Web Scraping (Yelp.ca)

Task

```
Task is to scrape reviews from yelp.com. You chose the topic (search term) for the reviews, that is, whether
it should be
reviews of for example restaurants, plumbers, nighlife or hair salons. The reviews should be written to a CSV
file.
```

```
I PICKED fALAFRL RESTURANTS (TOP) IN Toronto
Steps for the scraping :

 I import libraries
Then I import webpage of the falafel resturant in Toronto
Then I scraped all ten url of the ten resturant
Then I scraped names of the resturant
Then I scraped reviews,published dates of the review and rating
Then I converted the all four varirables(Name,PublishedDate,Review,rating) in One CSV File that is
Yelp_falafel.csv
```

Libraries and URL

In [22]:

```
#import libraries
import requests
#beautiful soup will help to scrape from website
from bs4 import BeautifulSoup
import csv
import sys
import numpy as np
import pandas as pd
import re

#Falahel resturants in toronto
html1 =
   requests.get("https://www.yelp.com/search?find_desc=Falafel&find_loc=Toronto%2C%20ON&sortby=rating")
falafel  = BeautifulSoup(html1.text, 'html.parser')
print(falafel)
```

```
<!DOCTYPE HTML>

<!--[if lt IE 7 ]> <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie6 ie ltie9 ltie8 no-js" l
ang="en"> <![endif]-->
<!--[if IE 7 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie7 ie ltie9 ltie8 no-js" l
ang="en"> <![endif]-->
<!--[if IE 8 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie8 ie ltie9 no-js" lang="e
n"> <![endif]-->
<!--[if IE 9 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie9 ie no-js" lang="en"> <!
[endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html class="no-js" lang="en" xmlns:fb="http://www.facebook.com/2008/fbm
l"> <!--<![endif]-->
<head>
<script nonce="a63606da">
            (function() {
                var main = null;

                var main=function(){window.onerror=function(k,a,c,i,f){var j=(document.getElementsByTagN
ame("html")[0].getAttribute("webdriver")==="true"||navigator.userAgent==="selenium");var h=f&&(f.name==
```

In [23]:
```
#Scraping All ten url


tops =
falafel.find_all
                ('a', attrs=
                            {'class' : "lemon--a__373c0__IEZFH link__373c0__1G70M
                            link-color--inherit__373c0__3dzpk link-size--inherit__373c0__1VFlE"})

toptens = tops[1:11]

links = []

for i in toptens:
  links.append("http://yelp.com" + i.get('href'))
links
```

Out[23]:
```
['http://yelp.com/biz/mystic-muffin-toronto?osq=Falafel',
 'http://yelp.com/biz/viva-shawarma-toronto-2?osq=Falafel',
 'http://yelp.com/biz/shawarma-frenzy-east-york?osq=Falafel',
 'http://yelp.com/biz/zezafoun-syrian-cuisine-toronto?osq=Falafel',
 'http://yelp.com/biz/arabesque-middle-eastern-foods-toronto?osq=Falafel',
 'http://yelp.com/biz/shawarma-empire-scarborough?osq=Falafel',
 'http://yelp.com/biz/salad-house-toronto?osq=Falafel',
 'http://yelp.com/biz/rikkochez-toronto?osq=Falafel',
 'http://yelp.com/biz/figs-and-olives-kitchen-toronto-2?osq=Falafel',
 'http://yelp.com/biz/papyrus-toronto-4?osq=Falafel']
```

Collecting the information of the names of the resturant.

In [24]:
```
#Scraping names of the resturants
Restaurant_Names = []
for i in toptens:
    Restaurant_Names.append(i.get('name'))
Restaurant_Names
```

Out[24]:
```
['Mystic Muffin',
 'Viva Shawarma',
 'Shawarma Frenzy',
 'Zezafoun Syrian Cuisine',
 'Arabesque Middle Eastern Foods',
 'Shawarma Empire',
 'Salad House',
 'Rikkochez',
 'Figs and Olives Kitchen',
 'Papyrus']
```

Collection of information of Reviews,PublishedDates and Ratings and append the resturant names and run for loop

In [26]:

```
#### Getting CSV ready and adding headers

header = ["Name","RatingValue","DatePublished","Review"]
with open("Yelp_Falafel.csv", "w", newline="") as WorkingFile:
    Data_to_WorkingFile =  csv.writer(WorkingFile, delimiter=",")
    Data_to_WorkingFile.writerow(header)


#### getting data and populating CSV file

Rest_Num = 0
AllData = []
for each_link in links:

    # scraping the reviews ,published dates and rating of the all ten url
    print(f"Getting from Restaurant {Restaurant_Names[Rest_Num]}: ", each_link)
    restaruants_f = requests.get(each_link)
    restaruant1_f = BeautifulSoup(restaruants_f.text, 'html.parser')
    ratings = restaruant1_f.find_all('div', attrs = {'role' : 'img'})


    review =

    restaruant1_f.find_all
    ("p", attrs =
             {"class":"lemon--p__373c0__3Qnnj text__373c0__2Kxyz
              comment__373c0__3EKjH text-color--normal__373c0__3xep9 text-align--left__373c0__2XGa-"})


    dates =
    restaruant1_f.find_all
    ('span', attrs =
             {'class' : 'lemon--span__373c0__3997G text__373c0__2Kxyz text-color--mid__373c0__jCeOG
              text-align--left__373c0__2XGa-'})




     # RatingValue
    ratings = ratings[1:11]
    rating = []
    for i in ratings:
```

```
        rating.append(i.get('aria-label'))
    score = []
    for i in range(len(rating)):
        temp = re.findall(r'\d+', rating[i])
        for i in temp:
            score.append(i)




    # 10 reviews of each Resturant

    toptenreviews = review[0:10]
    #toptenreviews
    reviewlist = []
    for i in toptenreviews:
        reviewlist.append(i.get_text())

    # Dates for reviews
    dates = dates[:10]
    date = []
    for i in dates:
        date.append(i.string)




    #compilation of all result in csv
    RestaurantData= {
        "Name": Restaurant_Names[Rest_Num],
        "RatingValue": score,
        "DatePublished": date,
        "Review":reviewlist
    }

    AllData.append(RestaurantData)
    dataframe = pd.DataFrame(RestaurantData)
    dataframe.to_csv("Yelp_Falafel.csv", index= False, mode= "a", header = False)
    Rest_Num += 1
```

Getting from Restaurant Mystic Muffin: http://yelp.com/biz/mystic-muffin-toronto?osq=Falafel (http://yelp.com/biz/mystic-muffin-toronto?osq=Falafel)

```
Getting from Restaurant Viva Shawarma:  http://yelp.com/biz/viva-shawarma-toronto-2?osq=Falafel (http://yelp.com/biz/viva-shawarma-toronto-2?osq=Falafel)
Getting from Restaurant Shawarma Frenzy:  http://yelp.com/biz/shawarma-frenzy-east-york?osq=Falafel (http://yelp.com/biz/shawarma-frenzy-east-york?osq=Falafel)
Getting from Restaurant Zezafoun Syrian Cuisine:  http://yelp.com/biz/zezafoun-syrian-cuisine-toronto?osq=Falafel (http://yelp.com/biz/zezafoun-syrian-cuisine-toronto?osq=Falafel)
Getting from Restaurant Arabesque Middle Eastern Foods:  http://yelp.com/biz/arabesque-middle-eastern-foods-toronto?osq=Falafel (http://yelp.com/biz/arabesque-middle-eastern-foods-toronto?osq=Falafel)
Getting from Restaurant Shawarma Empire:  http://yelp.com/biz/shawarma-empire-scarborough?osq=Falafel (http://yelp.com/biz/shawarma-empire-scarborough?osq=Falafel)
Getting from Restaurant Salad House:  http://yelp.com/biz/salad-house-toronto?osq=Falafel (http://yelp.com/biz/salad-house-toronto?osq=Falafel)
Getting from Restaurant Rikkochez:  http://yelp.com/biz/rikkochez-toronto?osq=Falafel (http://yelp.com/biz/rikkochez-toronto?osq=Falafel)
Getting from Restaurant Figs and Olives Kitchen:  http://yelp.com/biz/figs-and-olives-kitchen-toronto-2?osq=Falafel (http://yelp.com/biz/figs-and-olives-kitchen-toronto-2?osq=Falafel)
Getting from Restaurant Papyrus:  http://yelp.com/biz/papyrus-toronto-4?osq=Falafel (http://yelp.com/biz/papyrus-toronto-4?osq=Falafel)
```

Type *Markdown* and LaTeX: α^2

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: