ENGL 516X:
Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

29 March 2018

Class Outline

- Some code analysis (with random calls to explain)
- Continuing with BeautifulSoup
- Briefly about XML files

Question from last class -1

Taking list as a input from user

```
def readList():
    inp = list(input("Enter a list of numbers: ").replace(",",""))
    #Any of the below definitions of mylist variable do the same thing.
    #mylist = list(map(int,inp[1:-1]))
    #mylist = [int(i) for i in inp[1:-1]]
    mylist = []
    for item in inp[1:-1]:
        mylist.append(int(item))

    print(mylist)
    return mylist

def readList2():
    inp = input("Enter a list of numbers: ")[1:-1].split(",")
    return([int(i) for i in inp])
```

-What is the difference between these two functions? What does map() do?

Question from last class -2

Non-recursive binary search

```
#Modified from: http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBinarySearch.html
def NonRecursiveBinarySearch(alist, item):
    first = 0
    last = len(alist)-1
    found = -1
    while first <= last and found < 0:
        midpoint = (first + last)//2
        if alist[midpoint] == item:
            found = midpoint
        else:
            if item < alist[midpoint]:
                last = midpoint-1
            else:
                first = midpoint+1
    return found

mylist = sorted(readList2())
item = int(input("Enter an number to search in this list: "))
result = NonRecursiveBinarySearch(mylist,item)
if(result < 0):
    print("Item not found.")
else:
    print("Item is found at index", str(result), "in this list")
```

Question from last class -3

HTML scraping using BeautifulSoup - Taichi's solution

```
from bs4 import BeautifulSoup
import urllib.request
from urllib import request

def IntLang(url):
    url\_content = request.urlopen(url).read()
    soup = BeautifulSoup(url\_content,"html.parser")
    tags = soup.find\_all("a", class\_="interlanguage-link-target")
    for tag in tags:
        url = tag.get("href", None)
        if isinstance(url, str):
            if url.startswith("http"):
                print(url)

url = "https://en.wikipedia.org/wiki/Python\_(programming\_language)"

IntLang(url)
```

Question from last class -3

HTML scraping using BeautifulSoup - Brody's solution

```
from urllib import request
from bs4 import BeautifulSoup
import ssl

# this line prevents an error from occuring regarding Certificates
gcontext = ssl.SSLContext(ssl.PROTOCOL_TLSv1)

lang_list = []

def getAllUrls(url):
    url_content = request.urlopen(url, context=gcontext).read()
    soup = BeautifulSoup(url_content,"html.parser")
    tags = soup("a")
    for tag in tags:
        if tag.get("class", None) == ['interlanguage-link-target']:
            url = tag.get("href", None)
            title = tag.get("title",None)
            if isinstance(url,str):
                if url.startswith("http"):
                    print(url)
                    lang_list.append(title)
    print(lang_list)

getAllUrls("https://en.wikipedia.org/wiki/Python_(programming_language)")
```

Reading XML in Python

- After HTML, another common format of saving textual information is XML.
- XML is a structured markup, that is sometimes used to save corpora, database containing multiple fields per text etc.,
- Sometimes, it is also used by some programs that have a web query interface, to transmit results to another program.
- So, learning to parse XML gives you two benefits: to use corpora stored in xml format easily, and to make use of the API of some programs, so that we can build on their output.

Some example XML files

- Using XML to store a corpus (On browser: xml-example.xml, xml-example2.xml)
- Using XML to send a response over internet (On browser: Language Tool's output)
  `https://languagetool.org:8081/?language=en-US&text=my+texd`

Parsing XML in Python

ParsingXMLExample.py, ParsingXMLExample2.py - in Canvas.
One uses Python's XML parser, Another example uses Beautiful Soup
More examples on parsing xml in python at:
`http://www.diveintopython3.net/xml.html`

Practice exercise

Using Beautiful Soup, try to get the TOC from a Wikipedia page. If you finish that, try to do it without beautiful soup, using regular expressions (Not impossible).

Next Week

- Tuesday: Guest lecture by Jordan Smith on working with Beautiful Soup
- Generally: Back to text reader project code- some code analysis, practice with data bases etc.
- May be: accessing files on the computer, specifying paths correctly, etc.
- To Do for you: nothing specific - start thinking about your projects, spend time with python