In [1]:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import pandas as pd
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import WebDriverException
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import StaleElementReferenceException
```

In [2]:

```
# Set Chrome options for running in headless mode
chrome_options = Options()
chrome_options.add_argument("--headless")  # Enable headless mode
```

1.Scrape the details of most viewed videos on YouTube from Wikipedia.

Url = https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos You need to find following details: A) Rank B) Name C) Artist D) Upload date E) Views

In [3]:

```
# Create a new instance of the Chrome driver
driver = webdriver.Chrome(options=chrome_options)

# Open the given URL
driver.get("https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos")

# Find the table that contains the video details
table = driver.find_element(By.XPATH,'//table[@class="wikitable sortable jquery-tablesorter"]')

# Find the body of table in order to extract rows present in the table body
tbody = table.find_element(By.TAG_NAME,'tbody')

# Extract all rows present in the body.
rows = tbody.find_elements(By.TAG_NAME,'tr')

print("Top",len(rows),"Youtube vidoe details to be scrapped")#Print total no of data present.


## initialize empty lists for the details to be scrapped like : rank, name, artist, date, view
ranks=[]
names=[]
artists=[]
dates=[]
views=[]

try:
    for row in rows: #iterate through every row
        columns = row.find_elements(By.TAG_NAME,'td') # find all column data in every row
        if len(columns)>=6:  #no of columns in each row is 6.

            rank = columns[0].text  # scrapping the required data
            name = columns[1].text
            artist =columns[2].text
            date = columns[4].text
            view = columns[3].text


           # append all the scrapped to the lists defined.
            ranks.append(rank)
            names.append(name)
            artists.append(artist)
            dates.append(date)
            views.append(view)

except NoSuchElementException:  ## handling NoSuchElementException
    print("ERROR")
    pass


# close the driver
driver.quit()


## Put the data into a dictionary
top30_df = {"Rank":ranks,
           "Name":names,
           "Artist":artists,
           "Published Date": dates,
           "Views (In Billions)":views}

# Convert the dictonary to a dataframe
df = pd.DataFrame(top30_df)

df
```

```
Top 30 Youtube vidoe details to be scrapped
```

Out[3]:

	Rank	Name	Artist	Published Date	Views (In Billions)
0	1.	"Baby Shark Dance"[6]	Pinkfong Baby Shark - Kids' Songs & Stories	June 17, 2016	12.85
1	2.	"Despacito"[9]	Luis Fonsi	January 12, 2017	8.16
2	3.	"Johny Johny Yes Papa"[16]	LooLoo Kids	October 8, 2016	6.70
3	4.	"Bath Song"[17]	Cocomelon – Nursery Rhymes	May 2, 2018	6.20
4	5.	"Shape of You"[18]	Ed Sheeran	January 30, 2017	6.00
5	6.	"See You Again"[21]	Wiz Khalifa	April 6, 2015	5.89
6	7.	"Phonics Song with Two Words"[26]	ChuChu TV	March 6, 2014	5.30
7	8.	"Wheels on the Bus"[27]	Cocomelon – Nursery Rhymes	May 24, 2018	5.24
8	9.	"Uptown Funk"[28]	Mark Ronson	November 19, 2014	4.92
9	10.	"Learning Colors – Colorful Eggs on a Farm"[29]	Miroshka TV	February 27, 2018	4.89
10	11.	"Gangnam Style"[30]	Psy	July 15, 2012	4.80
11	12.	"Masha and the Bear – Recipe for Disaster"[35]	Get Movies	January 31, 2012	4.55
12	13.	"Dame Tu Cosita"[36]	El Chombo	April 5, 2018	4.35
13	14.	"Axel F"[37]	Crazy Frog	June 16, 2009	3.91
14	15.	"Sugar"[38]	Maroon 5	January 14, 2015	3.87
15	16.	"Roar"[39]	Katy Perry	September 5, 2013	3.80
16	17.	"Counting Stars"[40]	OneRepublic	May 31, 2013	3.79
17	18.	"Sorry"[41]	Justin Bieber	October 22, 2015	3.66
18	19.	"Baa Baa Black Sheep"[42]	Cocomelon – Nursery Rhymes	June 25, 2018	3.64
19	20.	"Thinking Out Loud"[43]	Ed Sheeran	October 7, 2014	3.60
20	21.	"Waka Waka (This Time for Africa)"[44]	Shakira	June 4, 2010	3.59
21	22.	"Dark Horse"[45]	Katy Perry	February 20, 2014	3.52
22	23.	"Lakdi Ki Kathi"[46]	Jingle Toons	June 14, 2018	3.48
23	24.	"Faded"[47]	Alan Walker	December 3, 2015	3.45
24	25.	"Perfect"[48]	Ed Sheeran	November 9, 2017	3.45
25	26.	"Let Her Go"[49]	Passenger	July 25, 2012	3.44
26	27.	"Girls Like You"[50]	Maroon 5	May 31, 2018	3.42
27	28.	"Humpty the train on a fruits ride"[51]	Kiddiestv Hindi – Nursery Rhymes & Kids Songs	January 26, 2018	3.41
28	29.	"Lean On"[52]	Major Lazer	March 22, 2015	3.38
29	30.	"Bailando"[53]	Enrique Iglesias	April 11, 2014	3.38

2. Scrape the details team India’s international fixtures from bcci.tv. Url = https://www.bcci.tv/.

You need to find following details: A) Match title (I.e. 1st ODI) B) Series C) Place D) Date E) Time Note: - From bcci.tv home page you have reach to the international fixture page through code.

In [4]:

```python
# Set up the Chrome driver

driver = webdriver.Chrome(options=chrome_options)  ## running chrome in headless mode

# Navigate to the BCCI website
driver.get("https://www.bcci.tv/")

# Find and click the "International Fixtures" link

driver.execute_script("arguments[0].click();", driver.find_element(By.XPATH, "//*[@id='navigation']/ul[1]/li[2]/a"))


time.sleep(5)

# Find more elements button and click

driver.execute_script("arguments[0].click();", driver.find_element(By.XPATH, '//button[@class="match-btn btn-red d-flex align-items-center

# Find the elements containing the fixture details

main = driver.find_element(By.XPATH,'//div[@class="fixture-tab-inner row"]')

# Find all elements in main
cards = main.find_elements(By.XPATH,'//div[@class="col-lg-4 col-md-6 col-sm-12 ng-scope"]')

print("No of data available: ", len(cards))
# Initialize empty lists to store the details
international_fixtures=[]
match_titles = []
series = []
places = []
dates = []
times = []

try:
    # Extract the details from each fixture element
    for element in cards:

       series_name_element = element.find_elements(By.XPATH,'//h5[@class="match-tournament-name ng-binding"]')

       place_element = element.find_elements(By.XPATH,'//div[@class="match-place ng-scope"]')

       date_element = element.find_elements(By.XPATH,'//div[@class="match-dates ng-binding"]')

       time_element = element.find_elements(By.XPATH,'//div[@class="match-time no-margin ng-binding"]')



    ## append the scrapped data
    for i in range(len(series_name_element)):

        series = series_name_element[i].text
        palces = place_element[i].text.split("-")[1]
        dates = date_element[i].text
        times=time_element[i].text
        match_titles = place_element[i].text.split("-")[0]


        international_fixtures.append([match_titles,series,palces,dates,times])

except NoSuchElementException:  ##handles no such element exception.
    pass

driver.quit()

## Display the data in a dataframe.
International_fixtures = pd.DataFrame(international_fixtures,columns=['Match_title','Series','Place','Date',
                                   'Time'])

International_fixtures
```

```
No of data available:  16
```

Out[4]:

	Match_title	Series	Place	Date	Time
0	1st T20I	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	9 JUL 2023	1:30 PM IST
1	2nd T20I	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	11 JUL 2023	1:30 PM IST
2	1st Test	INDIA TOUR OF WEST INDIES 2023	Windsor Park, Dominica	12 JUL 2023	7:30 PM IST
3	3rd T20I	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	13 JUL 2023	1:30 PM IST
4	1st ODI	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	16 JUL 2023	9:00 AM IST
5	2nd ODI	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	19 JUL 2023	9:00 AM IST
6	2nd Test	INDIA TOUR OF WEST INDIES 2023	Queen's Park Oval, Trinidad	20 JUL 2023	7:30 PM IST
7	3rd ODI	INDIA WOMEN TOUR OF BANGLADESH 2023	Shere Bangla National Stadium, Mirpur, Dhaka	22 JUL 2023	9:00 AM IST
8	1st ODI	INDIA TOUR OF WEST INDIES 2023	Kensington Oval, Barbados	27 JUL 2023	7:00 PM IST
9	2nd ODI	INDIA TOUR OF WEST INDIES 2023	Kensington Oval, Barbados	29 JUL 2023	7:00 PM IST
10	3rd ODI	INDIA TOUR OF WEST INDIES 2023	Brian Lara Stadium, Trinidad	1 AUG 2023	7:00 PM IST
11	1st T20I	INDIA TOUR OF WEST INDIES 2023	Brian Lara Stadium, Trinidad	3 AUG 2023	8:00 PM IST
12	2nd T20I	INDIA TOUR OF WEST INDIES 2023	National Stadium, Guyana	6 AUG 2023	8:00 PM IST
13	3rd T20I	INDIA TOUR OF WEST INDIES 2023	National Stadium, Guyana	8 AUG 2023	8:00 PM IST
14	4th T20I	INDIA TOUR OF WEST INDIES 2023	Central Broward Regional Park Stadium Turf Gr...	12 AUG 2023	8:00 PM IST
15	5th T20I	INDIA TOUR OF WEST INDIES 2023	Central Broward Regional Park Stadium Turf Gr...	13 AUG 2023	8:00 PM IST

3.Scrape the details of State-wise GDP of India from statisticstime.com. Url = http://statisticstimes.com/

You have to find following details: A) Rank B) State C) GSDP(18-19)- at current prices D) GSDP(19-20)- at current prices E) Share(18-19) F) GDP($ billion) Note: - From statisticstimes home page you have to reach to economy page through code.

In [5]:

```
# Set up the Chrome driver

driver = webdriver.Chrome(options=chrome_options)  ## running chrome in headless mode

# Navigate to the Statisticstimes website
driver.get("https://statisticstimes.com")

# Find and click the "Economy" link

driver.execute_script("arguments[0].click();", driver.find_element(By.XPATH, '//*[@id="top"]/div[2]/div[2]/div/a[3]'))

## find and click GDP by Indian States

driver.execute_script("arguments[0].click();", driver.find_element(By.LINK_TEXT, "» GDP of Indian states"))

## find all the rows containing the required details
rows = driver.find_elements(By.XPATH,'//table[@id="table_id"]//tbody//tr[@role="row"]')

print(len(rows),"datas are presented in the GDP by Indian States table.")

## Initialising an empty list
data=[]

## Before iterating handle nOsuch element exception.

try:
    for row in rows: ## Iterate through every row.
        cols = row.find_elements(By.TAG_NAME,"td") # find column data of every row.
        cols =[col.text.strip() for col in cols[:6]] #scrape evey data
        data.append(cols) # append the scrapped data into the list.

except NoSuchElementException:
    pass
# close the driver
driver.quit()

# Display the scrapped data into dataframe
GDP = pd.DataFrame(data,columns=['Rank','State','GSDP(19-20)Current Prices','GSDP(18-19)Current Prices',
                                 'Share(18-19)','GDP($billion)'])
GDP
```

```
33 datas are presented in the GDP by Indian States table.
```

Out[5]:

	Rank	State	GSDP(19-20)Current Prices	GSDP(18-19)Current Prices	Share(18-19)	GDP($billion)
0	1	Maharashtra	-	2,632,792	13.94%	399.921
1	2	Tamil Nadu	1,845,853	1,630,208	8.63%	247.629
2	3	Uttar Pradesh	1,687,818	1,584,764	8.39%	240.726
3	4	Gujarat	-	1,502,899	7.96%	228.290
4	5	Karnataka	1,631,977	1,493,127	7.91%	226.806
5	6	West Bengal	1,253,832	1,089,898	5.77%	165.556
6	7	Rajasthan	1,020,989	942,586	4.99%	143.179
7	8	Andhra Pradesh	972,782	862,957	4.57%	131.083
8	9	Telangana	969,604	861,031	4.56%	130.791
9	10	Madhya Pradesh	906,672	809,592	4.29%	122.977
10	11	Kerala	-	781,653	4.14%	118.733
11	12	Delhi	856,112	774,870	4.10%	117.703
12	13	Haryana	831,610	734,163	3.89%	111.519
13	14	Bihar	611,804	530,363	2.81%	80.562
14	15	Punjab	574,760	526,376	2.79%	79.957
15	16	Odisha	521,275	487,805	2.58%	74.098
16	17	Assam	-	315,881	1.67%	47.982
17	18	Chhattisgarh	329,180	304,063	1.61%	46.187
18	19	Jharkhand	328,598	297,204	1.57%	45.145
19	20	Uttarakhand	-	245,895	1.30%	37.351
20	21	Jammu & Kashmir	-	155,956	0.83%	23.690
21	22	Himachal Pradesh	165,472	153,845	0.81%	23.369
22	23	Goa	80,449	73,170	0.39%	11.115
23	24	Tripura	55,984	49,845	0.26%	7.571
24	25	Chandigarh	-	42,114	0.22%	6.397
25	26	Puducherry	38,253	34,433	0.18%	5.230
26	27	Meghalaya	36,572	33,481	0.18%	5.086
27	28	Sikkim	32,496	28,723	0.15%	4.363
28	29	Manipur	31,790	27,870	0.15%	4.233
29	30	Nagaland	-	27,283	0.14%	4.144
30	31	Arunachal Pradesh	-	24,603	0.13%	3.737
31	32	Mizoram	26,503	22,287	0.12%	3.385
32	33	Andaman & Nicobar Islands	-	-	-	-

4. Scrape the details of trending repositories on Github.com. Url = https://github.com/

You have to find the following details: A) Repository title B) Repository description C) Contributors count D) Language used

In [6]:

```
## Set the Chrome driver , run in headless mode.

driver = webdriver.Chrome(options=chrome_options)

## Handling WebDriverException taht occured , as the website's load is real slow
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## Find and click Top 100 songs
        driver.get("https://github.com/")

        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved , Please Check your internet connection")

driver.execute_script("arguments[0].click();", driver.find_element(By.XPATH,'/html/body/div[1]/div[1]/header/div/div[2]/div/nav/ul/li[3]/d

## Hold on the driver to find and select the Box containing the element
wait = WebDriverWait(driver, 10)
wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, "article.Box-row")))

## Find all boxes.
boxes = driver.find_elements(By.CSS_SELECTOR, "article.Box-row")

print("Total no of Trending repositories in Github : ",len(boxes))

## Initialise and empty list
data=[]


for box in boxes:

    g_data={} ## define an empty dictonary

    try:
        titles = box.find_element(By.XPATH,'.//h2[@class="h3 lh-condensed"]').text.strip("/")
    except NoSuchElementException:
        titles="-"  ## scrapping titles

    try:
        des= box.find_element(By.XPATH,'.//p[@class="col-9 color-fg-muted my-1 pr-4"]').text.strip()
    except NoSuchElementException:
        des="-"  ## scrapping description

    try:
        lan = box.find_element(By.XPATH,'.//span[@itemprop="programmingLanguage"]').text.strip()
    except NoSuchElementException:
        lan ="_" ## scrapping language

    try:
        ## To scrap CONTRIBUTORS COUNT , it is not presnet in the main page,  Steps followed :

        ## Step 1 : Find Urls for every repository and open them in a new window
        url = box.find_element(By.XPATH,'.//h2[@class="h3 lh-condensed"]//a').get_attribute("href")

        driver.execute_script(f"window.open('{url}', '_blank');")

        ## Switch Driver source to the new window
        driver.switch_to.window(driver.window_handles[1])

        try:
            ## Find all the elements presented in the side page
            x=driver.find_elements(By.XPATH,'//h2[@class="h4 mb-3"]')
            ## Out of the lists of elements select COntributors Count.
            count = x[-2].text.split('\n')[1]  ## For most of the links Contributors COunt is the second to last column.


        except:
            try:
                count= x[-1].text.split('\n')[1] ## for Few links Contributors Count is the last column.
            except:
                count ="-"

        ## Close the new window
        driver.close()
        ## Switch Back to the first window
        driver.switch_to.window(driver.window_handles[0])


    except:
         continue

    ## append all the scrapped details
    g_data["Title"] =titles
    g_data["Description"]=des
    g_data["Language"] = lan
    g_data["URL"]=url
    g_data["contributors_count"]=count

    data.append(g_data)

## Close the main driver.
driver.close()

## Display the data in Dataframe
data = pd.DataFrame(data)

## Make the URLs Clickable in dataframe
def make_clickable(val):
    # target _blank to open new window
    return '<a target="_blank" href="{}">{}</a>'.format(val, val)

## display dataframe
data.style.format({'URL': make_clickable})
```

```
Total no of Trending repositories in Github :  25
```

Out[6]:

	Title	Description	Language	URL	contributors_count
0	XingangPan / DragGAN	Official Code for DragGAN (SIGGRAPH 2023)	Python	https://github.com/XingangPan/DragGAN	10
1	THUDM / ChatGLM2-6B	ChatGLM2-6B: An Open Bilingual Chat LLM | 开源双语对话语言模型	Python	https://github.com/THUDM/ChatGLM2-6B	6
2	CASIA-IVA-Lab / FastSAM	Fast Segment Anything	Python	https://github.com/CASIA-IVA-Lab/FastSAM	10
3	ramonvc / freegpt-webui	GPT 3.5/4 with a Chat Web UI. No API key required.	Python	https://github.com/ramonvc/freegpt-webui	3
4	embedchain / embedchain	Framework to easily create LLM powered bots over any dataset.	Python	https://github.com/embedchain/embedchain	5
5	spacedriveapp / spacedrive	Spacedrive is an open source cross-platform file explorer, powered by a virtual distributed filesystem written in Rust.	Rust	https://github.com/spacedriveapp/spacedrive	64
6	xitanggg / open-resume	OpenResume is a powerful open-source resume builder and resume parser. https://open-resume.com/	TypeScript	https://github.com/xitanggg/open-resume	-
7	papers-we-love / papers-we-love	Papers from the computer science community to read and discuss.	Shell	https://github.com/papers-we-love/papers-we-love	247
8	sadmann7 / skateshop	An open source e-commerce skateshop build with everything new in Next.js 13.	TypeScript	https://github.com/sadmann7/skateshop	5
9	microsoft / Web-Dev-For-Beginners	24 Lessons, 12 Weeks, Get Started as a Web Developer	JavaScript	https://github.com/microsoft/Web-Dev-For-Beginners	205
10	sb-ocr / diy-spacemouse	A DIY navigation device for Fusion360	C++	https://github.com/sb-ocr/diy-spacemouse	-
11	THUDM / ChatGLM-6B	ChatGLM-6B: An Open Bilingual Dialogue Language Model | 开源双语对话语言模型	Python	https://github.com/THUDM/ChatGLM-6B	44
12	SizheAn / PanoHead	Code Repository for CVPR 2023 Paper "PanoHead: Geometry-Aware 3D Full-Head Synthesis in 360 degree"	Python	https://github.com/SizheAn/PanoHead	-
13	PlexPt / awesome-chatgpt-prompts-zh	ChatGPT 中文调教指南。各种场景使用指南。学习怎么让它听你的话。	_	https://github.com/PlexPt/awesome-chatgpt-prompts-zh	19
14	firstcontributions / first-contributions	🚀✨ Help beginners to contribute to open source projects	_	https://github.com/firstcontributions/first-contributions	5,000+
15	actualbudget / actual	A local-first personal finance system	JavaScript	https://github.com/actualbudget/actual	52
16	xtekky / gpt4free	The official gpt4free repository | various collection of powerful language models	Python	https://github.com/xtekky/gpt4free	83
17	sveltejs / svelte	Cybernetically enhanced web apps	JavaScript	https://github.com/sveltejs/svelte	610
18	OpenGVLab / DragGAN	Unofficial Implementation of DragGAN - "Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold" （DragGAN 全功能实现，在线Demo，本地部署试用，代码、模型已全部开源，支持Windows, macOS, Linux）	Python	https://github.com/OpenGVLab/DragGAN	9
19	OpenDriveLab / UniAD	[CVPR 2023 Best Paper] Planning-oriented Autonomous Driving	Python	https://github.com/OpenDriveLab/UniAD	6
20	qgis / QGIS	QGIS is a free, open source, cross platform (lin/win/mac) geographical information system (GIS)	C++	https://github.com/qgis/QGIS	491
21	chat2db / Chat2DB	🔥 🔥 🔥 An intelligent and versatile general-purpose SQL client and reporting tool for databases which integrates ChatGPT capabilities.(智能的通用数据库SQL客户端和报表工具)	Java	https://github.com/chat2db/Chat2DB	7
22	Kanaries / pygwalker	PyGWalker: Turn your pandas dataframe into a Tableau-style User Interface for visual analysis	Python	https://github.com/Kanaries/pygwalker	11
23	ggerganov / ggml	Tensor library for machine learning	C	https://github.com/ggerganov/ggml	48
24	StanGirard / quivr	🧠 Dump all your files and thoughts into your private GenerativeAI Second Brain and chat with it 🧠	TypeScript	https://github.com/StanGirard/quivr	28

5. Scrape the details of top 100 songs on billiboard.com. Url = https:/www.billboard.com/

You have to find the following details: A) Song name B) Artistname C) Last week rank D) Peak rank E) Weeks on board Note: - From the home page you have to click on the charts option then hot 100-page link through code.

In [7]:

```
## Set up Chrome driver and run in headless mode.
driver = webdriver.Chrome(options=chrome_options)

## Navigate to Billboard.com
driver.get("https://www.billboard.com")

## Find and click on Charts Option.
driver.execute_script("arguments[0].click();",driver.find_element(By.XPATH,'//*[@id="main-wrapper"]/header/div/div[2]/div/div/div[1]/div[1

## Handling WebDriverException that occured as, the website's load is real slow
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## Find and click Top 100 songs
        driver.execute_script("arguments[0].click();",driver.find_element(By.XPATH,'//*[@id="main-wrapper"]/div[9]/div/div/ul/li[1]/ul

        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved.")


## Find and select all element conatining conatiners.
boxes = driver.find_elements(By.CSS_SELECTOR, "div.o-chart-results-list-row-container")

## initialise empty lists for storage.
ranks =[]
songs=[]
artists=[]
last_week_ranks=[]
peak_ranks=[]
weeks_on_board=[]

##Iterate through every element box.
for box in boxes:
    try:
        ## Scrap the details
        rank = box.find_element(By.XPATH,'.//span[@class="c-label  a-font-primary-bold-l u-font-size-32@tablet u-letter-spacing-0080@table

        details= box.find_elements(By.XPATH,'.//ul[@class="lrv-a-unstyle-list lrv-u-flex lrv-u-height-100p lrv-u-flex-direction-column@mob

        if len(details)>=12:
            song_name = details[0].text.split('\n')[0]
            artist = details[0].text.split('\n')[1]
            last_wr =details[3].text
            peak_r = details[4].text
            weeks_ob= details[5].text

            # store the scrapped details
            songs.append(song_name)
            artists.append(artist)
            last_week_ranks.append(last_wr)
            peak_ranks.append(peak_r)
            weeks_on_board.append(weeks_ob)

        ranks.append(rank)

    except NoSuchElementException:
        pass

## Close the driver
driver.quit()

## Store the details in dictonary
billboard_hot_100={"Rank":ranks,
                  "Song":songs,
                  "Artist":artists,
                  "Last_Week_Rank":last_week_ranks,
                  "Peak_Rank":peak_ranks,
                  "Weeks_on_Board":weeks_on_board}
## Display the data in dataframe

df =pd.DataFrame(billboard_hot_100)
df
```

Out[7]:

	Rank	Song	Artist	Last_Week_Rank	Peak_Rank	Weeks_on_Board
0	1	Last Night	Morgan Wallen	1	1	21
1	2	Fast Car	Luke Combs	3	2	13
2	3	Calm Down	Rema & Selena Gomez	4	3	42
3	4	Flowers	Miley Cyrus	2	1	23
4	5	All My Life	Lil Durk Featuring J. Cole	5	2	6
...	...	...	...	...	...	...
95	96	Angel, Pt. 1	Kodak Black, NLE Choppa, Jimin, JVKE & Muni Long	-	65	2
96	97	Girl In Mine	Parmalee	-	97	1
97	98	Moonlight	Kali Uchis	90	80	11
98	99	Classy 101	Feid x Young Miko	-	99	1
99	100	Bluffin	Gucci Mane & Lil Baby	-	100	1

100 rows × 6 columns

6. Scrape the details of Highest sellingnovels.

Url = https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare

You have to find the following details:

A) Book name B) Author name C) Volumes sold D) Publisher E) Genre

In [8]:

```python
## Set up Chrome browser in headless mode
driver = webdriver.Chrome(options=chrome_options)
## Get the required link

## Handling WebDriverException taht occured , as the website's load is real slow
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## Find and click Top 100 songs
        driver.get("https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare")


        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved , Please Check your internet connection")


## Define empty lists for teh storage of scrapped data as required.
ranks =[]
titles=[]
authors=[]
v_s=[]
pubs=[]
genre=[]


try:
    for row in driver.find_elements(By.TAG_NAME,"tr"):
    # Extract the columns of each row
        columns = row.find_elements(By.TAG_NAME,"td")

    # Check if the row contains the required data
        if len(columns) >= 6:
        # Extract the details from the columns
            rank = columns[0].text.strip()
            title = columns[1].text.strip()
            author = columns[2].text.strip()
            Volume_sales = columns[3].text.strip()
            publisher = columns[4].text.strip()
            Genre = columns[5].text.strip()

           ## append the scrapped data
            ranks.append(rank)
            titles.append(title)
            authors.append(author)
            v_s.append(Volume_sales)
            pubs.append(publisher)
            genre.append(Genre)

except NoSuchElementException:
    pass

driver.quit()

## Put the data in dictonary
data={
     "Title":titles,
     "Author":authors,
     "Volume_Sold":v_s,
     "Publisher":pubs,
     "Genre":genre}
## display the data in dataframe

df=pd.DataFrame(data)
df
```

Out[8]:

	Title	Author	Volume_Sold	Publisher	Genre
0	Da Vinci Code,The	Brown, Dan	5,094,805	Transworld	Crime, Thriller & Adventure
1	Harry Potter and the Deathly Hallows	Rowling, J.K.	4,475,152	Bloomsbury	Children's Fiction
2	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4,200,654	Bloomsbury	Children's Fiction
3	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4,179,479	Bloomsbury	Children's Fiction
4	Fifty Shades of Grey	James, E. L.	3,758,936	Random House	Romance & Sagas
...	...	...	...	...	...
95	Ghost,The	Harris, Robert	807,311	Random House	General & Literary Fiction
96	Happy Days with the Naked Chef	Oliver, Jamie	794,201	Penguin	Food & Drink: General
97	Hunger Games,The:Hunger Games Trilogy	Collins, Suzanne	792,187	Scholastic Ltd.	Young Adult Fiction
98	Lost Boy,The:A Foster Child's Search for the L...	Pelzer, Dave	791,507	Orion	Biography: General
99	Jamie's Ministry of Food:Anyone Can Learn to C...	Oliver, Jamie	791,095	Penguin	Food & Drink: General

100 rows × 5 columns

7. Scrape the details most watched tv series of all time from imdb.com.

Url = https://www.imdb.com/list/ls095964455/ You have to find the following details: A) Name B) Year span C) Genre D) Run time E) Ratings F) Votes

In [9]:

```python
## Setup chrome browser in headless mode

driver =webdriver.Chrome(options=chrome_options)

## Handling WebDriverException taht occured , as the website's load is real slow
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## open imdb page

        driver.get("https://www.imdb.com/list/ls095964455/")

        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved , Please Check your internet connection")




## find all elements.
items = driver.find_elements(By.XPATH,'//div[@class="lister-item mode-detail"]')
## define and empty list for storage
imdb_df=[]

try:
    for item in items: ## iterate through items
        imdb={} ## define an empty dictonary
        ## scrape the required details
        title = item.find_element(By.XPATH,'.//h3[@class="lister-item-header"]//a').text
        year_span = item.find_element(By.XPATH,'.//span[@class="lister-item-year text-muted unbold"]').text
        genre = item.find_element(By.XPATH,'.//span[@class="genre"]').text
        runtime = item.find_element(By.XPATH,'.//span[@class="runtime"]').text
        rating = item.find_element(By.XPATH,'.//span[@class="ipl-rating-star__rating"]').text
        vote = item.find_element(By.XPATH,'.//p[@class="text-muted text-small"]//span[@name="nv"]').text


        ## append the scrapped deatils in dictonary
        imdb["Title"]=title
        imdb["Year_Span"]=year_span
        imdb["Genre"]=genre
        imdb["Runtime"]=runtime
        imdb["Rating"]=rating
        imdb["Vote"]= vote
        ## append the dictonary to the list
        imdb_df.append(imdb)
except NoSuchElementException:
    pass

## close the driver
driver.quit()

## display the list in dataframe
df = pd.DataFrame(imdb_df)
df
```

Out[9]:

	Title	Year_Span	Genre	Runtime	Rating	Vote
0	Game of Thrones	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,173,741
1	Stranger Things	(2016–2024)	Drama, Fantasy, Horror	51 min	8.7	1,251,569
2	The Walking Dead	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,032,509
3	13 Reasons Why	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	303,562
4	The 100	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	262,734
...	...	...	...	...	...	...
95	Reign	(2013–2017)	Drama	42 min	7.4	51,957
96	A Series of Unfortunate Events	(2017–2019)	Adventure, Comedy, Drama	50 min	7.8	63,995
97	Criminal Minds	(2005–)	Crime, Drama, Mystery	42 min	8.1	208,549
98	Scream	(2015–2019)	Comedy, Crime, Drama	45 min	7.1	43,403
99	The Haunting of Hill House	(2018)	Drama, Horror, Mystery	572 min	8.6	260,211

100 rows × 6 columns

8. Details of Datasets from UCI machine learning repositories. Url = https://archive.ics.uci.edu/

You have to find the following details: A) Dataset name B) Data type C) Task D) Attribute type E) No of instances F) No of attribute G) Year Note: - from the home page you have to go to the ShowAllDataset page through code.

In [10]:

```
## Set up a chrome browser
driver = webdriver.Chrome()

## Handling WebDriverException
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## open the given link
        driver.get("https://archive.ics.uci.edu/")

        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved , Please Check your internet connection")



## find and click All Datasets
driver.execute_script("arguments[0].click();",driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1]/div/div/div/a[1]'
```

In [11]:

```
## find and click expand all to scrappe the hidden details
expand = driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[2]/div[1]/div/label[2]/div[2]/span[1]')
driver.execute_script("arguments[0].click();", expand)
```

In [12]:

```
## Define empty lists

dataset_name=[]
task=[]
no_instance=[]
no_attribute=[]
data_type=[]
attribute_type=[]
year=[]

## Till the next page exists
while True:
    rows = driver.find_elements(By.XPATH,'//div[@role="row"]') ## find all element conatiners
    try:
        for row in rows:
            ## find dataset name
            d_name = row.find_element(By.XPATH,'.//h2[@class="truncate text-primary"]').text
            ## task , no of attribute and no of instance present under one column , so extarcting them one by one.
            cols = row.find_elements(By.XPATH,'.//div[@class="my-2 hidden gap-4 md:grid grid-cols-12"]/div')
            if len(cols)>=4:
                t = cols[0].text
                inst = cols[2].text
                att = cols[3].text
                ## append the scrapped data
                task.append(t)
                no_instance.append(inst)
                no_attribute.append(att)
            ## rest of the other features in other column by html design , extracting them one by one
            for trs in row.find_elements(By.TAG_NAME,'tr'):
                clms = trs.find_elements(By.TAG_NAME,'td')
                if len(clms)>=4:
                    d_type = clms[0].text
                    a_type = clms[1].text
                    y =clms[2].text.split("/")[-1]
                ## append the scrapped details accordingly
                    data_type.append(d_type)
                    attribute_type.append(a_type)
                    year.append(y)


            dataset_name.append(d_name)
    except StaleElementReferenceException: # handle stale element exceptin
            pass

        ## find and click next button
    next_button = driver.find_element(By.XPATH,'//button[@aria-label="Next Page"]')
        # check if next button is enabled
    if not next_button.is_enabled():
        break

    driver.execute_script("arguments[0].click();", next_button)

        #time.sleep(2)


## close the drievr
driver.quit()

## define the dictonary with scrapped data
data={"Dataset_Name":dataset_name,
     "Data Type":data_type,
     "Task":task,
     "No of Instance":no_instance,
     "No of attribute":no_attribute,
     "Attribute Type":attribute_type,
     "Year":year
     }

## Display the data in dataframe
df = pd.DataFrame(data)

df
```

Out[12]:

	Dataset_Name	Data Type	Task	No of Instance	No of attribute	Attribute Type	Year
0	Iris	Life Science	Classification	150 Instances	4 Attributes	Real	1988
1	Heart Disease	Life	Classification	303 Instances	13 Attributes	Categorical, Integer, Real	1988
2	Adult	Social	Classification	48.84K Instances	14 Attributes	Categorical, Integer	1996
3	Dry Bean Dataset	Computer	Classification	13.61K Instances	17 Attributes	Integer, Real	2020
4	Diabetes	Life			20 Attributes	Categorical, Integer	A
...	...	...	...	...	...	...	...
618	PMU-UD	Computer	Classification	5.18K Instances	9 Attributes		2018
619	Undocumented	Other				N/A	A
620	BAUM-2	Computer	Classification	1.05K Instances			2018
621	Connectionist Bench (Nettalk Corpus)	Other		20.01K Instances	4 Attributes	Categorical	1954
622	QtyT40I10D100K	Other		3.96M Instances	4 Attributes	Real	A

623 rows × 7 columns

9. Scrape the details of Data science recruiters

Url= https://www.naukri.com/hr-recruiters-consultants

You have to find the following details:

A) Name B) Designation C)Company D)Skills they hire for E) Location

Note: - From naukri.com homepage click on the recruiters option and the on the search pane type Data science and click on search. All this should be done through code.

In [13]:

```
# Setup the chrome browser in headless mode
driver = webdriver.Chrome()

## Load the given uRL.

## Handling WebDriverException
max_retries = 3 ##no of maximum retries
retry_delay = 2 ## retry dealy wait
for retry in range(max_retries):
    try:
        ## open the given link
        driver.get("https://www.naukri.com/hr-recruiters-consultants")

        break

    except WebDriverException as e: ## Handle the exception
        print("WebDriverException occurred on retry", retry + 1)
        print("Retrying in", retry_delay, "seconds...")
        time.sleep(retry_delay)
else:
    # If all retries fail, handle the exception
    print("All retries failed. WebDriverException could not be resolved , Please Check your internet connection")

## Define empty lists for storage.

names=[]
designations=[]
company_names=[]
skills=[]
locations=[]

while len(names)<=1000:
    ## Wait till the driver finds first job element
    wait = WebDriverWait(driver, 5)
    wait.until(EC.presence_of_element_located((By.XPATH,'//article[@class="jobTuple"]')))

    ## Select all jobs.
    jobs = driver.find_elements(By.XPATH,'//article[@class="jobTuple"]')

    ## Scrape all the required details
    try:
        for job in jobs:
            try:
                name = job.find_element(By.XPATH,'.//div[@class="info fleft"]//a').text
            except:
                name='-'
            try:
                des = job.find_element(By.XPATH,'.//div[@class="info fleft"]//a').text.split('HR')[1]
            except:
                des='-'
            try:
                company = job.find_element(By.XPATH,'.//div[@class="companyInfo subheading"]//a').text
            except:
                comapny='-'
            try:
                loc = job.find_element(By.XPATH,'.//li[@class="fleft br2 placeHolderLi location"]').text
            except:
                loc='-'
            try:

                skill = job.find_element(By.XPATH,'.//ul[@class="tags has-description"]').text.strip('/n')
            except:
                skill='-'
            ## Append all the scrapped details
            names.append(name)
            company_names.append(company)
            designations.append(des)
            locations.append(loc)
            skills.append(skill)

    ## If exception rise : continue
    except NoSuchElementException:
         continue
    ## try to find Next button on this page.
    try:
        ## Wait till next button is found
        wait_2 = WebDriverWait(driver, 10)
        wait_2.until(EC.presence_of_element_located((By.XPATH,'//a[@class="fright fs14 btn-secondary br2"]')))
        ## Click on next button
        next_button = driver.find_element(By.XPATH,'//a[@class="fright fs14 btn-secondary br2"]')
        driver.execute_script("arguments[0].click();", next_button)
    ## If Exception rises , try again
    except NoSuchElementException:
        max_retries = 2
        retry_delay = 2
        for retry in range(max_retries):
            next_button = driver.find_element(By.XPATH,'//a[@class="fright fs14 btn-secondary br2"]')

            if not next_button.is_enabled():
                break

            driver.execute_script("arguments[0].click();", next_button)
            time.sleep(2)

## Print no of jobs scarpped.
try:
    elements_displayed = driver.find_element(By.XPATH,'//div[@class="sortAndH1Cont"]').text.split()
    print(elements_displayed[2],"Out of",elements_displayed[4],"HR Jobs are scrapped" )

# If exception rises, wait yill driver finds the element and then print
except NoSuchElementException:
    wait_3 = WebDriverWait(driver, 10)

    wait_3.until(EC.presence_of_element_located((By.XPATH,'//div[@class="sortAndH1Cont"]')))

    elements_displayed = driver.find_element(By.XPATH,'//div[@class="sortAndH1Cont"]//div[@class="h1-wrapper"]').text.split()

    print("Exception raised and Handled")
    print(elements_displayed[0],"Out of",elements_displayed[4],"HR Jobs are scrapped" )

driver.quit()

# Store the scrapped details in dictonary
jobs_df={"Name":names,
         "Dsignation" : designations,
        "Company":company_names,

        "Location":locations,
        "Skills":skills}

## Display in dataframe
df = pd.DataFrame(jobs_df)
df
```

```
Exception raised and Handled
1021 Out of 15506 HR Jobs are scrapped
```

Out[13]:

	Name	Dsignation	Company	Location	Skills
0	Opening For Management Trainee / Executive - HR		Sahajanand Medical Technologies	Mumbai (All Areas)	Recruitment\nTalent Acquisition\nTraining\nMIS...
1	Hiring Freshers : HR Executive: Recruiter-Guru...	Executive: Recruiter-Gurugram : ACS	Advance Career Solutions	Gurgaon/ Gurugram, Haryana	communication skills\nRecruitment\nHiring\nAcd...
2	Executive/ Assistant Manager HR Generalist - P...	Generalist - Pune (Dress Code)	OASIS	Pune, Maharashtra(Koregaon Park)	hr generalist activities\nHR Information Syste...
3	Assistant Manager - HR (Field Level Recruitment)	(Field Level Recruitment)	Muthoot Microfin	Bhubaneswar, Odisha, Hubli, Karnataka, Sambalp...	NBFC\nrecruitment\nMass Hiring\nBulk Hiring\nL...
4	HR Recruiter	Recruiter	Symphoni Hr	Remote	Recruitment\nExit formalities\nTalent acquisit...
...	...	...	...	...	...
1015	HR Executive	Executive	Manpower Resources India	Jamshedpur, Jharkhand	HR Generalist Activities\nplant hr\nHR Operati...
1016	HR Exec/ Human Resources Executive/ Lead HR/ B...	Exec/ Human Resources Executive/ Lead	Selectica International Solutions Llp	Thane, Maharashtra, Pune, Maharashtra, Mumbai ...	BPO Hiring\nHR\nCampus hiring\nBPO\nBulk\nTale...
1017	Executive - HR & Compliance	& Compliance	Peoplepro Management Services	Kolkata, Durgapur, West Bengal, Howrah, West B...	Payroll Management\nLaw\nGeneralist Activities...
1018	Urgent requirement For HR Executives	Executives	Kamms Management Consultants	Chennai, Tamil Nadu	RECRUITER\nResource\nManagement\nHrsd\nRequire...
1019	HR Executive- Payroll	Executive- Payroll	Megma Services	Delhi / NCR	HR\nVerification\nProcess\nReconciliation\nHrs...

1020 rows × 5 columns