ENGL 516X:
Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

27 March 2018

Class Outline

- Assignment 5 discussion
- Assignment 6 description
- Working with webpages
- Working with external python libraries: case study with BeautifulSoup

Guiding questions for this week

- How to read information from different urls on the web?
- How to parse HTML (or other such structured text, eg. XML) without writing complicated regular expressions? (1 class)
- How do I read other people's code and make sense out of it? (continuation of last week)

Assignment 5 discussion

- First question -straight forward if you follow the instructions in the book
- Second question - not too difficult if you are comfortable with loops and conditionals
- Third question - slightly more challenging, only 2 groups attempted, I think (and 1 group seems to have got it fully correct)

I need volunteers to discuss their solutions (all people in the group should participate)

Assignment 6 description

- Group assignment
- Grade points: 10
- Deadline: 07 April 2018
- Task: combination of bottle + sqlite3
- You have all you need in the examples discussed in classes

Reading from a webpage

Python's urllib library

- urllib library in python is a collection of Python modules which allow us to read content from webpages.
- It has four modules:
  1. urllib.request - to open urls, read stuff from them
  2. urllib.error - deals with the errors arising out of the open/read process
  3. urllib.parse - for parsing urls. Inside a url, answers questions like: is this http or https? what are the parameters (e.g., xyz.com?user=a&name=b) or queries, and so on.
- Good intro and tutorial on urllib:
  `https://docs.python.org/3/library/urllib.html`

Reading from a webpage, line by line

```
import urllib.request
fhand = urllib.request.urlopen('http://www.py4inf.com/code/romeo.txt')
for line in fhand:
   print(line.decode(encoding="utf8"))
```

Reading from a webpage -2

What if the webpage is not a text file.. and it is a general HTML file?

```
import urllib.request
url = "http://www.theunixschool.com/2012/09/examples-how-to-change-delimiter-of-file-Linux.html"
fhand = urllib.request.urlopen('url')
for line in fhand:
   print(line.decode(encoding="utf8").strip())
```

This will print you all the html, and you need to write your regex or use other means to extract the text you want.

What if the url is an image?

```
import urllib.request
imagepath = "http://www.phdcomics.com/comics/archive/phd031813s.gif"
fhand = urllib.request.urlopen(imagepath)
for line in fhand:
   print(line)
```

What will happen now?

So how should I "download" the image?

```
import urllib.request
imagepath = "http://www.phdcomics.com/comics/archive/phd031813s.gif"
fhand = urllib.request.urlretrieve(imagepath)
```

Now, fhand will tell you where the image is in our computer. What will happen now?

Download and Save a image

```
img = urllib.request.urlopen('http://www.py4inf.com/cover.jpg').read()
fhand = open('cover.jpg', 'wb')
fhand.write(img)
fhand.close()
```

"Scraping" HTML

- One of the popular uses of urllib library is to extract text from webpages. This is called "scraping".
- Scraping is also what a search engine like google does, when it crawls all the webpages, and then, retrieves them when we query it.
- So, how do we scrape? One simple (not really simple) way: regular expressions.

"Scraping" HTML

- One of the popular uses of urllib library is to extract text from webpages. This is called "scraping".
- Scraping is also what a search engine like google does, when it crawls all the webpages, and then, retrieves them when we query it.
- So, how do we scrape? One simple (not really simple) way: regular expressions.
- We saw some regular expression HTML examples in the past few weeks

"Scraping" HTML

- One of the popular uses of urllib library is to extract text from webpages. This is called "scraping".
- Scraping is also what a search engine like google does, when it crawls all the webpages, and then, retrieves them when we query it.
- So, how do we scrape? One simple (not really simple) way: regular expressions.
- We saw some regular expression HTML examples in the past few weeks
- Another way is to use one of the several python libraries for HTML parsing.

Scraping HTML with Regular Expressions

Three important things to know:

1. How can I look at the html source of a webpage?

Scraping HTML with Regular Expressions

Three important things to know:

1. How can I look at the html source of a webpage?
2. How can I find what patterns I should target?

Scraping HTML with Regular Expressions

Three important things to know:

1. How can I look at the html source of a webpage?
2. How can I find what patterns I should target?
3. Third, but most important: how to write regular expressions!

Scraping HTML with Python Libraries

- There are a lot of Python libraries to do HTML scraping and parsing.
- Beautiful Soup is one of them, and very popular (It was popular in 2010 too, when I first used it)
- Lot of HTML on the web is broken.. but Beautiful Soup is very tolerant to such broken HTML and still gives you clean output.
- How to install? - install from PyCharm. Go the DIY way and figure out how to install.

BeautifulSoup

Installation

- How many of you successfully installed Beautiful Soup on your machines?

BeautifulSoup

Installation

- How many of you successfully installed Beautiful Soup on your machines?
- `https://www.youtube.com/watch?v=HJ9bTO5yYw0` - a video for doing it on PyCharm.
- only difference: in Python3, after installation, you write: "from bs4 import BeautifulSoup"

Using Beautiful Soup

BeautifulSoupBasics.py

How to work with Beautiful Soup

`http://www.crummy.com/software/BeautifulSoup/bs4/doc/`

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 Exercises

Write a program to read in any wikipedia page on some topic (eg: Python programming language), and get all other language pages related to that webpage as the output. You can use Beautiful Soup, or Regular Expressions. If you see a access forbidden error while trying to read a page, figure out how to fix it.