XML - Dateien

Liegen Daten im XML format vor, empfiehlt sich die Verwendung der [etree-Bibliothek (https://docs.python.org/3/library/xml.etree.elementtree.html)](https://docs.python.org/3/library/xml.etree.elementtree.html).

In [2]:
```
import xml.etree.ElementTree as etree

document = etree.parse('Data/country_data.xml')
country_data = document.getroot()

for country in country_data:
    print(f"{country.tag} '{country.attrib['name']}'")
    neighbors = []
    for country_child in country:
        if country_child.tag == 'neighbor':
            neighbors.append(country_child.attrib['name']) # let's collect all
        else:
            print(f"\t{country_child.tag}: {country_child.text}") # print othe

    print(f"\tneighbors: {', '.join(neighbors)}")
```

```
country 'Liechtenstein'
        rank: 1
        year: 2008
        gdppc: 141100
        neighbors: Austria, Switzerland
country 'Singapore'
        rank: 4
        year: 2011
        gdppc: 59900
        neighbors: Malaysia
country 'Panama'
        rank: 68
        year: 2011
        gdppc: 13600
        neighbors: Costa Rica, Colombia
SomeThingElse 'Winnie-the-pooh'
        type: Bear
        neighbors:
```

Das DOM (Domain Objeckt Model) kann auch geändert werden und dann geschrieben. Dies wird hier kurz gezeigt. Es wäre natürlich auch möglich, das XML direkt wie eine Text-Datei schreiben. Ich persönlich brauche dies fast häufiger.

In [3]:
```
del country_data[3] # remove 4th element, 'SomeThingElse'

# create a new entry 'NewElement' with one attribute 'name' and an sub-element
new_element = etree.Element('NewElement')
new_element.attrib['name'] = 'fooBar'
child = etree.Element('child1')
new_element.append(child)
child.text = "text"

country_data.append(new_element) # add the new element

# make sure document is indented and line-breaks are added for the whole docum
etree.indent(document, '    ')

# save the document to a file
document.write("Data/.2_modified.xml",)
```