Daniel Molina

CS 3750 – Section 2

2/25/2020

Python Project – Web Scraping

**Source Code:**

# ===========================================================================

# This program looks for the cheapest prices on Ebay

# product target: Nvidia GeForce 2080 Ti GPU

# condition: Brand New

# price search: below user's specified amount

# ===========================================================================

from bs4 import BeautifulSoup

import requests

# url to be used from ebay search 'nvidia geforce rtx 2080 ti'

URL = 'https://www.ebay.com/sch/i.html?\_from=R40&\_trksid=p2334524.m570.l1313.TR8.TRC1.A0.H0.Xnvidia+geforce+rtx+2080+ti.TRS0&\_nkw=nvidia+geforce+rtx+2080+ti&\_sacat=0&LH\_TitleDesc=0&\_osacat=0&\_odkw=2080+ti'

# Google ‘my user agent’ to find your User-Agent

headers = {"User-Agent": 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36'}

page = requests.get(URL, headers=headers)

soup = BeautifulSoup(page.content, 'html.parser')

all\_items = soup.find\_all('li' ,class\_ = 's-item')

# all\_items: all the items listed on the webpage (50 total)

item\_prices = []

brand\_new\_items = []

for item in all\_items:

if 'Brand New' in item.get\_text(): # check if the product is brand new

instance = item.find(class\_='s-item\_\_price').get\_text()

if '$' in instance: # check if item is money or text string

money\_value = float(instance.replace(',','').replace('$',''))

# money\_value: convert prices to float

brand\_new\_items.append(item)

item\_prices.append(money\_value)

# ===========================================================================

# Now, Extract links for certain items cheaper than user's specified amount

# Present that information back to the user

# ===========================================================================

max\_price = input('What is the max price you are willing to pay for a Brand New \nNvidia Geforce RTX 2080 TI? ')

try:

max\_price = float(max\_price)

print('\n\nHere is what I found for prices lower than $%.2f:' % max\_price)

print('----------------------------------------------------------\n')

items\_found = False # if item found for price range

item\_links = [] # link for found items lower than max\_price

index = 0 # price index

counter = 0 # number of Options available

est\_price = 0 # sum of prices

# loop find links and prices of products lower than user's max\_price

for price in item\_prices:

if price <= max\_price:

est\_price += price

counter+=1

print('Option #%d, Price $%.2f, link:' % (counter,price))

link = brand\_new\_items[index].find(class\_='s-item\_\_link').get('href')

# extract link for item under $800

item\_links.append(link)

print(link, end='\n\n')

items\_found = True

index += 1

# end of for loop

if (not items\_found):

print('Sorry .... Unfortunately I couldn\'t find anything cheaper than that.')

print('Maybe come back for Black Friday?...')

else:

print('----------------------------------------------------------')

print('Total Number of Options: %d, Average price found for results: $%.2f\n' % (counter, (est\_price/counter)))

print('Copy and paste your favorite link on the web for more information')

print('----------------------------------------------------------\n')

except:

print('That is not a valid price. Try Again.')

**Results:**

Highlighted values are user input

***[Trial #1] – Input is not a number***

What is the max price you are willing to pay for a Brand New

Nvidia Geforce RTX 2080 TI? Nothing

That is not a valid price. Try Again.

***[Trial #2] – Input is a number. Returns product links***

What is the max price you are willing to pay for a Brand New

Nvidia Geforce RTX 2080 TI? 800.00

Here is what I found for prices lower than $800.00:

----------------------------------------------------------

Option #1, Price $670.00, link:

https://www.ebay.com/itm/GIGABYTE-AORUS-GeForce-RTX-2080-Ti-Xtreme-Graphic-Card/392693557010?epid=4030516917&hash=item5b6e5c2712:g:rIEAAOSwAPpeTRd1

Option #2, Price $755.00, link:

https://www.ebay.com/itm/NVIDIA-GeForce-RTX-2080-TI-Founders-Edition/352976861044?epid=9026714548&hash=item522f0f7774:g:Q8YAAOSwgZteTZiC

Option #3, Price $690.00, link:

https://www.ebay.com/itm/New-PNY-GeForce-RTX-2080-Ti-11GB-Blower-Graphics-Card-PN-VCG2080T11BLMPB/264634799851?epid=27034355060&hash=item3d9d7696eb:g:MhcAAOSwUvhdqJRv

Option #4, Price $700.00, link:

https://www.ebay.com/itm/ASUS-NVIDIA-GeForce-RTX-2080-Ti-11GB-GDDR6-Graphics-Card-DUALRTX2080TIO11G/143534897598?epid=8030648475&hash=item216b58b9be:g:jZsAAOSwjKBeSepX

Option #5, Price $26.65, link:

https://www.ebay.com/itm/EVGA-PowerLink-Support-All-NVIDIA-Founders-Edition-All-GeForce-RTX-2080-Ti-20/312904959229?hash=item48da96c4fd:g:eqMAAOSwnw5d~-Sq

----------------------------------------------------------

Total Number of Options: 5, Average price found for results: $568.33

Copy and paste your favorite link on the web for more information

----------------------------------------------------------

***[Trial #3] – Input is a number, but nothing is found. Price too low***

What is the max price you are willing to pay for a Brand New

Nvidia Geforce RTX 2080 TI? 10

Here is what I found for prices lower than $10.00:

----------------------------------------------------------

Sorry .... Unfortunately I couldn't find anything cheaper than that.

Maybe come back for Black Friday?...

**Explanation/Comments:**

The following program above extracts the html code from an Ebay webpage that I specified to look at in python3. We are looking at the first page of the product search ‘nvidia geforce rtx 2080 ti’ on Ebay. I installed ‘bs4’ and imported the BeautifulSoup library, which allows you to look for given id’s, key words, classes, and different things within the html code of the webpage to perform web scraping. Web scraping is the ability to extract different things from a web page on the internet for your use (use with responsibility). After extracting prices and links for the given product, I proceeded to find the best prices for Brand New items of that GeForce graphics card. Finally, the code outputs all the option items for your specified price range, provides you with the link to the item, and the average amount of the price you’ll have to pay for the graphics card. I really liked this simple project because it served as my introduction into web scraping. Further improvements may include the automation of such results by making a never-ending loop that pauses for 24 hours to then check price fluctuations and sends an email to your Inbox about price drops for given items. You may also take dynamic web links and search for the specified item to find the cheapest prices. There is an Application Programming Interface(API) called “Honey” which automatically finds the best coupon codes and cheapest prices for your benefit. Essentially, it was a small step into recreating that same API, but for my own personal use and practice.