

WEB SCRAPPING ASSIGNMENT --- 3

Q1:-- Write a python program which searches all the product under a particular product from www.amazon.in (<http://www.amazon.in>). The product to be searched will be taken as input from user. For e.g. If user input is 'guitar'. Then search for guitars.

In [1]: !pip install selenium

```
Requirement already satisfied: selenium in c:\users\dell\anaconda3\lib\site-p
ackages (4.15.2)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\dell\anaconda3
\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\dell\anaconda3
\lib\site-packages (from selenium) (2022.12.7)
Requirement already satisfied: trio~=0.17 in c:\users\dell\anaconda3\lib\site
-packages (from selenium) (0.23.1)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\dell\anaco
nda3\lib\site-packages (from selenium) (1.26.14)
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in c:\users\dell\anac
onda3\lib\site-packages (from trio~=0.17->selenium) (1.1.3)
Requirement already satisfied: idna in c:\users\dell\anaconda3\lib\site-packa
ges (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\dell\anaconda3\lib
\site-packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: outcome in c:\users\dell\anaconda3\lib\site-pa
ckages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: attrs>=20.1.0 in c:\users\dell\anaconda3\lib\s
ite-packages (from trio~=0.17->selenium) (22.1.0)
Requirement already satisfied: sortedcontainers in c:\users\dell\anaconda3\li
b\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: cffi>=1.14 in c:\users\dell\anaconda3\lib\site
-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\dell\anaconda3\lib\s
ite-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\dell\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\dell\anaconda3\lib\site-
packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\dell\anaconda3\lib\s
ite-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
```

```
In [2]: 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 selenium.common.exceptions import NoSuchElementException
```

```
In [3]: driver = webdriver.Chrome()
```

```
In [4]: driver.get("https://www.amazon.in/")
```

```
In [5]: search_field= driver.find_element(By.ID, 'twotabsearchtextbox')
        search_field.send_keys(input("Enter any Product Name "))
```

Enter any Product Name shoes

```
In [6]: search_btn = driver.find_element(By.XPATH, '//span[@class="nav-search-submit-te
        search_btn.click()
```

Q2 :-- In the above question, now scrape the following details of each product listed in first 3 pages of your search results and save it in a data frame and csv. In case if any product has less than 3 pages in search results then scrape all the products available under that product name. Details to be scraped are: "Brand Name", "Name of the Product", "Price", "Return/Exchange", "Expected Delivery", "Availability" and "Product URL". In case, if any of the details are missing for any of the product then replace it by "-".

```
In [7]: title = []
delivery = []
MRP = []
product_url = []
availability = ["nAn"]
Return = ['NaN']
brand = []
start = 0
end = 3
for pages in range(start,end):
    #for Brand Name
    title_name = driver.find_elements(By.XPATH, '//span[@class="a-size-base-plu
    for i in title_name:
        title.append(i.text)

    delivery_date = driver.find_elements(By.XPATH, '//span[@class="a-color-base
    for i in delivery_date:
        delivery.append(i.text)

    price = driver.find_elements(By.XPATH, '//span[@class="a-price-whole"]')
    for i in price:
        MRP.append(i.text)
    url = driver.find_elements(By.XPATH, '//a[@class="a-link-normal s-underline
    for i in url:
        product_url.append(i.get_attribute('href'))
    brand_name = driver.find_elements(By.XPATH, '//h2[@class="a-size-mini s-lin
    for i in brand_name:
        brand.append(i.text)

    next_page = driver.find_element(By.XPATH, '//a[@class="s-pagination-item s-
    next_page.click()
    time.sleep(5)
```

```
In [8]: print(len(title),len(delivery),len(MRP),len(product_url),len(brand))
```

```
181 0 177 181 181
```

```
In [9]: df =pd.DataFrame({'Brand Name':brand})
df6 = pd.DataFrame({'Name of the Product':title})
df1 =pd.DataFrame({'Expected Delivery':delivery})
df2 =pd.DataFrame({'Price RS.':MRP})
df3 =pd.DataFrame({'Product URL':product_url})
df4 =pd.DataFrame({'Availability':availability, "Return/Exchange":Return})
df5=pd.concat([df,df6,df1,df2,df3,df4],axis=1)
df5.to_csv("Product list")
df5.head(180)
```

Out[9]:

	Brand Name	Name of the Product	Expected Delivery	Price RS.	Product URL	Availability	Return/Exchange
0	AVANT	Mens PulseSports Shoe	NaN	699	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
1	Bacca Bucci	Men Running Shoes	NaN	1,499	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
2	AVANT	Men's Dual Tone Knitted Running Shoes EVA Outs...	NaN	699	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
3	Bacca Bucci	Men's Running Shoe	NaN	1,399	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
4	Red Tape	Casual Sneaker Shoes for Men Enhanced Comfor...	NaN	1,679	https://www.amazon.in/Red-Tape-Cushioned-Slip-...	NaN	Return
...
175	Campus	Men's Strom PRO Running Shoes	NaN	3,749	https://www.amazon.in/Campus-Strom-Running-Sho...	NaN	Return
176	Bata	Mens Boss-Slick Formal Shoes, (8516051), Black	NaN	1,599	https://www.amazon.in/BATA-BOSS-Slick-Black-Un...	NaN	Return
177	Bacca Bucci	Men's Running Shoes	NaN	NaN	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
178	Bacca Bucci	Men's Running Shoes 5 1	NaN	NaN	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return
179	new balance	M480nb6Running Shoe	NaN	NaN	https://www.amazon.in/sspa/click?ie=UTF8&spc=M...	NaN	Return

180 rows × 7 columns

Q3:-- Write a python program to access the search bar and search button on images.google.com and scrape 10 images each

for keywords 'fruits', 'cars' and 'Machine Learning', 'Guitar', 'Cakes'.

```
In [10]: driver = webdriver.Chrome()
```

```
In [11]: driver.get("http://images.google.com/")
```

```
In [12]: search = driver.find_element(By.XPATH, '//*[@class="gLfyf"]')
search.send_keys('fruits')
```

```
In [16]: search_btn = driver.find_element(By.XPATH, '//*[@class="z1asCe MZy1Rb"]')
search_btn.click()
```

```
In [17]: fruits_images = []

url = driver.find_elements(By.XPATH, '//*[@class="fR600b isIir"]//img')
for i in url:
    fruits_images.append(i.get_attribute('src'))
```

```
In [18]: print(len(fruits_images))
```

48

```
In [19]: #For cars Images
search = driver.find_element(By.XPATH, '//*[@class="og3lId"]')
search.clear()
search.send_keys('cars')

search_btn = driver.find_element(By.XPATH, '//*[@class="XZ5MVe"]//span')
search_btn.click()

cars_images = []

url = driver.find_elements(By.XPATH, '//*[@class="fR600b isIir"]//img')
for i in url:
    cars_images.append(i.get_attribute('src'))

print(len(cars_images))
```

48

```
In [20]: #For Machine Learning Images
search = driver.find_element(By.XPATH, '//*[@class="og3lId"]')
search.clear()
search.send_keys('Machine Learning')

search_btn = driver.find_element(By.XPATH, '//*[@class="XZ5MVe"]//span')
search_btn.click()

ml_images = []

url = driver.find_elements(By.XPATH, '//*[@class="fR600b islr"]//img')
for i in url:
    ml_images.append(i.get_attribute('src'))
print(len(ml_images))
```

48

```
In [21]: #For Guitar Images
search = driver.find_element(By.XPATH, '//*[@class="og3lId"]')
search.clear()
search.send_keys('Guitar')

search_btn = driver.find_element(By.XPATH, '//*[@class="XZ5MVe"]//span')
search_btn.click()

guitar_images = []

url = driver.find_elements(By.XPATH, '//*[@class="fR600b islr"]//img')
for i in url:
    guitar_images.append(i.get_attribute('src'))

print(len(guitar_images))
```

48

```
In [22]: #For Cakes Images
search = driver.find_element(By.XPATH, '//*[@class="og3lId"]')
search.clear()
search.send_keys('Cakes')

search_btn = driver.find_element(By.XPATH, '//*[@class="XZ5MVe"]//span')
search_btn.click()

cakes_images = []

url = driver.find_elements(By.XPATH, '//*[@class="fR600b islr"]//img')
for i in url:
    cakes_images.append(i.get_attribute('src'))

print(len(cakes_images))
```

48

```
In [23]: df = pd.DataFrame({'Cars Images':cars_images,'Machine Learning':ml_images,'Gui
df1 = pd.DataFrame({'Fruits Images':fruits_images})
df2 = pd.concat([df1,df],axis=1)
df2.head(10)
```

Out[23]:

	Fruits Images	
0	...	
1	...	
2	...	
3	...	
4	...	
5	...	
6	...	
7	...	
8	...	
9	...	

Q4 :--Write a python program to search for a smartphone(e.g.: Oneplus Nord, pixel 4A, etc.) on www.flipkart.com (<http://www.flipkart.com>) and scrape following details for all the search results displayed on 1st page. Details to be scraped: “Brand Name”, “Smartphone name”, “Colour”, “RAM”, “Storage(ROM)”, “Primary Camera”, “Secondary Camera”, “Display Size”, “Battery Capacity”, “Price”, “Product URL”. Incase if any of the details is missing then replace it by “- “. Save your results in a dataframe and CSV.

```
In [24]: driver = webdriver.Chrome()
```

```
In [25]: driver.get("http://www.flipkart.com/")
```

```
In [26]: close_btn = driver.find_element(By.XPATH, '/html/body/div[3]/div/span')
close_btn.click()
```

```
In [27]: search_field= driver.find_element(By.CLASS_NAME, "Pke_EE")
search_field.send_keys("Oneplus Nord and Pixel 4A")
```

```
In [28]: search_btn = driver.find_element(By.XPATH, '/html/body/div[1]/div/div[1]/div/div')
search_btn.click()
```

```
In [29]: brand = []
smartphone =[]
ram_rom =[]
camera =[]
display =[]
battery = []
MRP =[]
product_url =[]

smart_name = driver.find_elements(By.XPATH, '//div[@class="_4rR01T"]')
for i in smart_name:
    smartphone.append(i.text)

brand_name = driver.find_elements(By.XPATH, '//div[@class="_4rR01T"]')
for i in brand_name:
    brand.append(i.text)

RAM = driver.find_elements(By.XPATH, '//ul[@class="_1xgFaf"]//li[1]')
for i in RAM:
    ram_rom.append(i.text)

display_size = driver.find_elements(By.XPATH, '//ul[@class="_1xgFaf"]//li[2]')
for i in display_size:
    display.append(i.text)

camera_pxl = driver.find_elements(By.XPATH, '//ul[@class="_1xgFaf"]//li[3]')
for i in camera_pxl:
    camera.append(i.text)

battery_mah = driver.find_elements(By.XPATH, '//ul[@class="_1xgFaf"]//li[4]')
for i in battery_mah:
    battery.append(i.text)

price = driver.find_elements(By.XPATH, '//div[@class="_25b18c"]//div[1]')
for i in price:
    MRP.append(i.text)

url = driver.find_elements(By.XPATH, '//a[@class="_1fQZEK"]')
for i in url:
    product_url.append(i.get_attribute('href'))
```



```
In [30]: df = pd.DataFrame({'Brand Name':brand,'SmartPhone Name':smartphone,'RAM/ROM':r  
df1 = pd.DataFrame({'Price':MRP})  
  
df2 = pd.concat([df,df1],axis=1)  
  
df2.to_csv("Scrapped data of Oneplus nord")  
df2
```

Out[30]:

	Brand Name	SmartPhone Name	RAM/ROM	Camera Front/Back	Battery	Display Size	Product URI
0	OnePlus Nord (Gray Onyx, 64 GB)	OnePlus Nord (Gray Onyx, 64 GB)	6 GB RAM 64 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-gray-ony..
1	OnePlus Nord (Gray Onyx, 128 GB)	OnePlus Nord (Gray Onyx, 128 GB)	8 GB RAM 128 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-gray-ony..
2	OnePlus Nord (Blue Marble, 128 GB)	OnePlus Nord (Blue Marble, 128 GB)	8 GB RAM 128 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-blue-mar..
3	OnePlus Nord (Blue Marble, 64 GB)	OnePlus Nord (Blue Marble, 64 GB)	6 GB RAM 64 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-blue-mar..
4	OnePlus Nord (Gray Onyx, 256 GB)	OnePlus Nord (Gray Onyx, 256 GB)	12 GB RAM 256 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-gray-ony..
5	OnePlus Nord (Blue Marble, 256 GB)	OnePlus Nord (Blue Marble, 256 GB)	12 GB RAM 256 GB ROM	48MP + 8MP 32MP + 8MP Dual Front Camera	4115 mAh Battery	16.36 cm (6.44 inch) Full HD+ Display	https://www.flipkart.com/oneplus-nord-blue-mar..
6	OnePlus Nord CE 2 5G (Bahama Blue, 128 GB)	OnePlus Nord CE 2 5G (Bahama Blue, 128 GB)	8 GB RAM 128 GB ROM	64MP Rear Camera	4500 mAh Battery	16.33 cm (6.43 inch) Display	https://www.flipkart.com/oneplus-nord-ce-2-5g-..
7	OnePlus Nord CE 2 5G (Gray Mirror, 128 GB)	OnePlus Nord CE 2 5G (Gray Mirror, 128 GB)	8 GB RAM 128 GB ROM	64MP Rear Camera	4500 mAh Battery	16.33 cm (6.43 inch) Display	https://www.flipkart.com/oneplus-nord-ce-2-5g-..

	Brand Name	SmartPhone Name	RAM/ROM	Camera Front/Back	Battery	Display Size	Product URI
8	OnePlus Nord CE 2 5G (Gray Mirror, 128 GB)	OnePlus Nord CE 2 5G (Gray Mirror, 128 GB)	6 GB RAM 128 GB ROM	64MP Rear Camera 16MP Front Camera	4500 mAh Battery	16.33 cm (6.43 inch) Display	https://www.flipkart.com/oneplus-nord-ce-2-5g-..
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Q5:-- Write a program to scrap geospatial coordinates (latitude, longitude) of a city searched on google maps.

In [31]: `!pip install geopy`

Requirement already satisfied: geopy in c:\users\dell\anaconda3\lib\site-packages (2.4.0)

Requirement already satisfied: geographiclib<3,>=1.52 in c:\users\dell\anaconda3\lib\site-packages (from geopy) (2.0)

In [32]: `from geopy.geocoders import ArcGIS`

In [33]: `name =ArcGIS()`

In [34]: `name.geocode(input("Enter any city Name "))`

Enter any city Name ludhiana

Out[34]: `Location(Ludhiana, Punjab, (30.90726, 75.8492, 0.0))`

Q6:--Write a program to scrap all the available details of best gaming laptops from digit.in.

In [35]: `driver = webdriver.Chrome()`

In [36]: `driver.get("https://www.digit.in/")`

```
In [37]: top = driver.find_element(By.ID, "menu-item-290390")
top.click()
```

```
In [39]: laptop= driver.find_element(By.XPATH, '/html/body/div[1]/div[3]/div/div[2]/div[1]')
laptop.click()
```

```
In [40]: title = []
os = []
display = []

title_name = driver.find_elements(By.XPATH, '//h3[@class="font130 mt0 mb10 mobi]')
for i in title_name:
    title.append(i.text)

os_name = driver.find_elements(By.XPATH, '//span[@class="meta_v_label mr5 rtlml1"]')
for i in os_name:
    os.append(i.text)

display_name = driver.find_elements(By.XPATH, '//span[@class="meta_v_value"]')
for i in display_name:
    display.append(i.text)
```

```
In [41]: df = pd.DataFrame({'Data 1':os, 'Data 2':display})
df1 =pd.DataFrame({'Laptop Name':title})

df1

#df2 = pd.concat([df1,df],axis = 1)
#df2
```

Out[41]:

	Laptop Name
0	Apple MacBook Air 2022 M2 MLY03HN/A
1	Samsung Galaxy Book3 Pro 360
2	Lenovo Yoga 9i13th Gen Core i7-1360P
3	Apple MacBook Pro M2 (2023-MPHK3HN/A)
4	Dell XPS 13 Plus D560075WIN9S 12th Gen Core i7...
5	Acer Swift X SFX16-51G 11th Gen Core i7-11390H

In [42]: df

Out[42]:

	Data 1	Data 2
0	Operating System:	iOS
1	Display Size :	13.6
2	Resolution :	2560 x 1600
3	Processor :	Apple M2
4	Operating System:	Windows 11 Home
5	Display Size :	14
6	Resolution :	2880 x 1800
7	Processor :	13th Gen Intel Core i7-1360P
8	Operating System:	Mac
9	Display Size :	14.2
10	Operating System:	Windows 11
11	Display Size :	13.4
12	Resolution :	3840 x 2400
13	Processor :	12th Gen Intel EVO Core i7-1260P
14	Operating System:	Windows 11 Home
15	Display Size :	16
16	Resolution :	1920 x 1080
17	Processor :	11th Gen Intel Core i7-11390H

Q7:-- Write a python program to scrape the details for all billionaires from www.forbes.com (<http://www.forbes.com>). Details to be scrapped: “Rank”, “Name”, “Net worth”, “Age”, “Citizenship”, “Source”, “Industry”.

In [43]: driver = webdriver.Chrome()

In [44]: driver.get("https://www.forbes.com/billionaires/")

```
In [45]: rank = []
name = []
net = []
age = []
country = []
source = []
industry = []

rank_name = driver.find_elements(By.XPATH, '//*[@class="Table_rank___YBhk Tab
for i in rank_name:
    rank.append(i.text)

Name = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km"]//div
for i in Name:
    name.append(i.text)

Net = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km"]//div[
for i in Net:
    net.append(i.text)

Age = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km"]//div[
for i in Age:
    age.append(i.text)

citizenship = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km
for i in citizenship:
    country.append(i.text)

Source = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km"]//d
for i in Source:
    source.append(i.text)

ind = driver.find_elements(By.XPATH, '//*[@class="TableRow_row___L-0Km"]//div[
for i in ind:
    industry.append(i.text)
```

```
In [46]: print(len(rank),len(name),len(net),len(age),len(country),len(source),len(indus

200 199 199 199 199 199 199
```

```
In [47]: df = pd.DataFrame({'Rank':rank})
df1 = pd.DataFrame({'Name':name, 'Net Worth':net, 'Age':age, 'Citizenship':country})
df2 =pd.concat([df,df1],axis=1)
df2
```

Out[47]:

	Rank	Name	Net Worth	Age	Citizenship	Source	Industry
0	1	Elon Musk	\$180 B	51	United States	Tesla, SpaceX	
1	2	Jeff Bezos	\$114 B	59	United States	Amazon	
2	3	Larry Ellison	\$107 B	78	United States	Oracle	
3	4	Warren Buffett	\$106 B	92	United States	Berkshire Hathaway	
4	5	Bill Gates	\$104 B	67	United States	Microsoft	
...
195	195	Luo Ligu & family	\$9.6 B	67	China	Chemicals	
196	195	Marijke Mars	\$9.6 B	58	United States	Candy, pet food	
197	195	Pamela Mars	\$9.6 B	62	United States	Candy, pet food	
198	195	Valerie Mars	\$9.6 B	64	United States	Candy, pet food	
199	195	NaN	NaN	NaN	NaN	NaN	NaN

200 rows × 7 columns

Q8:--Write a program to extract at least 500 Comments, Comment upvote and time when comment was posted from any YouTube Video.

```
In [48]: driver = webdriver.Chrome()
```

```
In [49]: driver.get("https://www.youtube.com/watch?v=rAbk5Xv1DMw")
```

```
In [50]: comment = []
vote = []
date = []

Comment = driver.find_elements(By.XPATH, '//div[@class="style-scope ytd-expande
for i in Comment:
    comment.append(i.text)

Vote = driver.find_elements(By.XPATH, '//span[@class="style-scope ytd-comment-a
for i in Vote:
    vote.append(i.text)

Date = driver.find_elements(By.XPATH, '//a[@class="yt-simple-endpoint style-sco
for i in Date:
    date.append(i.text)
```

```
In [51]: df = pd.DataFrame({'Comment':comment})
df1 = pd.DataFrame({'Comment Date':date})
df2 = pd.DataFrame({'Comments Vote':vote})

df3 = pd.concat([df,df1,df2],axis = 1)
df3.head(500)
```

Out[51]:

	Comment	Comment Date	Comments Vote
0	He भगवान विष्णु आप मुझे अपने श्री चरणों में स्...	Tilak	
1	काम,क्रोध,लोभ,मोह,वासना,इन विकरो से परे त्रिगु...		21
2	ऐसा कृष्ण मैने कही नहीं देखा, जैसे भगवान साक्षा...		
3	Bolo MahaVishnu Bhagwan Ki Jai	1 month ago	29
4	Hamare prabhu kitne dayalu aur kripalu hai ki ...	1 month ago	
...
495	Jai Shri Krishna	8 months ago	2
496	Jai shri Krishna	10 days ago	
497	Jay Shri Krishna	3 months ago	3
498	जय श्री कृष्ण भगवान	6 months ago	
499	Jay sri krishna Jay sri ram	9 months ago	1

500 rows × 3 columns

Q9:-- Write a python program to scrape a data for all available Hostels from

<https://www.hostelworld.com/>
(<https://www.hostelworld.com/>) in “London”
location. You have to scrape hostel name,
distance from city centre, ratings, total reviews,
overall reviews, privates from price, dorms
from price, facilities and property
description

```
In [52]: driver = webdriver.Chrome()
```

```
In [53]: driver.get("https://www.hostelworld.com/ ")
```

```
In [54]: search_field= driver.find_element(By.XPATH, '//div[@class="input-wrapper"]//inp  
search_field.send_keys('London')
```

```
In [56]: search_btn = driver.find_element(By.XPATH, '//li[@class="item is-two-row"]//but  
search_btn.click()
```

```
In [57]: search_btn2 = driver.find_element(By.XPATH, '//button[@class="btn-content large  
search_btn2.click()
```

```

In [58]: title = []
         rating = []
         tr = []
         overall_rating = []
         distance = []
         facility = []
         dorm = []

         Comment = driver.find_elements(By.XPATH, '//*[@class="property-name"]//span')
         for i in Comment:
             title.append(i.text)

         rate = driver.find_elements(By.XPATH, '//*[@class="number"]')
         for i in rate:
             rating.append(i.text)

         total = driver.find_elements(By.XPATH, '//*[@class="keyword"]')
         for i in total:
             tr.append(i.text)

         Over = driver.find_elements(By.XPATH, '//*[@class="review"]//span')
         for i in Over:
             overall_rating.append(i.text)

         city = driver.find_elements(By.XPATH, '//*[@class="property-distance"]//span')
         for i in city:
             distance.append(i.text)

         facilities = driver.find_elements(By.XPATH, '//*[@class="badge-wrapper theme-')
         for i in facilities:
             facility.append(i.text)

         dorm_p = driver.find_elements(By.XPATH, '//*[@class="property-accommodation-p')
         for i in dorm_p:
             dorm.append(i.text)

```

```

In [59]: print(len(title),len(rating),len(tr),len(overall_rating),len(distance),len(facility),len(dorm))

30 31 31 30 30 34 8

```

```
In [60]: df = pd.DataFrame({'Hostal Name':title})
df1 = pd.DataFrame({'Rating':tr})
df2 = pd.DataFrame({'Total Rating':rating})
df3 = pd.DataFrame({'Overall Ratings':overall_rating})
df4 = pd.DataFrame({'Distance From City Center':distance})
df5 = pd.DataFrame({'Fcility':facility})
df6 = pd.DataFrame({'Dorm':dorm})
df7 = pd.concat([df,df1,df2,df3,df4,df5,df6],axis =1)
df7.head(30)
```

Out[60]:

	Hostal Name	Rating	Total Rating	Overall Ratings	Distance From City Center	Facility	Dorm
0	St Christopher's Village	Fabulous	8.2	(12630)	- 1.8km from city centre		₹2,128
1	Onefam Notting Hill	Fabulous	8.2	(2386)	- 5.5km from city centre		₹1,571
2	Urbany Hostel London	Superb	9.6	(1013)	- 5.4km from city centre		₹2,105
3	Generator London	Superb	9.5	(7920)	- 3km from city centre		₹1,708
4	Safestay London Elephant & Castle	Very Good	7.5	(5194)	- 1.7km from city centre		₹1,620
5	Astor Hyde Park	Very Good	7.1	(12243)	- 4.3km from city centre		₹2,415
6	NX London Hostel	Fabulous	8.2	(2045)	- 6.1km from city centre		₹1,848
7	Safestay London Kensington Holland Park	Fabulous	8.3	(1718)	- 5.8km from city centre		₹1,850
8	Clink261	Good	6.9	(448)	- 3.2km from city centre		NaN
9	No.8 Willesden Hostel London	Fabulous	8.0	(5139)	- 10km from city centre		NaN
10	Smart Russell Square Hostel	Very Good	7.1	(10144)	- 2.6km from city centre		NaN
11	Destinations Hostels @ The Gallery	Very Good	7.5	(275)	- 1.7km from city centre		NaN
12	Smart Hyde Park Inn Hostel	Superb	9.1	(6766)	- 5km from city centre		NaN
13	Phoenix Hostel	Very Good	7.6	(4538)	- 4.2km from city centre		NaN
14	No.8 Seven Sisters	Very Good	7.1	(4116)	- 9km from city centre		NaN
15	Hostelle - women only hostel London	Good	6.5	(77)	- 5.1km from city centre		NaN
16	YHA London Oxford Street	Fabulous	8.8	(4673)	- 2.1km from city centre		NaN
17	St Christopher's Hammersmith	Superb	9.0	(4340)	- 7.5km from city centre		NaN
18	Prime Backpackers Angel	Very Good	7.6	(835)	- 3.6km from city centre		NaN
19	Barmy Badger Backpackers	Fabulous	8.1	(2009)	- 5.5km from city centre		NaN
20	London Backpackers	Superb	9.0	(4544)	- 11.9km from city centre		NaN
21	Astor Victoria	Fabulous	8.0	(15056)	- 1.8km from city centre		NaN
22	Saint James Backpackers	Very Good	7.1	(1940)	- 5.5km from city centre		NaN

	Hostal Name	Rating	Total Rating	Overall Ratings	Distance From City Center	Fcility	Dorm
23	Kabannas London St Pancras	Very Good	7.7	(2941)	- 3.3km from city centre		NaN
24	Book a Bed Hostels	Fabulous	8.3	(1288)	- 6.9km from city centre		NaN
25	St Christopher's Camden	Very Good	7.4	(4104)	- 4.3km from city centre		NaN
26	PubLove @ The Steam Engine, Waterloo	Very Good	7.4	(436)	- 0.5km from city centre		NaN
27	Bell House Hostel	Very Good	7.8	(112)	- 4.2km from city centre		NaN
28	Smart Hyde Park View Hostel	Very Good	7.1	(4730)	- 4.8km from city centre		NaN
29	PubLove @ The Rose & Crown	Very Good	7.7	(214)	- 1.6km from city centre		NaN

In []: