Web Scraper Using Python (for static web pages)

OVERVIEW

A fundamental project that gives you a better understanding of working with Python. Creation of a book directory, where endpoints are used and creation of it using four basic methods: GET, POST, PUT, and DELETE. we are going to build a REST API to manage books with Node.js and Express. REST APIs use different HTTP request methods, corresponding to the previously mentioned actions, to retrieve and manipulate data. Here we are using JSON file for the data collection purpose.

Problem Statement

Scrap data of 100+ restaurants and their information along with their phone numbers and addresses using python in less than 40 lines of code and export it as a CSV file format.

Software Requirements

1. Programming Language : Python

2. Environment: Jupyter Notebooks / Google Collab

3. Database: CSV(export type)

4. Operation System: Windows XP or above

5. Librarires Used: Beautiful Soup4, URLlib, Pandas

Creating the Scraper

1. Importing the library

- BeautifulSoup4 - for getting data out of HTML,XML documents
- urllib -for opening and reading urls
- pandas - for data analysis

```
import bs4 as bs
import urllib.request as url_x
import pandas as pd
```

2. Declaring Required Variables & Taking input of State Name

- Here we are declaring a list for every attribute of data you want to scrape from the web
- We use the input function to take the state name as an input from the user

```
BusinessNames=[]
Phone=[]
Address=[]
Urls=[]
state_name = input('Enter State name here:')
print('Process Ignited')
```

3. Declaring URL & post forwarding a variable

- Here we store the url of the website in a variable

- Initialize another variable and store the url along with the concatenated next string (next page url)

```
url='https://www.yelp.com/search?find_desc=Restaurants&find_near=
alabama-state-capitol-montgomery'

urlsource=''+url+'&next='
```

4. Main Function Process – Attaching Classes to Declared Variables

- First we declare the number of pages and then iterate using a for loop and fetch the url’s , using the urlopen method.
- Using BeautifulSoup4 module we initialize the HTML parser , the parser create parse tree through which you can access data from web
- We initialize another variable called mains in which we store the class name of the element that contains all the attributes that we require . (we use find_all method)
- In the try block we initialize a variable for each attribute using the find method we find the list of attributes through its class name and append this list of names to the respective list variables declared before.

```
no_of_pages=5
for iteration in range(no_of_pages):
  s=iteration*10
  if(s==0):
    s=1
  source = url_x.urlopen(urlsource+str(s))
  print(urlsource+str(s))

  page_soup = bs.BeautifulSoup(source, 'html.parser')
  mains = page_soup.find_all("div", {"class": " scrollablePhotos__09f24__1PpB8 arrange__09f24__AiSIM border-color--default__09f24__R1nRO"})
  for main in mains:
      try:
          busname = main.find("a", {"class" : " link__09f24__1kwXV link-color--inherit__09f24__3PYlA link-size--inherit__09f24__2Uj95"}).text
          BusinessNames.append(busname)
          pnumber = main.find("p", {"class" : " text__09f24__2tZKC text-color--black-extra-light__09f24__38DtK text-align--right__09f24__1TIxB text-size--small__09f24__1Z_UI"}).text
          Phone.append(pnumber)
          address = main.find("span", {"class" : " raw__09f24__3Obuy"}).text
          Address.append(address)
          url = main.find("a", {"class" : " link__09f24__1kwXV link-color--inherit__09f24__3PYlA link-size--inherit__09f24__2Uj95"})['href']
          Urls.append("yelp.com" + url)
      except:
          print(None)
  print('Loading......')
print('Done with processing')
```

5. Combining various variables into a single dictionary & data framing the Dictionary using Pandas
   - Here we convert the lists that contain the data into a dictionary structure
   - Now this dictionary is converted into data frame using pandas module

```
dictionary = {'BusinessNames': BusinessNames, 'Address': Address, 'State': state_name, 'Phone': Phone,  'Urls': Urls}

df=pd.DataFrame(dict([(k,pd.Series(v)) for k,v in dictionary.items()]))
```

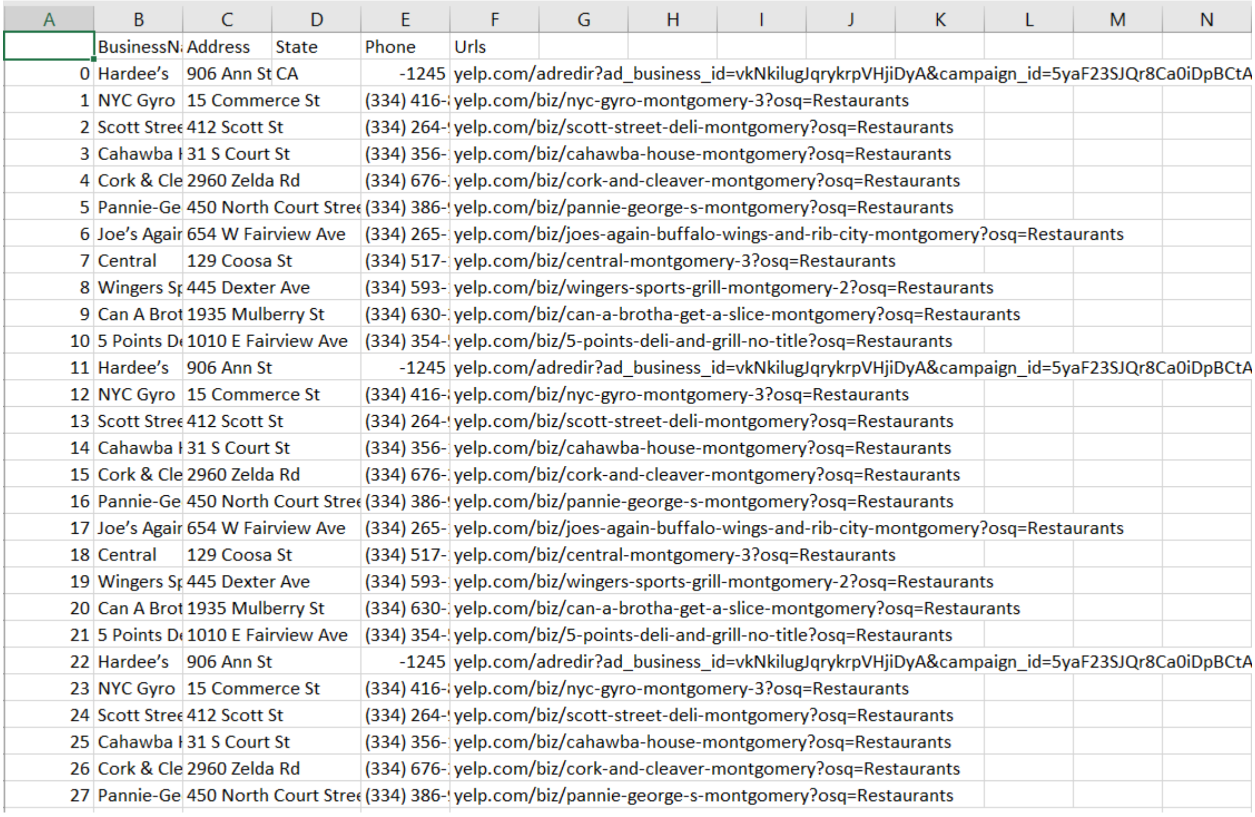
6. Converting the Data frames into CSV File
   - This is the final step here we convert data frame into CSV format

```
df.to_csv(''+state_name+'.csv',encoding='utf-8-sig')
print('saved as a file')
```

7. Downloading The CSV file from Google Collab

```
from google.colab import files
files.download(''+state_name+'.csv')
```

A Glimpse of the CSV File



A	B	C	D	E	F	G	H	I	J	K	L	M	N
	BusinessN	Address	State	Phone	Urls								
0	Hardee's	906 Ann St	CA	-1245	yelp.com/adredir?ad_business_id=vkNkilugJqrykrpVHjiDyA&campaign_id=5yaF23SJQr8Ca0iDpBCtA								
1	NYC Gyro	15 Commerce St		(334) 416-	yelp.com/biz/nyc-gyro-montgomery-3?osq=Restaurants								
2	Scott Stree	412 Scott St		(334) 264-	yelp.com/biz/scott-street-deli-montgomery?osq=Restaurants								
3	Cahawba H	31 S Court St		(334) 356-	yelp.com/biz/cahawba-house-montgomery?osq=Restaurants								
4	Cork & Cle	2960 Zelda Rd		(334) 676-	yelp.com/biz/cork-and-cleaver-montgomery?osq=Restaurants								
5	Pannie-Ge	450 North Court Stree		(334) 386-	yelp.com/biz/pannie-george-s-montgomery?osq=Restaurants								
6	Joe's Agair	654 W Fairview Ave		(334) 265-	yelp.com/biz/joes-again-buffalo-wings-and-rib-city-montgomery?osq=Restaurants								
7	Central	129 Coosa St		(334) 517-	yelp.com/biz/central-montgomery-3?osq=Restaurants								
8	Wingers Sp	445 Dexter Ave		(334) 593-	yelp.com/biz/wingers-sports-grill-montgomery-2?osq=Restaurants								
9	Can A Brot	1935 Mulberry St		(334) 630-	yelp.com/biz/can-a-brotha-get-a-slice-montgomery?osq=Restaurants								
10	5 Points De	1010 E Fairview Ave		(334) 354-	yelp.com/biz/5-points-deli-and-grill-no-title?osq=Restaurants								
11	Hardee's	906 Ann St		-1245	yelp.com/adredir?ad_business_id=vkNkilugJqrykrpVHjiDyA&campaign_id=5yaF23SJQr8Ca0iDpBCtA								
12	NYC Gyro	15 Commerce St		(334) 416-	yelp.com/biz/nyc-gyro-montgomery-3?osq=Restaurants								
13	Scott Stree	412 Scott St		(334) 264-	yelp.com/biz/scott-street-deli-montgomery?osq=Restaurants								
14	Cahawba H	31 S Court St		(334) 356-	yelp.com/biz/cahawba-house-montgomery?osq=Restaurants								
15	Cork & Cle	2960 Zelda Rd		(334) 676-	yelp.com/biz/cork-and-cleaver-montgomery?osq=Restaurants								
16	Pannie-Ge	450 North Court Stree		(334) 386-	yelp.com/biz/pannie-george-s-montgomery?osq=Restaurants								
17	Joe's Agair	654 W Fairview Ave		(334) 265-	yelp.com/biz/joes-again-buffalo-wings-and-rib-city-montgomery?osq=Restaurants								
18	Central	129 Coosa St		(334) 517-	yelp.com/biz/central-montgomery-3?osq=Restaurants								
19	Wingers Sp	445 Dexter Ave		(334) 593-	yelp.com/biz/wingers-sports-grill-montgomery-2?osq=Restaurants								
20	Can A Brot	1935 Mulberry St		(334) 630-	yelp.com/biz/can-a-brotha-get-a-slice-montgomery?osq=Restaurants								
21	5 Points De	1010 E Fairview Ave		(334) 354-	yelp.com/biz/5-points-deli-and-grill-no-title?osq=Restaurants								
22	Hardee's	906 Ann St		-1245	yelp.com/adredir?ad_business_id=vkNkilugJqrykrpVHjiDyA&campaign_id=5yaF23SJQr8Ca0iDpBCtA								
23	NYC Gyro	15 Commerce St		(334) 416-	yelp.com/biz/nyc-gyro-montgomery-3?osq=Restaurants								
24	Scott Stree	412 Scott St		(334) 264-	yelp.com/biz/scott-street-deli-montgomery?osq=Restaurants								
25	Cahawba H	31 S Court St		(334) 356-	yelp.com/biz/cahawba-house-montgomery?osq=Restaurants								
26	Cork & Cle	2960 Zelda Rd		(334) 676-	yelp.com/biz/cork-and-cleaver-montgomery?osq=Restaurants								
27	Pannie-Ge	450 North Court Stree		(334) 386-	yelp.com/biz/pannie-george-s-montgomery?osq=Restaurants								

Conclusion

Therefore, we have successfully scraped the Data of 100+ restaurants along with their mobile numbers, addresses & URLs using Python