write a program to extract and display all the header tags from wikepedia.org.

In [ ]:
```
```

In [ ]:
```
from urllib. request import urlopen
```

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

In [ ]:
```
html=urlopen('https://en.wikipedia.org/wiki/main_page')
```

In [ ]:
```
page
```

In [ ]:
```
bs=BeautifulSoup(html,"html.parser")
```

In [ ]:
```
titles=bs.find_all(['h1','h2','h3','h4','h5','h6'])
```

In [ ]:
```
print('list all the header:',*titles,sep='\n\n')
```

write a program to display IMDB'S top rated movies data(i.e name,rating) and make data frame.

In [ ]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [ ]:
```
## request page source from url
```

In [ ]:
```
url="https://www.imdb.com/chart/top/"
```

In [ ]:
```
page=requests.get(url)
page
```

In [ ]:
```
## display the page source code
page.content
```

In [ ]:
```
Soup=BeautifulSoup(page.content,"html.parser")
print(Soup.prettify())
```

In [ ]:
```
# scrap movies name
```

In [ ]:
```
scraped_movies=Soup.find_all('td', class_="titleColumn")
scraped_movies
```

In [ ]:
```
#parse movies name
movies=[]
for movie in scraped_movies:
    movies.append(movie.get_text().strip())
movies
```

In [ ]:
```
# scrap rating for movies
scraped_ratings=Soup.find_all('td',class_="ratingColumn imdbRating")
scraped_ratings
```

In [ ]:
```
# parse ratings
ratings=[]
for rating in scraped_ratings:
    rating=rating.get_text().replace('\n','')
    ratings.append(rating)
ratings
```

In [ ]:
```
# make data frame
```

In [ ]:
```
data=pd.DataFrame()
data['Movie Names']=movies
data['Ratings']=ratings
data.head(n=101)
```

write a python program to display IMDB'S top rated indian movies data(i.e name,rating,year of realease) and make data frame.

In [ ]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [ ]:
```
## request page source from url
```

In [ ]:
```
url="https://www.imdb.com/list/ls084312846/"
```

In [ ]:
```
page=requests.get(url)
page
```

In [ ]:
```
## display the page source code
page.content
```

In [ ]:
```
Soup=BeautifulSoup(page.content,"html.parser")
print(Soup.prettify())
```

In [ ]:
```
# scrap movies name
```

In [ ]:
```
scraped_movies=Soup.find_all('h3',class_="lister-item-header")
scraped_movies
```

In [ ]:
```
movies=[]
for movie in scraped_movies:
    movies.append(movie.get_text().strip())
movies
```

In [ ]:
```
# scrap rating for movies
scraped_ratings=Soup.find_all('div',class_="ipl-rating-star__rating")
scraped_ratings
```

In [ ]:
```
# parse ratings
ratings=[]
for rating in scraped_ratings:
    rating=rating.get_text().replace('\n','')
    ratings.append(rating)
ratings
```

In [ ]:
```
#make data frame
data=pd.DataFrame()
data['Movie Names']=movies

data.head(n=45)
```

write a python program to scrape first product details which include

write a python program to scrape first product details which include product name,price,image url from https://www.bewakoof.com/women-tshirts?ga_q=tshirts.

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

In [ ]:
```
page=requests.get('https://www.bewakoof.com/women-tshirts?ga_q=tshirts')
```

In [ ]:
```
page
```

In [ ]:
```
#page content
Soup=BeautifulSoup(page.content)
Soup
```

scraping first name

In [ ]:
```
first_title=Soup.find('div',class_="productCardDetail")
first_title
```

In [ ]:
```
first_title.text
```

In [ ]:
```
scraped_title=Soup.find_all('div',class_="productCardDetail")
scraped_title
```

In [ ]:
```
# parse title name
title=[]
for title in scraped_title:
    title.append(title.get_text().strip())
title
```

In [ ]:
```
scraping first price
```

In [ ]:
```
sta=Soup.find('span',class_="discountedPriceText")
sta.text.split()[1]
```

In [ ]:
```
scraping the multiple price
```

In [ ]:
```
price=[] # empty list
for i in Soup.find_all('span',class_="discountedPriceText"):
    price.append(i.text.replace('rs',''))
price
```

In [ ]:
```
images=[]
for i in Soup.find_all("a",class_="col-sm-4 col-xs-6"):
    images.append(i)
images
```

write a python program to scrape mentioned details from dineout.co.in(restaurant name,cuisine,location,ratings,imageurl.

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

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

In [ ]:
```
page
```

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

In [ ]:
```
first_title=Soup.find('div', class_="restnt-info cursor")
first_title
```

scraping first name

In [ ]:
```
first_title.text
```

scraping first location

In [ ]:
```
loc=Soup.find('div', class_="restnt-loc ellipsis")
loc.text
```

scraping first price

In [ ]:
```
sta=Soup.find('span',class_="double-line-ellipsis")
sta.text.split()[1]
```

scraping multiple locations

In [ ]:
```
location=[]#empty list
for i in Soup.find_all('div',class_="restnt-loc ellipsis"):
    location.append(i.text)
```

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

write a python program to scrape house datail from mentioned url.it should include house title,location,area,emi and price from https://www.nobroker.in/. enter three localities which are indira nagar,jayanagar,rajaji nagar.

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

In [ ]:
```
page=requests.get('https://www.nobroker.in/')
```

In [ ]:
```
page
```

page content

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

scraping first name

In [ ]:
```
first_title=Soup.find('div', class_="flex")

first_title
```

In [ ]:
```
first_title.text
```

scraping first location

In [ ]:
```
loc=Soup.find('div',class_="flex")
loc.text
```

In [ ]:
```
area=Soup.find('div',class_="font-semi-bold heading-6")
```

```
area.text
```

In [ ]:
```
Emi=Soup.find('div',class_="font-semi-bold heading-6")
Emi.text
```

In [ ]:
```
price=Soup.find('div',class_="nb__7nqQI")
price.text
```

Write a python program to scrape details of all the posts from coreyms.com. Scrape the heading, date, content and the code for the video from the link for the youtube video from the post.

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

In [ ]:
```
page7=requests.get('https://coreyms.com')
```

In [ ]:
```
page7
```

In [ ]:
```
Soup7=BeautifulSoup(page7.content)
Soup7
```

In [ ]:
```
heading=[]
for i in Soup7.find_all('h2',class_='entry-title'):
    heading.append(i.text)
heading
```

In [ ]:
```
date=[]
for i in Soup7.find_all('time',class_='entry-time'):
    date.append(i.text)
date
```

In [ ]:
```
content=[]
for i in Soup7.find_all('div',class_='entry-content'):
    content.append(i.text)
content
```

In [ ]:
```
videolink=[]
for i in Soup7.find_all('iframe',class_='youtube-player'):
    videolink.append(i['src'])
videolink
```

In [ ]:
```
print(len(heading),len(date),len(content),len(videolink))
```

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 along 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.

and

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

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import re
import pandas as pd
```

In [ ]:
```
headers={
    "user-agent":
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101 Firefox/97.0"
}
```

In [ ]:
```
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",
]
```

In [ ]:
```
final_result_file_name="All Ranking List.csv"

final_column_names = ["Ranking Type", "Position", "Player Name", "Team Name", "Rating", "Career Best Rating", "Cr
pd.DataFrame(columns=final_column_names).to_csv(final_result_file_name, sep="\t", index=False, encoding="utf-8")
```

In [ ]:
```
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"))
```

In [ ]:
```
    ranking_type = soup_object.select_one(".rankings-block__title-container > h4").text
```

In [ ]:
```
    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")
```

In [ ]:
```
    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()
```

In [ ]:
```
        pd.DataFrame([data_dict], columns=column_names).to_csv(result_file_name, sep="\t", index=False, header=Fa
```

In [ ]:
```
pd.DataFrame([data_dict], columns=final_column_names).to_csv(final_result_file_name, sep="\t", index=False, heade
```

Write a python program to scrape product name, price and discounts from https://meesho.com/bags-ladies/pl/p7vbp

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

```
import requests
```

In [ ]:
```
page=requests.get("https://meesho.com/bags-ladies/pl/p7vbp")
```

In [ ]:
```
page
```

page content

Soup=BeautifulSoup(page.content) Soup

scraping product name

In [ ]:
```
first_title=Soup.find('p',class_= "Text__StyledText-sc-oo0kvp-0 bWSOET NewProductCard__ProductTitle_Desktop-sc-j0
first_title
```

In [ ]:
```
first_title.text
```

scraping all price

In [ ]:
```
price=[]
for i in Soup.find_all('h5',class_="Text__StyledText-sc-oo0kvp-0 hiHdyy"):
    price.append(i.text.replace('rs',''))
price
```

discount price

In [ ]:
```
price=[]
for i in Soup.find_all('p',class_="Text__StyledText-sc-oo0kvp-0 fCJVtz NewProductCard__DiscountTextParagraph-sc-j
    price.append(i.text.replace('rs',''))
price
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```