

```
In [4]: 1 import requests
2 from bs4 import BeautifulSoup
3 import pandas as pd
4 import re
```

```
In [7]: 1 url = "http://m.imdb.com/list/ls009997493/"
2 r=requests.get(url)
3 print(r)
4
```

<Response [403]>

```
In [13]: 1 soup = BeautifulSoup(r.text, 'lxml')
2
3 movies = soup.select('li.list-item-header')
4 ratings = soup.select('span.ipl-rating-star__rating')
5 years = soup.select('span.list-item-year')
6 movie_titles = [movie.get_text() for movie in movies]
7 movie_ratings = [rating.get_text() for rating in ratings]
8 movie_years = [year.get_text().strip('(') for year in years]
9
10 data = {'Movie Title': movie_titles, 'Rating': movie_ratings, 'Year of
11 df = pd.DataFrame(data)
12
13 print(df)
```

Empty DataFrame

Columns: [Movie Title, Rating, Year of Release]

Index: []

```
In [ ]: 1
```

question 2

```
In [29]: 1 url=("https://www.patreon.com/coreyms")
2 r=requests.get(url)
3 print(r)
```

<Response [200]>

```
In [30]: 1 Soup =BeautifulSoup(r.content,'lxml')
2         print(Soup)
3
```

```
In [45]: 1 heading = Soup.find_all ("div",class_="post-title")
          2 for i in heading :
          3     print(i.text)
```

```
In [44]: 1 heading.text
```

```

AttributeError                                Traceback (most recent call last)
t)
Cell In[44], line 1
----> 1 heading.text

File ~\anaconda\Lib\site-packages\bs4\element.py:2428, in ResultSet.__getattr__(self, key)
    2426 def __getattr__(self, key):
    2427     """Raise a helpful exception to explain a common code fix."""
-> 2428     raise AttributeError(
    2429         "ResultSet object has no attribute '%s'. You're probably treating a list of elements like a single element. Did you call find_all() when you meant to call find()?" % key
    2430     )

AttributeError: ResultSet object has no attribute 'text'. You're probably treating a list of elements like a single element. Did you call find_all() when you meant to call find()?

```

```
In [41]: 1 date=Soup.find("span",class_="sc-kfPuZi doifNG")
          2 date
```

In []: 1

```
In [ ]: 1
```

```
In [ ]: 1
```

```
In [ ]: 1
```

question 3

```
In [27]: 1 url=("https://www.nobroker.in/delhi/buy")
2 r=requests.get(url)
3 print(r)
```

<Response [200]>

```
In [29]: 1 Soup=BeautifulSoup(r.content,"lxml")
2 Soup
```

```
<link href="http://dynamic.criteo.com" rel="dns-prefetch"/>
<link as="font" crossorigin="" href="https://fonts.gstatic.com/s/sourcesanspro/v13/6xKydsBYKcSV-LCoeQqfX1RY0o3ik4zwlxdu.woff2" rel="preload" type="font/woff2"/>
<link as="font" crossorigin="" href="https://fonts.gstatic.com/s/sourcesanspro/v13/6xK3dSBYKcSV-LCoeQqfX1RY0o3q0K71.woff2" rel="preload" type="font/woff2"/>
<link as="font" crossorigin="" href="https://fonts.gstatic.com/s/sourcesanspro/v13/6xKydsBYKcSV-LCoeQqfX1RY0o3i54rwlxdu.woff2" rel="preload" type="font/woff2"/>
<style>
    @font-face {
        font-family: 'Source Sans Pro';
        font-style: normal;
        font-weight: 300;
        font-display: swap;
        src: local('Source Sans Pro Light'), local('SourceSansPro-Light'), url(https://fonts.gstatic.com/s/sourcesanspro/v13/6xKydsBYKcSV-LCoeQqfX1RY0o3ik4zwlxdu.woff2) format('woff2');
        unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02
```

```
In [30]: 1 first_title=Soup.find('div',class_="heading-6 flex items-center font-se
2 first_title
```

```
In [62]: 1 loc=Soup.find('div',class_="relative bg-white md:top-5 md:sticky z-1")
2 loc
```

```
In [ ]: 1
```

```
In [ ]: 1
```

```
In [ ]: 1
```

question 4

```
In [7]: 1 url=("https://www.bewakoof.com/bestseller?sort=popular%20")
        2 r=requests.get(url)
        3 print(r)
```

<Response [200]>

```
In [8]: 1 Soup=BeautifulSoup(r.content,"lxml")
        2 Soup
```

```
(()=>n});const n="1.260.1",i="PROD",o="CDN",a="2.0.0-alpha.12"},385:(e,t,r)=>{"use strict";r.d(t,{FN:()=>u,IF:()=>f,LW:()=>s,Nk:()=>p,Tt:()=>d,_A:()=>a,iS:()=>c,il:()=>i,sK:()=>g,ux:()=>l,v6:()=>o,w1:()=>h});var n=r(7894);const i="undefined"!==typeof window&&!window.document,o="undefined"!==typeof WorkerGlobalScope&&("undefined"!==typeof self&&self instanceof WorkerGlobalScope&&self.navigator instanceof WorkerNavigator||"undefined"!==typeof globalThis&&globalThis instanceof WorkerGlobalScope&&globalThis.navigator instanceof WorkerNavigator),a=i?window:"undefined"!==typeof WorkerGlobalScope&&("undefined"!==typeof self&&self instanceof WorkerGlobalScope&&self||"undefined"!==typeof globalThis&&globalThis instanceof WorkerGlobalScope&&globalThis),s="complete"===a?.document?.readyState,c=Boolean("hidden"===a?.document?.visibilityState),u=""+"a?.location,d=/iPad|iPhone|iPod/.test(a.navigator?.userAgent),l=d&&"undefined"===typeof SharedWorker,f=((()=>{const e=a.navigator?.userAgent?.match(/Firefox[/\s](\d+\.\d+)/);return Array.isArray(e)&&e.length>2?e[1]:0})(),h=Boolean(i&&window.document.documentMode),p=!a.navigator?.sendBeacon,g=Date.now()-(0,n.z)()),1117:(e,t,r)=>{"use strict";r.d(t,{w:()=>o});var n=r(50);const i={agentIdentifier:"",ee:void 0};class o{constructor(e){try{if("object"!==typeof e)return(0,n.Z)("shared context requires an object as input"):this.sharedContext=f}.Object.assign(this.sharedContext,i).Object.ent
```

```
In [9]: 1 title = Soup.title
        2 print(title.text)
```

Bestseller

```
In [10]: 1 product_detail=Soup.find('div',class_="productNaming bkf-ellipsis")
        2 product_detail
```

```
Out[10]: <div class="productNaming bkf-ellipsis"><h3 class="brand-name undefined">Bewakoof®</h3><h2 class="clr-shade4 h3-p-name undefined false">Men's Grey Oversized Cargo Pants</h2></div>
```

```
In [11]: 1 product_detail.text
```

```
Out[11]: "Bewakoof®Men's Grey Oversized Cargo Pants"
```

```
In [14]: 1 product=[]
2
3 for i in Soup.find_all('div',class_="productName bkf-ellipsis"):
4     product.append(i.text)
5
6 product
7
```

```
Out[14]: ["bewakoof x star warsMen's Black Storm Trooper Graphic Printed Oversized
T-shirt",
"Bewakoof®Men's Red Hunter Graphic Printed Oversized T-shirt",
"Bewakoof Air® 1.0Men's Black Oversized Cargo Joggers",
"Bewakoof®Women's Orange Skool Graphic Printed Boyfriend T-shirt",
"Bewakoof®Men's Blue Rider Vroom Panda Graphic Printed T-shirt",
"bewakoof x friendsWomen's Black Friends Typography Boyfriend T-shirt",
"Bewakoof®Men's White T-shirt",
"bewakoof x tom & jerryMen's Red Pocket Jerry Graphic Printed T-shirt",
"bewakoof x dcMen's Black Batman Outline Logo (BML) Printed T-shirt",
"bewakoof x disneyWomen's Red Sarcastic One (DL) Boyfriend Graphic Printe
d T-shirt"]
```

```
In [20]: 1
2
3 prices= Soup.find_all("div", class_="discountedPriceText clr-p-black
4
5 for i in prices :
6
7     print(i.text)
8
9
10
```

```
In [ ]: 1
```

```
In [24]: 1 # images = []
2
3 images = Soup.find_all("img", class_="productImgTag")
4 #     src = img.get('src')
5 #     images.append(src)
6 #     images
7 for i in images :
8
9     print(i.text)
```

```
In [ ]: 1
```

In []: 1

question 5

```
In [29]: 1 url = ("https://www.cnbc.com/world/?region=world")
          2 r=requests.get(url)
          3 r
```

Out[29]: <Response [200]>

```
In [30]: 1 Soup=BeautifulSoup(r.content,"lxml")
          2 Soup
          3
          4 display:flex;flex-direction:row;font-size:14px;font-weight:600;height:40px;
          5 height:50px!important;justify-content:center;letter-spacing:1px;margin:0 auto;max-width:260px!important;padding:0 20px;padding:0!important;text-align:center;text-transform:uppercase;transition:all .15s linear;width:100%;width:100%!important}.AuthForms-container .AuthForms-accountButton>span,.AuthForms-container .AuthForms-submitButton>span{display:block;line-height:12px}.AuthForms-container .AuthForms-accountButton [class*="icon-"],.AuthForms-container .AuthForms-accountButton [class^=icon-],.AuthForms-container .AuthForms-submitButton [class*="icon-"],.AuthForms-container .AuthForms-submitButton [class^=icon-]{height:12px;margin-left:5px;position:relative}.AuthForms-container .AuthForms-accountButton [class*="icon-"]:before,.AuthForms-container .AuthForms-accountButton [class^=icon-]:before,.AuthForms-container .AuthForms-submitButton [class*="icon-"]:before,.AuthForms-container .AuthForms-submitButton [class^=icon-]:before{line-height:12px}.AuthForms-container .AuthForms-accountButton:focus,.AuthForms-container .AuthForms-accountButton:hover,.AuthForms-container .AuthForms-submitButton:focus,.AuthForms-container .AuthForms-submitButton:hover{background-color:#002f6c;border-color:#002f6c;color:#fff}.AuthForms-container .AuthForms-accountButton[class*=touchButton]:hover,.AuthForms-container .AuthForms-submitButton[class*=touchButton]:hove
```

In [65]:

```
1 headings = []
2 dates = []
3 news_links = []
4
5 articles = Soup.find_all('div', class_='Card-titleContainer')
6 for article in articles:
7
8     heading = article.find('a', class_='Card-titleLink')
9     headings.append(heading.text.strip() if heading else 'No heading')
10
11
12     date = article.find('time')
13     dates.append(date.text.strip() if date else 'No date found')
14
15     link = article.find('a', class_='Card-titleLink')['href'] if ar
16     news_links.append(link)
17
18 return headings, dates, news_links
19
20 if __name__ == "__main__":
21
22     headings, dates, news_links = scrape_cnbc_world_news()
23
24     print("Headings:")
25     for idx, heading in enumerate(headings, start=1):
26         print(f"{idx}. {heading}")
27
28     print("\nDates:")
29     for date in dates:
30         print(date)
31
32     print("\nNews Links:")
33     for link in news_links:
34         print(link)
```

Input In [65]

return headings, dates, news_links

^

SyntaxError: 'return' outside function

```

In [63]: 1 def scrape_cnbc_world_news():
2         url = "https://www.cnbc.com/world/?region=world"
3         response = requests.get(url)
4         soup = BeautifulSoup(response.content, 'lxml')
5
6         headings = []
7         dates = []
8         news_links = []
9
10        articles = soup.find_all('div', class_='BreakingNews-titleContainer')
11
12        for article in articles:
13
14            heading = article.find('a', class_='ArticleHeader-eyebrow')
15            headings.append(heading.text.strip)
16
17            date = article.find('ArticleHeader-time')
18            dates.append(date.text.strip)
19
20
21            link = article.find('a', class_='WatchLive-container')['href']
22            news_links.append(link)
23
24        return headings, dates, news_links
25
26    if __name__ == "__main__":
27        headings, dates, news_links = scrape_cnbc_world_news()
28
29        print("Headings:")
30        for idx, heading in enumerate(headings, start=1):
31            print(f"{idx}. {heading}")
32
33        print("\nDates:")
34        for date in dates:
35            print(date)
36
37        print("\nNews Links:")
38        for link in news_links:
39            print(link)
40

```

Headings:

Dates:

News Links:

```

In [ ]: 1

```

```

In [ ]: 1

```