ASSIGNMENT-1

WEB SCRAPING

In all the following questions, you have to use BeautifulSoup to scrape different websites and collect data as per

the requirement of the question.

Every answer to the question should be in form of a python function which should take URL as the parameter.

Use Jupyter Notebooks to program, upload it on your GitHub and send the link of the Jupyter notebook to

your SME.

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

Ans.

from urllib.request import urlopen

from bs4 import BeautifulSoup

html = urlopen('https://en.wikipedia.org/wiki/Main\_Page')

bs = BeautifulSoup(html, "html.parser")

titles = bs.find\_all(['h1', 'h2','h3','h4','h5','h6'])

print('List all the header tags :', \*titles, sep='\n\n')

1. **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.

from bs4 import BeautifulSoup

import requests

import re

import pandas as pd

# Downloading imdb top 250 movie's data

url = 'http://www.imdb.com/chart/top'

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

movies = soup.select('td.titleColumn')

crew = [a.attrs.get('title') for a in soup.select('td.titleColumn a')]

ratings = [b.attrs.get('data-value')

for b in soup.select('td.posterColumn span[name=ir]')]

# create a empty list for storing

# movie information

list = []

# Iterating over movies to extract

# each movie's details

for index in range(0, len(movies)):

# Separating movie into: 'place',

# 'title', 'year'

movie\_string = movies[index].get\_text()

movie = (' '.join(movie\_string.split()).replace('.', ''))

movie\_title = movie[len(str(index))+1:-7]

year = re.search('\((.\*?)\)', movie\_string).group(1)

place = movie[:len(str(index))-(len(movie))]

data = {"place": place,

"movie\_title": movie\_title,

"rating": ratings[index],

"year": year,

"star\_cast": crew[index],

}

list.append(data)

# printing movie details with its rating.

for movie in list:

print(movie['place'], '-', movie['movie\_title'], '('+movie['year'] +

') -', 'Starring:', movie['star\_cast'], movie['rating'])

##.......##

df = pd.DataFrame(list)

df.to\_csv('imdb\_top\_250\_movies.csv',index=False)

3) Write a python program to display IMDB’s Top rated 100 Indian movies’ data (i.e. name, rating, year of

release) and make data frame.

Ans.

from bs4 import BeautifulSoup

import requests

import re

# Download IMDB's Top 250 Indian movies

url = 'https://www.imdb.com/india/top-rated-indian-movies/'

response = requests.get(url)

soup = BeautifulSoup(response.text, 'lxml')

movies = soup.select('td.titleColumn')

links = [a.attrs.get('href') for a in soup.select('td.titleColumn a')]

crew = [a.attrs.get('title') for a in soup.select('td.titleColumn a')]

ratings = [b.attrs.get('data-value') for b in soup.select('td.posterColumn span[name=ir]')]

votes = [b.attrs.get('data-value') for b in soup.select('td.ratingColumn strong')]

imdb = []

# Store each item into dictionary (data), then put those into a list (imdb)

for index in range(0, len(movies)):

# Seperate movie into: 'place', 'title', 'year'

movie\_string = movies[index].get\_text()

movie = (' '.join(movie\_string.split()).replace('.', ''))

movie\_title = movie[len(str(index))+1:-7]

year = re.search('\((.\*?)\)', movie\_string).group(1)

place = movie[:len(str(index))-(len(movie))]

data = {"movie\_title": movie\_title,

"year": year,

"place": place,

"star\_cast": crew[index],

"rating": ratings[index],

"vote": votes[index],

"link": links[index]}

imdb.append(data)

for item in imdb:

print(item['place'], '-', item['movie\_title'], '('+item['year']+') -', 'Starring:', item['star\_cast'])

4) Write s python program to display list of respected former presidents of India(i.e. Name , Term of office)

from <https://presidentofindia.nic.in/former-presidents.htm>

Ans.

from bs4 import BeautifulSoup

import requests

import re

import pandas as pd

# Downloading formar presidentof india data

url = 'https://presidentofindia.nic.in/former-presidents.htm'

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

president= soup.select('td.titleColumn')

crew = [a.attrs.get('title') for a in soup.select('td.titleColumn a')]

ratings = [b.attrs.get('data-value')

for b in soup.select('td.posterColumn span[name=ir]')]

# create a empty list for storing

# president information

list = []

# Iterating over to president name extract

# each president details

for index in range(0, len(president)):

# Separating president into: 'place',

# 'name', 'year'

president\_string = president[index].get\_text()

president = (' '.join(prisendent\_string.split()).replace('.', ''))

president\_name = president[len(str(index))+1:-7]

year = re.search('\((.\*?)\)', president).group(1)

place = president[:len(str(index))-(len(president))]

data = {"place": place,

"president\_name": president\_name,

"former": former[index],

"year": year,

"discreption": discreption[index],

}

list.append(data)

# printing president details with its rating.

for president in list:

print(president['place'], '-', president['president\_name'], '('+year['year'] +

') -', 'name:', name['former\_name'], president['former'])

##.......##

df = pd.DataFrame(list)

df.to\_csv('https://presidentofindia.nic.in/former-presidents.htm',index=False)

5) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape:

a) Top 10 ODI teams in men’s cricket alo ng with the records for matches, points and rating.

b) Top 10 ODI Batsmen along with the records of their team and rating.

c) Top 10 ODI bowlers along with the records of their team and rating.

Ans

import requests

from bs4 import BeautifulSoup

import re

import pandas as pd

headers = {

"User-Agent": "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36"

}

urls = [

"https://www.icc-cricket.com/rankings/mens/player-rankings/odi/batting",

"https://www.icc-cricket.com/rankings/mens/player-rankings/odi/bowling",

"https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting",

]

final\_result\_file\_name = "All Ranking List.csv"

final\_column\_names = ["Ranking Type", "Position", "Player Name", "Team Name", "Rating", "Career Best Rating", "Crawl URL"]

pd.DataFrame(columns=final\_column\_names).to\_csv(final\_result\_file\_name, sep="\t", index=False, encoding="utf-8")

for url in urls:

request\_object = requests.get(url, headers=headers)

html\_content = request\_object.text

print(request\_object.status\_code, "->", url)

soup\_object = BeautifulSoup(html\_content, "lxml")

for element in soup\_object.select('[class="ranking-pos up"], [class="ranking-pos down"]'):

element.replace\_with(BeautifulSoup("<" + element.name + "></" + element.name + ">", "html.parser"))

ranking\_type = soup\_object.select\_one(".rankings-block\_\_title-container > h4").text

result\_file\_name = ranking\_type + ".csv"

column\_names = ["Position", "Player Name", "Team Name", "Rating", "Career Best Rating", "Crawl URL"]

pd.DataFrame(columns=column\_names).to\_csv(result\_file\_name, sep="\t", index=False, encoding="utf-8")

for element in soup\_object.select('table[class="table rankings-table"] tr'):

if(element.find("th")):

continue

data\_dict = dict()

data\_dict["Crawl URL"] = url

data\_dict["Ranking Type"] = ranking\_type

if(element.select\_one('[class\*="position"]')):

data\_dict["Position"] = element.select\_one('[class\*="position"]').text

for player\_name in (element.select('a[href\*="/player-rankings"]')):

if(player\_name.text.strip()):

data\_dict["Player Name"] = player\_name.text

if(element.select\_one('[class^="flag-15"]')):

data\_dict["Team Name"] = element.select\_one('[class^="flag-15"]')["class"][-1]

if(element.select\_one('[class$="rating"]')):

data\_dict["Rating"] = element.select\_one('[class$="rating"]').text

if(element.select\_one('td.u-hide-phablet')):

data\_dict["Career Best Rating"] = element.select\_one('td.u-hide-phablet').text

for key in data\_dict.keys():

data\_dict[key] = re.sub(r"\s+", " ", data\_dict[key])

data\_dict[key] = data\_dict[key].strip()

pd.DataFrame([data\_dict], columns=column\_names).to\_csv(result\_file\_name, sep="\t", index=False, header=False, encoding="utf-8", mode="a")

pd.DataFrame([data\_dict], columns=final\_column\_names).to\_csv(final\_result\_file\_name, sep="\t", index=False, header=False, encoding="utf-8", mode="a")

**6) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape:**

**a) Top 10 ODI teams in women’s cricket along with the records for matches, points and rating.**

**b) Top 10 women’s ODI Batting players along with the records of their team and rating.**

**c) Top 10 women’s ODI all-rounder along with the records of their team and rating.**

Ans.

import requests

from bs4 import BeautifulSoup

import re

import pandas as pd

headers = {

"User-Agent": "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36"

}

urls = [

"https://www.icc-cricket.com/rankings/womens/player-rankings/odi/batting",

"https://www.icc-cricket.com/rankings/womens/player-rankings/t20i/batting",

"https://www.icc-cricket.com/rankings/womens/player-rankings/odi/bowling",

"https://www.icc-cricket.com/rankings/womens/player-rankings/t20i/bowling",

]

final\_result\_file\_name = "All Ranking List.csv"

final\_column\_names = ["Ranking Type", "Position", "Player Name", "Team Name", "Rating", "Career Best Rating", "Crawl URL"]

pd.DataFrame(columns=final\_column\_names).to\_csv(final\_result\_file\_name, sep="\t", index=False, encoding="utf-8")

for url in urls:

request\_object = requests.get(url, headers=headers)

html\_content = request\_object.text

print(request\_object.status\_code, "->", url)

soup\_object = BeautifulSoup(html\_content, "lxml")

for element in soup\_object.select('[class="ranking-pos up"], [class="ranking-pos down"]'):

element.replace\_with(BeautifulSoup("<" + element.name + "></" + element.name + ">", "html.parser"))

ranking\_type = soup\_object.select\_one(".rankings-block\_\_title-container > h4").text

result\_file\_name = ranking\_type + ".csv"

column\_names = ["Position", "Player Name", "Team Name", "Rating", "Career Best Rating", "Crawl URL"]

pd.DataFrame(columns=column\_names).to\_csv(result\_file\_name, sep="\t", index=False, encoding="utf-8")

for element in soup\_object.select('table[class="table rankings-table"] tr'):

if(element.find("th")):

continue

data\_dict = dict()

data\_dict["Crawl URL"] = url

data\_dict["Ranking Type"] = ranking\_type

if(element.select\_one('[class\*="position"]')):

data\_dict["Position"] = element.select\_one('[class\*="position"]').text

for player\_name in (element.select('a[href\*="/player-rankings"]')):

if(player\_name.text.strip()):

data\_dict["Player Name"] = player\_name.text

if(element.select\_one('[class^="flag-15"]')):

data\_dict["Team Name"] = element.select\_one('[class^="flag-15"]')["class"][-1]

if(element.select\_one('[class$="rating"]')):

data\_dict["Rating"] = element.select\_one('[class$="rating"]').text

if(element.select\_one('td.u-hide-phablet')):

data\_dict["Career Best Rating"] = element.select\_one('td.u-hide-phablet').text

for key in data\_dict.keys():

data\_dict[key] = re.sub(r"\s+", " ", data\_dict[key])

data\_dict[key] = data\_dict[key].strip()

pd.DataFrame([data\_dict], columns=column\_names).to\_csv(result\_file\_name, sep="\t", index=False, header=False, encoding="utf-8", mode="a")

pd.DataFrame([data\_dict], columns=final\_column\_names).to\_csv(final\_result\_file\_name, sep="\t", index=False, header=False, encoding="utf-8", mode="a")

7) Write a python program to scrape mentioned news details from https://www.cnbc.com/world/?region=world :

i) Headline

ii) Time

iii) News Link

Ans.

8) Write a python program to scrape the details of most downloaded articles from AI in last 90 days.

https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles

Scrape below mentioned details :

i) Paper Title

ii) Authors

iii) Published Date

iv) Paper URL

Ans.

9) 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.

10) Write a python program to scrape the details of top publications from Google Scholar from

https://scholar.google.com/citations?view\_op=top\_venues&hl=en

i) Rank

ii) Publication

iii) h5-index

iv) h5-median

Ans.