Assignment1 Tyler Medina

1/26/17

**1 Question 1**

* 1. **The URI**

For part 1 of this assignment, I decided to use a simple php form method that I created to post the form data. It consists of two files, test.php and welcome.php. Figure 0 and Figure 1 show these two files respectively.

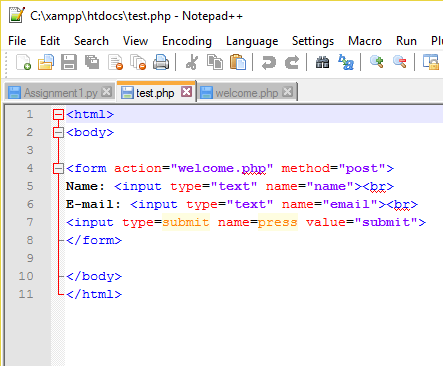
****

Figure 1. Html displayed in test.php

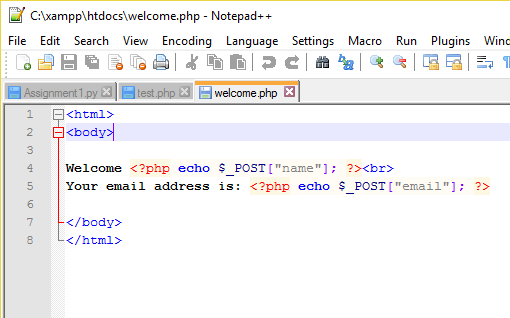


Figure . Html displayed in welcome.php

These files work to create a simple webpage that displays the user’s name and email once filled in and the submit button is clicked.

**1.2 Environment**

I utilized the windows command shell to post data with the curl command. It was originally unable to work, so I had to download and install curl. To test the php files, a localhost needed to be created with apache. This led to the easy to type url localhost/test.php.

**1.3 Test and Results**

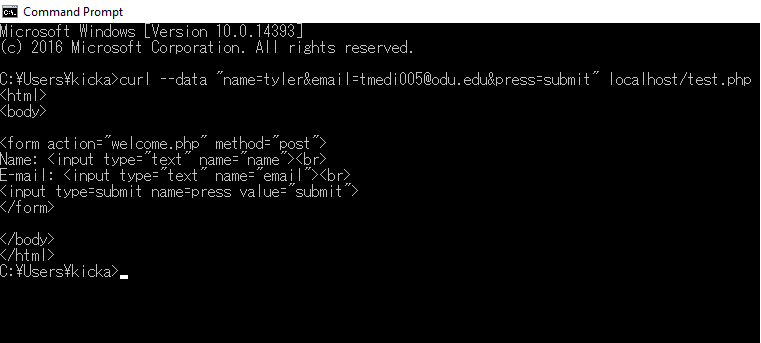
The command used to post the data was curl –data. I filled the name value with my name and the email value with my ODU email. The html was tight so there wasn’t a need to encode with %20 for spaces. Figure 3 displays the command input into the command line. 

Figure . Command line input and the response from the server

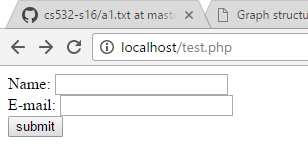
The next two figures ( Figure 4 and Figure 5 show the two php files used in a web browser while running on a local host.

Figure . Browser display of test.php

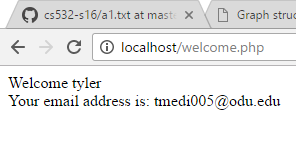


Figure . Browser display of welcome.php and specific form data

**2 Question 2**

**2.1** **Program libraries**

The 5 libraries used were BeautifulSoup, urllib2, requests, and sys. The sys library was chosen to import command line arguments to run the program. Requests was mainly for practice. I couldn’t utilize request correctly so I ended up just commenting out most of my attempts at requests but left the library intact in case I could figure out how to tie them in later. The urllib2 library was to get the info of the uri and BeautifulSoup was to find all the links on the uri.

**2.2 Explaining Logic and Functionality**

To open the uri, I used the urllib2open function to open the parameter from the command line argument. Immediately following the read() function was used to read the data. BeautifulSoup fetched all the links on the page. The find\_all(‘a’) function found all instances in the html code where the character ‘a’ was used to post a link. Once found, a simple get(‘href’) got all the links.

After all the links had been acquired, the pdf files had to be separated. To separate them, the header info needs to be retrieved to determine the content type. Again, urlopen is used to open the link. The info() function returns the metadata information. Once this information is retrieved, getheader(‘Content-Type’) gets the content type of the link. To print all the pdfs, and if statement is created that only runs if the link is of the “application/pdf” type. Once the pdf file is determined, getheader(‘content-legnth’) gets the size in bytes. All that is left to do is print out the link and the size.

**2.3 Results**

The three links used were :

1. <http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html>

2. <http://www.cs.odu.edu/~mln/teaching/>

3. http://www.cs.odu.edu/~mln/pubs/all.html

The results are displayed in figures 6,7, and 8:

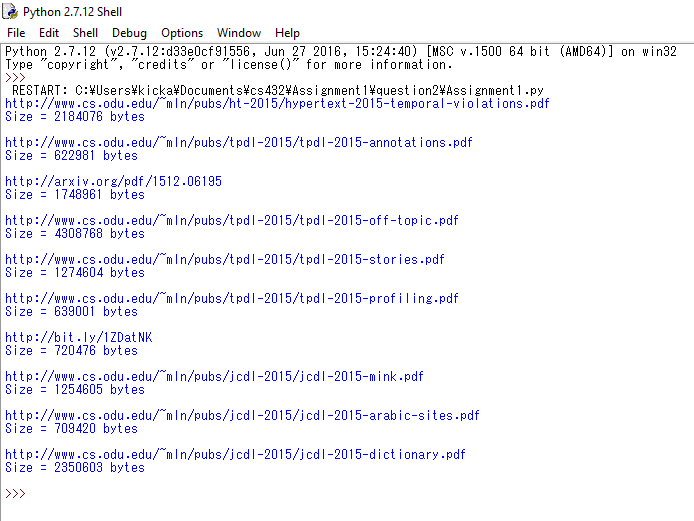


Figure . PDF files from link 1

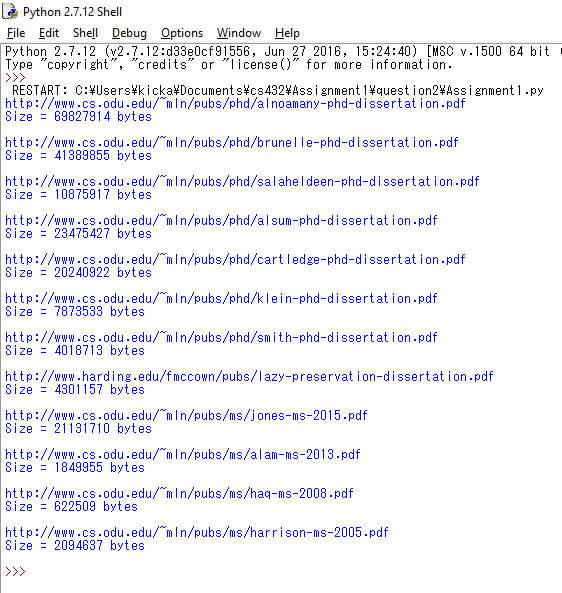


Figure . PDFS from link 2

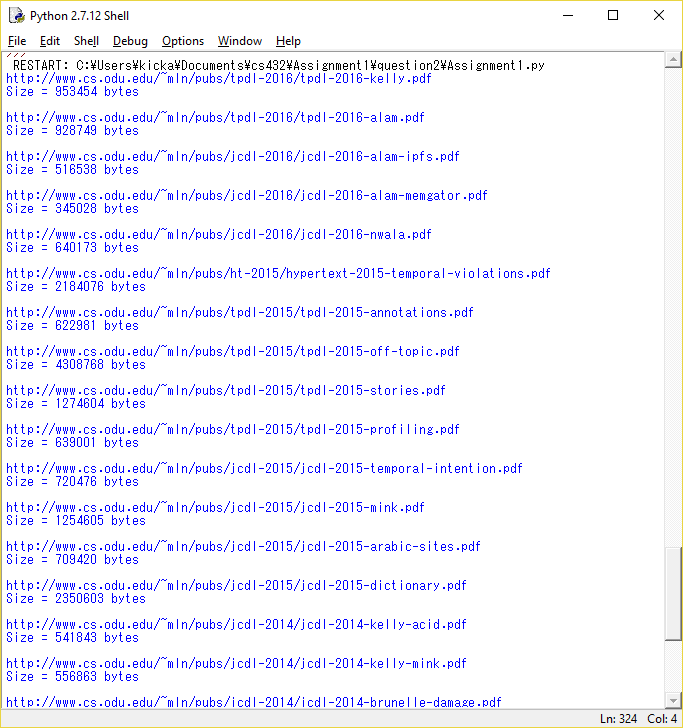


Figure . PDFs from link 3

**2.4 Issues**

I ran into a multitude of issues. The one of most annoyance was inconsistent indentation. I was unable to properly take in command arguments to run my program, so all the data is from me manually testing inside the Assignment1.py file. The original uri and the final uri could not be output because I couldn’t figure out how to follow the redirects.

**3 Question 3**

**3.1 Graph and Answers**

The graph (Figure 9) was handrawn and connected all the nodes that were given in the assignment. Connecting the data points I came up with the following answers:

IN: O, M, P

SCC: A, B, C, G

OUT: D, H

Tendrils: I, K, L

Tubes: N

Disconnected: F, E

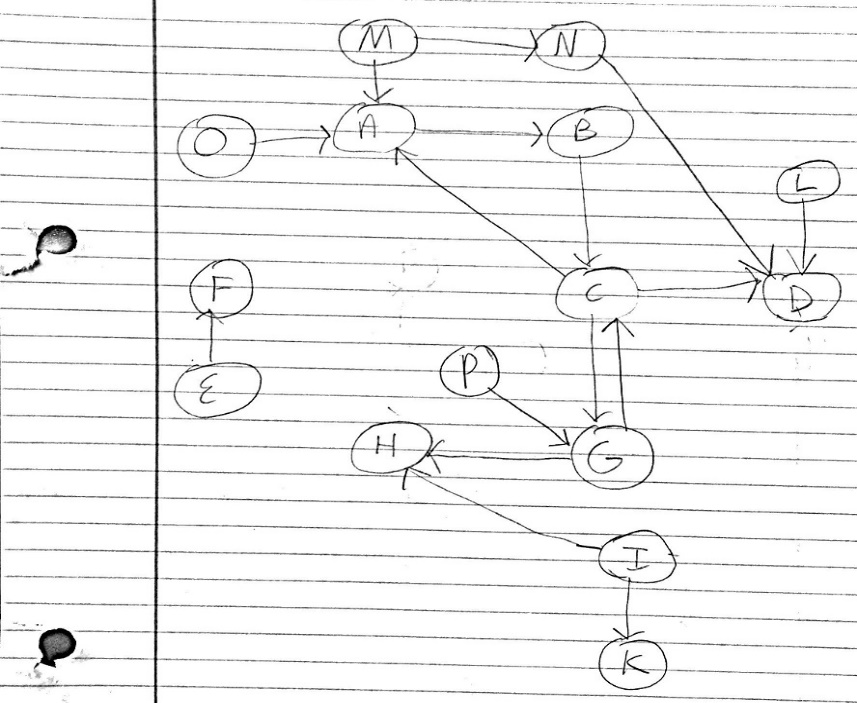
The letters O, M, and P are IN because they feed into the SCC nodes. Since A, B, C, and G have either a node connected that is IN or OUT they are SCC. Tendrils are the nodes that branch off of IN and OUT nodes, so I, K, and L are tendrils.

Figure . Bowtie graph

N is a tube because it serves as the bridge an IN (M) and an OUT (D) nodes. F and E are completely separated from all the other nodes which makes them disconnected.