Semantic Web – Tutorial #4

Isabelle Kuhlmann

Institute for Web Science & Technologies
University of Koblenz-Landau

May 20, 2021 | Assignment 3

1: XML Validation and XML Schema

1: XML Validation and XML Schema

Task: Is the following XML document valid?

```
1:  ?xml version="1.1" encoding="UTF-8" ?>
2:  <!DOCTYPE book [
3:      <!ELEMENT book (title | author | publisher | year) >
4:      <!ELEMENT title (#PCDATA)>
5:      <!ELEMENT author (#PCDATA) >
6:      <!ELEMENT publisher (#PCDATA)>
7:      <!ELEMENT year (EMPTY)>
8:  ]>
9:  <book>
10:     <author><publisher>Macmillan</author></publisher>
11:     <title>Alice Adventures in Wonderland</title>
12:     <year>1865</year>
13: </book>
```

It is **not** valid:

- Author closes before publisher (line 10)
- Year should be empty (line 12)
- Title and year are not expected on lines 11 and 12 (since author appeared before)

WeST People and Knowledge Networks

2: XML-DTD

2: XML-DTD

Domain Specification

A director, who has a first name and a last name, directs multiple films. A film has a single title, a publishing year, a studio, and it might be based on a book. A book has a title as well, and it has one or multiple authors.

Task: Write a DTD document for the movie domain above. The outermost class must be director.

Task: Translate your DTD document for the movie domain above into a XML Schema. The outermost class must be director. Your must specify at least one mandatory attribute, and at least two entities.

2: XML-DTD

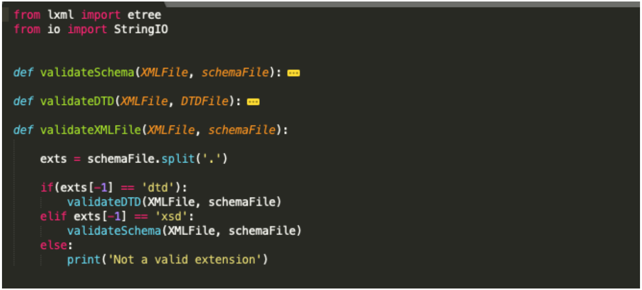
Task: Write an XML document that conforms with the XML Schema you created in the previous task. Your XML document must encode the following information:

> *David Fincher is an American director. Famous examples of his work are Se7en (production studio: New Line Cinema) from 1995 and Fight Club from 1999 (20th Century Fox). Although the movie Fight Club is well-known around the globe, the fact that it is actually based on a novel under the same name, written by Chuck Palahniuk, is lesser known.*

3: Programming

Question 3.1

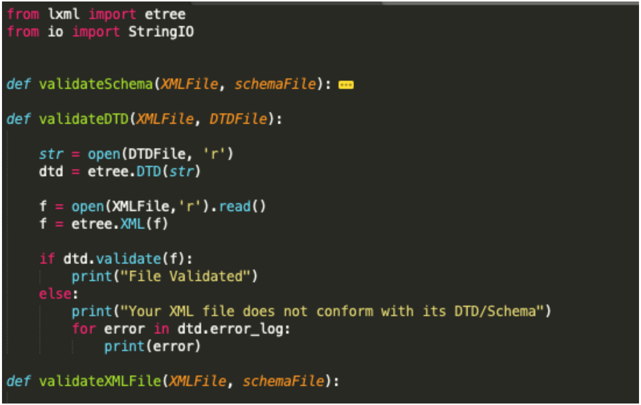
Create a function validadeXMLFile(XMLfile, schemaFile) that validates the XM file against the schemaFile, and print: *"File Validated"*, if the XML file conforms with its DTD or XML Schema; otherwise, it must print *"Your XML file does not conform with its DTD/Schema"*.



```
from lxml import etree
from io import StringIO


def validateSchema(XMLFile, schemaFile): ...

def validateDTD(XMLFile, DTDFile): ...

def validateXMLFile(XMLFile, schemaFile):

    exts = schemaFile.split('.')

    if(exts[-1] == 'dtd'):
        validateDTD(XMLFile, schemaFile)
    elif exts[-1] == 'xsd':
        validateSchema(XMLFile, schemaFile)
    else:
        print('Not a valid extension')
```

Question 3.1

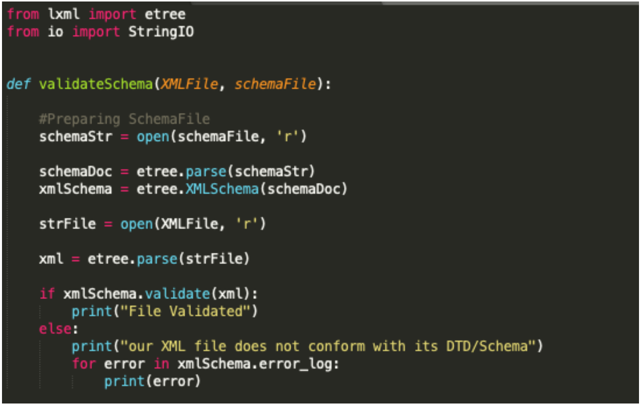
Create a function validadeXMLFile(XMLfile, schemaFile) that validates the XM file against the schemaFile, and print: *"File Validated"*, if the XML file conforms with its DTD or XML Schema; otherwise, it must print *"Your XML file does not conform with its DTD/Schema"*.



```python
from lxml import etree
from io import StringIO


def validateSchema(XMLFile, schemaFile):

def validateDTD(XMLFile, DTDFile):

    str = open(DTDFile, 'r')
    dtd = etree.DTD(str)

    f = open(XMLFile,'r').read()
    f = etree.XML(f)

    if dtd.validate(f):
        print("File Validated")
    else:
        print("Your XML file does not conform with its DTD/Schema")
        for error in dtd.error_log:
            print(error)

def validateXMLFile(XMLFile, schemaFile):
```

Question 3.1

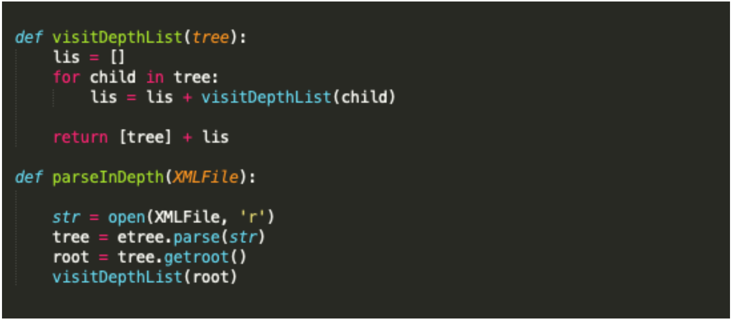
Create a function validadeXMLFile(XMLfile, schemaFile) that validates the XM file against the schemaFile, and print: *"File Validated"*, if the XML file conforms with its DTD or XML Schema; otherwise, it must print *"Your XML file does not conform with its DTD/Schema"*.



```python
from lxml import etree
from io import StringIO


def validateSchema(XMLFile, schemaFile):

    #Preparing SchemaFile
    schemaStr = open(schemaFile, 'r')

    schemaDoc = etree.parse(schemaStr)
    xmlSchema = etree.XMLSchema(schemaDoc)

    strFile = open(XMLFile, 'r')

    xml = etree.parse(strFile)

    if xmlSchema.validate(xml):
        print("File Validated")
    else:
        print("our XML file does not conform with its DTD/Schema")
        for error in xmlSchema.error_log:
            print(error)
```

Question 3.2

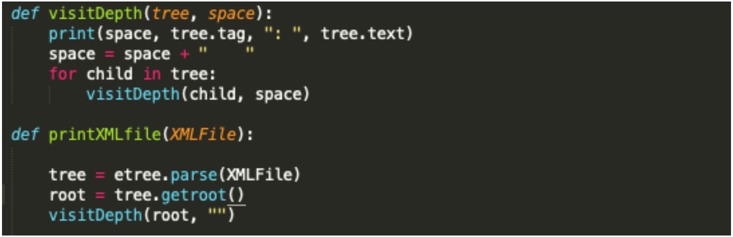
Create a function with the following signature: parseInDepth(XMLFile). The function must visit each node of the XMLfile document top-down in a depth-first policy, and return a list of these nodes in the visiting order.



```
def visitDepthList(tree):
    lis = []
    for child in tree:
        lis = lis + visitDepthList(child)

    return [tree] + lis

def parseInDepth(XMLFile):

    str = open(XMLFile, 'r')
    tree = etree.parse(str)
    root = tree.getroot()
    visitDepthList(root)
```

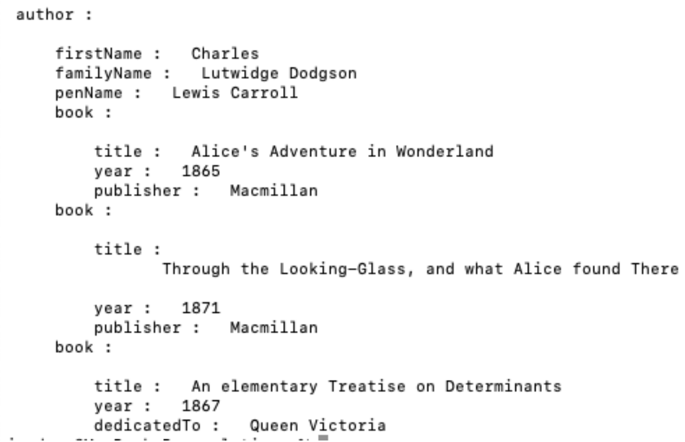
Question 3.3

Create a function with the following signature: printXMLFile(XMLFile). The function must print in a human readable representation (e.g. an indented list) the XMLFile.



```
def visitDepth(tree, space):
    print(space, tree.tag, ": ", tree.text)
    space = space + "    "
    for child in tree:
        visitDepth(child, space)

def printXMLfile(XMLFile):

    tree = etree.parse(XMLFile)
    root = tree.getroot()
    visitDepth(root, "")
```

Question 3.3



```
author :

     firstName :   Charles
     familyName :   Lutwidge Dodgson
     penName :   Lewis Carroll
     book :

         title :   Alice's Adventure in Wonderland
         year :   1865
         publisher :   Macmillan
     book :

         title :
                Through the Looking-Glass, and what Alice found There

         year :   1871
         publisher :   Macmillan
     book :

         title :   An elementary Treatise on Determinants
         year :   1867
         dedicatedTo :   Queen Victoria
```