```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


pip install requests


# In[2]:


pip install bs4


# In[5]:


pip install html5lib


# In[8]:


import requests
from bs4 import BeautifulSoup
s=BeautifulSoup(r.content,'html5lib')
print(s.prettify())


# In[1]:


##1. Write a python program to display all the header tags from
wikipedia.org
import pandas as pd
import requests
from bs4 import BeautifulSoup

# scraping a wikipedia article
url_link = 'https://www.wikipedia.org/'
request = requests.get(url_link)

s = BeautifulSoup(request.text, 'lxml')

# creating a list of all common heading tags
heading_tags = ["h1", "h2", "h3","h4","h5","h6","h7"]
for i in s.find_all(heading_tags):
    print(i.name + ' -> ' + i.text.strip())


# In[5]:


import pandas as pd
data=[['Sri',35],['Deep',32],['Rind',19],['poos',22],['Krish',16]]
df=pd.DataFrame(data,columns=['Name','Age'])
df
```

```
# In[ ]:


##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
import pandas as pd
from bs4 import BeautifulSoup
import requests
import re


# Downloading imdb top 100 movie's data
url = 'http://www.imdb.com/chart/top'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'lxml')
#print(soup)
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')]

list = []

# create a empty list for storing
# movie information
list = []

# Iterating over movies to extract
# each movie's details
for index in range(0, 100):

    # 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)

    data={movie_title,year,place,ratings[index]}
    print(data)
    dataf=pd.DataFrame(data)#,columns=['movietitle','year','ratings'])
    dataf


# In[29]:


##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.
import pandas as pd
from bs4 import BeautifulSoup
```

```
import requests
import re


# Downloading imdb top 100 movie's data
url = 'https://www.imdb.com/india/top-rated-indian-movies/'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'lxml')
#print(soup)
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')]

list = []

# create a empty list for storing
# movie information
list = []

# Iterating over movies to extract
# each movie's details
for index in range(0, 100):

    # 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)

    data={movie_title,year,place,ratings[index]}
    print(data)
    dataf=pd.DataFrame(data)#,columns=['movietitle','year','ratings'])
    dataf


# In[31]:


import pandas as pd
from bs4 import BeautifulSoup
import requests
import re


url = 'https://meesho.com/bags-ladies/pl/p7vbp/'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'lxml')
print(soup)
#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')]
```

```
list = []

# create a empty list for storing
# movie information
list = []

# Iterating over movies to extract
# each movie's details
#for index in range(0, 100):



# In[34]:


##5. Write a python program to scrape cricket rankings from icc-
cricket.com
##6. Write a python program to scrape cricket rankings from icc-
cricket.com
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/test/batting",
"https://www.icc-cricket.com/rankings/mens/player-rankings/test/bowling",
"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/mens/player-rankings/t20i/batting",
"https://www.icc-cricket.com/rankings/mens/player-rankings/t20i/bowling",
"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")


# In[ ]:


##8. Write a python program to scrape house details from mentioned URL.
It should include house title, location,area, EMI and price from
https://www.nobroker.in/

from bs4 import BeautifulSoup
```

```
import requests
import pandas as pd
import time

# Creating time string to give fie name
timestr = time.strftime("%Y%m%d-%H%M%S")

# Creating empty list
BHK = []
Area = []
Latitude = []
Longitude = []
Size = []
Deposit = []
Rent = []
Type = []
Age = []
For = []
Possesion = []
Link = []


# Function to scrape
def scrape_NoBroker(n):
    print(f'Exporting {n} rows!!!')

    try:
        for page in range(int(n / 10)):

            try:
                print(f'{(page + 1) * 10} rows added!!!')

                # Requesting URL
                url = requests.get(

'https://www.nobroker.in/property/rent/bangalore/Bangalore/?searchParam=W
3sibGF0IjoxMi45NzE1OTg3LCJsb24iOjc3LjU5NDU2MjcsInBsYWNlSWQiOiJDaElKYlU2MH
lYQVdyanNSNEU5LVVlakQzX2ciLCJwbGFjZU5hbWUiOiJCYW5nYWxvcmUifV0=&sharedAcco
modation=0&orderBy=nbRank,desc&radius=2&traffic=true&travelTime=30&proper
tyType=rent&pageNo=' + str(
                        page)).text

                # Converting from HTML tag to BeautifulSoup object
                soup = BeautifulSoup(url, 'lxml')

                # Finding all the div tag wich contains all the info
                houses = soup.find_all('div', class_='card')

                # Looping through each div tag to get individual content
                for house in houses:
                    BHK.append(house.find('a', class_='card-link-
detail')['title'][:1])
                    Area_raw = house.find('a', class_='card-link-
detail')['title']
                    if ',' in Area_raw:
                        Area.append(Area_raw.split(',')[-1])
                    else:
                        Area.append(Area_raw.split('in', 1)[-1])
```

```
                    Latitude.append(house.find('meta',
itemprop='latitude')['content'])
                    Longitude.append(house.find('meta',
itemprop='longitude')['content'])
                    Size.append(house.find_all('meta',
itemprop='value')[0]['content'])
                    Deposit.append(house.find_all('meta',
itemprop='value')[1]['content'])
                    Rent.append(house.find_all('meta',
itemprop='value')[2]['content'])
                    Type.append(house.find_all('h5', class_="semi-
bold")[0].text)
                    Age.append(house.find_all('h5', class_="semi-
bold")[1].text)
                    For.append(house.find_all('h5', class_="semi-
bold")[2].text.replace('\n', ''))
                    Possesion.append(house.find_all('h5', class_="semi-
bold")[3].text.replace('\n', ''))
                    Link.append(house.find('a', class_='card-link-
detail')['href'])
            except:
                print(f'Row number {(page + 1) * 10} failed. Trying next
one!!!')
    except:
        pass

    # Creating DataFrame and storing data
    df = pd.DataFrame(list(zip(BHK, Area, Latitude, Longitude, Size,
Deposit, Rent, Type, Age, For, Possesion, Link)),
                      columns=['BHK', 'Address', 'Latitude', 'Longitude',
'Size(Acres)', 'Deposit(Rs)', 'Rent(Rs)',
                               'Furnishing', 'Property Age', 'Available
For', ' Immediate Possesion', 'Link'])

    # Exporting DataFrame in form of CSV file
    File_name = "House_Data_" + timestr + ".csv"
    df.to_csv(File_name, index=False)
    print("File Exported Sucessfully!!!!")

# Calling fuction to export 10000 rows
scrape_NoBroker(10000)


# In[2]:


##9. Write a python program to scrape mentioned details from
dineout.co.in :

import requests
from bs4 import BeautifulSoup
import re
import pandas as pd
url = 'https://www.dineout.co.in/delhi-restaurants/buffet-special/'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'lxml')
print(soup)
```

```
# In[18]:


import requests
from bs4 import BeautifulSoup
import pandas as pd

rest_list = []
for page in range(1,3):
    print(f'getting page, {page}')

    s = requests.Session()

    url = f"https://www.dineout.co.in/delhi-
restaurants?search_str=biryani&p={page}"  # URL of the website
    header = {'User-Agent':'Mozilla/5.0 (X11; CrOS x86_64 8172.45.0)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.64
Safari/537.36'} # Temporary user agent
    r = s.get(url, headers=header)
    soup = BeautifulSoup(r.content, 'html.parser')

    divs = soup.find_all('div', class_ = 'restnt-card restaurant')

    for item in divs:
        code = item.find('a')['href'].split('-')[-1] # restaurant code
        print(f'Getting details for {code}')
        data =
s.get(f'https://www.dineout.co.in/get_rdp_data_main/delhi/{code}/restaura
nt_detail_main').json()

        info = data['header']
        info.pop('share') #clean up csv
        info.pop('options')
        rest_list.append(info)

df = pd.DataFrame(rest_list)
df.to_csv('dehli_rest.csv',index=False)


# In[20]:


##10 Write a python program to scrape first 10 product details which
include product name , price , Image URL from
https://www.bewakoof.com/women-tshirts?ga_q=tshirts .
from bs4 import *
import requests
import os

# CREATE FOLDER
def folder_create(images):
    try:
        folder_name = input("Enter Folder Name:- ")
        # folder creation
        os.mkdir(folder_name)

    # if folder exists with that name, ask another name
```

```
    except:
        print("Folder Exist with that name!")
        folder_create()

    # image downloading start
    download_images(images, folder_name)


# DOWNLOAD ALL IMAGES FROM THAT URL
def download_images(images, folder_name):

    # initial count is zero
    count = 0

    # print total images found in URL
    print(f"Total {len(images)} Image Found!")

    # checking if images is not zero
    if len(images) != 0:
        for i, image in enumerate(images):
            # From image tag ,Fetch image Source URL

                        # 1.data-srcset
                        # 2.data-src
                        # 3.data-fallback-src
                        # 4.src

            # Here we will use exception handling

            # first we will search for "data-srcset" in img tag
            try:
                # In image tag ,searching for "data-srcset"
                image_link = image["data-srcset"]

            # then we will search for "data-src" in img
            # tag and so on..
            except:
                try:
                    # In image tag ,searching for "data-src"
                    image_link = image["data-src"]
                except:
                    try:
                        # In image tag ,searching for "data-fallback-src"
                        image_link = image["data-fallback-src"]
                    except:
                        try:
                            # In image tag ,searching for "src"
                            image_link = image["src"]

                        # if no Source URL found
                        except:
                            pass

            # After getting Image Source URL
            # We will try to get the content of image
            try:
                r = requests.get(image_link).content
                try:
```

```
                    # possibility of decode
                    r = str(r, 'utf-8')

                except UnicodeDecodeError:

                    # After checking above condition, Image Download
start
                    with open(f"{folder_name}/images{i+1}.jpg", "wb+") as
f:
                        f.write(r)

                    # counting number of image downloaded
                    count += 1
            except:
                pass

        # There might be possible, that all
        # images not download
        # if all images download
        if count == len(images):
            print("All Images Downloaded!")

        # if all images not download
        else:
            print(f"Total {count} Images Downloaded Out of
{len(images)}")

# MAIN FUNCTION START
def main(url):

    # content of URL
    r = requests.get(url)

    # Parse HTML Code
    soup = BeautifulSoup(r.text, 'html.parser')

    # find all images in URL
    images = soup.findAll('img')

    # Call folder create function
    folder_create(images)


# take url
url = input("Enter URL:- ")

# CALL MAIN FUNCTION
main(url)


# In[2]:


pip install nbconvert


# In[1]:
```

```
jupyter-nbconvert --to PDFviaHTML Webscrape.ipynb

# In[ ]:
```