In [3]:

```
import selenium
from selenium import webdriver
import pandas as pd
from selenium.webdriver.common.by import By
import warnings
warnings.filterwarnings("ignore")
import time

from bs4 import BeautifulSoup
import pandas as pd
from tabulate import tabulate
import requests
import urllib.request
```

In [ ]:

```
#Name-neha somvanshi
#project_name- web scraping assignment 4
#company_name- FlipRobo
#Batch number- DS2307
```

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

In [12]:

```
url='https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos'
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_content=[]
content=soup.find_all('table',class_="wikitable")
for x in content:
    table_content.append(x.text)

df=pd.read_html(str(content))[0]
df=df.drop(df.index[-1])
df.index = df.index + 1
df.drop(df.columns[-1], axis=1, inplace=True)
df.drop(df.columns[-1], axis=1, inplace=True)
df
```

Out[12]:

	No.	Video name	Uploader	Views (billions)	Publication date
1	1.	"Baby Shark Dance"[6]	Pinkfong Baby Shark - Kids' Songs & Stories	13.36	June 17, 2016
2	2.	"Despacito"[9]	Luis Fonsi	8.26	January 12, 2017
3	3.	"Johny Johny Yes Papa"[16]	LooLoo Kids - Nursery Rhymes and Children's Songs	6.80	October 8, 2016
4	4.	"Bath Song"[17]	Cocomelon - Nursery Rhymes	6.41	May 2, 2018
5	5.	"Shape of You"[18]	Ed Sheeran	6.08	January 30, 2017
6	6.	"See You Again"[21]	Wiz Khalifa	6.03	April 6, 2015
7	7.	"Wheels on the Bus"[26]	Cocomelon - Nursery Rhymes	5.56	May 24, 2018
8	8.	"Phonics Song with Two Words"[27]	ChuChu TV Nursery Rhymes & Kids Songs	5.48	March 6, 2014
9	9.	"Uptown Funk"[28]	Mark Ronson	5.03	November 19, 2014
10	10.	"Learning Colors – Colorful Eggs on a Farm"[29]	Miroshka TV	4.97	February 27, 2018
11	11.	"Gangnam Style"[30]	officialpsy	4.90	July 15, 2012
12	12.	"Masha and the Bear – Recipe for Disaster"[35]	Get Movies	4.56	January 31, 2012
13	13.	"Dame Tu Cosita"[36]	Ultra Records	4.44	April 5, 2018
14	14.	"Axel F"[37]	Crazy Frog	4.06	June 16, 2009
15	15.	"Sugar"[38]	Maroon 5	3.93	January 14, 2015
16	16.	"Counting Stars"[39]	OneRepublic	3.87	May 31, 2013
17	17.	"Roar"[40]	Katy Perry	3.87	September 5, 2013
18	18.	"Baa Baa Black Sheep"[41]	Cocomelon - Nursery Rhymes	3.78	June 25, 2018
19	19.	"Waka Waka (This Time for Africa)"[42]	Shakira	3.73	June 4, 2010
20	20.	"Sorry"[43]	Justin Bieber	3.71	October 22, 2015
21	21.	"Lakdi Ki Kathi"[44]	Jingle Toons	3.69	June 14, 2018
22	22.	"Thinking Out Loud"[45]	Ed Sheeran	3.66	October 7, 2014
23	23.	"Dark Horse"[46]	Katy Perry	3.59	February 20, 2014
24	24.	"Humpty the train on a fruits ride"[47]	Kiddiestv Hindi - Nursery Rhymes & Kids Songs	3.55	January 26, 2018
25	25.	"Perfect"[48]	Ed Sheeran	3.54	November 9, 2017
26	26.	"Faded"[49]	Alan Walker	3.51	December 3, 2015

	No.	Video name	Uploader	Views (billions)	Publication date
27	27.	"Let Her Go"[50]	Passenger	3.51	July 25, 2012
28	28.	"Girls Like You"[51]	Maroon 5	3.48	May 31, 2018
29	29.	"Lean On"[52]	Major Lazer Official	3.46	March 22, 2015
30	30.	"Bailando"[53]	Enrique Iglesias	3.45	April 11, 2014

2. Scrape the details team India’s international fixtures from bcci.tv.

In [23]:

```
driver=webdriver.Chrome()
```

In [24]:

```
driver.get('https://www.bcci.tv/')
```

In [25]:

```
international=driver.find_element(By.XPATH,'/html/body/nav/div[1]/div[2]/ul[1]/li[2
international.click()
```

In [46]:

```
Match_title=[]
title=driver.find_elements(By.XPATH,'//h5[@class="match-tournament-name ng-binding'
for x in title:
    Match_title.append(x.text)
```

In [47]:

```
Match_series=[]
series=driver.find_elements(By.XPATH,'//div[@class="match-card-middle__inner d-fle
for x in series:
    Match_series.append(x.text)
```

In [48]:

```
Match_place=[]
place=driver.find_elements(By.XPATH,'//span[@class="ng-binding"]')
for x in place:
    Match_place.append(x.text)
```

In [49]:

```
Match_date=[]
date=driver.find_elements(By.XPATH,'//div[@class="match-dates ng-binding"]')
for x in date:
    Match_date.append(x.text)
```

In [50]:

```
Match_time=[]
time=driver.find_elements(By.XPATH,'//div[@class="match-time no-margin ng-binding"]
for x in time:
    Match_time.append(x.text)
```

In [51]:

```
df=pd.DataFrame({'Match_title':Match_title,'Match_series':Match_series,'Match_place
df
```

Out[51]:

	Match_title	Match_series	Match_place	Match_date	Match_time
0	19TH ASIAN GAMES HANGZHOU 2022	India Women\nvs\nTBD	Hangzhou	21 SEP 2023	6:30 AM IST
1	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Mohali	22 SEP 2023	1:30 PM IST
2	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Indore	24 SEP 2023	1:30 PM IST
3	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Rajkot	27 SEP 2023	1:30 PM IST
4	ICC MENS WORLD CUP 2023 WARM-UP MATCHES	India\nvs\nEngland	Guwahati	30 SEP 2023	2:00 PM IST
5	19TH ASIAN GAMES HANGZHOU 2022	India\nvs\nTBD	Hangzhou	3 OCT 2023	6:30 AM IST
6	ICC MENS WORLD CUP 2023 WARM-UP MATCHES	India\nvs\nNetherlands	Thiruvananthapuram	3 OCT 2023	2:00 PM IST
7	ICC MENS WORLD CUP 2023	India\nvs\nAustralia	Chennai	8 OCT 2023	2:00 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)

In [57]:

```
url='https://www.statisticstimes.com/economy/india/indian-states-gdp.php'
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_content=[]
content=soup.find_all('table',class_="display")
for x in content:
    table_content.append(x.text)

df=pd.read_html(str(content))[0]
df=df.drop(df.index[-1])
df.index=df.index+1
df
```

Out[57]:

	Rank	State	GSDP (Cr INR at Current prices)		Share	GDP ($billion)	GSDP (Cr INR at 2011-12 prices)	
	Rank	State	19-20	18-19	18-19	2019	19-20	18-19
1	1.0	Maharashtra	-	2632792	13.94%	399.921	-	2039074
2	2.0	Tamil Nadu	1845853	1630208	8.63%	247.629	1312929	1215307
3	3.0	Uttar Pradesh	1687818	1584764	8.39%	240.726	1166817	1123982
4	4.0	Gujarat	-	1502899	7.96%	228.290	-	1186379
5	5.0	Karnataka	1631977	1493127	7.91%	226.806	1156039	1091077
6	6.0	West Bengal	1253832	1089898	5.77%	165.556	793223	739525
7	7.0	Rajasthan	1020989	942586	4.99%	143.179	711627	677428
8	8.0	Andhra Pradesh	972782	862957	4.57%	131.083	672018	621301
9	9.0	Telangana	969604	861031	4.56%	130.791	663258	612828
10	10.0	Madhya Pradesh	906672	809592	4.29%	122.977	561801	522009
11	11.0	Kerala	-	781653	4.14%	118.733	-	559412
12	12.0	Delhi	856112	774870	4.10%	117.703	634408	590569
13	13.0	Haryana	831610	734163	3.89%	111.519	572240	531085
14	14.0	Bihar	611804	530363	2.81%	80.562	414977	375651
15	15.0	Punjab	574760	526376	2.79%	79.957	418868	397669
16	16.0	Odisha	521275	487805	2.58%	74.098	396499	376877
17	17.0	Assam	-	315881	1.67%	47.982	-	234048
18	18.0	Chhattisgarh	329180	304063	1.61%	46.187	243477	231182
19	19.0	Jharkhand	328598	297204	1.57%	45.145	240036	224986
20	20.0	Uttarakhand	-	245895	1.30%	37.351	-	193273
21	21.0	Jammu & Kashmir	-	155956	0.83%	23.690	-	112755
22	22.0	Himachal Pradesh	165472	153845	0.81%	23.369	124403	117851
23	23.0	Goa	80449	73170	0.39%	11.115	63408	57787
24	24.0	Tripura	55984	49845	0.26%	7.571	40583	36963
25	25.0	Chandigarh	-	42114	0.22%	6.397	-	31192
26	26.0	Puducherry	38253	34433	0.18%	5.230	25093	23013
27	27.0	Meghalaya	36572	33481	0.18%	5.086	26695	24682
28	28.0	Sikkim	32496	28723	0.15%	4.363	20017	18722
29	29.0	Manipur	31790	27870	0.15%	4.233	20673	19300
30	30.0	Nagaland	-	27283	0.14%	4.144	-	17647
31	31.0	Arunachal Pradesh	-	24603	0.13%	3.737	-	16676
32	32.0	Mizoram	26503	22287	0.12%	3.385	18797	16478

	Rank	State	GSDP (Cr INR at Current prices)		Share	GDP ($billion)	GSDP (Cr INR at 2011-12 prices)	
	Rank	State	19-20	18-19	18-19	2019	19-20	18-19
33	33.0	Andaman & Nicobar Islands	-	-	-	-	-	-

4. Scrape the details of trending repositories on Github.com.

In [25]:

```
driver=webdriver.Chrome()
```

In [26]:

```
driver.get('https://github.com/')
```

In [32]:

```
repository_title=[]
title=driver.find_elements(By.XPATH,'//h2[@class="h3 lh-condensed"]')[:20]
for x in title:
    repository_title.append(x.text)
```

In [33]:

```
repository_description=[]
description=driver.find_elements(By.XPATH,'//p[@class="col-9 color-fg-muted my-1 pr
for x in description:
    repository_description.append(x.text)
```

In [34]:

```
Contributors_count=[]
count=driver.find_elements(By.XPATH,'//span[@class="d-inline-block float-sm-right"]
for x in count:
    Contributors_count.append(x.text)
```

In [35]:

```
Language_used=[]
language=driver.find_elements(By.XPATH,'//span[@itemprop="programmingLanguage"]')[
for x in language:
    Language_used.append(x.text)
```

In [36]:

```
df=pd.DataFrame({'repository_title':repository_title,'repository_description':repos
df[:20]
```

Out[36]:

	repository_title	repository_description	Contributors_count	Language_used
0	hyperdxio / hyperdx	Resolve production issues, fast. An open sourc...	573 stars today	TypeScript
1	williamyang1991 / Rerender_A_Video	[SIGGRAPH Asia 2023] Rerender A Video: Zero-Sh...	740 stars today	Jupyter Notebook
2	NExT-GPT / NExT-GPT	Code and models for NExT-GPT: Any-to-Any Multi...	317 stars today	Python
3	makeplane / plane	🔥 🔥 🔥 Open Source JIRA, Linear and Height Alte...	398 stars today	TypeScript
4	grafana / beyla	eBPF-based autoinstrumentation of HTTP and HTT...	97 stars today	C
5	CorentinJ / Real-Time-Voice-Cloning	Clone a voice in 5 seconds to generate arbitra...	182 stars today	Python
6	mastodon / mastodon	Your self-hosted, globally interconnected micr...	17 stars today	Ruby
7	AntonioErdeljac / next13-lms-platform	Self-hosted AI coding assistant	29 stars today	TypeScript
8	TabbyML / tabby	Deploy web apps anywhere.	327 stars today	TypeScript
9	basecamp / kamal	Godot Engine – Multi-platform 2D and 3D game e...	199 stars today	Ruby
10	godotengine / godot	The React Framework	468 stars today	C++
11	vercel / next.js	A list of SaaS, PaaS and IaaS offerings that h...	116 stars today	JavaScript
12	ripienaar / free-for-dev	An Open-source Framework for Autonomous Langua...	299 stars today	HTML
13	aiwaves-cn / agents	Data set of top third party web domains with r...	343 stars today	Python
14	duckduckgo / tracker-radar	Elegant HTTP Networking in Swift	68 stars today	JavaScript
15	Alamofire / Alamofire	🐸 💬 - a deep learning toolkit for Text-to-Speec...	12 stars today	Swift
16	coqui-ai / TTS	[ICCV 2023] ProPainter: Improving Propagation ...	977 stars today	Python
17	sczhou / ProPainter	Toy Gaussian Splatting visualization in Unity	155 stars today	Python
18	aras-p / UnityGaussianSplatting	「Java学习+面试指南」一份涵盖大部分 Java 程序员所需要掌握的核心知识。准备 Jav...	35 stars today	C#
19	Snailclimb / JavaGuide	The paper list of the 86-page paper "The Rise ...	45 stars today	Java

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

In [43]:

```
driver=webdriver.Chrome()
```

In [44]:

```
driver.get('https://www.billboard.com/charts/hot-')
```

In [ ]:

```
click_chart_2=driver.find_element(By.XPATH,'/html/body/div[3]/div[9]/div/div/div/u
click_chart_2.click()
```

In [48]:

```
Song_name=[]
song=driver.find_elements(By.XPATH,'//h3[@class="c-title  a-no-trucate a-font-prima
for x in song:
    Song_name.append(x.text)
```

In [46]:

```
Artist_name=[]
name=driver.find_elements(By.XPATH,'//span[@class="c-label  a-no-trucate a-font-pri
for x in name:
    Artist_name.append(x.text)
```

In [49]:

```
df=pd.DataFrame({'Song_name':Song_name,'Artist_name':Artist_name})
df
```

Out[49]:

	Song_name	Artist_name
0	Paint The Town Red	Doja Cat
1	I Remember Everything	Zach Bryan Featuring Kacey Musgraves
2	Fast Car	Luke Combs
3	Cruel Summer	Taylor Swift
4	Last Night	Morgan Wallen
...	...	...
94	Lagunas	Peso Pluma & Jasiel Nunez
95	Sittin' On Top Of The World	Burna Boy
96	Rubicon	Peso Pluma
97	TQM	Fuerza Regida
98	Amargura	Karol G

99 rows × 2 columns

6.Scrape the details of Highest selling novels.

In [8]:

```
url='https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-t
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_content=[]
content=soup.find_all('table',class_="in-article sortable")
for x in content:
    table_content.append(x.text)
```

```
df=pd.read_html(str(content))[0]
df=df.drop(df.index[-1])
df.index=df.index+1
df
```

Out[8]:

	Rank	Title	Author	Volume Sales	Publisher	Genre
1	1	Da Vinci Code,The	Brown, Dan	5094805	Transworld	Crime, Thriller & Adventure
2	2	Harry Potter and the Deathly Hallows	Rowling, J.K.	4475152	Bloomsbury	Children's Fiction
3	3	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4200654	Bloomsbury	Children's Fiction
4	4	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4179479	Bloomsbury	Children's Fiction
5	5	Fifty Shades of Grey	James, E. L.	3758936	Random House	Romance & Sagas
...	...	...	...	...	...	...
96	96	Ghost,The	Harris, Robert	807311	Random House	General & Literary Fiction
97	97	Happy Days with the Naked Chef	Oliver, Jamie	794201	Penguin	Food & Drink: General
98	98	Hunger Games,The:Hunger Games Trilogy	Collins, Suzanne	792187	Scholastic Ltd.	Young Adult Fiction
99	99	Lost Boy,The:A Foster Child's Search for the L...	Pelzer, Dave	791507	Orion	Biography: General
100	100	Jamie's Ministry of Food:Anyone Can Learn to C...	Oliver, Jamie	791095	Penguin	Food & Drink: General

100 rows × 6 columns

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

In [18]:
```
driver=webdriver.Chrome()
```

In [19]:
```
driver.get('https://www.imdb.com/list/ls095964455/')
```

In [30]:
```
name_shows=[]
name=driver.find_elements(By.XPATH,'//h3[@class="lister-item-header"]')
for x in name:
    name_shows.append(x.text)
```

In [22]:
```
Year_span=[]
year=driver.find_elements(By.XPATH,'//span[@class="lister-item-year text-muted unbo
for x in year:
    Year_span.append(x.text)
```

In [23]:
```
Genre_shows=[]
Genre=driver.find_elements(By.XPATH,'//span[@class="genre"]')
```

```
for x in Genre:
    Genre_shows.append(x.text)
```

In [24]:

```
Run_time=[]
time=driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for x in time:
    Run_time.append(x.text)
```

In [25]:

```
Ratings_show=[]
Ratings=driver.find_elements(By.XPATH,'//div[@class="ipl-rating-star small"]')
for x in Ratings:
    Ratings_show.append(x.text)
```

In [27]:

```
Votes_shows=[]
Votes=driver.find_elements(By.XPATH,'//span[@name="nv"]')
for x in Votes:
    Votes_shows.append(x.text)
```

In [32]:

```
df=pd.DataFrame({'name_shows':name_shows,'Year_span':Year_span,'Genre_shows':Genre_
df
```

Out[32]:

	name_shows	Year_span	Genre_shows	Run_time	Ratings_show	Votes_shows
0	1. Game of Thrones (2011–2019)	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,204,953
1	2. Stranger Things (2016–2025)	(2016–2025)	Drama, Fantasy, Horror	51 min	8.7	1,276,338
2	3. The Walking Dead (2010–2022)	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,046,200
3	4. 13 Reasons Why (2017–2020)	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	307,473
4	5. The 100 (2014–2020)	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	266,553
...	...	...	...	...	...	...
95	96. Reign (2013–2017)	(2013–2017)	Drama	42 min	7.5	52,745
96	97. A Series of Unfortunate Events (2017–2019)	(2017–2019)	Adventure, Comedy, Drama	50 min	7.8	64,808
97	98. Criminal Minds (2005–)	(2005–)	Crime, Drama, Mystery	42 min	8.1	211,103
98	99. Scream: The TV Series (2015–2019)	(2015–2019)	Comedy, Crime, Drama	45 min	7	43,917
99	100. The Haunting of Hill House (2018)	(2018)	Drama, Horror, Mystery	572 min	8.6	266,172

100 rows × 6 columns

8. Details of Datasets from UCI machine learning repositories.

In [17]:

```
driver=webdriver.Chrome()
```

In [18]:

```
driver.get('https://archive.ics.uci.edu/')
```

In [20]:

```
#data is not scrapable
```