How to handle XML?

XML = eXtensible Markup Language

- Data are structured in a hierarchy of node
- Nodes can be elements, text nodes or attributes
- It can be used to represent structures with complex details

XML: an example

```
<breakfast_menu>
   <food>
      <name>Belgian Waffles</name>
      <price>$5.95</price>
      <description>Two of our famous Belgian Waffles with...</description>
      <calories>650</calories>
   </food>
</breakfast_menu>
```

- `<breakfast_menu>` : the root element
- `<food>` : an element
- **Belgian Waffles**: a text content

XML: targeting element thanks to XPath

- One can query the document thanks to **XPath** expressions
- **XPath** takes the path from the root element to the targeted node
- Each level is represented by a `/`

XML: Read a document using python (native option)

```
import xml.etree.ElementTree as xmlReader
# read from wml
tree = xmlReader.parse('menu.xml')
```

XML: Read a document using python (with lxml)

```
from lxml import etree
# read from xml
tree = etree.parse('menu.xml')
root = tree.getroot()
print(root)
```

`lxml` has a more extensive support of XPath, and it is really convenient

XML: get the elements in a list using xpath

```
elems = root.findall('./food')
data = [[elem.find("./name").text,
        elem.find("./price").text
        ] for elem in elems]

print(data)
```

XPATH 101:

- When a query starts with : `/` : the query looks data from the root
- query starts with `./` : path is relatively taken from the current path
- one can filter data thanks to predicates, example

```
./food[starts-with(./name/text(), 'Be')]
```

Exercise : load this xml file from your preferred python environment, then do the same in Orange (using **Python Script** widget)

Bind xml results to an Orange data table

- It is possible to bind "raw data" in Orange tables
- this is available thanks to this code snippet (from Orange docs):

```
from Orange.data import *
data = [
    ['green', 4, 1.2, 'apple'],
    ['orange', 5, 1.1, 'orange'],
    ['yellow', 4, 1.0, 'peach']
]
color = DiscreteVariable('color', values=set([row[0] for row in data]))
calories = ContinuousVariable('calories')
fiber = ContinuousVariable('fiber')
fruit = DiscreteVariable('fruit', values=set([row[3] for row in data]))

domain = Domain([color, calories, fiber], class_vars=fruit)

table = Table.from_list(domain, data)
```

XML exercises with Orange

- In Orange, import the previous file through a widget named "Python Script" and transform it to a data table.
- Store this data table in a variable named "output_data".
- Then put a table widget after the python script widget

XML exercises with Orange (2)

- Consider the zip: file, it contains data about article published on ML over time in XML format
- try yourself on extracting data on one file
- extract the year, month, title, and add a column "topic" that contains always ML
- iterate over the files present in this folder to extract all the information, and then integrate it in orange thanks to the Python script widget
- Analyse data thanks to orange (visualization)