Webscraping with Python BeautifulSoup4

1. Install Anaconda3 in C:\Anaconda3 folder.
2. Open Windows command line.
3. To install BeautifulSoup, go to C:\Anaconda3\Scripts. Type,
   ```
   C:\Anaconda3\Scripts>pip.exe install beautifulsoup4
   ```
   You will get,
   ```
   Requirement already satisfied: beautifulsoup4 in
   c:\anaconda3\lib\site-packages
   ```
4. To know if BeautifulSoup is successfully installed, type,
   ```
   C:\Anaconda3\Scripts>python
   ```
   You will get,
   ```
   Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016,
   11:57:41) [MSC v.1
   900 64 bit (AMD64)] on win32
   Type "help", "copyright", "credits" or "license" for more
   information.
   ```
   Then type,
   ```
   >>> import bs4
   ```
5. To grab a html page, type,
   ```
   >>> from urllib.request import urlopen as uReq
   ```
6. To parse html tags, call BeautifulSoup by typing,
   ```
   >>> from bs4 import BeautifulSoup as soup
   ```
7. To define the html page's url, type
   ```
   >>> my_url = 'http://stackoverflow.com/questions/19957194/install-
   beautiful-soup-using-pip'
   ```
8. To check contents of my_url variable, type
   ```
   >>> my_url
   ```
   You will get,
   ```
   'http://stackoverflow.com/questions/19957194/install-beautiful-soup-
   using-pip'
   ```
9. To open a connection to the web page and downloading into our machine, type,
   ```
   >>> uClient = uReq(my_url)
   ```
10. To read the scraped contents, type,
    ```
    >>> page_html = uClient.read()
    ```
    Warning, do not view the contents at this point in time, because if the web page is huge, the command prompt will crash.
11. To close the connection, type,
    ```
    >>> uClient.close()
    ```
12. To parse the contents, type,
    ```
    >>> page_soup = soup(page_html,"html.parser")
    ```
13. To view the header of the contents, type,
    ```
    >>> page_soup.h1
    ```
    You will get,
    ```
    <h1 itemprop="name"><a class="question-hyperlink"
    href="/questions/19957194/inst
    all-beautiful-soup-using-pip">install beautiful soup using
    pip</a></h1>
    ```
14. To view any paragraph in the contents, type,
    ```
    >>> page_soup.p
    ```
    You will get,

```
<p>I am trying to install BeautifulSoup using <code>pip</code> in
Python 2.7. I keep getting an error message, and can't understand
why.</p>
```

15. To know what elements and tags that the webpage has, right mouse click on Chrome and choose Inspect.
16. To check tags in the content's <body>, type,
    ```
    >>> page_soup.body
    ```
    Again, not advisable unless the body is short.
17. To check what is in the , type,
    ```
    >>> page_soup.body.span
    ```
    You will get,
    ```
    <span class="-img">Stack Overflow</span>
    ```
18. To focus on a particular part of the webpage, simply highlight the text and right mouse click to choose Inspect.
19. Identify html class to be passed.
20. To parse a particular class into a variable, type,
    ```
    >>>containers = page_soup.findAll("div",{"class":"item-container"})
    ```
21. To check the length of a variable, type,
    ```
    >>> len(containers)
    ```
22. To read what is in the variable, type,
    ```
    >>> containers[0]
    ```
23. To grab title from the following,
    ```
    <img alt="EVGA" class="lazy-img" data-effect="blab la" title="EVGA"
    ```
    Type,
    ```
    >>> container.div.div.a.img["title"]
    ```

24. To run our py file,

    ```
    C:\ITS480 - Business Data Analytics\Notes\Webscraping>python
    my_webscraper.py
    ```

```
1  from urllib.request import urlopen as uReq
2  from bs4 import BeautifulSoup as soup
3
4  my_url = 'http://www.newegg.com/Video-Cards-Video-Devices/Category/ID-38?Tpk=grap
5
6  # opening up connection, grabbing the page
7  uClient = uReq(my_url)
8  page_html = uClient.read()
9  uClient.close()
10
11 # html parsing
12 page_soup = soup(page_html, "html.parser")
13
14 # grabs each product
15 containers = page_soup.findAll("div",{"class":"item-container"})
16
17 for container in containers:
18     brand = container.div.div.a.img["title"]
```

```
10
11 # html parsing
12 page_soup = soup(page_html, "html.parser")
13
14 # grabs each product
15 containers = page_soup.findAll("div",{"class":"item-container"})
16
17 for container in containers:
18     brand = container.div.div.a.img["title"]
19
20     title_container = container.findAll("a", {"class":"item-title"})
21     product_name = title_container[0].text
22
23     shipping_container = container.findAll("li", {"class":"price-ship"})
24     shipping = shipping_container[0].text.strip()
```

```
13
14 # grabs each product
15 containers = page_soup.findAll("div",{"class":"item-container"})
16
17 for container in containers:
18     brand = container.div.div.a.img["title"]
19
20     title_container = container.findAll("a", {"class":"item-title"})
21     product_name = title_container[0].text
22
23     shipping_container = container.findAll("li", {"class":"price-ship"})
24     shipping = shipping_container[0].text.strip()
25
26     print("brand: " + brand)
27     print("product_name: " + product_name)
28     print("shipping: " + shipping)
29
```

```
17 filename = "products.csv"
18 f = open(filename, "w")
19
20 headers = "brand, product_name, shipping\n"
21
22 f.write(headers)
23
24 for container in containers:
25     brand = container.div.div.a.img["title"]
26
27     title_container = container.findAll("a", {"class":"item-title"})
28     product_name = title_container[0].text
29
30     shipping_container = container.findAll("li", {"class":"price-ship"})
31     shipping = shipping_container[0].text.strip()
32
```

```
    brand = container.div.div.a.img["title"]

    title_container = container.findAll("a", {"class":"item-title"})
    product_name = title_container[0].text

    shipping_container = container.findAll("li", {"class":"price-ship"})
    shipping = shipping_container[0].text.strip()

    print("brand: " + brand)
    print("product_name: " + product_name)
    print("shipping: " + shipping)

    f.write(brand + "," + product_name.replace(",", "|") + "," + shipping + "\n")

f.close()
```