

Assignment 1

Ryan Dill
CS 432
January 25, 2018

Question 1: Demonstrate that you know how to use "curl" well enough to correctly POST data to a form. Show that the HTML response that is returned is "correct". That is, the server should take the arguments you POSTed and build a response accordingly. Save the HTML response to a file and then view that file in a browser and take a screen shot.

```
ryan@ryan-VirtualBox: ~  
ryan@ryan-VirtualBox:~$ curl -I http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php  
HTTP/1.1 200 OK  
Server: nginx  
Date: Fri, 26 Jan 2018 02:38:05 GMT  
Content-Type: text/html  
Connection: keep-alive  
ryan@ryan-VirtualBox:~$ curl -i http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php  
HTTP/1.1 200 OK  
Server: nginx  
Date: Fri, 26 Jan 2018 02:38:30 GMT  
Content-Type: text/html  
Transfer-Encoding: chunked  
Connection: keep-alive  
Vary: Accept-Encoding  
<!DOCTYPE html>  
<html>  
<body>  
  
<br />  
<br />  
<b>fname Posted: </b><br />  
<b>lname Posted: </b><br />  
  
</body>  
</html>  
ryan@ryan-VirtualBox:~$ curl -d "fname=Ryan&lname=Dill" http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php  
^[[C<!DOCTYPE html>  
<html>  
<body>  
  
<br />  
<br />  
<b>fname Posted: </b>Ryan<br />  
<b>lname Posted: </b>Dill<br />  
  
</body>  
</html>  
ryan@ryan-VirtualBox:~$
```

Question 2: Write a Python program that:

1. takes as a command line argument a web page
2. extracts all the links from the page
3. lists all the links that result in PDF files, and prints out the bytes for each of the links. (note: be sure to follow all the redirects until the link terminates with a "200 OK".)
4. show that the program works on 3 different URIs, one of which needs to be:
<http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html>

The code used to extract links and find .pdf files

```
import sys
from bs4 import BeautifulSoup
import urllib.request
import requests
import re
if len(sys.argv) == 2:
    url = sys.argv[1]

else:
    print("Not enough arguments. Please retry.")
    sys.exit()

website = urllib.request.urlopen(url)

beauty = BeautifulSoup(website, "html.parser")

link_array = []

for link in beauty.findAll('a', attrs={'href': re.compile("^http")}):
    link_array.append(link.get('href'))

print("\nTotal URL's is ", len(link_array), "\n")
print(*link_array, sep='\n')

print("\n")

pdf_array = [s for s in link_array if ".pdf" in s]
print("The number of links which lead to .pdf files is: ", len(pdf_array), "\n")

print(*pdf_array, sep='\n')

print("\n")

for index, newurl in enumerate(pdf_array):
    response = requests.get(newurl)
    print(newurl, "is ", response.headers['content-length'], "bytes long")
```

Screen-shots from three separate websites:

```
ryan@ryan-VirtualBox: ~/Desktop
ryan@ryan-VirtualBox:~/Desktop$ python3 assignment1.py http://www.cs.odu.edu/~nln/teaching/cs532-s17/test/pdfs.html
Total URL's is 20
http://twitter.com/webscdl
http://www.dlib.org/dlib/november15/vandesompel/11vandesompel.html
http://arxiv.org/abs/1508.02315
http://arxiv.org/abs/1508.02315
http://www.cs.odu.edu/~nln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
http://arxiv.org/pdf/1512.06195
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-stories.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
http://dx.doi.org/10.1007/s00799-015-0150-6
http://www.cs.odu.edu/~nln/pubs/jcdl-2014/jcdl-2014-brunelle-danage.pdf
http://arxiv.org/abs/1506.06279
http://dx.doi.org/10.1007/s00799-015-0155-1
http://bit.ly/1ZDatNK
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-mink.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
http://bit.ly/jcdl-pdf
http://dx.doi.org/10.1007/s00799-015-0140-8

The number of links which lead to .pdf files is: 9
http://www.cs.odu.edu/~nln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-stories.pdf
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2014/jcdl-2014-brunelle-danage.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-mink.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf

http://www.cs.odu.edu/~nln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf is 2184076 bytes long
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-annotations.pdf is 622981 bytes long
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf is 4308768 bytes long
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-stories.pdf is 1274604 bytes long
http://www.cs.odu.edu/~nln/pubs/tpdl-2015/tpdl-2015-profiling.pdf is 639001 bytes long
http://www.cs.odu.edu/~nln/pubs/jcdl-2014/jcdl-2014-brunelle-danage.pdf is 2205546 bytes long
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-mink.pdf is 1254605 bytes long
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf is 709420 bytes long
http://www.cs.odu.edu/~nln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf is 2350603 bytes long
ryan@ryan-VirtualBox:~/Desktop$
ryan@ryan-VirtualBox:~/Desktop$
ryan@ryan-VirtualBox:~/Desktop$
ryan@ryan-VirtualBox:~/Desktop$
ryan@ryan-VirtualBox:~/Desktop$
ryan@ryan-VirtualBox:~/Desktop$
```

```
ryan@ryan-VirtualBox: ~/Desktop
urllib.error.HTTPError: HTTP Error 403: Forbidden
ryan@ryan-VirtualBox:~/Desktop$ python3 assignment1.py http://www.google.com
Total URL's is 11
http://www.google.com/inghp?hl=en&tab=wl
http://maps.google.com/maps?hl=en&tab=wl
https://play.google.com/?hl=en&tab=w8
http://www.youtube.com/?gl=US&tab=w1
http://news.google.com/nwshp?hl=en&tab=wn
https://mail.google.com/mail/?tab=wm
https://drive.google.com/?tab=wo
https://www.google.com/intl/en/options/
http://www.google.com/history/optout?hl=en
https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=http://www.google.com/
https://plus.google.com/116899029375914044550
The number of links which lead to .pdf files is: 0
```

```
ryan@ryan-VirtualBox: ~/Desktop
ryan@ryan-VirtualBox:~/Desktop$ python3 assignment1.py http://webnovel.com
Total URL's is 12
https://forum.webnovel.com/d/2593-2018-s-china-original-literature-billboard-top-10
https://www.webnovel.com/book/829959786003805/Cultivation-Chat-Group
https://forum.webnovel.com/d/2164-keep-running-out-of-spirit-stones-now-you-can-get-more
https://www.webnovel.com/book/8060642606003005/Full-Marks-Hidden-Marriage%3A-Pick-Up-a-Son%2C-Get-a-Free-Husband
https://forum.webnovel.com/d/1661-the-man-behind-the-scenes
https://www.webnovel.com/book/683865602002005
https://forum.webnovel.com/d/2573-about-facebook-login-issues
https://forum.webnovel.com/d/2573-about-facebook-login-issues
https://business.facebook.com/ilovewebnovel
https://twitter.com/QidianReading
https://plus.google.com/118428022004113337450
http://www.qidian.com
The number of links which lead to .pdf files is: 0
```

3. Consider the "bow-tie" graph in the Broder et al. paper (fig 9):
<http://www9.org/w9cdrom/160/160.html>

Now consider the following graph:

```
A --> B
B --> C
C --> D
C --> A
C --> G
E --> F
```

G --> C
G --> H
I --> H
I --> K
L --> D
M --> A
M --> N
N --> D
O --> A
P --> G

For the above graph, give the values for:

IN:
SCC:
OUT:
Tendrils:
Tubes:
Disconnected:

IN: I, P, O, M, N, L
SCC: A, B, C, G
OUT: H, D
TENDRILS: I \rightarrow K
TUBES: L \rightarrow D
DISCONNECTED: E \rightarrow F

