

name = Divyanshu singh

batch = DS2306

id = 56

```
In [1]: import selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.common.exceptions import NoSuchElementException, StaleElementReferenceException
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
import time
import re

import warnings
warnings.filterwarnings('ignore')
```

.

1

```
In [2]: driver = webdriver.Chrome()
driver.get("https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos")
```

```
In [8]: Rank = []
Ra = driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tr')
for i in Ra:
    Rank.append(i.text)
```

```
In [9]: Rank
```

```
Out[9]: ['Top 30 most-viewed YouTube videos[5]\nNo. Video name Uploader Views (billions) Publication date Note\n1. "Baby Shark Dance"[6] Pinkfong Baby Shark - children's songs 1 3.18 June 17, 2016 [A]\n2. "Despacito"[9] Luis Fonsi 8.23 January 12, 2017 [B]\n3. "J ohny Johny Yes Papa"[16] LooLoo Kids - nursery rhymes 6.76 October 8, 2016\n4. "Bath Song"[17] Cocomelon - nursery rhymes 6.33 May 2, 2018\n5. "Shape of You"[18] Ed Sheeran 6.05 January 30, 2017 [C]\n6. "See You Again"[21] Wiz Khalifa 5.98 April 6, 2015 [D]\n7. "Wheels on the Bus"[26] Cocomelon - nursery rhymes 5.46 May 24, 2018\n8. "Phonics Song with Two Words"[27] ChuChu TV - children's songs 5.42 March 6, 2014\n9. "U ptown Funk"[28] Mark Ronson 5.00 November 19, 2014\n10. "Learning Colors - Colorful Eggs on a Farm"[29] Miroshka TV - children's songs 4.94 February 27, 2018\n11. "Gangnam Style"[30] Psy 4.86 July 15, 2012 [E]\n12. "Masha and the Bear - Recipe for Disaster"[35] Get Movies - children's songs 4.55 January 31, 2012\n13. "Dame Tu Cosita"[36] El Chombo 4.41 April 5, 2018\n14. "Axel F"[37] Crazy Frog 4.00 June 16, 2009\n15. "Sugar"[38] Maroon 5 3.91 January 14, 2015\n16. "Roar"[39] Katy Perry 3.84 September 5, 2013\n17. "Counting Stars"[40] OneRepublic 3.84 May 31, 2013\n18. "Baa Baa Black Sheep"[41] Cocomelon - nursery rhymes 3.73 June 25, 2018\n19. "Sorry"[42] Justin Bieber 3.69 October 22, 2015\n20. "Waka Waka (This Time for Africa)"[43] Shakira 3.68 June 4, 2010\n21. "Thinking Out Loud"[44] Ed Sheeran 3.63 October 7, 2014\n22. "Lakdi Ki Kathi"[45] Jingle Toons 3.63 June 14, 2018\n23. "Dark Horse"[46] Katy Perry 3.56 February 20, 2014\n24. "Perfect"[47] Ed Sheeran 3.51 November 9, 2017\n25. "Faded"[48] Alan Walker 3.49 December 3, 2015\n26. "Let Her Go"[49] Passenger 3.48 July 25, 2012\n27. "Humpty the train on a fruits ride"[50] Kiddiestv Hindi - children's songs 3.51 January 26, 2018\n28. "Girls Like You"[51] Maroon 5 3.45 May 31, 2018\n29. "Bailando"[52] Enrique Iglesias 3.43 April 11, 2014\n30. "Lean On"[53] Major Lazer 3.43 March 22, 2015\nAs of August 8, 2023',
```

'Progression of the most-viewed video on YouTube\nVideo name Uploader Views at\nevent* Publication date Date achieved Days after\ncreation Days\neheld Takedown\ndate Ref Notes\n"Baby Shark Dance"[6] Pinkfong Baby Shark - Kids' Songs & Stories 7,046,700,000 June 17, 2016 November 2, 2020 1600 1,019\n"Despacito"[9] Luis Fonsi 2,993,700,000 January 12, 2017 August 4, 2017 206 1,186 [65] [B]\n"See You Again"[21] Wiz Khalifa 2,894,000,000 April 6, 2015 July 10, 2017 826 25 [22]\n"Gangnam Style"*[30] Psy 803,700,000 July 15, 2012 November 24, 2012 134 1,689 [31] [E]\n"Baby"*[66] Justin Bieber 245,400,000 February 19, 2010 July 16, 2010 149 862 [67] [F]\n"Bad Romance"[70] Lady Gaga 178,400,000 November 24, 2009 April 14, 2010 143 93 [71] [G]\n"Charlie Bit My Finger"#[74] HDCYT 128,900,000 May 22, 2007 October 25, 2009 887 171 late July 2021[75]\n[76]\n"Evolution of Dance"[77] Judson Laippy 118,900,000 April 6, 2006 May 2, 2009 1062 176 [78]\n"Girlfriend"#[79][80] RCA Records 92,600,000 February 27, 2007 July 17, 2008 508 289 no data [81] [H]\n"Evolution of Dance"[77] Judson Laippy 78,400,000 April 6, 2006 March 15, 2008 651 124 [83]\n"Music Is My Hot Hot Sex"#[84] CLARUSB ARTEL72 76,600,000 April 9, 2007 March 1, 2008 327 14 mid-2008[85] [86] [I]\n"Evolution of Dance"*[77] Judson Laippy 10,600,000 April 6, 2006 May 19, 2006 16 652 [89] [J]\n"Pokémon Theme Music Video"#[90] Smosh 4,300,000 November 28, 2005 March 12, 2006 105 68 June 2007[91] [92] [K]\n"MySpace - The Movie"#[95][96] eggtea 2,700,000 January 31, 2006 February 18, 2006 18 22 mid-2006[97] [98]\n"Phony Photo Booth"#[99] mugenized 3,400,000 December 1, 2005 January 21, 2006 50 28 no data [100] [L]\n"The Chronic of Narnia Rap"#[101] youtubedude 2,300,000 December 18, 2005 January 9, 2006 22 12 no data [102]\n"Cross Bar"#[103] joeB 1,000,000 October 21, 2005 December 10, 2005 50 30 no data [104][103]\n"Ronaldinho: Touch of Gold"#[105] Nikesoccer 255,000 October 21, 2005 October 31, 2005 10 40 no data [106] [M]\n"I/O Brush"#[108] larfus 247,000 October 5, 2005 October 29, 2005 24 2 January to April 2007[109] [110]\n"Me at the Zoo"#[111] jawed 1 April 23, 2005 April 23, 2005 0 189 [112]\nAs of August 18, 2023'

In []:

In []:

- A) Rank
- B) Name
- C) Artist
- D) Upload date
- E) Views

```
In [ ]: driver.close()
```

2

1

```
In [2]: driver = webdriver.Chrome()
        driver.get("https://www.bcci.tv")
```

```
In [7]: view_all = driver.find_element(By.XPATH, '/html/body/div[5]/div/div[2]/div/div[2]/div/c  
view_all.click()
```

```
In [55]: df = pd.DataFrame({'Match_title':Match_title})  
df
```

Out[55]: **Match_title**

0	1st T20I -
1	2nd T20I -
2	3rd T20I -
3	1st ODI -
4	2nd ODI -
5	1st ODI -
6	2nd ODI -
7	3rd ODI -
8	1st T20I -
9	2nd T20I -
10	3rd T20I -
11	1st ODI -
12	2nd ODI -
13	1st ODI -
14	2nd ODI -
15	3rd ODI -
16	1st T20I -
17	2nd T20I -
18	3rd T20I -
19	1st ODI -
20	2nd ODI -
21	1st ODI -
22	2nd ODI -
23	3rd ODI -

```
In [77]: Place = []
start = 0
end = 3
for page in range(start,end):
    Pla = driver.find_elements(By.XPATH, '//span[@class="ng-binding"]')
    for i in Pla:
```

```
Place.append(i.text)
next_button = driver.find_element(By.XPATH, '/html/body/div[2]/div[2]/div/div/div/c
```

```
In [75]: Date = []
start = 0
end = 3
for page in range(start,end):
    Da = driver.find_elements(By.XPATH, '//div[@class="match-dates ng-binding"]')
    for i in Da:
        Date.append(i.text)
next_button = driver.find_element(By.XPATH, '/html/body/div[2]/div[2]/div/div/div/c
```

```
In [76]: Time = []
start = 0
end = 3
for page in range(start,end):
    Ti = driver.find_elements(By.XPATH, '//div[@class="match-time no-margin ng-binding"]')
    for i in Ti:
        Time.append(i.text)
next_button = driver.find_element(By.XPATH, '/html/body/div[2]/div[2]/div/div/div/c
```

- ```
In []: A) Match title (I.e. 1 ODI)
B) Series
C) Place
D) Date
E) Time
```

```
In [78]: df = pd.DataFrame({'Match_title':Match_title,'Series':Series,'Place':Place,'Date':Date,
df
```

|    | Match_title                               | Series    | Place       | Date        | Time |
|----|-------------------------------------------|-----------|-------------|-------------|------|
| 0  | 1st T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 18 AUG 2023 | 7:00 AM PDT |      |
| 1  | 2nd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 20 AUG 2023 | 7:00 AM PDT |      |
| 2  | 3rd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 23 AUG 2023 | 7:00 AM PDT |      |
| 3  | 1st ODI - ASIA CUP 2023                   | Pallekele | 1 SEP 2023  | 9:30 PM PDT |      |
| 4  | 2nd ODI - ASIA CUP 2023                   | Pallekele | 3 SEP 2023  | 9:30 PM PDT |      |
| 5  | 1st ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Mohali    | 22 SEP 2023 | 1:00 AM PDT |      |
| 6  | 2nd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Indore    | 24 SEP 2023 | 1:00 AM PDT |      |
| 7  | 3rd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Rajkot    | 27 SEP 2023 | 1:00 AM PDT |      |
| 8  | 1st T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 18 AUG 2023 | 7:00 AM PDT |      |
| 9  | 2nd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 20 AUG 2023 | 7:00 AM PDT |      |
| 10 | 3rd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 23 AUG 2023 | 7:00 AM PDT |      |
| 11 | 1st ODI - ASIA CUP 2023                   | Pallekele | 1 SEP 2023  | 9:30 PM PDT |      |
| 12 | 2nd ODI - ASIA CUP 2023                   | Pallekele | 3 SEP 2023  | 9:30 PM PDT |      |
| 13 | 1st ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Mohali    | 22 SEP 2023 | 1:00 AM PDT |      |
| 14 | 2nd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Indore    | 24 SEP 2023 | 1:00 AM PDT |      |
| 15 | 3rd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Rajkot    | 27 SEP 2023 | 1:00 AM PDT |      |
| 16 | 1st T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 18 AUG 2023 | 7:00 AM PDT |      |
| 17 | 2nd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 20 AUG 2023 | 7:00 AM PDT |      |
| 18 | 3rd T20I - INDIA TOUR OF IRELAND 2023     | Dublin    | 23 AUG 2023 | 7:00 AM PDT |      |
| 19 | 1st ODI - ASIA CUP 2023                   | Pallekele | 1 SEP 2023  | 9:30 PM PDT |      |
| 20 | 2nd ODI - ASIA CUP 2023                   | Pallekele | 3 SEP 2023  | 9:30 PM PDT |      |
| 21 | 1st ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Mohali    | 22 SEP 2023 | 1:00 AM PDT |      |
| 22 | 2nd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Indore    | 24 SEP 2023 | 1:00 AM PDT |      |
| 23 | 3rd ODI - AUSTRALIA TOUR OF INDIA 2023-24 | Rajkot    | 27 SEP 2023 | 1:00 AM PDT |      |

In [79]: `df.to_csv('team_India's_international_fixtures.csv')`

In [80]: `driver.close()`

3

```
In [2]: driver = webdriver.Chrome()
driver.get('http://statisticstimes.com/')
```

```
In [3]: economy = driver.find_element(By.XPATH, '/html/body/div[2]/div[1]/div[2]/div[2]/button')
economy.click()
india = driver.find_element(By.XPATH, '/html/body/div[2]/div[1]/div[2]/div[2]/div/a[3]')
india.click()
```

```
In [5]: gdp_states = driver.find_element(By.XPATH, '/html/body/div[2]/div[2]/div[2]/ul/li[1]/a')
gdp_states.click()
```

```
In [6]: Rank = []
Ra = driver.find_elements(By.XPATH, '//td[@class="data1"]')
for i in Ra:
 Rank.append(i.text)
```

```
In [71]: Rank = Rank[:33]
Rank
```

```
Out[71]: ['1',
'2',
'3',
'4',
'5',
'6',
'7',
'8',
'9',
'10',
'11',
'12',
'13',
'14',
'15',
'16',
'17',
'18',
'19',
'20',
'21',
'22',
'23',
'24',
'25',
'26',
'27',
'28',
'29',
'30',
'31',
'32',
'33']
```

```
In [8]: State = []
St = driver.find_elements(By.XPATH, '//td[@class="name"]')
for i in St:
 State.append(i.text)
```

```
In [72]: State = State[:33]
State
```

```
Out[72]: ['Maharashtra',
 'Tamil Nadu',
 'Uttar Pradesh',
 'Gujarat',
 'Karnataka',
 'West Bengal',
 'Rajasthan',
 'Andhra Pradesh',
 'Telangana',
 'Madhya Pradesh',
 'Kerala',
 'Delhi',
 'Haryana',
 'Bihar',
 'Punjab',
 'Odisha',
 'Assam',
 'Chhattisgarh',
 'Jharkhand',
 'Uttarakhand',
 'Jammu & Kashmir',
 'Himachal Pradesh',
 'Goa',
 'Tripura',
 'Chandigarh',
 'Puducherry',
 'Meghalaya',
 'Sikkim',
 'Manipur',
 'Nagaland',
 'Arunachal Pradesh',
 'Mizoram',
 'Andaman & Nicobar Islands']
```

```
In [21]: GSDP1819 = []
GS = driver.find_elements(By.XPATH,'//td[@class="data_sorting_1"]')
for i in GS:
 GSDP1819.append(i.text)
```

```
In [73]: GSDP1819 = GSDP1819[:33]
GSDP1819
```

```
Out[73]: ['2,632,792',
 '1,630,208',
 '1,584,764',
 '1,502,899',
 '1,493,127',
 '1,089,898',
 '942,586',
 '862,957',
 '861,031',
 '809,592',
 '781,653',
 '774,870',
 '734,163',
 '530,363',
 '526,376',
 '487,805',
 '315,881',
 '304,063',
 '297,204',
 '245,895',
 '155,956',
 '153,845',
 '73,170',
 '49,845',
 '42,114',
 '34,433',
 '33,481',
 '28,723',
 '27,870',
 '27,283',
 '24,603',
 '22,287',
 '-']
```

```
In [75]: GSDP1920 = []
DP = driver.find_elements(By.XPATH, '//tr[@class="odd" or@class="even"]/td')
for i in DP:
 GSDP1920.append(i.text)
```

```
In [76]: GSDP1920 = GSDP1920[:264]
GSDP1920 = GSDP1920[2::8]
GSDP1920
```

```
Out[76]: ['_',
'_1,845,853',
'_1,687,818',
'_',
'_1,631,977',
'_1,253,832',
'_1,020,989',
'_972,782',
'_969,604',
'_906,672',
'_',
'_856,112',
'_831,610',
'_611,804',
'_574,760',
'_521,275',
'_',
'_329,180',
'_328,598',
'_',
'_',
'_165,472',
'_80,449',
'_55,984',
'_',
'_38,253',
'_36,572',
'_32,496',
'_31,790',
'_',
'_',
'_26,503',
'_']
```

```
In [78]: share1819 = []
sh = driver.find_elements(By.XPATH, '//tr[@class="odd" or@class="even"]/td')
for i in sh:
 share1819.append(i.text)
```

```
In [79]: share1819 = share1819[:264]
share1819 = share1819[4::8]
share1819
```

```
Out[79]: ['13.94%',
 '8.63%',
 '8.39%',
 '7.96%',
 '7.91%',
 '5.77%',
 '4.99%',
 '4.57%',
 '4.56%',
 '4.29%',
 '4.14%',
 '4.10%',
 '3.89%',
 '2.81%',
 '2.79%',
 '2.58%',
 '1.67%',
 '1.61%',
 '1.57%',
 '1.30%',
 '0.83%',
 '0.81%',
 '0.39%',
 '0.26%',
 '0.22%',
 '0.18%',
 '0.18%',
 '0.15%',
 '0.15%',
 '0.14%',
 '0.13%',
 '0.12%',
 ' - ']
```

```
In [80]: GDP =[]
G = driver.find_elements(By.XPATH,'//tr[@class="odd" or@class="even"]/td')
for i in G:
 GDP.append(i.text)
```

```
In [81]: GDP = GDP[:264]
GDP = GDP[4::8]
GDP
```

```
Out[81]: ['13.94%',
 '8.63%',
 '8.39%',
 '7.96%',
 '7.91%',
 '5.77%',
 '4.99%',
 '4.57%',
 '4.56%',
 '4.29%',
 '4.14%',
 '4.10%',
 '3.89%',
 '2.81%',
 '2.79%',
 '2.58%',
 '1.67%',
 '1.61%',
 '1.57%',
 '1.30%',
 '0.83%',
 '0.81%',
 '0.39%',
 '0.26%',
 '0.22%',
 '0.18%',
 '0.18%',
 '0.15%',
 '0.15%',
 '0.14%',
 '0.13%',
 '0.12%',
 ' - ']
```

In [ ]:

- 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 [84]: df3 = pd.DataFrame({'Rank':Rank, 'State':State, 'GSDP1819':GSDP1819, 'GSDP1920':GSDP1920, df3

| Out[84]:  | Rank | State                     | GSDP1819  | GSDP1920  | Share1819 | GDP    |
|-----------|------|---------------------------|-----------|-----------|-----------|--------|
| <b>0</b>  | 1    | Maharashtra               | 2,632,792 | -         | 13.94%    | 13.94% |
| <b>1</b>  | 2    | Tamil Nadu                | 1,630,208 | 1,845,853 | 8.63%     | 8.63%  |
| <b>2</b>  | 3    | Uttar Pradesh             | 1,584,764 | 1,687,818 | 8.39%     | 8.39%  |
| <b>3</b>  | 4    | Gujarat                   | 1,502,899 | -         | 7.96%     | 7.96%  |
| <b>4</b>  | 5    | Karnataka                 | 1,493,127 | 1,631,977 | 7.91%     | 7.91%  |
| <b>5</b>  | 6    | West Bengal               | 1,089,898 | 1,253,832 | 5.77%     | 5.77%  |
| <b>6</b>  | 7    | Rajasthan                 | 942,586   | 1,020,989 | 4.99%     | 4.99%  |
| <b>7</b>  | 8    | Andhra Pradesh            | 862,957   | 972,782   | 4.57%     | 4.57%  |
| <b>8</b>  | 9    | Telangana                 | 861,031   | 969,604   | 4.56%     | 4.56%  |
| <b>9</b>  | 10   | Madhya Pradesh            | 809,592   | 906,672   | 4.29%     | 4.29%  |
| <b>10</b> | 11   | Kerala                    | 781,653   | -         | 4.14%     | 4.14%  |
| <b>11</b> | 12   | Delhi                     | 774,870   | 856,112   | 4.10%     | 4.10%  |
| <b>12</b> | 13   | Haryana                   | 734,163   | 831,610   | 3.89%     | 3.89%  |
| <b>13</b> | 14   | Bihar                     | 530,363   | 611,804   | 2.81%     | 2.81%  |
| <b>14</b> | 15   | Punjab                    | 526,376   | 574,760   | 2.79%     | 2.79%  |
| <b>15</b> | 16   | Odisha                    | 487,805   | 521,275   | 2.58%     | 2.58%  |
| <b>16</b> | 17   | Assam                     | 315,881   | -         | 1.67%     | 1.67%  |
| <b>17</b> | 18   | Chhattisgarh              | 304,063   | 329,180   | 1.61%     | 1.61%  |
| <b>18</b> | 19   | Jharkhand                 | 297,204   | 328,598   | 1.57%     | 1.57%  |
| <b>19</b> | 20   | Uttarakhand               | 245,895   | -         | 1.30%     | 1.30%  |
| <b>20</b> | 21   | Jammu & Kashmir           | 155,956   | -         | 0.83%     | 0.83%  |
| <b>21</b> | 22   | Himachal Pradesh          | 153,845   | 165,472   | 0.81%     | 0.81%  |
| <b>22</b> | 23   | Goa                       | 73,170    | 80,449    | 0.39%     | 0.39%  |
| <b>23</b> | 24   | Tripura                   | 49,845    | 55,984    | 0.26%     | 0.26%  |
| <b>24</b> | 25   | Chandigarh                | 42,114    | -         | 0.22%     | 0.22%  |
| <b>25</b> | 26   | Puducherry                | 34,433    | 38,253    | 0.18%     | 0.18%  |
| <b>26</b> | 27   | Meghalaya                 | 33,481    | 36,572    | 0.18%     | 0.18%  |
| <b>27</b> | 28   | Sikkim                    | 28,723    | 32,496    | 0.15%     | 0.15%  |
| <b>28</b> | 29   | Manipur                   | 27,870    | 31,790    | 0.15%     | 0.15%  |
| <b>29</b> | 30   | Nagaland                  | 27,283    | -         | 0.14%     | 0.14%  |
| <b>30</b> | 31   | Arunachal Pradesh         | 24,603    | -         | 0.13%     | 0.13%  |
| <b>31</b> | 32   | Mizoram                   | 22,287    | 26,503    | 0.12%     | 0.12%  |
| <b>32</b> | 33   | Andaman & Nicobar Islands | -         | -         | -         | -      |

```
In [85]: df3.to_csv('State-wise_GDP_of_India.csv')
```

```
In [107]: driver.close()
```

1

4

```
In [27]: driver = webdriver.Chrome()
 driver.get('https://www.github.com')
```

```
In [35]: product = driver.find_element(By.XPATH, '/html/body/div[1]/div[1]/header/div/div[2]/div')
product.click()
```

```
In [36]: search = driver.find_element(By.XPATH, '/html/body/div[1]/div[1]/header/div/div[2]/div/search.click()
```

```
In [37]: Repository_title = []
Reposit = driver.find_elements(By.XPATH,'//span[@class="text-normal"]')
for i in Reposit:
 Repository title.append(i.text)
```

```
In [46]: opening_url = []
url = driver.find_elements(By.XPATH,'//a[@class="Link"]')
for i in url:
 opening_url.append(i.get_attribute('href'))
```

```
In [47]: Repository_description = []
for p in opening_url:
 driver.get(p)
 time.sleep(5)
 try:
 Repo= driver.find_element(By.XPATH,'/html/body/div[1]/div[4]/div/main/turbo-fr
 Repository_description.append(Repo.text)
 except NoSuchElementException:
 Repository_description.append(' ')
```

```
In [48]: Repository description
```

```
Out[48]: ['A MIT-licensed, deployable starter kit for building and customizing your own version of AI town - a virtual town where AI characters live, chat and socialize.',
 "The OpenTF Manifesto expresses concern over HashiCorp's switch of the Terraform license from open-source to the Business Source License (BSL) and calls for the tool's return to a truly open-source license.",
 'Focus on prompting and generating',
 'An extensible, easy-to-use, and portable diffusion web UI 😊\u200d😊',
 'FaceChain is a deep-learning toolchain for generating your Digital-Twin.',
 'A sample app for the Retrieval-Augmented Generation pattern running in Azure, using Azure Cognitive Search for retrieval and Azure OpenAI large language models to power ChatGPT-style and Q&A experiences.',
 'Langchain-Chatchat (formerly langchain-ChatGLM), local knowledge based LLM (like ChatGLM) QA app with langchain | 基于 Langchain 与 ChatGLM 等语言模型的本地知识库问答',
 'Rift: an AI-native language server for your personal AI software engineer',
 'Official implementation of "Neuralangelo: High-Fidelity Neural Surface Reconstruction" (CVPR 2023)',
 ' ',
 'A curated list of modern Generative Artificial Intelligence projects and services',
 '2023 HVV情报速递~',
 'AWS zero to hero repo for devops engineers to learn AWS in 30 Days. This repo includes projects, presentations, interview questions and real time examples.',
 'Specify what you want it to build, the AI asks for clarification, and then builds it.',
 '</> htmx - high power tools for HTML',
 'The modern web developer's platform',
 'Official Code for DragGAN (SIGGRAPH 2023)',
 'GoogleTest - Google Testing and Mocking Framework',
 'A well-designed cross-platform ChatGPT UI (Web / PWA / Linux / Win / MacOS). 一键拥有你自己的跨平台 ChatGPT 应用。',
 'A list of Free Software network services and web applications which can be hosted on your own servers',
 'The React Framework',
 ':🔥🔥🔥 现代化、开源的 Linux 服务器运维管理面板。',
 'GPT based autonomous agent that does online comprehensive research on any given topic',
 '.NET MAUI is the .NET Multi-platform App UI, a framework for building native device applications spanning mobile, tablet, and desktop.',
 "A game where you are a computer's OS and you have to manage processes, memory and I/O events."]
```

```
In [58]: Language_used = []
Contributors_count = []
for p in opening_url:
 driver.get(p)
 time.sleep(5)

 try:
 Language= driver.find_element(By.XPATH,'//span[@class="color-fg-default text-t')
 Language_used.append(Language.text)
 except NoSuchElementException:
 Language_used.append('_')

 try:
 Contributors= driver.find_element(By.XPATH,'/html/body/div[1]/div[4]/div/main')
 Contributors_count.append(Contributors.text)
 except NoSuchElementException:
 Contributors_count.append('_')
```

```
In [59]: df4 = pd.DataFrame({'Repository_title':Repository_title,'Repository_description':Repos
df4
```

Out[59]:

|    | Repository_title     | Repository_description                                              | Contributors_count | Language_used |
|----|----------------------|---------------------------------------------------------------------|--------------------|---------------|
| 0  | a16z-infra /         | A MIT-licensed, deployable starter kit for building AI products.    | -                  | TypeScript    |
| 1  | opentffoundation /   | The OpenTF Manifesto expresses concern over Hashicorp's...          | 239                | HTML          |
| 2  | Illyasviel /         | Focus on prompting and generating                                   | -                  | Python        |
| 3  | varunshenoy /        | An extensible, easy-to-use, and portable diffutils-like tool for... | 6                  | JavaScript    |
| 4  | modelscope /         | FaceChain is a deep-learning toolchain for generating...            | 7                  | Python        |
| 5  | Azure-Samples /      | A sample app for the Retrieval-Augmented Generative...              | -                  | Python        |
| 6  | chatchat-space /     | Langchain-Chatchat (formerly langchain-ChatGLM...)                  | 74                 | Python        |
| 7  | morph-labs /         | Rift: an AI-native language server for your personal...             | 13                 | Python        |
| 8  | NVlabs /             | Official implementation of "Neuralangelo": High-quality...          | -                  | Python        |
| 9  | ProfSynapse /        |                                                                     | -                  | -             |
| 10 | steven2358 /         | A curated list of modern Generative Artificial...                   | -                  | -             |
| 11 | ibaiw /              | 2023 HVV情报速递~                                                       | -                  | -             |
| 12 | iam-veeramalla /     | AWS zero to hero repo for devops engineers to learn AWS...          | -                  | Python        |
| 13 | AntonOsika /         | Specify what you want it to build, the AI asks...                   |                    | Python        |
| 14 | bigskysoftware /     | </> htmx - high power tools for HTML                                |                    | JavaScript    |
| 15 | angular /            | The modern web developer's platform                                 |                    | TypeScript    |
| 16 | XingangPan /         | Official Code for DragGAN (SIGGRAPH 2023)                           | -                  | Python        |
| 17 | google /             | GoogleTest - Google Testing and Mocking Framework                   | 406                | C++           |
| 18 | Yidadaa /            | A well-designed cross-platform ChatGPT UI (WebAssembly)             | 118                | TypeScript    |
| 19 | awesome-selfhosted / | A list of Free Software network services and web applications...    | 1,254              | Makefile      |
| 20 | vercel /             | The React Framework                                                 | 2,824              | JavaScript    |
| 21 | 1Panel-dev /         | 🔥🔥🔥 现代化、开源的 Linux 服务器运维管理面板。                                        | 28                 | Go            |
| 22 | assafelovic /        | GPT based autonomous agent that does online co...                   | -                  | Python        |

|    | Repository_title | Repository_description                            | Contributors_count | Language_used |
|----|------------------|---------------------------------------------------|--------------------|---------------|
| 23 | dotnet /         | .NET MAUI is the .NET Multi-platform App UI, a... | 1                  | C#            |
| 24 | plbrault /       | A game where you are a computer's OS and you h... | 2                  | Python        |

```
In [60]: df4.to_csv('trending_repositories.csv')
```

```
In [62]: '''A) Repository_title
B) Repository_description
C) Contributors_count
D) Language_used'''
```

```
Out[62]: 'A) Repository_title \nB) Repository_description \nC) Contributors_count \nD) Languag e_used'
```

```
In [105... driver.close()
```

## 5

```
In [104... driver = webdriver.Chrome()
driver.get('https://www.billboard.com/')
```

```
In [68]: option = driver.find_element(By.XPATH, '/html/body/div[3]/header/div/div[4]/div/div[1],
option.click()
```

```
In [69]: chart = driver.find_element(By.XPATH, '/html/body/div[3]/div[9]/div/div/div/ul/li[1]/h3
chart.click()
```

```
In [70]: hot100 = driver.find_element(By.XPATH, '/html/body/div[3]/main/div[2]/div[1]/div[1]/div
hot100.click()
```

```
In [71]: Song_name = []
Song = driver.find_elements(By.XPATH, '//li[@class="lrv-u-width-100p"]/ul/li/h3')
for i in Song:
 Song_name.append(i.text)
```

```
In [74]: Artist_name = []
Artist = driver.find_elements(By.XPATH, '//li[@class="lrv-u-width-100p"]/ul/li/h3/span'
for i in Artist:
 Artist_name.append(i.text)
```

```
In [75]: Artist_name
```

```
Out[75]: []
```

```
In [90]: Last_week_rank = []
Last_week = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item //')
for i in Last_week:
 Last_week_rank.append(i.text)
```

```
In [91]: Last_week_rank = Last_week_rank[::2]
```

```
In [94]: Weeks_on_board = []
Weeks = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-ch')
for i in Weeks:
 Weeks_on_board.append(i.text)
```

```
In [95]: Weeks_on_board = Weeks_on_board[1::2]
```

```
In [96]: Peak_rank = []
Peak = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-ch')
for i in Peak:
 Peak_rank.append(i.text)
```

```
In [97]: Peak_rank = Peak_rank[1::2]
```

- In [ ]:
- A) Song\_name
  - B) Artist\_name
  - C) Last\_week\_rank
  - D) Peak\_rank
  - E) Weeks\_on\_board

```
In [98]: df5 = pd.DataFrame({'Song_name':Song_name,'Last_week_rank':Last_week_rank,'Peak_rank':Peak_rank,'Weeks_on_board':Weeks_on_board})
```

Out[98]:

|     | Song_name                    | Last_week_rank | Peak_rank | Weeks_on_board |
|-----|------------------------------|----------------|-----------|----------------|
| 0   | Last Night                   | 1              | 1         | 28             |
| 1   | Fast Car                     | 2              | 2         | 20             |
| 2   | Cruel Summer                 | 4              | 3         | 14             |
| 3   | Calm Down                    | 6              | 3         | 49             |
| 4   | Fukumean                     | 7              | 4         | 8              |
| ... | ...                          | ...            | ...       | ...            |
| 95  | Lagunas                      | -              | 77        | 6              |
| 96  | Overdrive                    | 68             | 47        | 3              |
| 97  | Bzrp Music Sessions, Vol. 55 | 99             | 31        | 10             |
| 98  | Dawns                        | -              | 42        | 15             |
| 99  | Rubicon                      | -              | 63        | 6              |

100 rows × 4 columns

```
In [99]: df5.to_csv('top_100_songs.csv')
```

```
In [100... driver.close()
```

6

## THIS LINK IS NOT OPENING

```
In [107... driver = webdriver.Chrome()
driver.get('https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-a')
```

7

```
In [108... driver = webdriver.Chrome()
driver.get('https://www.imdb.com/list/ls095964455/')
```

```
In [109... Name = []
Na = driver.find_elements(By.XPATH,'//h3[@class="lister-item-header"]')
for i in Na:
 Name.append(i.text)
```

```
In [110... Year_span = []
Year = driver.find_elements(By.XPATH,'//span[@class="lister-item-year text-muted unbold"]')
for i in Year:
 Year_span.append(i.text)
```

```
In [111... Genre = []
Gen = driver.find_elements(By.XPATH,'//span[@class="genre"]')
for i in Gen:
 Genre.append(i.text)
```

```
In [112... Run_time = []
Run = driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for i in Run:
 Run_time.append(i.text)
```

```
In [113... Ratings = []
Ratin = driver.find_elements(By.XPATH,'//div[@class="ipl-rating-star small"]')
for i in Ratin:
 Ratings.append(i.text)
```

In [114...]

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

In [116...]

```
'''A) Name
B) Year_span
C) Genre
D) Run_time
E) Ratings
F) Votes'''
```

Out[116]:

```
'A) Name \nB) Year_span \nC) Genre \nD) Run_time \nE) Ratings \nF) Votes '
```

In [117...]

```
df7 = pd.DataFrame({'Name':Name, 'Year_span':Year_span, 'Genre':Genre, 'Run_time':Run_time, 'Ratings':Ratings, 'Votes':Votes})
```

Out[117]:

|     | Name                                           | Year_span   | Genre                    | Run_time | Ratings | Votes     |
|-----|------------------------------------------------|-------------|--------------------------|----------|---------|-----------|
| 0   | 1. Game of Thrones (2011–2019)                 | (2011–2019) | Action, Adventure, Drama | 57 min   | 9.2     | 2,193,366 |
| 1   | 2. Stranger Things (2016–2024)                 | (2016–2024) | Drama, Fantasy, Horror   | 51 min   | 8.7     | 1,267,099 |
| 2   | 3. The Walking Dead (2010–2022)                | (2010–2022) | Drama, Horror, Thriller  | 44 min   | 8.1     | 1,041,100 |
| 3   | 4. 13 Reasons Why (2017–2020)                  | (2017–2020) | Drama, Mystery, Thriller | 60 min   | 7.5     | 305,991   |
| 4   | 5. The 100 (2014–2020)                         | (2014–2020) | Drama, Mystery, Sci-Fi   | 43 min   | 7.6     | 265,074   |
| ... | ...                                            | ...         | ...                      | ...      | ...     | ...       |
| 95  | 96. Reign (2013–2017)                          | (2013–2017) | Drama                    | 42 min   | 7.4     | 52,435    |
| 96  | 97. A Series of Unfortunate Events (2017–2019) | (2017–2019) | Adventure, Comedy, Drama | 50 min   | 7.8     | 64,462    |
| 97  | 98. Criminal Minds (2005– )                    | (2005– )    | Crime, Drama, Mystery    | 42 min   | 8.1     | 210,123   |
| 98  | 99. Scream: The TV Series (2015–2019)          | (2015–2019) | Comedy, Crime, Drama     | 45 min   | 7       | 43,693    |
| 99  | 100. The Haunting of Hill House (2018)         | (2018)      | Drama, Horror, Mystery   | 572 min  | 8.6     | 263,707   |

100 rows × 6 columns

In [118...]

```
df7.to_csv('most watched tv series of all time.csv')
```

In [119...]

```
driver.close()
```

.

8

```
try:
 att2= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1]
 No_of_attribute.append(att2.text)
except NoSuchElementException:
 No_of_attribute.append('_')

try:
 yea= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1]
 Year.append(yea.text)
except NoSuchElementException:
 Year.append('_')
```

- In [ ]:
- A) Dataset\_name
  - B) Data\_type
  - C) Task
  - D) Attribute\_type
  - E) No\_of\_instances
  - F) No\_of\_attribute
  - G) Year

In [135...]: df8 = pd.DataFrame({'Dataset\_name':Dataset\_name,'Data\_type':Data\_type,'Task':Task,'Attribute\_type':Attribute\_type,'No\_of\_instances':No\_of\_instances,'No\_of\_attribute':No\_of\_attribute,'Year':Year})

|   | Dataset_name                               | Data_type                    | Task           | Attribute_type                | No_of_instances | No_of_attribute | Y                 |
|---|--------------------------------------------|------------------------------|----------------|-------------------------------|-----------------|-----------------|-------------------|
| 0 | Iris                                       | Multivariate                 | Classification | Real                          | 150             | 4               | Donat<br>6/30/19  |
| 1 | Heart Disease                              | Multivariate                 | Classification | Categorical,<br>Integer, Real | 303             | 13              | Donat<br>6/30/19  |
| 2 | Adult                                      | Multivariate                 | Classification | Categorical,<br>Integer       | 48842           | 14              | Donat<br>4/30/19  |
| 3 | Dry Bean<br>Dataset                        | Multivariate                 | Classification | Integer, Real                 | 13611           | 16              | Donat<br>9/13/20  |
| 4 | Diabetes                                   | Multivariate,<br>Time-Series | -              | Categorical,<br>Integer       | -               | 20              |                   |
| 5 | Wine                                       | Multivariate                 | Classification | Integer, Real                 | 178             | 13              | Donat<br>6/30/19  |
| 6 | Breast Cancer<br>Wisconsin<br>(Diagnostic) | Multivariate                 | Classification | Real                          | 569             | 30              | Donat<br>10/31/19 |
| 7 | Rice (Cammeo<br>and Osmancik)              | Multivariate                 | Classification | Real                          | 3810            | 8               | Donat<br>10/5/20  |
| 8 | Car Evaluation                             | Multivariate                 | Classification | Categorical                   | 1728            | 6               | Donat<br>5/31/19  |
| 9 | Mushroom                                   | Multivariate                 | Classification | Categorical                   | 8124            | 22              | Donat<br>4/26/19  |

In [136...]

`df8.to_csv('dataset.csv')`

In [137...]

`driver.close()`