

XPATH

Xpath is a query language for XML, it is used to access information contained in a XML file.

It is:

- used to select parts of the XML file
- to navigate the hierarchy of the XML file
- to check if some condition are satisfied.

In Xpath we have 3 different types of expressions:

1. Value expressions, to define proposition on values
2. Path expression, to select parts of the XML file
3. Node set expressions, used to combine the results of different expressions

PATH EXPRESSIONS

They can be:

- relative
- absolute if they starts with /

Are made by a sequence of steps separated by /

Each one step has 3 different parts:

1. an axis
2. a node-test
3. 0 or more predicates

```
axis::node-test [predicates]
```

At each one step the context (the result) is modified following the axis, then is restricted to the elements with the name of the node-test and eventually filtered by predicates.

AXIS

The axis modify the context.

We have different valid axis that are:

- child (default) and descendant (roughly /)
- parent (..) and ancestor
- following-sibling and preceding-sibling
- self (.), descendant-or-self and ancestor-or-self
- following and preceding
- attribute (@) and namespace

Examples:

All ancestors of professors who were professors:

```
//professor/ancestor::professor
```

NODE-TESTS

They are used to restrict the context

Valid node-tests are:

- The name of an element
- The wildcard *
- `comment()`
- `text()`
- `processing-instruction()`
- `node()`

When we use * we are restricting the contexts to elements only.

Using node we restrict it to elements and attributes only.

PREDICATES

They are used to filter the context.

We can use in it any possible value expressions.

Non boolean expressions are interpreted as follows:

- integer, are equivalent to the `position()`
- strings, is true if the string is not empty
- node sets, are true if the node sets are not empty

VALUE EXPRESSIONS

Value expressions include

- strings and numeric literals: “home”, 5, 1.3...
- variable reference: \$x, \$y ...
- function invocation
 - fn:upper-case(\$x) ...
- logic and arithmetic comparison: \$x <10 ...
- Path expressions

- `last()`: position of the last element in the context
- `position()`: position of each element in the context
- `string(arg)`, `concat(s1,s2,...)`,
`starts-with(s1,s2)`, `contains(s1,s2)`,
`substring(s, start, len?)`, `string-length(s)`,
`normalize-space(s)`
- `Boolean(arg)`, `not(arg)`, `true()`, `false()`
- `lang(lang)`: true if the language of the current node is that given in the argument
- `number(arg)`, `floor(number)`, `ceiling(number)`,
`round(number)`
- `count(arg,arg,...)`, `sum(arg,arg,...)`,
`min(arg,arg,...)`, `max(arg,arg,...)`,
`avg(arg,arg,...)`

For convention we use fn: before the functions

NODE-SETS EXPRESSIONS

Allow to combine contexts obtained by several expressions. They are:

- **| (or union)**
- **intersect**
- **except**
- **, (or concatenation)**

EXAMPLES

```

<Library title="Alpha">
  <Book title="Bravo">
    <Chapter title="Charlie">Traveling with a poodle</Chapter>
    <Chapter title="Delta">Mouth of the Mississippi</Chapter>
  </Book>
  <Book title="Echo">
    <Chapter title="Foxtrot">Dance to four-quarters time</Chapter>
    <Part title="Golf">
      <Chapter title="Hotel">Check in, but not out</Chapter>
      <Chapter title="India">Indus to the Ganges</Chapter>
    </Part>
  </Book>
  <Book title="Juliet">
    <Part title="Kilo">
      <Chapter title="Lima">Peru is here too</Chapter>
      <Chapter title="Mike">Decorated Sistine Chapel</Chapter>
    </Part>
    <Part title="November">
      <Chapter title="Oscar">Academy Awards</Chapter>
      <Chapter title="Papa">To me he was so wonderful</Chapter>
    </Part>
  </Book>
</Library>

```

For example:

- //Chapter[contains(@title, "r")]
 - we select all the chapters that contains in the title the r, starting from the root of the document //
- //Chapter[@title="Papa"]/..
 - we get the chapter with title Papa and we select the ancestor of it (Part with title November)
- //Part[@title="Kilo"]/Chapter
 - we select Chapter Lima and Chapter Mike
- /Library/Book/Chapter
 - we select the chapter under Book tag, so Charlie, Delta, Foxtrot (not the others)