WEB SCRAPING ASSIGNMENT

NAME – SUBHAM

BATCH – DS2401

1. **Write a python program to display all the header tags from wikipedia.org. Write a python program to display all the header tags from wikipedia.org.**

Ans)

```
In [1]: !pip install bs4
        !pip install requests
```

```
Collecting bs4
  Obtaining dependency information for bs4 from https://files.pythonhosted.org/packages/51/bb/bf7aab772a159614954d84aa832c12962
4ba6c32faa559dfb200a534e50b/bs4-0.0.2-py2.py3-none-any.whl.metadata
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in c:\users\dell\anaconda3\lib\site-packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\dell\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.4)
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dell\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests) (2023.7.22)
```

1) install requests and bs4 library

```
In [2]: from bs4 import BeautifulSoup
        import requests
```

```
In [3]: page=requests.get('https://en.wikipedia.org/wiki/Main_Page')
        page
```

```
Out[3]: <Response [200]>
```

2) import Beautiful Soup and request library

3) sending the request to the server and storing it in a variable and the response is [200] meaning we can scrape it's data

```
In [4]: soup=BeautifulSoup(page.content)
        soup
```

```
Out[4]: <!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vecto
r-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feat
ure-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-f
eature-custom-font-size-clientpref-0 vector-feature-client-preferences-disabled vector-feature-client-prefs-pinned-disabled v
ector-feature-night-mode-disabled skin-night-mode-clientpref-0 vector-toc-not-available" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>Wikipedia, the free encyclopedia</title>
<script>(function(){var className="client-js vector-feature-language-in-header-enabled vector-feature-language-in-main-page-h
eader-disabled vector-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clie
ntpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-cont
ent-enabled vector-feature-custom-font-size-clientpref-0 vector-feature-client-preferences-disabled vector-feature-client-pre
fs-pinned-disabled vector-feature-night-mode-disabled skin-night-mode-clientpref-0 vector-toc-not-available";var cookie=docum
ent.cookie.match(/(?:^|; )enwikimwclientpreferences=([^;]+)/);if(cookie){cookie[1].split('%2C').forEach(function(pref){classN
ame=className.replace(new RegExp('(^| )'+pref.replace(/-clientpref-\w+$|[^\w-]+/g,'')+'-clientpref-\\w+( |$)'),'$1'+pref
+'$2');});}document.documentElement.className=className;}());RLCONF={"wgBreakFrames":false,"wgSeparatorTransformTable":
["",""],
"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","Ju
```

4) downloading the page content and storing it in a variable

5) check the headers by visiting website and right click on the header and click on inspect -> click on class and copy it (ctrl + c) -> filter (ctrl + f) -> paste (ctrl + v) -> put brackets [] at beginning and end

Since it comes in header tags (<h> </h>) these are headers

There are 9 headings here

```
In [8]: headers=[]
        for i in soup.find_all('span',class_="mw-headline"):
            headers.append(i.text)

        headers
```

```
Out[8]: ['Welcome to Wikipedia',
         "From today's featured article",
         'Did you know\xa0...',
         'In the news',
         'On this day',
         "Today's featured picture",
         'Other areas of Wikipedia',
         "Wikipedia's sister projects",
         'Wikipedia languages']
```

6) Using for loop and find_all function we can scrap all the 9 headings.

2. **Write a python program to display IMDB's Top rated 100 movies' data (i.e. name, rating, year of release) and make data frame.**

Ans) IMDB site request response is 403 means forbidden. Meaning we cannot scrape from this site

```
In [11]: IMDB=requests.get('https://www.imdb.com/chart/top/')
         IMDB

Out[11]: <Response [403]>

In [12]: IMDB1=BeautifulSoup(IMDB.content)
         IMDB1

Out[12]: <html>
         <head><title>403 Forbidden</title></head>
         <body>
         <center><h1>403 Forbidden</h1></center>
         </body>
         </html>

In [14]: names=[]
         for i in IMDB1.find_all('h3',class_="ipc-title__text"):
                 names.append(i.text)

         names

Out[14]: []
```

3. **Write a python program to scrape mentioned details from dineout.co.in : i) Restaurant name ii) Cuisine iii) Location iv) Ratings v) Image URL.**

Ans)

```
In [15]: dine=requests.get('https://www.dineout.co.in/delhi-restaurants/buffet-special')
         dine

Out[15]: <Response [200]>

In [17]: dine1=BeautifulSoup(dine.content)
         dine1
```

```
Out[17]: <!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8"/><meta content="IE=edge" http-equiv="X-UA-Compatible"/><meta content="width=devic
e-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport"/><link href="/manifest.json" rel="manifest"/
><style type="text/css">
            @font-face {
                font-family: 'dineicon';
                src:  url('/fonts/dineicon.eot');
                src:  url('/fonts/dineicon.eot#iefix') format('embedded-opentype'),
                url('/fonts/dineicon.ttf') format('truetype'),
                url('/fonts/dineicon.woff') format('woff'),
                url('/fonts/dineicon.svg#dineicon') format('svg');
                font-weight: normal;
                                font-style: normal;
                                font-display: swap;
            }
            .hide {
                display: none !important;
            }
            .async-hide{
```

```
In [18]: names=[]
         for i in dine1.find_all('div',class_="restnt-info cursor"):
             names.append(i.text)

         names

Out[18]: ['Castle BarbequeConnaught Place, Central Delhi',
          'Cafe KnoshThe Leela Ambience Convention Hotel,Shahdara, East Delhi',
          'India GrillHilton Garden Inn,Saket, South Delhi',
          'The Barbeque CompanyGardens Galleria,Sector 38A, Noida',
          'Delhi BarbequeTaurus Sarovar Portico,Mahipalpur, South Delhi',
          'The Monarch - Bar Be Que VillageIndirapuram Habitat Centre,Indirapuram, Ghaziabad',
          'The Barbeque TimesM2K Corporate Park,Sector 51, Gurgaon']

In [19]: location=[]
         for i in dine1.find_all('div',class_="restnt-loc ellipsis"):
             location.append(i.text)

         location

Out[19]: ['Connaught Place, Central Delhi',
          'The Leela Ambience Convention Hotel,Shahdara, East Delhi',
          'Hilton Garden Inn,Saket, South Delhi',
          'Gardens Galleria,Sector 38A, Noida',
          'Taurus Sarovar Portico,Mahipalpur, South Delhi',
          'Indirapuram Habitat Centre,Indirapuram, Ghaziabad',
          'M2K Corporate Park,Sector 51, Gurgaon']
```

In [22]:
```
ratings=[]
for i in dine1.find_all('div',class_="restnt-rating rating-4"):
    ratings.append(i.text)

ratings
```

Out[22]:
```
['4', '4.3', '3.9', '3.9', '3.7', '3.8', '4.1']
```

In [28]:
```
images=[]
for i in dine1.find_all('img',class_="no-img"):
    images.append(i['data-src'])
images
```

Out[28]:
```
['https://im1.dineout.co.in/images/uploads/restaurant/sharpen/8/k/b/p86792-16062953735fbe1f4d3fb7e.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/4/p/m/p406-15438184745c04ccea491bc.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/2/q/t/p2687-169589385765154961ea87c.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/7/p/k/p79307-16051787755fad1597f2bf9.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/5/d/i/p52501-1661855212630de5eceb6d2.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/3/n/o/p34822-15599107305cfa594a13c24.jpg?tr=tr:n-medium',
 'https://im1.dineout.co.in/images/uploads/restaurant/sharpen/1/u/r/p106428-166073786162fcd945925a9.jpg?tr=tr:n-medium']
```

In [30]:
```
cuisine=[]
for i in dine1.find_all('span',class_="double-line-ellipsis"):
    cuisine.append(i.text)
cuisine
```

Out[30]:
```
['₹ 2,000 for 2 (approx) | Chinese, North Indian',
 '₹ 3,000 for 2 (approx) | Italian, Continental',
 '₹ 2,400 for 2 (approx) | North Indian, Italian',
 '₹ 1,700 for 2 (approx) | North Indian, Chinese',
 '₹ 1,800 for 2 (approx) | North Indian',
 '₹ 1,900 for 2 (approx) | North Indian',
 '₹ 1,500 for 2 (approx) | North Indian, Continental, Chinese, South Indian']
```

In [31]:
```
import pandas as pd
df=pd.DataFrame({'resturant name':names,'price and cuisine':cuisine,'location':location,'ratings':ratings,'image url':images})
df
```

Out[31]:

	resturant name	price and cuisine	location	ratings	image url
0	Castle BarbequeConnaught Place, Central Delhi	₹ 2,000 for 2 (approx) | Chinese, North Indian	Connaught Place, Central Delhi	4	https://im1.dineout.co.in/images/uploads/resta...
1	Cafe KnoshThe Leela Ambience Convention Hotel,...	₹ 3,000 for 2 (approx) | Italian, Continental	The Leela Ambience Convention Hotel,Shahdara, ...	4.3	https://im1.dineout.co.in/images/uploads/resta...
2	India GrillHilton Garden Inn,Saket, South Delhi	₹ 2,400 for 2 (approx) | North Indian, Italian	Hilton Garden Inn,Saket, South Delhi	3.9	https://im1.dineout.co.in/images/uploads/resta...
3	The Barbeque CompanyGardens Galleria,Sector 38...	₹ 1,700 for 2 (approx) | North Indian, Chinese	Gardens Galleria,Sector 38A, Noida	3.9	https://im1.dineout.co.in/images/uploads/resta...
4	Delhi BarbequeTaurus Sarovar Portico,Mahipalpu...	₹ 1,800 for 2 (approx) | North Indian	Taurus Sarovar Portico,Mahipalpur, South Delhi	3.7	https://im1.dineout.co.in/images/uploads/resta...
5	The Monarch - Bar Be Que VillageIndirapuram Ha...	₹ 1,900 for 2 (approx) | North Indian	Indirapuram Habitat Centre,Indirapuram, Ghaziabad	3.8	https://im1.dineout.co.in/images/uploads/resta...
6	The Barbeque TimesM2K Corporate Park,Sector 51...	₹ 1,500 for 2 (approx) | North Indian, Contine...	M2K Corporate Park,Sector 51, Gurgaon	4.1	https://im1.dineout.co.in/images/uploads/resta...

4. **Write a python program to display list of respected former finance minister of India(i.e. Name , Term of office) from https://presidentofindia.nic.in/former-presidents.htm and make data frame.**

Ans)

```
In [32]: poi=requests.get('https://presidentofindia.nic.in/former-presidents')
         poi

Out[32]: <Response [200]>
```

```
In [33]: poi1=BeautifulSoup(poi.content)
         poi1
```

```
Out[33]: <!DOCTYPE html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<noscript><meta content="0; URL=/big_pipe/no-js?destination=/former-presidents" http-equiv="Refresh"/>
</noscript><meta content="Former Presidents of India - | President of India" name="description"/>
<meta content="President of India | Former Presidents of India" name="keywords"/>
<link href="http://presidentofindia.nic.in/former-presidents" rel="canonical"/>
<link href="/manifest.json" rel="manifest"/>
<meta content="" name="theme-color"/>
<meta content="Drupal 9 (https://www.drupal.org)" name="Generator"/>
<meta content="width" name="MobileOptimized"/>
<meta content="true" name="HandheldFriendly"/>
<meta content="width=device-width, initial-scale=1.0" name="viewport"/>
<script>var _paq = _paq || [];(function(){var u=(("https:" == document.location.protocol) ? "" : "http://presidentofindia.ni
c.in");_paq.push(["setSiteId", 1]);_paq.push(["setTrackerUrl", u+"/visitors/_track"]);_paq.push(["setUserId", 0]);_paq.push
(["setCustomVariable", 7, "route", "view.former_presidents_listing.page_1", "visit"]);_paq.push(["setCustomVariable", 8, "pat
h", "\/former-presidents", "visit"]);if (!window.matomo_search_results_active) {_paq.push(["trackPageView"]);}var d=document,
g=d.createElement("script"),s=d.getElementsByTagName("script")[0];g.type="text/javascript";g.defer=true;g.async=true;g.src=u
```

```
In [42]: pname=[]
         for i in poi1.find_all('div',class_="desc-sec"):
             pname.append(i.h3.text)

         pname

Out[42]: ['Shri Ram Nath Kovind',
          'Shri Pranab Mukherjee',
          'Smt Pratibha Devisingh Patil',
          'DR. A.P.J. Abdul Kalam',
          'Shri K. R. Narayanan',
          'Dr Shankar Dayal Sharma',
          'Shri R Venkataraman',
          'Giani Zail Singh',
          'Shri Neelam Sanjiva Reddy',
          'Dr. Fakhruddin Ali Ahmed',
          'Shri Varahagiri Venkata Giri',
          'Dr. Zakir Husain',
          'Dr. Sarvepalli Radhakrishnan',
          'Dr. Rajendra Prasad']
```

```
In [44]: TOO=[]
         for i in poi1.find_all('div',class_="desc-sec"):
             TOO.append(i.h5.text)

         TOO

Out[44]: ['14th President of India',
          '13th President of India',
          '12th President of India',
          '11th President of India',
          '10th President of India',
          '9th  President of India',
          '8th President of India',
          '7th President of India',
          '6th President of India',
          '5th President of India',
          '4th President of India',
          '3rd President of India',
          '2nd President of India',
          '1st President of India']
```

```
In [47]: df1=pd.DataFrame({'President Names':pname,'Term Of Office':TOO})
         df1

Out[47]:
```

	President Names	Term Of Office
0	Shri Ram Nath Kovind	14th President of India
1	Shri Pranab Mukherjee	13th President of India
2	Smt Pratibha Devisingh Patil	12th President of India
3	DR. A.P.J. Abdul Kalam	11th President of India
4	Shri K. R. Narayanan	10th President of India
5	Dr Shankar Dayal Sharma	9th President of India
6	Shri R Venkataraman	8th President of India
7	Giani Zail Singh	7th President of India
8	Shri Neelam Sanjiva Reddy	6th President of India
9	Dr. Fakhruddin Ali Ahmed	5th President of India
10	Shri Varahagiri Venkata Giri	4th President of India
11	Dr. Zakir Husain	3rd President of India
12	Dr. Sarvepalli Radhakrishnan	2nd President of India
13	Dr. Rajendra Prasad	1st President of India

Term of office was not present on the former president webpage but it is present in the individual president webpages. So I have taken numbering of president (e.g 1st ,2nd president of India etc).