

A first few XPath functions

David J. Birnbaum, Martin Holmes

A first few XPath functions

- Already covered
 - # `count()` (Kay 733)
 - # `not()`
 - # `position()` and `last()` (Kay 854)
- Functions that operate on strings
 - # `normalize-space()` (Kay 845)
 - # `string-length()`
 - # `replace()` (Kay 862)
 - # `translate()` (Kay 897)
 - # `concat()`
- functions that operate on sets of numbers
 - # `avg()` (Kay 718)
 - # `max()` (Kay 830)
 - # `min()` (Kay 830)
 - # `sum()` (Kay 889)
- `distinct-values()` (Kay 749)
- `current-date()` (Kay 738)

What is a function?

- Functions in XPath work just like functions in other programming languages.
- A function is called by **name**, with following **parentheses**, containing **parameters**.
- Parameters are "things you want the function to work on".
- A call to a function will be replaced by the result from the function.

```
<xsl:value-of select="count(//sp)"/>
```

Hamlet again...

Open the hamlet.xml example file, which you can find here: <http://web.uvic.ca/~mholmes/dhoxss2013/examples/hamlet.xml>

Open the `xpath_functions.xsl` example file, which you can find here: http://web.uvic.ca/~mholmes/dhoxss2013/examples/xpath_functions.xsl

Switch to the XSLT debugger, so you can experiment with functions. Set the output to XHTML.

not ()

- Syntax: `not(boolean)`
- Semantic: reverses boolean truth value
- Example: `match="persName[not(surname)]"`
- Task: find the last line in *Hamlet*. (Hint: the last line is the one which has no following lines.)

normalize-space() and string-length()

- Trims leading and trailing whitespace; reduces runs of any remaining space to single blank
- Example:

```
# string-length( //l[@xml:id eq 'sha-ham301055']/text() ) # 70
# string-length( normalize-space( //l[@xml:id eq 'sha-ham301055']/text() ) ) # 42
```
- Use `normalize-space()` to clean up messy text, especially due to pretty-printing.

replace()

- Syntax: `replace(input-string, regex-pattern, replacement-string, flags)`
- Example: `replace('Syd', 'y', 'i')`
- Task: Retrieve all roles in *Hamlet* and replace “a” with “Hi, Mom!”
 - # Complication: What’s wrong with `replace(//role, 'a', 'Hi, Mom!')`?
 - # Clever trick: Use an path expression with a dot to refer to the context node, e.g., `//role/upper-case(.)` means “find all the `<role>` elements and then upper-case the context node (the `<role>` element you just found)”
- Uses

- # Regex patterns (advanced topic), such as “transform all dates like MM/DD/YYYY to YYYY-MM-DD”
- # One-to-many or many-to-one transformations (e.g., change LC Cyrillic romanization of “#” as “ch” into linguistic “#” [or vice versa])

translate()

- Syntax: `translate(input-string, characters-to-match, replacement-characters)`
- Limitation: performs only one-for-one replacements (use `replace()` for more complex replacements)
- Example:
`translate('Syd', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyza')`
Cf. `upper-case()` and `lower-case()`
- Task: Replace all vowels (aeiou) in the names of *Hamlet* characters with exclamation marks
- Uses
 - # One-to-one transformations
 - # Change unix paths to Windows paths
 - # Change European numerical punctuation to US
 - # Convert text from legacy character sets to Unicode

concat()

- Syntax: `concat('str1', 'str2', 'str3') # str1str2str3`
- Example: `concat(/TEI/descendant::title[1], ' by ', /TEI/descendant::author[1])`
- Uses
 - # Constructing content
 - # Presenting the results of queries in human-readable sentences

avg(), max(), min(), sum()

- Syntax: `avg(sequence-of-numbers)`
- Example: `avg(1,2,3,4,5,6)`

A first few XPath functions

- # What's wrong with: `avg(1, 2, 3, 4, 5, 6)`?
- Similarly: `max()`, `min()`, and `sum()`
- Uses
 - # Basic arithmetic operations on sets of numerical data items
- Task: Find the average length of a speech in *Hamlet*, and then the longest and shortest lengths
- Hints
 - # The function `string-length(string)` returns the length of the string
 - # The dot (`.`) refers to the current item
 - # `//sp/string-length(.)` will return a sequence consisting of the length of each string (try it)

distinct-values()

- `distinct-values(//speaker)`
 - # Returns 38 items
 - # What does `//speaker` return?
- `//role`
 - # Returns 37 items
 - # What does `distinct-values(//role)` return?
 - # Food for thought: What does the discrepancy between speakers and roles mean?

position() and last()

- Returns the position of an item in a sequence
- Syntax: `item/position()`
- Example: `//role/position()` returns the number of each role in the list of roles
- Task: find the stage direction in the last line Hamlet speaks
- Note: the `last()` function returns the sequence number of the last node in the set
- Why does this fail? `//sp[@who=' #Hamlet']//l[position() = last()]/stage`

current-date()

- Syntax: `current-date()`
- Returns the current date; useful for date-stamping
- Cf. `current-dateTime()`, `current-time()`
- May be used with `format-date()` (`-dateTime`, `-time`) to massage output (Kay 781)
`format-date()` works only in XSLT, not in the XPath browser of <Oxygen/>
- Example: `current-date()`
- Example: `format-date(current-date(), '[Y]-[M01]-[D01]')`
- Example: `format-date(current-date(), '[FNn], [MNn] [D1o], [Y] [E]')`

Other functions

- There are many, many other XPath functions.
- Whatever it is you want to do, there's probably a function for it.
- Find a small selection here: http://www.w3schools.com/xpath/xpath_functions.asp
- Find the full specification here: <http://www.w3.org/TR/xpath-functions/>