

Advanced XML 1: Namespaces

Namespaces

- A way to associate XML documents with schemas.
- A way to disambiguate elements, attributes, and other features of XML that might have the same name in different application.

Namespace Syntax

- Two Parts: prefix and URI
- The prefix is arbitrary and can be just about anything. Often, we use the same abbreviations as the name of the language.
- The URI doesn't actually get anything from that location. Rather, the processor sees the namespace and then looks for a schema associated with that namespace.

Examples of Elements with Prefixes

- `<svg:g />` an SVG element for grouping.
- `<tei:g />` a TEI “Glyph” element.

(Don't confuse the prefix with the namespace. Often, elements and attributes with no prefix will belong to a namespace that they have inherited.)

	Namespace URI
xml	*
html5	*
kml	http://www.opengis.net/kml/2.2
svg	http://www.w3.org/2000/svg
tei	http://www.tei-c.org/ns/1.0/
xslt	http://www.w3.org/1999/XSL/Transform
xpath math functions	http://www.w3.org/2005/xpath-functions/math
(custom)	(your URI)
...	(any more examples)

Namespaces as Taxonomy

Genus	Species	Common Name or Description
Homo-	Sapiens	Human
Picea-	Glauca	White Spruce
Agoseris-	Glauca	Prairie Agoseris
Pilosocereus-	Leucocephalus	A fuzzy cactus
Haliaeetus-	Leucocephalus	Bald Eagle
Namespace	Element	Description
svg:	g	An SVG element for grouping objects.
tei:	g	A “Glyph” element in TEI.
html:	head	The part of the HTML file used for metadata.
tei:	head	An element to describe (e.g. chapter) headings in the body of a TEI text.

** Adapted from a chart of unknown provenance shared on Twitter, possibly originating from Julia Flanders and Syd Bauman.*

How to Declare Namespaces in XML Files

Without a prefix

We attach the namespace node (without any prefix information) to an element.

- This sets the default namespace for the scope of the given element.
- That element and all its children are interpreted as belonging to that namespace.

Example:

```
<svg
xmlns="http://www.w3.org/2000/svg">
    <defs />
    <desc />
</svg>
```

With a prefix

We attach a namespace node with prefix information to an element.

- Only those elements which have the prefix *and* which are nested within that element are interpreted as belonging to that namespace.
- Conventionally, these namespace nodes are attached to the root element.

```
<tei.2 xmlns:svg="http://www.w3.org/2000/svg"
xmlns="http://www.tei-c.org/ns/1.0/">
    <teiHeader> ... .. </teiHeader>
    <text>
        <body>
            <svg:svg>
                <svg:defs />
                <svg:desc>
                    <svg:circle cx="" cy="" r="" />
                </svg:desc>
            </svg:svg>
        </body>
    </text>
</tei.2>
```

Advanced XML 2: Datatypes

Datatypes

- A way to restrict the types of data that a processor will accept.
- Useful for standardization.

Why do these matter?

- Databases / schemas / consistent data
- e.g. entering data into forms on the web.

List of Especially Useful Datatypes		
Type	Description	example
string	Some text, numbers, etc. (Order is alpha-numeric.)	I am some text. 012345.
boolean	True or False	*
date	A simple date format, as YYYY-MM-DD	2016-06-16
dateTime	A combination of date and time that orders correctly. *	2003-01-02T11:30:00-05:00
decimal	One of several datatypes used for numbers.	2 2.0 323423 .0001
anyURI	Universal Resource Identifier – Identifies a resource by its location in a file system or web address.	http://www.tei-c.org

And eighteen more ...

Advanced XML 3: Types of Nodes

Most of these XML nodes can be accessed, manipulated, or generated using XPath and Xquery.

There are more than just elements and attributes...

Seven Types of Nodes in XML	
Name	Example
document	myfile.xml
element	<div>
attribute	cx="45"
text	It was the best of times, it was the blurst of times.
Namespace*	xmlns="http://www.opengis.net/kml/2.2"
processing instruction	<?xml version="1.0"?>
comment	<!-- Note to future self... -->

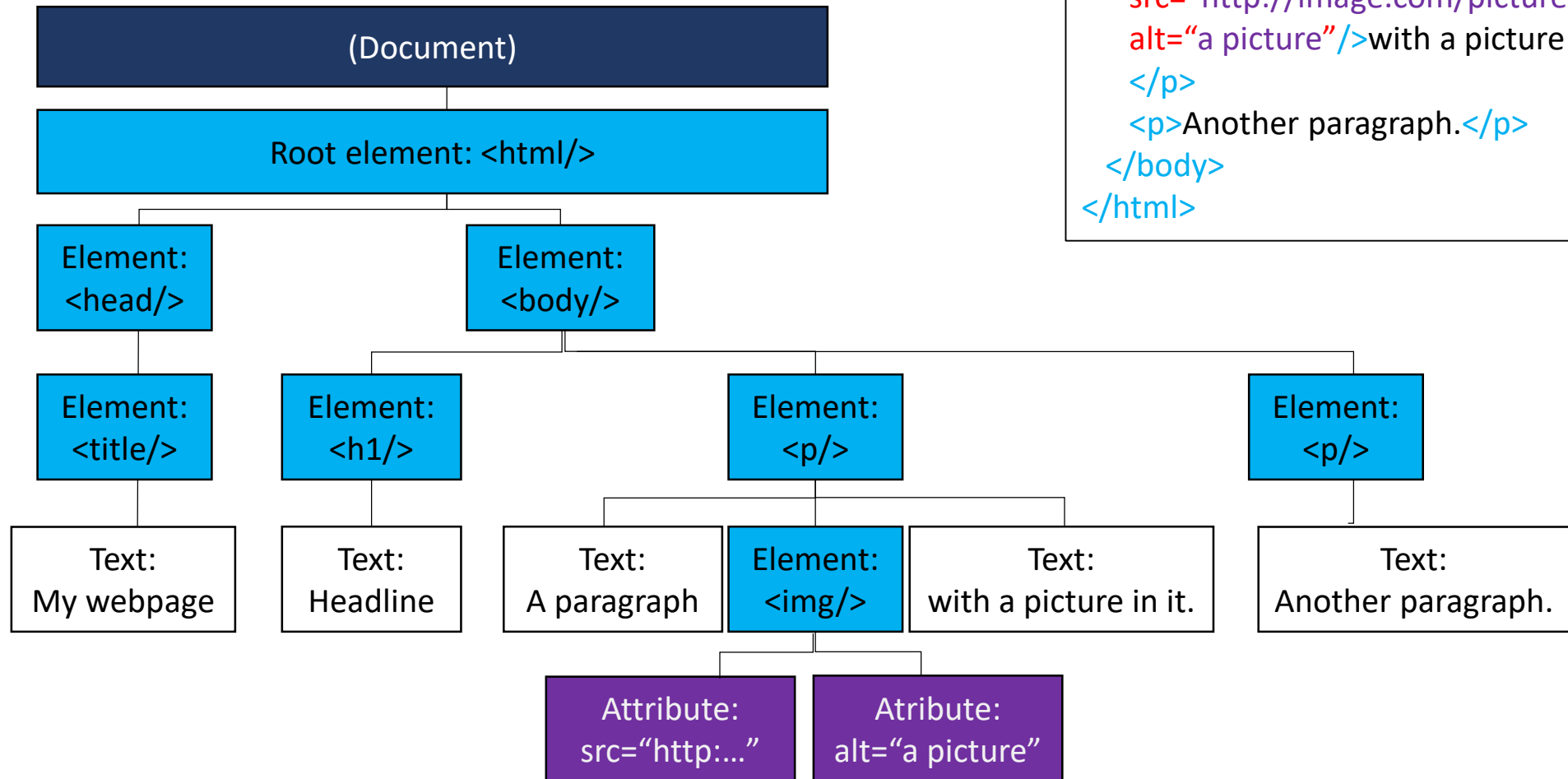
* not accessed via XPath/XQuery.

Manipulating XML 1: XPath

XML Path Language (XPath)

- A way of navigating the XML tree.
- An important component of XQuery and XSLT (more on these later).
- Similar in utility to CSS or jQuery selectors.
- Similar in look to file paths on your computer and web addresses.
- Includes functions for doing things with XML documents.
- Also used in Python libraries for working with XML and HTML, such as BeautifulSoup and lxml. (... but not fully implemented)

The XML Tree Structure (Again!)



```
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body>
    <h1>Headline</h1>
    <p>A paragraphwith a picture in it.
    </p>
    <p>Another paragraph.</p>
  </body>
</html>
```

XML Path Operators

We can move the context around a document with path expressions, which make use of operators like those to the right.

We can also manipulate our documents by putting functions in our Xpath

(We will see some examples later.)

Operator	How it changes the context
/	Selects the root element. (When placed at the beginning of the path expression.)
/	Takes a step along some axis. (When following another operator)
.	Selects the current context; similar to jQuery \$(this).
*	Accepts anything and everything it can.
//	Takes as many steps as is necessary along the descendant axis. Abbr. for /*/
@	Selects an attribute
[]	“Predicate Filter”: Filters a selection.
	A union of two sets.
..	Specifies the parent.
function()	Calls an XPath function at the context.
text()	Specifies a text node.
... and more!	

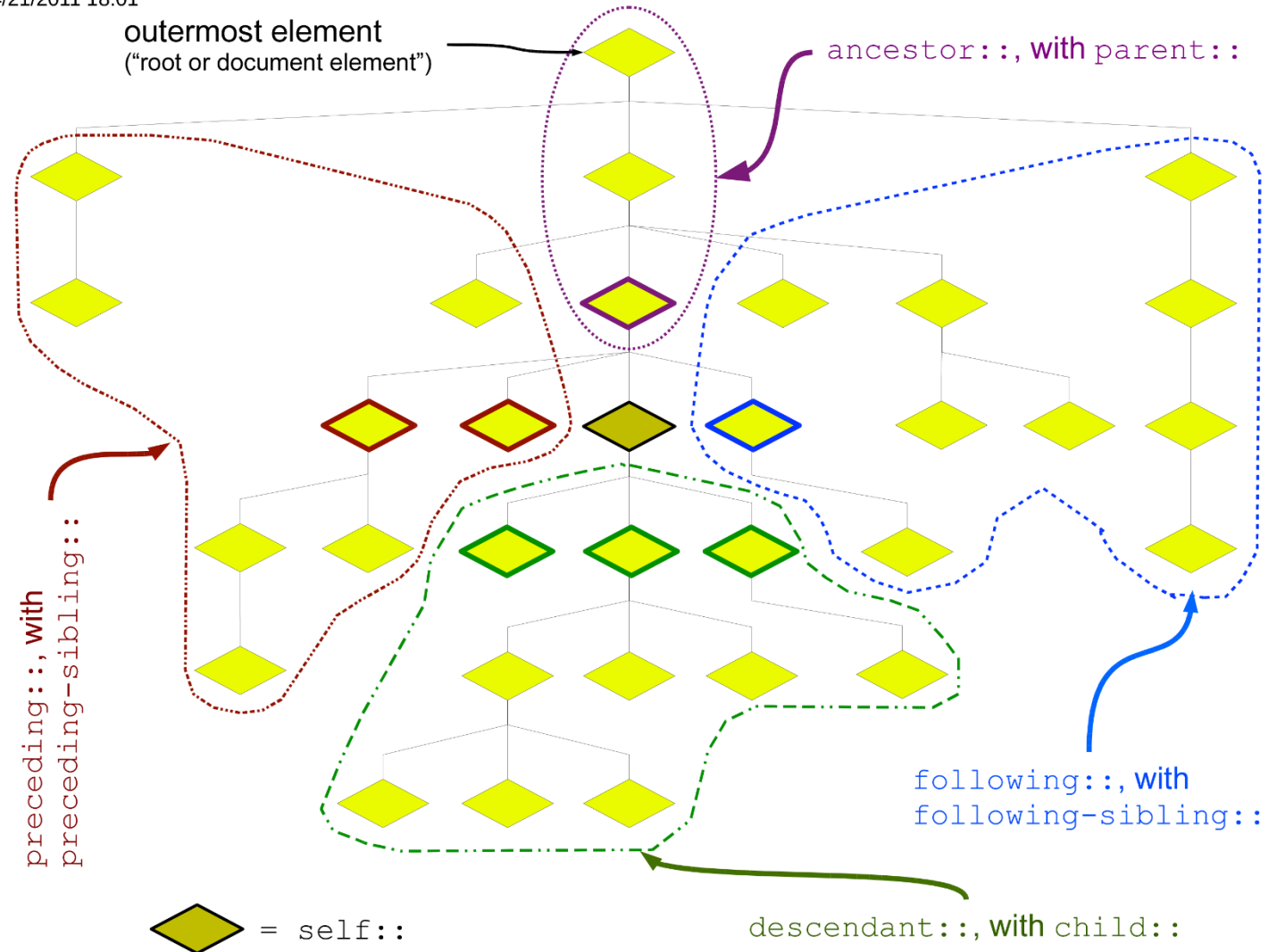
XML Path Axes

There are 13 XPath Axes.

They can be accessed with special operators, including:

- Parent and Ancestor
- Preceding
- Following
- Preceding Sibling
- Following Sibling
- Child and Descendant
- Self
- Attribute

04/21/2011 18:01



XPath Expressions and CSS Selectors Compared

XPath Expression	Description	CSS Selector
kml//Placemark	Select all <Placemark> elements which are descendants of a <kml> element.	kml Placemark
svg/defs	Select all <defs> elements which are children of an <svg> element.	svg > defs
p quote	Select all elements which are either <p> or <quote>.	p, quote
path[@class="dark"]	Select all <path> elements which have the attribute class="dark".	path.dark
p[1]	Selects all <p> elements which are the first <p> among their siblings. (e.g. the first paragraph of a section.)	p:first-of-type
path/@class	Select all class attributes that are attached to a <path> element.	<i>Can't select attributes.</i>