The SUB-RDD XSLT style guide

Purpose

This document serves to establish consistent XSLT code style for all projects (from June 2018 on) in the SUB Research and Development Department.

The structure of this document is inspired by Google's HTML and CSS style guide. Best practices have been collected from:

- https://superdevelopment.com/2013/08/23/xslt-best-practices/
- https://www.onenaught.com/posts/23/xslt-tips-for-cleaner-code-and-better-performance
- own experiences

General Style Rules

All of the style rules that are specified in the RDD technical reference also apply to the formatting and styling of XSLT code.

Capitalization

All code has to be lower-case. The only exception are XML elements that contain upperspace characters, for example `tei:TEI`.

Use a hyphen for template names instead of camel case, e.g.

```
<xsl:template name="your-template-name">
  <!--- code -->
</xsl:template>
```

Trailing whitespaces

Avoid trailing whitespaces. Files should end with an empty line.

Meta rules

Comments

Write your code in a way that minimizes the necessity for comments, i.e. using variable and template names that express clearly what they are for.

In case this isn't possible, write comments that explain your code in a way that a developer new to the project could understand what and why you are doing what you do.

Mark TODOs

Mark todos with `TODO` and provide information about what should be done.

XSLT specific rules

Structure of the XSLT

The XSLT should start with general information about the output method, parameters, elements that strip/preserve space, imports etc.

Afterwards the templating rules should be listed in a top down order (where applicable), e.g. when serializing TEI-P5:

```
<xsl:template match="tei:TEI">
  <!--- code -->
</xsl:template>

<xsl:template match="tei:teiHeader">
  <!--- code -->
</xsl:template>

<xsl:template match="tei:body">
  <!--- code -->
</xsl:template>

<xsl:template match="tei:div">
  <!--- code -->
</xsl:template>
```

...

Formatting

Use a new line after each of these instructions, and indent every child element:

- xsl:template
- xsl:call-template
- xsl:if
- xsl:choose
- xsl:when
- xsl:otherwise
- xsl:for-each

Code that belongs together shouldn't be vertically separated. Every time you begin a new "thought" or conceptual unit, insert a vertical space in form of an empty line.

Line-length

Although the general RDD style guides apply, be mindful of the length of any XQuery path. Keep them as short as possible, since they are hard to read and comprehend when they are too long.

Quotation marks

Use double (`" "`) quotation marks at the top level, e.g. attributes within the xsl namespace. Use apostrophes (`' '`) for parameters of XQuery functions or any other second level. Include decimal unicode references (`&#34; = "` and `&#39; = '`) when there is a need for single or double quotations marks at any higher level.

Special Characters

Include visible special characters directly whenever it is possible. Use decimal unicode references for non-standard whitespace characters and combining characters.

Mind performance

In order to improve the performance of your transformation, stick to the following:

- keep the scanning of an XML document (e.g. by using //someXPath) to a minimum
- when the output format is XML or HTML, insert `indent=no` to the respective `<xsl:output method="..."/>`
- use `<xsl:template match="">` instead of `xsl:for-each` where possible