

1. While researching the how to handle large files of text I came across <http://stackoverflow.com/questions/1131220/get-md5-hash-of-big-files-in-python>. Initially I didn't know of the hashlib or md5, but upon researching those I came across <https://docs.python.org/2/library/hashlib.html> which further informed me of how to work with the library and its resources. I constructed a python scrip by merging my original draft to process the raw data(shown below) along with a block of code I found from the second link(also shown below).

Original code

```
1 import sys
2     import fileinput
3     import commands
4 import os
5
6 print(sys.version)
7
8 # RAW DATA
9 input = open('test.txt', 'r')
10 for line in input:
11     url = line
12     print(line)
13
14 counter_1 = 0
15
16 for line in input:
17
18     filename = str(counter_1)
19     print str(line)
20     curling = ' curl ' + '"' + str(line).rstrip('\n') + '"' + '> ./rawData/' + filename
21     print curling
22     output_1 = commands.getoutput(curling)
23
24 input.close()
```

Code from link

```
>>> import hashlib
>>> m = hashlib.md5()
>>> m.update("Nobody inspects")
>>> m.update(" the spammish repetition")
>>> m.digest()
'\xbbd\x9c\x83\xdd\x1e\xa5\xc9\xd9\xde\xc9\xa1\x8d\xf0\xff\xe9'
>>> m.digest_size
16
>>> m.block_size
64
```

Bryan Carey

CS432 Assignment3

The end result is-

```
1  import ...
5
6  input = open("URIfinal.txt", 'r')
7
8  for line in input:
9      url = line
10     url = url.replace('\n', '')
11
12     # referenced from http://stackoverflow.com/questions/1131220/get-md5-hash-of-big-files-in-python
13     def convert_to_md5(content):
14         m = hashlib.md5()
15         m.update(content)
16         return m.hexdigest()
17
18     output = convert_to_md5(url)
19     print output
20
21     # % wget -O www.cnn.com http: // www.cnn.com / provided on assignment prompt
22     os.system(" wget -O /bcarey/HTML" + output + ".txt " + url)
```

This document simply downloads the raw content of each URL.

I simply copied this file into a separate .py file, and altered the final line with the lynx command provided on the prompt as well to strip away everything except for the text file

```
% lynx -dump -force_html www.cnn.com > www.cnn.com.processed
```

Both files(page_content.py & text_only.py) were ran through Putty. It took roughly 30 minutes for both processes to be completed.

2. For question two I had to review grep, as I had not used it since CS252. I found tutorials on grep at http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_04_02.html & <http://www.computerhope.com/unix/ugrep.htm> . Once I did that it became fairly easy.

-l,
--files-with-matches

Instead of the normal output, print the name of each input file from which output would normally have been printed. The scanning will stop on the first match.

I used the previous command from computerhope.com and dumped all matches into a text file. My query term was "Cena" as in John Cena. I then used grep again to yield the individual word counts for Cena. I placed all files in my Top 10 folder, and grep -c for the ones with the most hits. I removed all other files from my Top10 folder, all that remains are the 10 pages with the highest occurrence of Cena and the Top10.txt with their names. This was done by-

Bryan Carey

CS432 Assignment3

```
atria.cs.odu.edu - PuTTY
login as: bcarey
bcarey@atria.cs.odu.edu's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Fri Feb 24 20:58:41 2017 from 172.18.4.146
atria:~> cd CS432
atria:~/CS432> cd Assignment3Boost
atria:~/CS432/Assignment3Boost> ls
CenaHits.txt      downloadText.py  HTML           ~$Report.docx   Top10
Cena.txt          getHashes.py    junk           Report.docx      URIfinal.txt
downloadPage.py  hashes.txt      PROCESSED      Thumbs.db        wordCount.py
atria:~/CS432/Assignment3Boost> grep -c "Cena" Top10/* > CenaHits.txt
atria:~/CS432/Assignment3Boost>
```

All generated .txt files can be found inside of my Q2 folder.

3. For part three I chose to use http://www.prchecker.info/check_page_rank.php

Web Page URL: <http://www.muscleandfitness.com/athletes-celebrities/news/john-cena-no-reason-bring-back-attitude-era>

The Page Rank: 0/10

(the page rank value is 0 from 10 possible points)

Web Page URL: <http://wwe.com>

The Page Rank: 6/10

(the page rank value is 6 from 10 possible points)

Web Page URL: <http://bleacherreport.com/wwe>

The Page Rank: 4/10

(the page rank value is 4 from 10 possible points)

Web Page URL: <http://www.skysports.com/wwe/news/14203/10760037/wwe-smackdown-john-cena-beats-randy-orton-thanks-to-luke-harper>

The Page Rank: 0/10

(the page rank value is 0 from 10 possible points)

Web Page URL: <http://www.cagesideseats.com/>

The Page Rank: 6/10

(the page rank value is 6 from 10 possible points)

Web Page URL: <http://www.wrestlinginc.com/wi/news/2015/0505/594002/triple-h-comments-on-sami-zayn-vs-john-cena/>

The Page Rank: 0/10

(the page rank value is 0 from 10 possible points)

Web Page URL: <http://www.foxsports.com/wwe/story/john-cena-promises-fans-isnt-ditching-wwe-hollywood-122716>

The Page Rank: 0/10

(the page rank value is 0 from 10 possible points)

Web Page URL: <http://hollywoodlife.com/2016/08/21/aj-styles-beats-john-cena-summerlam-2016-wwe-results/#!>

The Page Rank: 0/10

(the page rank value is 0 from 10 possible points)

Web Page URL: <http://shop.wwe.com/>

The Page Rank:  **5/10**

(the page rank value is **5** from 10 possible points)

Web Page URL: https://www.washingtonpost.com/news/early-lead/wp/2016/04/04/shaquille-oneal-john-cena-make-surprise-ring-appearances-at-wrestlemania-32/?utm_term=.9968e3103fce

The Page Rank:  **0/10**

(the page rank value is **0** from 10 possible points)

Web Page URL: <http://www.wrestlinginc.com/wi/news/wwe-news/>

The Page Rank:  **0/10**

(the page rank value is **0** from 10 possible points)