Programming

with python™

By

Rami Tailakh

Senior Software Engineer and Data Science Practitioner

MSc in Applied Computing and Information Technology

Day-7 Agenda

- Continue with Regex

- XML Query (XPath)

XML XPath

- XPath is a query language used to search through an XML
- XPath stands for XML Path Language used to identify and navigate nodes in an XML document
- The importance of XPath is to scanning and populating XMLs
- XPath expressions can be used to specify more useful searches
- XPath through an xml like SQL through a database

XML XPath-Continue

- The ElementTree package supports XPath for locating elements in a tree
- With **ElementTree** and **For Loop**, you can parse, explore, and populate XML files
- Starting with Python 2.7, ElementTree has a better support for XPath queries
- The functions findall() and find() are popularly used

XPath Expressions

- The
- XPath expressions are used to select nodes or node-sets in an XML document.
- These path expressions look like the path in computer file systems
- XPath expressions can be used in Python and and lots of other languages

XPath Nodes

Node Type	Description
Root	Root element node of an XML Document.
Element	Element node
Text	Text of an element node.
Attribute	Attribute of an element node.
Comment	Comment node

XPath Operators

Operator	Description
<node-name> or <tag>	Select all nodes with the given name "nodename". This works with the first element root.findall("server")
/	Selection starts from the root node root.findall("./server")

XPath Operators-Continue

Operator	Description
.	Selects the current node. This is mostly useful at the beginning of the path (root). root.findall(".")
..	Selects the parent of the current node (element) root.findall("server/..")
//	Selection starts from the current node that match the selection root.findall(".//ip")

XPath Operators-Continue

Operator	Description
[@attrib]	Selects all elements that have the given attribute. root.findall(".server/[@version]")
[@attrib='value']	Selects all elements for which the given attribute has the given value. root.findall(".server/[@name='server-1']")
[tag]	Selects all elements that have a child named tag

XPath Operators-Continue

Operator	Description

[tag='text']	Selects all elements that have a child named tag whose complete text content equals the given text.
	root.findall(".//*[status='down']")
	root.findall(".//*[status='down']/status")

[position]	Selects all elements that are located at the given position. It starts from 1.
	root.findall("./server/[1]")
	root.findall("./server/[2]")
	root.findall("./server/[last()]")
	root.findall("./server/[last()-1]")

XPath Practice

Go through the books.xml document on Github to query the following:

- Find the book with title: ‘Calculus’

```
>>> tree.find("Books/Book[Title='Calculus']")
```

- Find the book with title: ‘Calculus’

```
>>> tree.find("Books/Book[Title='Calculus']")
```

XPath Practice-Continue

- Find the firstly indexed book:

```
>>> tree.find("Books/Book[1]")
```

Or

```
>>> tree.find("Books/Book[@id='5']")
```

- Find all books:

```
>>> [b.text for b in tree.findall(".//Title")]
```

- Find all author names:

```
>>> [a.text for a in tree.findall(".//Author")]
```

- Find all paid books:

```
>>> [b.text for b in tree.findall("./Books/Book[@price]/Title")]
```

XPath Practice-Continue

- Find all books whose price is **5.50** and volume **1**:

```
>>> [b.text for b in tree.findall("./Books/Book[@price='5.50'][@volume='1']/Title")]
```

```
>>> # Try the practice with the following line of code:
>>> [b.text for b in tree.findall("./Books/Book[@price='5.50' and @volume='1']/Title")]
```

Challenge

1. Using the books.xml document, find the all the information about books written by an author named Rob.

   Hint: Use filter, lambda, and ElementTree.

2. Using the books.xml document, find the free books (no price attribute).
3. Convert the data in the books.xml document into a tabular form.
4. Find the answer of the first challenge by another package called: **lxml.etree**, which has ‘contains’ capability.
5. Rewrite all practices and challenges using the **lxml.etree** package.