic error or deviant spelling. So the word *smal* in the paragraph above is misspelled. This makes it harder to run NLP processes, and it makes it harder to search. So it is useful to have a regularized version of the word; but we do not want to modify the

transcription - the misspelling might be relevant, esp. in the case of historic, dialectal, or learner texts. So instead, we provide the regularized orthography as an attribute:

```
<tok reg="small">smal</tok>
```

Correction can be added in the raw XML - but editing tokenized XML is difficult. Therefore, almost all editing in TEITOK is done via a graphical interface, in the case of correcting the annotations on a token by simply clicking on the token. This will open an HTML form, with fields for the relevant annotations.