XML Namespaces

Understanding XPath and Namespaces

XML Elements and Namespaces

XML elements have a name (e.g. `body`) and may also have a namespace URI (e.g. `http://www.w3.org/1999/xhtml`). The two together are the proper name of the element.

XML documents don't specify the namespace at the location of each element. Instead, they specify it at one place, and assign it a prefix, which acts as a shortcut. For example, `<... xmlns:h="http://www.w3.org/1999/xhtml">` would indicate that within that element, `h:body` is a shortcut for `(http://www.w3.org/1999/xhtml, body)`. Alternatively, they can specify a *default namespace*, which applies to any elements that lack a prefix. For example, `<... xmlns="http://www.w3.org/1999/xhtml">` would indicate that within that element, `body` really means `(http://www.w3.org/1999/xhtml, body)`.

It's important to realize that **a default namespace is still a namespace**. Specifically, `(, body)` - the `body` element in an empty is *completely different* from `(http://www.w3.org/1999/xhtml, body)`.

Finally, it's critical to realize that a namespace prefix has no semantic value. It's a shortcut to the parser, which is read to determine the namespace, and then thrown away. For instance, if a document defines two namespace prefixes, `a` and `b`, and makes them both shortcuts for the same namespace, then, in that document, `a:body` is the exact same element as `b:body`. No XPath query can distinguish them!

XPath and Namespaces

By default, XPath queries refers to elements in the empty namespace. If you're looking for an element in a different namespace (i.e. any URI), you need to specify the namespace. In lxml, this is done by passing a namespace dictionary as a second argument, and using prefixes from that dictionary:

```
# From
http://stackoverflow.com/questions/297239/why-doesnt-xpath-work-when-processing-an-xht
ml-document-with-lxml-in-python
>>> tree.getroot().xpath(
... "//xhtml:img",
... namespaces={'xhtml':'http://www.w3.org/1999/xhtml'}
... )
[<Element {http://www.w3.org/1999/xhtml}img at 11a29e0>]
```

Or, another example:

```
# From http://lxml.de/xpathxslt.html#namespaces-and-prefixes
>>> r = doc.xpath('/t:foo/b:bar',
... namespaces={'t': 'http://codespeak.net/ns/test1',
...              namespaces={'t': 'http://codespeak.net/ns/test1',
...                          'b': 'http://codespeak.net/ns/test2'})
```

Note that the prefix used in the query and dictionary must match each other (of course), but *do not need to match the document prefix whatsoever*. The document prefix is entirely irrelevant: As explained above, it's a shortcut to the parser, but then thrown out.

Likewise, *if the document uses a default namespace (with no prefix), you still need to specify the namespace in an XPath query, using a prefix and dictionary*.

Further examples

For more examples:
* http://docs.oracle.com/cd/E22345_01/doc.111/e22309/xpath.htm#insertedID3
* http://blog.davber.com/2006/09/17/xpath-with-namespaces-in-java/
* http://stackoverflow.com/questions/297239/why-doesnt-xpath-work-when-processing-an-xhtml-document-with-lxml-in-python

* http://stackoverflow.com/questions/11565285/xpath-issue-using-lxml-library-in-an-xml-file-with-namespaces/11565736#11565736
* http://stackoverflow.com/questions/5123521/xpath-and-xml-multiple-namespaces