C996 Task Prompt Responses

Jesse Ashby

Western Governors University

C996 Task Prompt Responses

A. Explain how the Python program extracts the web links from the HTML code of the "Current Estimates," found in web links section.

This script uses the requests module for Python 3 originally written by Kennth Reitz (Reitz, 2019) to make an HTTP request to the website assigned. Python file operations are then used to write the document in the response to a plain text file called "responseHtml.html". Once the document has been created on the machine running the script, the BeautifulSoup module (Richardson, 2019) applies Python 3's default html parser to the document to parse the tags from the document.

In the HTML on this page, external web links are created with the href attribute of the "a" tag. The value of this attribute contains the address of the web link. Link extraction starts with an empty list called uriList. The BeautifulSoup find_all iterable method is then used to iterate over the document's "a" tags so that BeautifulSoup's get method can assign the value of the href attribute of each tag to the uri variable. In some "a" tags, there is no href attribute and the uri is assigned None. One such tag is found on line 740 of the responseHtml.html file created by the script. Other tags move around the document, but don't leave it. These will start with a '#' character. An example of this can be found on line 4,092. The script needs to examine the contents of the uri variable teach time it is assigned for these reasons. There were two types of valid web link addresses found in the document, absolute URLs that could be navigated to after removing the context of the site they were on and relative URLs that directed a visitor to another page within the same site and rely on the location of the page they are used in.

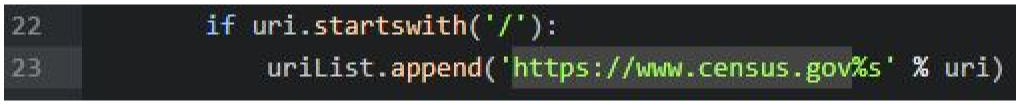
All of the absolute URLs begin with "http" and all of the relative URLs begin with "/". All valid weblink values of any href attribute found in an "a" tag on this page begin with either a "http" or "/". On each iteration, the uri variable is examined for its type first. If it is a string, then the content of that string is examined. If the string begins with "http" it is an absolute URL and written to the uriList list. If it begins with "/", then "https://www.census.gov" is added to the beginning of the string and that result is written to uriList. As a final measure, the list is iterated over to remove and trailing "/" characters found as these could potentially cause logical duplication that are not literal duplicate strings. For example, www.google.com is the same as www.google.com/.

B. Explain the criteria you used to determine if a link is a locator to another HTML page. Identify the code segment that executes this action as part of your explanation.

Once the final output was written to uriList.csv, I examined each of the links and found that each of them contained either .html or did not specify the file. For those that did not specify the file, I navigated to those links and saw that the doctype on the default page was indeed HTML.

C. Explain how the program ensures that relative links are saved as absolute URIs in the output file. Identify the code segment that executes this action as part of your explanation.

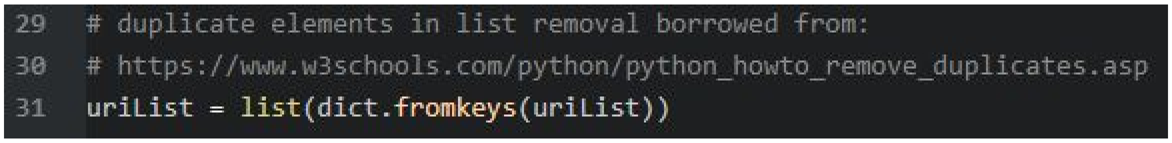
Relative links in this file begin with a "/". I did not see any relative links that were filename only. Any href attribute that began with a "/" was turned into an absolute URI by prefixing the string with "https://www.census.gov". I tested several of these independently and was able to load the proper page. Lines 17 - 23 in c996.py handle this, with the part that tests for the relative link and prefixing it on lines 22 and 23.



```
22          if uri.startswith('/'):
23              uriList.append('https://www.census.gov%s' % uri)
```

D. Explain how the program ensures that there are no duplicated links in the output file. Identify the code that executes this action as part of your explanation.

W3schools had an interesting method for this. Python dictionaries do not allow for duplicate keys and will remove those duplicates instead of halting the script with an exception (W3schools, 2019). That means it's possible to convert a list to a dictionary that uses the list values as keys, lose the duplicate keys (elements from the original list) and then convert the dictionary back into a list. As no keys are given values, only the old dictionary keys are included in the new list. Lines 29-31 handle this.



```
29  # duplicate elements in list removal borrowed from:
30  # https://www.w3schools.com/python/python_howto_remove_duplicates.asp
31  uriList = list(dict.fromkeys(uriList))
```

E. Provide the Python code you wrote to extract all the unique web links from the HTML code of the "Current Estimates" (in the web links section), that point out to other HTML pages.

Included as c996.py.

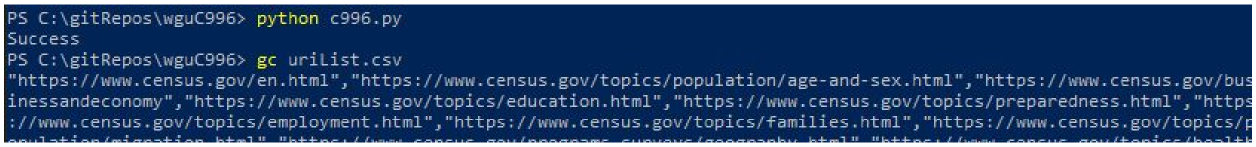
F. Provide the HTML code of the "Current Estimates" web page scrapped at the time when the scraper was run and the CSV file was generated.

Included as responseHtml.html.

G. Provide the CSV file that your script created.

Included as uriList.csv.

H. Run your script and provide a screenshot of the successfully executed results.



```
PS C:\gitRepos\wguC996> python c996.py
Success
PS C:\gitRepos\wguC996> gc uriList.csv
"https://www.census.gov/en.html","https://www.census.gov/topics/population/age-and-sex.html","https://www.census.gov/bus
inessandeconomy","https://www.census.gov/topics/education.html","https://www.census.gov/topics/preparedness.html","https
://www.census.gov/topics/employment.html","https://www.census.gov/topics/families.html","https://www.census.gov/topics/p
opulation/migration.html","https://www.census.gov/programs-surveys/geography.html","https://www.census.gov/topics/health
```

References

Reitz, Kenneth. (2019). Requests: HTTP for Humans. Retrieved from https://requests.kennethreitz.org/en/master/

Richardson, Leonard. (2019). Beautiful Soup. Retrieved from https://www.crummy.com/software/BeautifulSoup/

Sentdex. (2016). Web scraping and parsing with Beautiful Soup & Python Introduction p.1. *Pythonprogramming.net*. Retrieved from https://www.youtube.com/watch?v=aIPqt-OdmS0

W3schools. (2019). How to Remove Duplicates From a Python List. *Refsnes Data*. Retreived from https://www.w3schools.com/python/python_howto_remove_duplicates.asp