{ "cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Web Scraping Assignment 3 : Solutions"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\Srishti Maan\\anaconda3\\lib\\site-packages\\requests\\\_\_init\_\_.py:89: RequestsDependencyWarning: urllib3 (1.26.6) or chardet (3.0.4) doesn't match a supported version!\n",

" warnings.warn(\"urllib3 ({}) or chardet ({}) doesn't match a supported \"\n"

]

}

],

"source": [

"# Importing Libraries\n",

"import selenium\n",

"import pandas as pd\n",

"import time\n",

"from bs4 import BeautifulSoup\n",

"\n",

"# Importing selenium webdriver \n",

"from selenium import webdriver\n",

"\n",

"# Importing required Exceptions which needs to handled\n",

"from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException\n",

"\n",

"#Importing requests\n",

"import requests\n",

"\n",

"# importing regex\n",

"import re"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q1- Write a python program which searches all the product under a particular product vertical\n",

"from www.amazon.in. The product verticals to be searched will be taken as input from user.\n",

"For e.g. If user input is â€˜guitarâ€™. Then search for guitars."

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Enter the product you want to search : Guitar\n"

]

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(3)\n",

"\n",

"# Opening the homepage of Amazon.in\n",

"url = \"https://www.amazon.in/\"\n",

"driver.get(url)\n",

"\n",

"time.sleep(2)\n",

"# Asking the user to input the keywords he/she wants to search\n",

"user\_inp = input('Enter the product you want to search : ')"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [],

"source": [

"search\_bar = driver.find\_element\_by\_id(\"twotabsearchtextbox\") # Locating searc\_bar by id\n",

"search\_bar.clear() # clearing search\_bar\n",

"search\_bar.send\_keys(user\_inp) # sending user input to search bar\n",

"search\_button = driver.find\_element\_by\_xpath('//div[@class=\"nav-search-submit nav-sprite\"]/span/input') # Locating search\_button by xpath\n",

"search\_button.click() # Clicking the button to start search"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q2- In the above question, now scrape the following details of each product listed in first 3 pages \n",

"of your search results and save it in a dataframe and csv. In case if any product vertical has \n",

"less than 3 pages in search results then scrape all the products available under that product \n",

"vertical. Details to be scraped are: \"Brand Name\", \"Name of the Product\", \"Rating\", \"No. of \n",

"Ratings\", \"Price\", \"Return/Exchange\", \"Expected Delivery\", \"Availability\", \"Other Details\" \n",

"and â€œProduct URLâ€. In case, if any of the details are missing for any of the product then \n",

"replace it by â€œ-â€œ."

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Brand</th>\n",

" <th>Name</th>\n",

" <th>Rating</th>\n",

" <th>No. of ratings</th>\n",

" <th>Price</th>\n",

" <th>Return/Exchange</th>\n",

" <th>Expected Delivery</th>\n",

" <th>Availability</th>\n",

" <th>Other Details</th>\n",

" <th>URL</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Visit the JUAREZ Store</td>\n",

" <td>JuÃ¢rez Acoustic Guitar, 38 Inch Cutaway, 038C ...</td>\n",

" <td>3.9 out of 5</td>\n",

" <td>9,448 global ratings</td>\n",

" <td>â‚¹ 2,369.00</td>\n",

" <td>7 Days Replacement</td>\n",

" <td></td>\n",

" <td>In stock.</td>\n",

" <td>Black Glossy Finish, Number of Frets: 18, Acou...</td>\n",

" <td>https://www.amazon.in/Juarez-Acoustic-Cutaway-...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Visit the Intern Store</td>\n",

" <td>Intern INT-38C Acoustic Guitar Kit, With Bag, ...</td>\n",

" <td>3.9 out of 5</td>\n",

" <td>5,347 global ratings</td>\n",

" <td>â‚¹ 2,249.00</td>\n",

" <td>7 Days Replacement</td>\n",

" <td>This item cannot be shipped to your selected d...</td>\n",

" <td>In stock.</td>\n",

" <td>Great looks with an innovative design to produ...</td>\n",

" <td>https://www.amazon.in/Intern-INT-38C-Acoustic-...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Visit the JUAREZ Store</td>\n",

" <td>Juarez JRZ38C Acoustic Guitar, 38 Inch Cutaway...</td>\n",

" <td>4 out of 5</td>\n",

" <td>3,236 global ratings</td>\n",

" <td>â‚¹ 2,499.00</td>\n",

" <td>7 Days Replacement</td>\n",

" <td>This item cannot be shipped to your selected d...</td>\n",

" <td>In stock.</td>\n",

" <td>Black glossy finish, number of frets: 18, acou...</td>\n",

" <td>https://www.amazon.in/JUAREZ-JRZ38C-Acoustic-S...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Visit the Kadence Store</td>\n",

" <td>Kadence Acoustica Series Semi Acoustic Ash Woo...</td>\n",

" <td>3.9 out of 5</td>\n",

" <td>444 global ratings</td>\n",

" <td>â‚¹ 7,999.00</td>\n",

" <td>7 Days Replacement</td>\n",

" <td>This item cannot be shipped to your selected d...</td>\n",

" <td>In stock.</td>\n",

" <td>Product Type : Acoustic Guitar Fretboard mater...</td>\n",

" <td>https://aax-eu.amazon.in/x/c/QktLOrBU6p1H9iGsd...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Visit the Mexa Store</td>\n",

" <td>Mexa Acoustic Guitar Bag Foam Padded For 38; 3...</td>\n",

" <td>4.3 out of 5</td>\n",

" <td>924 global ratings</td>\n",

" <td>â‚¹ 950.00</td>\n",

" <td>7 Days Replacement</td>\n",

" <td>This item cannot be shipped to your selected d...</td>\n",

" <td>Only 1 left in stock.</td>\n",

" <td>10 mm thick foam padding used for extra protec...</td>\n",

" <td>https://www.amazon.in/Yamaha-Guitars-Cover-Sta...</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Brand Name \\\n",

"0 Visit the JUAREZ Store JuÃ¢rez Acoustic Guitar, 38 Inch Cutaway, 038C ... \n",

"1 Visit the Intern Store Intern INT-38C Acoustic Guitar Kit, With Bag, ... \n",

"2 Visit the JUAREZ Store Juarez JRZ38C Acoustic Guitar, 38 Inch Cutaway... \n",

"3 Visit the Kadence Store Kadence Acoustica Series Semi Acoustic Ash Woo... \n",

"4 Visit the Mexa Store Mexa Acoustic Guitar Bag Foam Padded For 38; 3... \n",

"\n",

" Rating No. of ratings Price Return/Exchange \\\n",

"0 3.9 out of 5 9,448 global ratings â‚¹ 2,369.00 7 Days Replacement \n",

"1 3.9 out of 5 5,347 global ratings â‚¹ 2,249.00 7 Days Replacement \n",

"2 4 out of 5 3,236 global ratings â‚¹ 2,499.00 7 Days Replacement \n",

"3 3.9 out of 5 444 global ratings â‚¹ 7,999.00 7 Days Replacement \n",

"4 4.3 out of 5 924 global ratings â‚¹ 950.00 7 Days Replacement \n",

"\n",

" Expected Delivery Availability \\\n",

"0 In stock. \n",

"1 This item cannot be shipped to your selected d... In stock. \n",

"2 This item cannot be shipped to your selected d... In stock. \n",

"3 This item cannot be shipped to your selected d... In stock. \n",

"4 This item cannot be shipped to your selected d... Only 1 left in stock. \n",

"\n",

" Other Details \\\n",

"0 Black Glossy Finish, Number of Frets: 18, Acou... \n",

"1 Great looks with an innovative design to produ... \n",

"2 Black glossy finish, number of frets: 18, acou... \n",

"3 Product Type : Acoustic Guitar Fretboard mater... \n",

"4 10 mm thick foam padding used for extra protec... \n",

"\n",

" URL \n",

"0 https://www.amazon.in/Juarez-Acoustic-Cutaway-... \n",

"1 https://www.amazon.in/Intern-INT-38C-Acoustic-... \n",

"2 https://www.amazon.in/JUAREZ-JRZ38C-Acoustic-S... \n",

"3 https://aax-eu.amazon.in/x/c/QktLOrBU6p1H9iGsd... \n",

"4 https://www.amazon.in/Yamaha-Guitars-Cover-Sta... "

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# scrape all product urls\n",

"product\_urls = []\n",

"start=0\n",

"end=3\n",

"for page in range(start,end):#for loop for scrapping 3 page\n",

" url=driver.find\_elements\_by\_xpath('//a[@class=\"a-link-normal a-text-normal\"]') #scraping urls\n",

" for i in url:\n",

" product\_urls.append(i.get\_attribute(\"href\")) #appending the urls in product\_urls list\n",

" nxt\_button=driver.find\_element\_by\_xpath(\"//li[@class='a-last']//a\") #scraping the list of buttons from the page\n",

" nxt\_button.click()\n",

" time.sleep(2)\n",

"\n",

"#Make empty lists\n",

"Brand = [] \n",

"Name = []\n",

"Rating = []\n",

"no\_of\_ratings = []\n",

"Price = []\n",

"Return = []\n",

"expected\_delivery = []\n",

"Availability = [] \n",

"Other\_Details = []\n",

"\n",

"for url in product\_urls:\n",

" driver.get(url)\n",

" time.sleep(2)\n",

" \n",

" # Extracting Brand from xpath\n",

" try:\n",

" brand = driver.find\_element\_by\_xpath('//a[@id=\"bylineInfo\"]') \n",

" Brand.append(brand.text)\n",

" except NoSuchElementException:\n",

" Brand.append('-') \n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting name from id\n",

" try:\n",

" name = driver.find\_element\_by\_id('productTitle') \n",

" Name.append(name.text)\n",

" except NoSuchElementException:\n",

" Name.append('-')\n",

"\n",

"\n",

" time.sleep(1) \n",

" # Extracting Ratings from xpath \n",

" try:\n",

" rating = driver.find\_element\_by\_xpath('//span[@class=\"a-size-base a-nowrap\"]//span') \n",

" Rating.append(rating.text)\n",

" except NoSuchElementException:\n",

" Rating.append('-')\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting No of Ratings from xpath\n",

" try:\n",

" no\_rating = driver.find\_element\_by\_xpath('//span[@class=\"a-size-base a-color-secondary\"]') \n",

" no\_of\_ratings.append(no\_rating.text)\n",

" except NoSuchElementException:\n",

" no\_of\_ratings.append('-')\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting price from xpath \n",

" try:\n",

" price = driver.find\_element\_by\_xpath('//td[@class=\"a-span12\"]') \n",

" Price.append(price.text)\n",

" except NoSuchElementException:\n",

" Price.append('-')\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting Return from xpath\n",

" try:\n",

" return\_ = driver.find\_element\_by\_xpath('//a[@class=\"a-size-small a-link-normal a-text-normal\"]') \n",

" Return.append(return\_.text)\n",

" except NoSuchElementException:\n",

" Return.append('-')\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting expected\_delivery from xpath\n",

" try:\n",

" exptd\_dlvry= driver.find\_element\_by\_xpath('//span[@class=\"a-color-error\"]') \n",

" expected\_delivery.append(exptd\_dlvry.text)\n",

" except NoSuchElementException:\n",

" expected\_delivery.append('-')\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting Availability from xpath \n",

" try:\n",

" avl= driver.find\_element\_by\_xpath('//span[@class=\"a-size-medium a-color-success\"]') \n",

" Availability.append(avl.text)\n",

" except NoSuchElementException:\n",

" Availability.append('-')\n",

"\n",

"\n",

"\n",

" time.sleep(1)\n",

" # Extracting Other Details from xpath \n",

" try:\n",

" othr\_dtls= driver.find\_element\_by\_xpath('//ul[@class=\"a-unordered-list a-vertical a-spacing-mini\"]') \n",

" Other\_Details.append(othr\_dtls.text)\n",

" except NoSuchElementException:\n",

" Other\_Details.append('-')\n",

"\n",

"time.sleep(2) \n",

"# Make dataframe\n",

"prod\_df = pd.DataFrame({'Brand':Brand,'Name':Name,'Rating':Rating,'No. of ratings':no\_of\_ratings,'Price':Price,\n",

" 'Return/Exchange':Return,'Expected Delivery':expected\_delivery,'Availability':Availability,\n",

" 'Other Details':Other\_Details,'URL':product\_urls})\n",

"prod\_df.head(5) "

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* 3-Write a python program to access the search bar and search button on images.google.com and \n",

"scrape 10 images each for keywords â€˜fruitsâ€™, â€˜carsâ€™ and â€˜Machine Learningâ€™,â€˜Guitarâ€™, â€˜Cakesâ€™."

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"<ipython-input-21-b45e2335bbd1>:2: DeprecationWarning: executable\_path has been deprecated, please pass in a Service object\n",

" driver=webdriver.Chrome(\"chromedriver.exe\")\n"

]

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(3)\n",

"\n",

"# Opening the google images\n",

"url = \"https://images.google.com/\"\n",

"driver.get(url)\n",

"search\_bar = driver.find\_element\_by\_xpath('//\*[@id=\"sbtc\"]/div/div[2]/input') # Finding the search bar using it's xpath\n",

"search\_bar.send\_keys(\"cakes\") # Inputing \"cakes\" keyword to search images\n",

"search\_button = driver.find\_element\_by\_xpath('//\*[@id=\"sbtc\"]/button') # Finding the xpath of search button\n",

"search\_button.click() # Clicking the search button"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Downloading 0 of 10 images\n",

"Downloading 1 of 10 images\n",

"Downloading 2 of 10 images\n",

"Downloading 3 of 10 images\n",

"Downloading 4 of 10 images\n",

"Downloading 5 of 10 images\n",

"Downloading 6 of 10 images\n",

"Downloading 7 of 10 images\n",

"Downloading 8 of 10 images\n",

"Downloading 9 of 10 images\n",

"Downloading 10 of 10 images\n"

]

}

],

"source": [

"\n",

"#20 time we scroll down by 100 in order to generate more images on the website\n",

"for \_ in range(20):\n",

" driver.execute\_script(\"window.scrollBy(0,1000)\")\n",

" \n",

"images = driver.find\_elements\_by\_xpath('//img[@class=\"rg\_i Q4LuWd\"]')\n",

"\n",

"img\_urls = []\n",

"img\_data = []\n",

"for image in images:\n",

" source= image.get\_attribute('src')\n",

" if source is not None:\n",

" if(source[0:4] == 'http'):\n",

" img\_urls.append(source)\n",

" \n",

" \n",

"for i in range(len(img\_urls)):\n",

" if i > 10:\n",

" break\n",

" print(\"Downloading {0} of {1} images\" .format(i, 10))\n",

" response= requests.get(img\_urls[i])\n",

" file = open(r\"E:\\AssignmentFlipRobo\"+str(i)+\".jpg\", \"wb\")\n",

" file.write(response.content)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q4. Write a python program to search for a smartphone(e.g.: Oneplus Nord, pixel 4A, etc.) on www.flipkart.com and scrape following details for all the search results displayed on 1st page. Details to be scraped: â€œBrand Nameâ€, â€œSmartphone nameâ€, â€œColourâ€, â€œRAMâ€, â€œStorage(ROM)â€, â€œPrimary Cameraâ€, â€œSecondary Cameraâ€, â€œDisplay Sizeâ€, â€œDisplay Resolutionâ€, â€œProcessorâ€, â€œProcessor Coresâ€, â€œBattery Capacityâ€, â€œPriceâ€, â€œProduct URLâ€. Incase if any of the details is missing then replace it by â€œ- â€œ. Save your results in a dataframe and CSV."

]

},

{

"cell\_type": "code",

"execution\_count": 67,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" Enter the name of the mobile phone you want to search : pixel\n"

]

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(3)\n",

"\n",

"# opning flipkart.com\n",

"driver.get('https://www.flipkart.com/')\n",

"time.sleep(3)\n",

"try:\n",

" login\_X\_button = driver.find\_element\_by\_xpath('//button[@class=\"\_2KpZ6l \_2doB4z\"]') # Button to close login popup\n",

" login\_X\_button.click()\n",

"except NoSuchElementException : \n",

" print(\"No Login page\")\n",

"search\_bar = driver.find\_element\_by\_xpath('//\*[@id=\"container\"]/div/div[1]/div[1]/div[2]/div[2]/form/div/div/input') # Finding the search bar using it's xpath\n",

"search\_bar.clear() # Clearing the search bar\n",

"search\_bar.send\_keys(\"pixel\") # Inputing keyword to search\n",

"search\_button = driver.find\_element\_by\_xpath('//button[@class=\"L0Z3Pu\"]') # Finding the xpath of search button\n",

"search\_button.click() # Clicking the search button"

]

},

{

"cell\_type": "code",

"execution\_count": 68,

"metadata": {},

"outputs": [],

"source": [

"# Fetching urls of phones coming on 1st page\n",

"flip\_urls = []\n",

"urls = driver.find\_elements\_by\_xpath('//a[@class=\"\_1fQZEK\"]')\n",

"for url in urls:\n",

" flip\_urls.append(url.get\_attribute(\"href\"))\n",

" \n",

" \n",

" \n",

"#Make empty lists \n",

"flip\_dict = {}\n",

"flip\_dict[\"Brand\"] = []\n",

"flip\_dict[\"Smartphone\"] = []\n",

"flip\_dict[\"Colour\"] = []\n",

"flip\_dict[\"RAM\"] = []\n",

"flip\_dict[\"Storage(ROM)\"] = []\n",

"flip\_dict[\"Primary Camera\"] = []\n",

"flip\_dict[\"Secondary Camera\"] = []\n",

"flip\_dict[\"Display Size\"] = []\n",

"flip\_dict[\"Display Resolution\"] = []\n",

"flip\_dict[\"Processor\"] = []\n",

"flip\_dict[\"Processor Cores\"] = []\n",

"flip\_dict[\"Battery Capacity\"] = []\n",

"flip\_dict[\"Battery Type\"] = []\n",

"flip\_dict[\"Price\"] = []\n",

"flip\_dict[\"URL\"] = []"

]

},

{

"cell\_type": "code",

"execution\_count": 71,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Scraping URL = https://www.flipkart.com/google-pixel-3a-clearly-white-64-gb/p/itmfgk4jfgstaack?pid=MOBFFGFPJSCEXMSG&lid=LSTMOBFFGFPJSCEXMSGODGRZE&marketplace=FLIPKART&srno=s\_1\_1&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFFGFPJSCEXMSG.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/google-pixel-3a-just-black-64-gb/p/itmfgk4jfgstaack?pid=MOBFFGFP7UHHJUZU&lid=LSTMOBFFGFP7UHHJUZUW7182D&marketplace=FLIPKART&srno=s\_1\_2&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFFGFP7UHHJUZU.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/lg-g8x-black-128-gb/p/itme8a4f5f473aa4?pid=MOBFZKQWFRFMHKQK&lid=LSTMOBFZKQWFRFMHKQKVOP4L4&marketplace=FLIPKART&srno=s\_1\_3&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFZKQWFRFMHKQK.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/samsung-galaxy-m21-raven-black-64-gb/p/itm771ab543df368?pid=MOBFSF867ET7DDMZ&lid=LSTMOBFSF867ET7DDMZCGQLQ7&marketplace=FLIPKART&srno=s\_1\_4&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFSF867ET7DDMZ.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pro-interstellar-black-128-gb/p/itm0418537d115ba?pid=MOBFUZYNSYP64TQK&lid=LSTMOBFUZYNSYP64TQKRLCVXM&marketplace=FLIPKART&srno=s\_1\_5&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUZYNSYP64TQK.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-prime-space-blue-128-gb/p/itmcf3b9282de0fd?pid=MOBFUT9YYVEXJHF2&lid=LSTMOBFUT9YYVEXJHF2BN6GOP&marketplace=FLIPKART&srno=s\_1\_6&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUT9YYVEXJHF2.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-sky-blue-64-gb/p/itm4fb151383983b?pid=MOBFV5FPUS8UHG9H&lid=LSTMOBFV5FPUS8UHG9HWEZHFA&marketplace=FLIPKART&srno=s\_1\_7&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFV5FPUS8UHG9H.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-carbon-black-64-gb/p/itm4fb151383983b?pid=MOBFVHMPGATFZXWR&lid=LSTMOBFVHMPGATFZXWRAMCBY4&marketplace=FLIPKART&srno=s\_1\_8&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFVHMPGATFZXWR.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/karbonn-kx3/p/itm60cf67471b066?pid=MOBFKVM39K9MDYFV&lid=LSTMOBFKVM39K9MDYFVQ1TN2Y&marketplace=FLIPKART&srno=s\_1\_9&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFKVM39K9MDYFV.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/itel-vision1-gradation-green-32-gb/p/itmabec72dce88ad?pid=MOBFV6FGB3MXFQN4&lid=LSTMOBFV6FGB3MXFQN4A1Z9F8&marketplace=FLIPKART&srno=s\_1\_10&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFV6FGB3MXFQN4.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/kall-k6300-new/p/itm16befd1f65f1e?pid=MOBFTYU5NDX6SXDT&lid=LSTMOBFTYU5NDX6SXDTCPUHKN&marketplace=FLIPKART&srno=s\_1\_11&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFTYU5NDX6SXDT.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-prime-matteblack-128-gb/p/itmb9bf25ba21cb7?pid=MOBFUSED4AYBFUVH&lid=LSTMOBFUSED4AYBFUVH9CLUIA&marketplace=FLIPKART&srno=s\_1\_12&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUSED4AYBFUVH.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-prime-mintgreen-128-gb/p/itmb9bf25ba21cb7?pid=MOBFUT5YRUHBGQH3&lid=LSTMOBFUT5YRUHBGQH37N6QIB&marketplace=FLIPKART&srno=s\_1\_13&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUT5YRUHBGQH3.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-9-prime-space-blue-64-gb/p/itmb9bf25ba21cb7?pid=MOBFUT9M84NDDYQN&lid=LSTMOBFUT9M84NDDYQNDPDOI1&marketplace=FLIPKART&srno=s\_1\_14&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUT9M84NDDYQN.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/samsung-m01-core-black-16-gb/p/itmcc59efe93b40b?pid=MOBFUAEBGZKRFGHX&lid=LSTMOBFUAEBGZKRFGHX4PPGAD&marketplace=FLIPKART&srno=s\_1\_15&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUAEBGZKRFGHX.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/samsung-galaxy-m21-midnight-blue-64-gb/p/itm322eb5a51f9a2?pid=MOBFSF86NH2HVZAW&lid=LSTMOBFSF86NH2HVZAWWGLN03&marketplace=FLIPKART&srno=s\_1\_16&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFSF86NH2HVZAW.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-aqua-green-128-gb/p/itm12201c2ee63e5?pid=MOBFU3ZFY6FGK4ZU&lid=LSTMOBFU3ZFY6FGK4ZUM4KTNY&marketplace=FLIPKART&srno=s\_1\_17&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFU3ZFY6FGK4ZU.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/samsung-m01-core-blue-16-gb/p/itma9e257004962e?pid=MOBFUAEBBSAWG6HG&lid=LSTMOBFUAEBBSAWG6HGDMUUN8&marketplace=FLIPKART&srno=s\_1\_18&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUAEBBSAWG6HG.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/samsung-galaxy-m31-ocean-blue-64-gb/p/itm1268b57512fb8?pid=MOBFPNPS6GTGZHE4&lid=LSTMOBFPNPS6GTGZHE48OUZUH&marketplace=FLIPKART&srno=s\_1\_19&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFPNPS6GTGZHE4.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pro-aurora-blue-128-gb/p/itma84d60532d415?pid=MOBFUUN6B5XSFYXT&lid=LSTMOBFUUN6B5XSFYXTBNVX5C&marketplace=FLIPKART&srno=s\_1\_20&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUUN6B5XSFYXT.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pro-glacier-white-64-gb/p/itma84d60532d415?pid=MOBFPZXU9QDDSKNU&lid=LSTMOBFPZXU9QDDSKNUZWULL1&marketplace=FLIPKART&srno=s\_1\_21&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFPZXU9QDDSKNU.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pro-champagne-gold-128-gb/p/itma84d60532d415?pid=MOBFVNYYZZ6AJT44&lid=LSTMOBFVNYYZZ6AJT445SIQD7&marketplace=FLIPKART&srno=s\_1\_22&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFVNYYZZ6AJT44.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pro-glacier-white-128-gb/p/itma84d60532d415?pid=MOBFUSGFZDDJHWMG&lid=LSTMOBFUSGFZDDJHWMGKFPUF1&marketplace=FLIPKART&srno=s\_1\_23&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFUSGFZDDJHWMG.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n",

"Scraping URL = https://www.flipkart.com/redmi-note-9-pebble-grey-64-gb/p/itm12201c2ee63e5?pid=MOBFU3ZFQ3GAGANG&lid=LSTMOBFU3ZFQ3GAGANGMKROVO&marketplace=FLIPKART&srno=s\_1\_24&otracker=search&otracker1=search&fm=SEARCH&iid=a5e3e22a-6fdf-4ba3-b771-a604ebad02a9.MOBFU3ZFQ3GAGANG.SEARCH&ppt=sp&ppn=sp&ssid=6lybxosh2o0000001605705507073&qH=ab4086ecd47c568d\n"

]

}

],

"source": [

"# Scraping data from each url\n",

"for url in flip\_urls:\n",

" driver.get(url) # Saving url \n",

" print(\"Scraping URL = \", url)\n",

" flip\_dict['URL'].append(url) # Loading the webpage by url\n",

" time.sleep(2)\n",

" \n",

" try:\n",

" read\_more = driver.find\_element\_by\_xpath('//button[@class=\"\_2KpZ6l \_1FH0tX\"]') # Button for expanding the specs\n",

" read\_more.click()\n",

" except NoSuchElementException:\n",

" print(\"Exception Occured. Moving to next page\")\n",

" \n",

" try:\n",

" brand = driver.find\_element\_by\_xpath('//span[@class=\"B\_NuCI\"]') # Extracting Brand from xpath\n",

" flip\_dict[\"Brand\"].append(brand.text.split()[0])\n",

" except NoSuchElementException:\n",

" flip\_dict['Brand'].append('-')\n",

" \n",

" try:\n",

" price = driver.find\_element\_by\_xpath('//div[@class=\"\_30jeq3 \_16Jk6d\"]') # Extracting Price from xpath\n",

" flip\_dict['Price'].append(price.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Price'].append('-')\n",

" \n",

" try:\n",

" name = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][1]/table/tbody/tr[3]/td[2]/ul/li') # Extracting Name from xpath\n",

" flip\_dict['Smartphone'].append(name.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Smartphone'].append('-')\n",

" \n",

" try:\n",

" color = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][1]/table/tbody/tr[4]/td[2]/ul/li') # Extracting colour from xpath\n",

" flip\_dict['Colour'].append(color.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Colour'].append('-')\n",

" \n",

" try:\n",

" disp\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][2]/div')\n",

" if disp\_chk.text != \"Display Features\" : raise NoSuchElementException\n",

" disp\_size = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][2]/table[1]/tbody/tr[1]/td[2]/ul/li') # Extracting Display Size from xpath\n",

" flip\_dict['Display Size'].append(disp\_size.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Display Size'].append('-')\n",

" \n",

" try:\n",

" disp\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][2]/div')\n",

" if disp\_chk.text != \"Display Features\" : raise NoSuchElementException\n",

" disp\_res = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][2]/table[1]/tbody/tr[2]/td[2]/ul/li') # Extracting Display Resolution from xpath\n",

" flip\_dict['Display Resolution'].append(disp\_res.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Display Resolution'].append('-')\n",

" \n",

" try:\n",

" pro\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[2]/td[1]')\n",

" if pro\_chk.text != \"Processor Type\" : raise NoSuchElementException\n",

" processor = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[2]/td[2]/ul/li') # Extracting processor from xpath\n",

" Processor.append(processor.text)\n",

" except NoSuchElementException:\n",

" Processor.append('-')\n",

" \n",

" try: # Extracting Processor Cores from xpath\n",

" core\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[3]/td[1]')\n",

" if core\_chk.text != \"Processor Core\" :\n",

" core\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[2]/td[1]')\n",

" if core\_chk.text != \"Processor Core\" : \n",

" raise NoSuchElementException\n",

" else :\n",

" cores = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[2]/td[2]/ul/li')\n",

" else :\n",

" cores = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][3]/table[1]/tbody/tr[3]/td[2]/ul/li')\n",

" flip\_dict['Processor Cores'].append(cores.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Processor Cores'].append('-')\n",

" \n",

" try:\n",

" rom = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][4]/table[1]/tbody/tr[1]/td[2]/ul/li') # Extracting Storage(ROM) from xpath\n",

" flip\_dict['Storage(ROM)'].append(rom.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Storage(ROM)'].append('-')\n",

" \n",

" try:\n",

" ram = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][4]/table[1]/tbody/tr[2]/td[2]/ul/li') # Extracting RAM from xpath\n",

" flip\_dict['RAM'].append(ram.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['RAM'].append('-')\n",

" \n",

" try: \n",

" pri\_cam = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][5]/table[1]/tbody/tr[2]/td[2]/ul/li') # Extracting Camera from xpath\n",

" flip\_dict['Primary Camera'].append(pri\_cam.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Primary Camera'].append('-')\n",

" \n",

" try: # Extracting Secondary Camera from xpath\n",

" cam\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][5]/table[1]/tbody/tr[6]/td[1]')\n",

" if cam\_chk != \"Secondary Camera\" : \n",

" if driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][5]/table[1]/tbody/tr[5]/td[1]').text == \"Secondary Camera\":\n",

" sec\_cam = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][5]/table[1]/tbody/tr[5]/td[2]/ul/li')\n",

" else :\n",

" raise NoSuchElementException\n",

" else :\n",

" sec\_cam = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][5]/table[1]/tbody/tr[6]/td[2]/ul/li')\n",

" flip\_dict['Secondary Camera'].append(sec\_cam.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Secondary Camera'].append('-')\n",

" \n",

" try:\n",

" if driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/div').text != \"Battery & Power Features\" : # Extracting Battery Capacity from xpath\n",

" if driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/div').text == \"Battery & Power Features\" :\n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/table/tbody/tr/td[1]')\n",

" if bat\_chk.text != \"Battery Capacity\" : raise NoSuchElementException\n",

" bat\_cap = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/table/tbody/tr/td[2]/ul/li') \n",

" elif driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/div').text == \"Battery & Power Features\" : \n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/table/tbody/tr/td[1]')\n",

" if bat\_chk.text != \"Battery Capacity\" : raise NoSuchElementException\n",

" bat\_cap = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/table/tbody/tr/td[2]/ul/li')\n",

" else:\n",

" raise NoSuchElementException\n",

" else :\n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/table/tbody/tr/td[1]')\n",

" if bat\_chk.text != \"Battery Capacity\" : raise NoSuchElementException\n",

" bat\_cap = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/table/tbody/tr/td[2]/ul/li') # Extracting Availability from xpath\n",

" flip\_dict['Battery Capacity'].append(bat\_cap.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Battery Capacity'].append('-')\n",

" \n",

" try:\n",

" if driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/div').text != \"Battery & Power Features\" :\n",

" if driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/div').text == \"Battery & Power Features\" :\n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/table/tbody/tr[2]/td[1]')\n",

" if bat\_chk.text != \"Battery Type\" : raise NoSuchElementException\n",

" bat\_typ = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][9]/table/tbody/tr[2]/td[2]/ul/li')\n",

" elif driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/div').text == \"Battery & Power Features\" :\n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/table/tbody/tr[2]/td[1]')\n",

" if bat\_chk.text != \"Battery Type\" : raise NoSuchElementException\n",

" bat\_typ = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][8]/table/tbody/tr[2]/td[2]/ul/li')\n",

" else:\n",

" raise NoSuchElementException\n",

" else :\n",

" bat\_chk = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/table/tbody/tr[2]/td[1]')\n",

" if bat\_chk.text != \"Battery Type\" : raise NoSuchElementException\n",

" bat\_typ = driver.find\_element\_by\_xpath('//div[@class=\"\_3k-BhJ\"][10]/table/tbody/tr[2]/td[2]/ul/li') # Extracting Battery Type from xpath\n",

" flip\_dict['Battery Type'].append(bat\_typ.text)\n",

" except NoSuchElementException:\n",

" flip\_dict['Battery Type'].append('-')\n",

" \n",

" "

]

},

{

"cell\_type": "code",

"execution\_count": 73,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"24 24 24 24 24\n"

]

}

],

"source": [

"print(len(flip\_dict[\"Brand\"]), len(flip\_dict[\"Smartphone\"]), len(flip\_dict[\"Processor\"]), len(flip\_dict[\"Price\"]), len(flip\_dict['URL']))"

]

},

{

"cell\_type": "code",

"execution\_count": 74,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Brand</th>\n",

" <th>Smartphone</th>\n",

" <th>Colour</th>\n",

" <th>RAM</th>\n",

" <th>Storage(ROM)</th>\n",

" <th>Primary Camera</th>\n",

" <th>Secondary Camera</th>\n",

" <th>Display Size</th>\n",

" <th>Display Resolution</th>\n",

" <th>Processor</th>\n",

" <th>Processor Cores</th>\n",

" <th>Battery Capacity</th>\n",

" <th>Battery Type</th>\n",

" <th>Price</th>\n",

" <th>URL</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Google</td>\n",

" <td>Pixel 3a</td>\n",

" <td>Clearly White</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>12.2MP Rear Camera</td>\n",

" <td>8MP Front Camera</td>\n",

" <td>14.22 cm (5.6 inch)</td>\n",

" <td>2220 x 1080 pixels</td>\n",

" <td>Qualcomm Snapdragon 670</td>\n",

" <td>Octa Core</td>\n",

" <td>3000 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹30,999</td>\n",

" <td>https://www.flipkart.com/google-pixel-3a-clear...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Google</td>\n",

" <td>Pixel 3a</td>\n",

" <td>Just Black</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>12.2MP Rear Camera</td>\n",

" <td>8MP Front Camera</td>\n",

" <td>14.22 cm (5.6 inch)</td>\n",

" <td>2220 x 1080 pixels</td>\n",

" <td>Qualcomm Snapdragon 670</td>\n",

" <td>Octa Core</td>\n",

" <td>3000 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹30,999</td>\n",

" <td>https://www.flipkart.com/google-pixel-3a-just-...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>LG</td>\n",

" <td>G8X</td>\n",

" <td>Black</td>\n",

" <td>6 GB</td>\n",

" <td>128 GB</td>\n",

" <td>12MP + 13MP</td>\n",

" <td>-</td>\n",

" <td>16.26 cm (6.4 inch)</td>\n",

" <td>2340 x 1080 pixel</td>\n",

" <td>Qualcomm Snapdragonâ„¢ 855 Octa-core (up to 2.84...</td>\n",

" <td>Octa Core</td>\n",

" <td>4000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹39,990</td>\n",

" <td>https://www.flipkart.com/lg-g8x-black-128-gb/p...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Samsung</td>\n",

" <td>Galaxy M21</td>\n",

" <td>Raven Black</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>48MP + 8MP + 5MP</td>\n",

" <td>-</td>\n",

" <td>16.26 cm (6.4 inch)</td>\n",

" <td>2340 x 1080$$Pixel</td>\n",

" <td>Exynos 9611-Octa Core processor</td>\n",

" <td>Octa Core</td>\n",

" <td>6000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹15,982</td>\n",

" <td>https://www.flipkart.com/samsung-galaxy-m21-ra...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9 Pro</td>\n",

" <td>Interstellar Black</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>48MP + 8MP</td>\n",

" <td>-</td>\n",

" <td>16.94 cm (6.67 inch)</td>\n",

" <td>2400 x 1080$$pixel</td>\n",

" <td>Qualcomm Snapdragon 720G</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹15,199</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pro-inte...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>5</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9 Prime</td>\n",

" <td>Space Blue</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>13MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>-</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹11,335</td>\n",

" <td>https://www.flipkart.com/redmi-9-prime-space-b...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9</td>\n",

" <td>Sky Blue</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>13MP + 8MP</td>\n",

" <td>8MP Front Camera</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>720 x 1600$$pixel</td>\n",

" <td>MediaTek Helio G35</td>\n",

" <td>-</td>\n",

" <td>5000 mAh</td>\n",

" <td>Li-Po</td>\n",

" <td>â‚¹9,672</td>\n",

" <td>https://www.flipkart.com/redmi-9-sky-blue-64-g...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>7</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9</td>\n",

" <td>Carbon Black</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>13MP + 2MP</td>\n",

" <td>5MP Front Camera</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>720 x 1600$$pixel</td>\n",

" <td>MediaTek Helio G35</td>\n",

" <td>Octa Core</td>\n",

" <td>5000 mAh</td>\n",

" <td>Li-Po</td>\n",

" <td>â‚¹9,750</td>\n",

" <td>https://www.flipkart.com/redmi-9-carbon-black-...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8</th>\n",

" <td>Karbonn</td>\n",

" <td>KX3</td>\n",

" <td>Black,Red</td>\n",

" <td>32 MB</td>\n",

" <td>32 MB</td>\n",

" <td>0.3MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>4.57 cm (1.8 inch)</td>\n",

" <td>128 x 160$$pixel</td>\n",

" <td>-</td>\n",

" <td>-</td>\n",

" <td>800 mAh</td>\n",

" <td>Li-ion</td>\n",

" <td>â‚¹799</td>\n",

" <td>https://www.flipkart.com/karbonn-kx3/p/itm60cf...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9</th>\n",

" <td>Itel</td>\n",

" <td>VISION1</td>\n",

" <td>GRADATION GREEN</td>\n",

" <td>2 GB</td>\n",

" <td>32 GB</td>\n",

" <td>8MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>15.46 cm (6.088 inch)</td>\n",

" <td>1560 x 720$$pixel</td>\n",

" <td>UNISOC SC9863A OCTA CORE</td>\n",

" <td>Octa Core</td>\n",

" <td>4000 mAh</td>\n",

" <td>LI POLY</td>\n",

" <td>â‚¹6,499</td>\n",

" <td>https://www.flipkart.com/itel-vision1-gradatio...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>10</th>\n",

" <td>I</td>\n",

" <td>k6300 NEW</td>\n",

" <td>Green</td>\n",

" <td>64 MB</td>\n",

" <td>64 MB</td>\n",

" <td>Primary Camera</td>\n",

" <td>-</td>\n",

" <td>7.11 cm (2.8 inch)</td>\n",

" <td>240 x 320$$pixel</td>\n",

" <td>-</td>\n",

" <td>-</td>\n",

" <td>-</td>\n",

" <td>-</td>\n",

" <td>â‚¹1,169</td>\n",

" <td>https://www.flipkart.com/kall-k6300-new/p/itm1...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>11</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9 Prime</td>\n",

" <td>MatteBlack</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>13MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>-</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹11,576</td>\n",

" <td>https://www.flipkart.com/redmi-9-prime-mattebl...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>12</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9 Prime</td>\n",

" <td>MintGreen</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>13MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>-</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹11,536</td>\n",

" <td>https://www.flipkart.com/redmi-9-prime-mintgre...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>13</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi 9 Prime</td>\n",

" <td>Space Blue</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>13MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>-</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>-</td>\n",

" <td>â‚¹10,998</td>\n",

" <td>https://www.flipkart.com/redmi-9-prime-space-b...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14</th>\n",

" <td>Samsung</td>\n",

" <td>M01 core</td>\n",

" <td>Black</td>\n",

" <td>1 GB</td>\n",

" <td>16 GB</td>\n",

" <td>8MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>13.46 cm (5.3 inch)</td>\n",

" <td>1480 x 720$$pixel</td>\n",

" <td>MediaTek | MT6739WW quad core</td>\n",

" <td>Octa Core</td>\n",

" <td>3000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹5,059</td>\n",

" <td>https://www.flipkart.com/samsung-m01-core-blac...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>15</th>\n",

" <td>Samsung</td>\n",

" <td>Galaxy M21</td>\n",

" <td>Midnight Blue</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>48MP + 8MP + 5MP</td>\n",

" <td>-</td>\n",

" <td>16.26 cm (6.4 inch)</td>\n",

" <td>2340 x 1080$$Pixel</td>\n",

" <td>Exynos 9611-Octa Core processor</td>\n",

" <td>Octa Core</td>\n",

" <td>6000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹15,986</td>\n",

" <td>https://www.flipkart.com/samsung-galaxy-m21-mi...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>16</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9</td>\n",

" <td>Aqua Green</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>48MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>MediaTek Helio G85</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>Lithium Polyme</td>\n",

" <td>â‚¹14,399</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-aqua-gre...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>17</th>\n",

" <td>Samsung</td>\n",

" <td>M01 core</td>\n",

" <td>Blue</td>\n",

" <td>1 GB</td>\n",

" <td>16 GB</td>\n",

" <td>8MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>13.46 cm (5.3 inch)</td>\n",

" <td>1480 x 720$$pixel</td>\n",

" <td>MediaTek | MT6739WW quad core</td>\n",

" <td>Octa Core</td>\n",

" <td>3000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹5,022</td>\n",

" <td>https://www.flipkart.com/samsung-m01-core-blue...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>Samsung</td>\n",

" <td>Galaxy M31</td>\n",

" <td>Ocean Blue</td>\n",

" <td>6 GB</td>\n",

" <td>64 GB</td>\n",

" <td>64MP + 8MP + 5MP + 5MP</td>\n",

" <td>32MP Front Camera</td>\n",

" <td>16.26 cm (6.4 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>Samsung Exynos 9 Octa 9611</td>\n",

" <td>Octa Core</td>\n",

" <td>6000 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹17,474</td>\n",

" <td>https://www.flipkart.com/samsung-galaxy-m31-oc...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9 Pro</td>\n",

" <td>Aurora Blue</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>48MP + 8MP</td>\n",

" <td>-</td>\n",

" <td>16.94 cm (6.67 inch)</td>\n",

" <td>2400 x 1080$$pixel</td>\n",

" <td>Qualcomm Snapdragon 720G</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹15,448</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pro-auro...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9 Pro</td>\n",

" <td>Glacier White</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>48MP + 8MP + 5MP + 2MP</td>\n",

" <td>16MP Front Camera</td>\n",

" <td>16.94 cm (6.67 inch)</td>\n",

" <td>2400 x 1080 pixel</td>\n",

" <td>QualcommÂ® Snapdragonâ„¢ 720G</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹14,958</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pro-glac...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9 Pro</td>\n",

" <td>Champagne Gold</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>48MP + 8MP + 5MP + 2MP</td>\n",

" <td>16MP Front Camera</td>\n",

" <td>16.94 cm (6.67 inch)</td>\n",

" <td>2400 x 1080$$pixel</td>\n",

" <td>QualcommÂ® Snapdragonâ„¢ 720G</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹14,990</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pro-cham...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9 Pro</td>\n",

" <td>Glacier White</td>\n",

" <td>4 GB</td>\n",

" <td>128 GB</td>\n",

" <td>48MP + 8MP + 5MP + 2MP</td>\n",

" <td>16MP Front Camera</td>\n",

" <td>16.94 cm (6.67 inch)</td>\n",

" <td>2400 x 1080$$pixel</td>\n",

" <td>QualcommÂ® Snapdragonâ„¢ 720G</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>lithium-ion</td>\n",

" <td>â‚¹14,910</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pro-glac...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>Redmi</td>\n",

" <td>Redmi Note 9</td>\n",

" <td>Pebble Grey</td>\n",

" <td>4 GB</td>\n",

" <td>64 GB</td>\n",

" <td>48MP Rear Camera</td>\n",

" <td>-</td>\n",

" <td>16.59 cm (6.53 inch)</td>\n",

" <td>2340 x 1080$$pixel</td>\n",

" <td>MediaTek Helio G85</td>\n",

" <td>Octa Core</td>\n",

" <td>5020 mAh</td>\n",

" <td>Lithium Polyme</td>\n",

" <td>â‚¹12,272</td>\n",

" <td>https://www.flipkart.com/redmi-note-9-pebble-g...</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Brand Smartphone Colour RAM Storage(ROM) \\\n",

"0 Google Pixel 3a Clearly White 4 GB 64 GB \n",

"1 Google Pixel 3a Just Black 4 GB 64 GB \n",

"2 LG G8X Black 6 GB 128 GB \n",

"3 Samsung Galaxy M21 Raven Black 4 GB 64 GB \n",

"4 Redmi Redmi Note 9 Pro Interstellar Black 4 GB 128 GB \n",

"5 Redmi Redmi 9 Prime Space Blue 4 GB 128 GB \n",

"6 Redmi Redmi 9 Sky Blue 4 GB 64 GB \n",

"7 Redmi Redmi 9 Carbon Black 4 GB 64 GB \n",

"8 Karbonn KX3 Black,Red 32 MB 32 MB \n",

"9 Itel VISION1 GRADATION GREEN 2 GB 32 GB \n",

"10 I k6300 NEW Green 64 MB 64 MB \n",

"11 Redmi Redmi 9 Prime MatteBlack 4 GB 128 GB \n",

"12 Redmi Redmi 9 Prime MintGreen 4 GB 128 GB \n",

"13 Redmi Redmi 9 Prime Space Blue 4 GB 64 GB \n",

"14 Samsung M01 core Black 1 GB 16 GB \n",

"15 Samsung Galaxy M21 Midnight Blue 4 GB 64 GB \n",

"16 Redmi Redmi Note 9 Aqua Green 4 GB 128 GB \n",

"17 Samsung M01 core Blue 1 GB 16 GB \n",

"18 Samsung Galaxy M31 Ocean Blue 6 GB 64 GB \n",

"19 Redmi Redmi Note 9 Pro Aurora Blue 4 GB 128 GB \n",

"20 Redmi Redmi Note 9 Pro Glacier White 4 GB 64 GB \n",

"21 Redmi Redmi Note 9 Pro Champagne Gold 4 GB 128 GB \n",

"22 Redmi Redmi Note 9 Pro Glacier White 4 GB 128 GB \n",

"23 Redmi Redmi Note 9 Pebble Grey 4 GB 64 GB \n",

"\n",

" Primary Camera Secondary Camera Display Size \\\n",

"0 12.2MP Rear Camera 8MP Front Camera 14.22 cm (5.6 inch) \n",

"1 12.2MP Rear Camera 8MP Front Camera 14.22 cm (5.6 inch) \n",

"2 12MP + 13MP - 16.26 cm (6.4 inch) \n",

"3 48MP + 8MP + 5MP - 16.26 cm (6.4 inch) \n",

"4 48MP + 8MP - 16.94 cm (6.67 inch) \n",

"5 13MP Rear Camera - 16.59 cm (6.53 inch) \n",

"6 13MP + 8MP 8MP Front Camera 16.59 cm (6.53 inch) \n",

"7 13MP + 2MP 5MP Front Camera 16.59 cm (6.53 inch) \n",

"8 0.3MP Rear Camera - 4.57 cm (1.8 inch) \n",

"9 8MP Rear Camera - 15.46 cm (6.088 inch) \n",

"10 Primary Camera - 7.11 cm (2.8 inch) \n",

"11 13MP Rear Camera - 16.59 cm (6.53 inch) \n",

"12 13MP Rear Camera - 16.59 cm (6.53 inch) \n",

"13 13MP Rear Camera - 16.59 cm (6.53 inch) \n",

"14 8MP Rear Camera - 13.46 cm (5.3 inch) \n",

"15 48MP + 8MP + 5MP - 16.26 cm (6.4 inch) \n",

"16 48MP Rear Camera - 16.59 cm (6.53 inch) \n",

"17 8MP Rear Camera - 13.46 cm (5.3 inch) \n",

"18 64MP + 8MP + 5MP + 5MP 32MP Front Camera 16.26 cm (6.4 inch) \n",

"19 48MP + 8MP - 16.94 cm (6.67 inch) \n",

"20 48MP + 8MP + 5MP + 2MP 16MP Front Camera 16.94 cm (6.67 inch) \n",

"21 48MP + 8MP + 5MP + 2MP 16MP Front Camera 16.94 cm (6.67 inch) \n",

"22 48MP + 8MP + 5MP + 2MP 16MP Front Camera 16.94 cm (6.67 inch) \n",

"23 48MP Rear Camera - 16.59 cm (6.53 inch) \n",

"\n",

" Display Resolution Processor \\\n",

"0 2220 x 1080 pixels Qualcomm Snapdragon 670 \n",

"1 2220 x 1080 pixels Qualcomm Snapdragon 670 \n",

"2 2340 x 1080 pixel Qualcomm Snapdragonâ„¢ 855 Octa-core (up to 2.84... \n",

"3 2340 x 1080$$Pixel Exynos 9611-Octa Core processor \n",

"4 2400 x 1080$$pixel Qualcomm Snapdragon 720G \n",

"5 2340 x 1080$$pixel - \n",

"6 720 x 1600$$pixel MediaTek Helio G35 \n",

"7 720 x 1600$$pixel MediaTek Helio G35 \n",

"8 128 x 160$$pixel - \n",

"9 1560 x 720$$pixel UNISOC SC9863A OCTA CORE \n",

"10 240 x 320$$pixel - \n",

"11 2340 x 1080$$pixel - \n",

"12 2340 x 1080$$pixel - \n",

"13 2340 x 1080$$pixel - \n",

"14 1480 x 720$$pixel MediaTek | MT6739WW quad core \n",

"15 2340 x 1080$$Pixel Exynos 9611-Octa Core processor \n",

"16 2340 x 1080$$pixel MediaTek Helio G85 \n",

"17 1480 x 720$$pixel MediaTek | MT6739WW quad core \n",

"18 2340 x 1080$$pixel Samsung Exynos 9 Octa 9611 \n",

"19 2400 x 1080$$pixel Qualcomm Snapdragon 720G \n",

"20 2400 x 1080 pixel QualcommÂ® Snapdragonâ„¢ 720G \n",

"21 2400 x 1080$$pixel QualcommÂ® Snapdragonâ„¢ 720G \n",

"22 2400 x 1080$$pixel QualcommÂ® Snapdragonâ„¢ 720G \n",

"23 2340 x 1080$$pixel MediaTek Helio G85 \n",

"\n",

" Processor Cores Battery Capacity Battery Type Price \\\n",

"0 Octa Core 3000 mAh - â‚¹30,999 \n",

"1 Octa Core 3000 mAh - â‚¹30,999 \n",

"2 Octa Core 4000 mAh lithium-ion â‚¹39,990 \n",

"3 Octa Core 6000 mAh lithium-ion â‚¹15,982 \n",

"4 Octa Core 5020 mAh lithium-ion â‚¹15,199 \n",

"5 Octa Core 5020 mAh - â‚¹11,335 \n",

"6 - 5000 mAh Li-Po â‚¹9,672 \n",

"7 Octa Core 5000 mAh Li-Po â‚¹9,750 \n",

"8 - 800 mAh Li-ion â‚¹799 \n",

"9 Octa Core 4000 mAh LI POLY â‚¹6,499 \n",

"10 - - - â‚¹1,169 \n",

"11 Octa Core 5020 mAh - â‚¹11,576 \n",

"12 Octa Core 5020 mAh - â‚¹11,536 \n",

"13 Octa Core 5020 mAh - â‚¹10,998 \n",

"14 Octa Core 3000 mAh lithium-ion â‚¹5,059 \n",

"15 Octa Core 6000 mAh lithium-ion â‚¹15,986 \n",

"16 Octa Core 5020 mAh Lithium Polyme â‚¹14,399 \n",

"17 Octa Core 3000 mAh lithium-ion â‚¹5,022 \n",

"18 Octa Core 6000 mAh lithium-ion â‚¹17,474 \n",

"19 Octa Core 5020 mAh lithium-ion â‚¹15,448 \n",

"20 Octa Core 5020 mAh lithium-ion â‚¹14,958 \n",

"21 Octa Core 5020 mAh lithium-ion â‚¹14,990 \n",

"22 Octa Core 5020 mAh lithium-ion â‚¹14,910 \n",

"23 Octa Core 5020 mAh Lithium Polyme â‚¹12,272 \n",

"\n",

" URL \n",

"0 https://www.flipkart.com/google-pixel-3a-clear... \n",

"1 https://www.flipkart.com/google-pixel-3a-just-... \n",

"2 https://www.flipkart.com/lg-g8x-black-128-gb/p... \n",

"3 https://www.flipkart.com/samsung-galaxy-m21-ra... \n",

"4 https://www.flipkart.com/redmi-note-9-pro-inte... \n",

"5 https://www.flipkart.com/redmi-9-prime-space-b... \n",

"6 https://www.flipkart.com/redmi-9-sky-blue-64-g... \n",

"7 https://www.flipkart.com/redmi-9-carbon-black-... \n",

"8 https://www.flipkart.com/karbonn-kx3/p/itm60cf... \n",

"9 https://www.flipkart.com/itel-vision1-gradatio... \n",

"10 https://www.flipkart.com/kall-k6300-new/p/itm1... \n",

"11 https://www.flipkart.com/redmi-9-prime-mattebl... \n",

"12 https://www.flipkart.com/redmi-9-prime-mintgre... \n",

"13 https://www.flipkart.com/redmi-9-prime-space-b... \n",

"14 https://www.flipkart.com/samsung-m01-core-blac... \n",

"15 https://www.flipkart.com/samsung-galaxy-m21-mi... \n",

"16 https://www.flipkart.com/redmi-note-9-aqua-gre... \n",

"17 https://www.flipkart.com/samsung-m01-core-blue... \n",

"18 https://www.flipkart.com/samsung-galaxy-m31-oc... \n",

"19 https://www.flipkart.com/redmi-note-9-pro-auro... \n",

"20 https://www.flipkart.com/redmi-note-9-pro-glac... \n",

"21 https://www.flipkart.com/redmi-note-9-pro-cham... \n",

"22 https://www.flipkart.com/redmi-note-9-pro-glac... \n",

"23 https://www.flipkart.com/redmi-note-9-pebble-g... "

]

},

"execution\_count": 74,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"flip\_df = pd.DataFrame.from\_dict(flip\_dict)\n",

"flip\_df"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"URL Extracted: https://www.google.co.in/maps/place/New+Delhi,+Delhi/@28.5272803,77.0688983,11z/data=!3m1!4b1!4m5!3m4!1s0x390cfd5b347eb62d:0x52c2b7494e204dce!8m2!3d28.6139391!4d77.2090212\n",

"Latitude = 28.5272803, Longitude = 77.0688983\n"

]

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(2)\n",

"\n",

"# opening google maps\n",

"url = \"https://www.google.co.in/maps\"\n",

"driver.get(url)\n",

"time.sleep(2)\n",

"\n",

"\n",

"search = driver.find\_element\_by\_id(\"searchboxinput\") # locating search bar\n",

"search.clear() # clearing search bar\n",

"time.sleep(2)\n",

"search.send\_keys(\"New Delhi\") # entering values in search bar\n",

"button = driver.find\_element\_by\_id(\"searchbox-searchbutton\") # locating search button\n",

"button.click() # clicking search button\n",

"time.sleep(3)\n",

"\n",

"try:\n",

" url\_string = driver.current\_url\n",

" print(\"URL Extracted: \", url\_string)\n",

" lat\_lng = re.findall(r'@(.\*)data',url\_string)\n",

" if len(lat\_lng):\n",

" lat\_lng\_list = lat\_lng[0].split(\",\")\n",

" if len(lat\_lng\_list)>=2:\n",

" lat = lat\_lng\_list[0]\n",

" lng = lat\_lng\_list[1]\n",

" print(\"Latitude = {}, Longitude = {}\".format(lat, lng))\n",

"\n",

"except Exception as e:\n",

" print(\"Error: \", str(e))"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"- Q6.\tWrite a program to scrap details of all the funding deals for second quarter (i.e. Jan 21 â€“ March 21) from trak.in."

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"<ipython-input-23-fb8f63c1dfb2>:2: DeprecationWarning: executable\_path has been deprecated, please pass in a Service object\n",

" driver=webdriver.Chrome(\"chromedriver.exe\")\n"

]

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(2)\n",

"\n",

"# opening trak.in\n",

"url = \"https://trak.in/\"\n",

"driver.get(url)\n",

"time.sleep(2)\n",

"\n",

"# do click on funding deals\n",

"button = driver.find\_element\_by\_xpath('//li[@id=\"menu-item-51510\"]/a').get\_attribute('href')\n",

"driver.get(button)"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Startup Name</th>\n",

" <th>Industry/Vertical</th>\n",

" <th>Sub-Vertical</th>\n",

" <th>Location</th>\n",

" <th>Investor</th>\n",

" <th>Investment Type</th>\n",

" <th>Amount(in USD)</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>15/01/2021</td>\n",

" <td>Digit Insurance</td>\n",

" <td>Financial Services</td>\n",

" <td>Insurance Services</td>\n",

" <td>Bengaluru</td>\n",

" <td>A91 Partners, Faering Capital, TVS Capital Funds</td>\n",

" <td>Venture</td>\n",

" <td>1,80,00,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>28/01/2021</td>\n",

" <td>Bombay Shaving Company</td>\n",

" <td>Consumer Goods Company</td>\n",

" <td>Shave care, beard care, and skincare products</td>\n",

" <td>New Delhi</td>\n",

" <td>Reckitt Benckiser</td>\n",

" <td>Venture</td>\n",

" <td>6,172,258.50</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>19/01/2021</td>\n",

" <td>DeHaat</td>\n",

" <td>AgriTech Startup</td>\n",

" <td>online marketplace for farm products and services</td>\n",

" <td>Patna</td>\n",

" <td>Prosus Ventures</td>\n",

" <td>Series C</td>\n",

" <td>30,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>19/01/2021</td>\n",

" <td>Darwinbox</td>\n",

" <td>SaaS</td>\n",

" <td>HR Tech</td>\n",

" <td>Mumbai</td>\n",

" <td>Salesforce Ventures</td>\n",

" <td>Seed</td>\n",

" <td>15,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>18/01/2021</td>\n",

" <td>mfine</td>\n",

" <td>Health Tech Startup</td>\n",

" <td>AI-powered telemedicine mobile app</td>\n",

" <td>Bengaluru</td>\n",

" <td>Heritas Capital Management</td>\n",

" <td>Venture Round</td>\n",

" <td>16,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>5</th>\n",

" <td>18/01/2021</td>\n",

" <td>Udayy</td>\n",

" <td>EdTech</td>\n",

" <td>Online learning platform for kids in class 1-5</td>\n",

" <td>Gurgaon</td>\n",

" <td>Sequoia Capital</td>\n",

" <td>Seed Funding</td>\n",

" <td>15,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6</th>\n",

" <td>11/01/2021</td>\n",

" <td>True Elements</td>\n",

" <td>Food Startup</td>\n",

" <td>Whole Food plant based Nashta</td>\n",

" <td>Pune</td>\n",

" <td>SIDBI Venture Capital</td>\n",

" <td>Series</td>\n",

" <td>100,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>7</th>\n",

" <td>13/01/2021</td>\n",

" <td>Saveo</td>\n",

" <td>B2B E-commerce</td>\n",

" <td>Pharmacies</td>\n",

" <td>Bengaluru</td>\n",

" <td>Matrix Partners India, RTP Global, others</td>\n",

" <td>Seed</td>\n",

" <td>4,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8</th>\n",

" <td>11/02/2021</td>\n",

" <td>Doubtnut</td>\n",

" <td>Edu Tech</td>\n",

" <td>E-Learning Platform</td>\n",

" <td>Gurgaon</td>\n",

" <td>SIG Global, Sequoia Capital, WaterBridge Ventu...</td>\n",

" <td>Series B</td>\n",

" <td>2,500,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9</th>\n",

" <td>22/02/2021</td>\n",

" <td>Zomato</td>\n",

" <td>Hospitality</td>\n",

" <td>Online Food Delivery Platform</td>\n",

" <td>Gurgaon</td>\n",

" <td>Tiger Global, Kora</td>\n",

" <td>Venture</td>\n",

" <td>250,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>10</th>\n",

" <td>19/02/2021</td>\n",

" <td>Fingerlix</td>\n",

" <td>Hospitality</td>\n",

" <td>Semi-cooked food delivery app</td>\n",

" <td>Mumbai</td>\n",

" <td>Rhodium Trust, Accel Partners and Swiggy</td>\n",

" <td>Series C</td>\n",

" <td>2,747,045.20</td>\n",

" </tr>\n",

" <tr>\n",

" <th>11</th>\n",

" <td>17/02/2021</td>\n",

" <td>Zolve</td>\n",

" <td>FinTech</td>\n",

" <td>Global Neobank Venture</td>\n",

" <td>Mumbai</td>\n",

" <td>Accel Partners and Lightspeed Venture Partners</td>\n",

" <td>Seed</td>\n",

" <td>1,50,00,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>12</th>\n",

" <td>15/02/2021</td>\n",

" <td>KreditBee</td>\n",

" <td>Finance</td>\n",

" <td>Digital lending platform</td>\n",

" <td>Bengaluru</td>\n",

" <td>Azim Premjiâ€™s PremjiInvest and South Koreaâ€™s M...</td>\n",

" <td>Series C</td>\n",

" <td>75,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>13</th>\n",

" <td>12/02/2021</td>\n",

" <td>Pepperfry</td>\n",

" <td>E-commerce</td>\n",

" <td>Multi-brand furniture brand</td>\n",

" <td>Mumbai</td>\n",

" <td>InnoVen Capital</td>\n",

" <td>Debt Financing</td>\n",

" <td>4,773,958</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14</th>\n",

" <td>12/02/2021</td>\n",

" <td>Grofers</td>\n",

" <td>E-Commerce</td>\n",

" <td>Online supermarket</td>\n",

" <td>Gurgaon</td>\n",

" <td>SoftBank Vision Fund (SVF)</td>\n",

" <td>Unspecified</td>\n",

" <td>55,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>15</th>\n",

" <td>09/02/2021</td>\n",

" <td>Nothing</td>\n",

" <td>Technology</td>\n",

" <td>Consumer Technology Venture</td>\n",

" <td>London</td>\n",

" <td>GV</td>\n",

" <td>Series A</td>\n",

" <td>15,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>16</th>\n",

" <td>09/02/2021</td>\n",

" <td>SplashLearn</td>\n",

" <td>EdTech</td>\n",

" <td>Game-based learning programme</td>\n",

" <td>Gurgaon</td>\n",

" <td>Owl Ventures</td>\n",

" <td>Series C</td>\n",

" <td>18,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>17</th>\n",

" <td>04/03/2021</td>\n",

" <td>DealShare</td>\n",

" <td>E-commerce</td>\n",

" <td>Online shopping platform</td>\n",

" <td>Jaipur, Rajasthan</td>\n",

" <td>Innoven Capital</td>\n",

" <td>Debt Financing</td>\n",

" <td>250,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>31/03/2021</td>\n",

" <td>Uniphore</td>\n",

" <td>Technology</td>\n",

" <td>Conversational Service Automation (CSA)</td>\n",

" <td>Palo Alto</td>\n",

" <td>Sorenson Capital Partners</td>\n",

" <td>Series D</td>\n",

" <td>140,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>30/03/2021</td>\n",

" <td>Dunzo</td>\n",

" <td>E-commerce</td>\n",

" <td>Hyper-local delivery app</td>\n",

" <td>Bengaluru</td>\n",

" <td>Krishtal Advisors Pte Ltd</td>\n",

" <td>Series E</td>\n",

" <td>8,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>30/03/2021</td>\n",

" <td>BYJUâ€™S</td>\n",

" <td>Edu-tech</td>\n",

" <td>Online tutoring</td>\n",

" <td>Bengaluru</td>\n",

" <td>MC Global Edtech, B Capital, Baron, others</td>\n",

" <td>Series F</td>\n",

" <td>460,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>23/03/2021</td>\n",

" <td>SkilloVilla</td>\n",

" <td>Edu-tech</td>\n",

" <td>Career and job-oriented upskilling.</td>\n",

" <td>Bengaluru</td>\n",

" <td>Titan Capital, others</td>\n",

" <td>Seed</td>\n",

" <td>300,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>25/03/2021</td>\n",

" <td>CityMall</td>\n",

" <td>E-commerce</td>\n",

" <td>Social ecommerce and online grocery platform</td>\n",

" <td>Gurgaon</td>\n",

" <td>Accel Partners</td>\n",

" <td>Series A</td>\n",

" <td>11,000,000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>26/03/2021</td>\n",

" <td>DotPe</td>\n",

" <td>FinTech</td>\n",

" <td>Commerce and payments platform to offline ente...</td>\n",

" <td>Gurgaon</td>\n",

" <td>PayU</td>\n",

" <td>Series A</td>\n",

" <td>27,500,000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Date Startup Name Industry/Vertical \\\n",

"0 15/01/2021 Digit Insurance Financial Services \n",

"1 28/01/2021 Bombay Shaving Company Consumer Goods Company \n",

"2 19/01/2021 DeHaat AgriTech Startup \n",

"3 19/01/2021 Darwinbox SaaS \n",

"4 18/01/2021 mfine Health Tech Startup \n",

"5 18/01/2021 Udayy EdTech \n",

"6 11/01/2021 True Elements Food Startup \n",

"7 13/01/2021 Saveo B2B E-commerce \n",

"8 11/02/2021 Doubtnut Edu Tech \n",

"9 22/02/2021 Zomato Hospitality \n",

"10 19/02/2021 Fingerlix Hospitality \n",

"11 17/02/2021 Zolve FinTech \n",

"12 15/02/2021 KreditBee Finance \n",

"13 12/02/2021 Pepperfry E-commerce \n",

"14 12/02/2021 Grofers E-Commerce \n",

"15 09/02/2021 Nothing Technology \n",

"16 09/02/2021 SplashLearn EdTech \n",

"17 04/03/2021 DealShare E-commerce \n",

"18 31/03/2021 Uniphore Technology \n",

"19 30/03/2021 Dunzo E-commerce \n",

"20 30/03/2021 BYJUâ€™S Edu-tech \n",

"21 23/03/2021 SkilloVilla Edu-tech \n",

"22 25/03/2021 CityMall E-commerce \n",

"23 26/03/2021 DotPe FinTech \n",

"\n",

" Sub-Vertical Location \\\n",

"0 Insurance Services Bengaluru \n",

"1 Shave care, beard care, and skincare products New Delhi \n",

"2 online marketplace for farm products and services Patna \n",

"3 HR Tech Mumbai \n",

"4 AI-powered telemedicine mobile app Bengaluru \n",

"5 Online learning platform for kids in class 1-5 Gurgaon \n",

"6 Whole Food plant based Nashta Pune \n",

"7 Pharmacies Bengaluru \n",

"8 E-Learning Platform Gurgaon \n",

"9 Online Food Delivery Platform Gurgaon \n",

"10 Semi-cooked food delivery app Mumbai \n",

"11 Global Neobank Venture Mumbai \n",

"12 Digital lending platform Bengaluru \n",

"13 Multi-brand furniture brand Mumbai \n",

"14 Online supermarket Gurgaon \n",

"15 Consumer Technology Venture London \n",

"16 Game-based learning programme Gurgaon \n",

"17 Online shopping platform Jaipur, Rajasthan \n",

"18 Conversational Service Automation (CSA) Palo Alto \n",

"19 Hyper-local delivery app Bengaluru \n",

"20 Online tutoring Bengaluru \n",

"21 Career and job-oriented upskilling. Bengaluru \n",

"22 Social ecommerce and online grocery platform Gurgaon \n",

"23 Commerce and payments platform to offline ente... Gurgaon \n",

"\n",

" Investor Investment Type \\\n",

"0 A91 Partners, Faering Capital, TVS Capital Funds Venture \n",

"1 Reckitt Benckiser Venture \n",

"2 Prosus Ventures Series C \n",

"3 Salesforce Ventures Seed \n",

"4 Heritas Capital Management Venture Round \n",

"5 Sequoia Capital Seed Funding \n",

"6 SIDBI Venture Capital Series \n",

"7 Matrix Partners India, RTP Global, others Seed \n",

"8 SIG Global, Sequoia Capital, WaterBridge Ventu... Series B \n",

"9 Tiger Global, Kora Venture \n",

"10 Rhodium Trust, Accel Partners and Swiggy Series C \n",

"11 Accel Partners and Lightspeed Venture Partners Seed \n",

"12 Azim Premjiâ€™s PremjiInvest and South Koreaâ€™s M... Series C \n",

"13 InnoVen Capital Debt Financing \n",

"14 SoftBank Vision Fund (SVF) Unspecified \n",

"15 GV Series A \n",

"16 Owl Ventures Series C \n",

"17 Innoven Capital Debt Financing \n",

"18 Sorenson Capital Partners Series D \n",

"19 Krishtal Advisors Pte Ltd Series E \n",

"20 MC Global Edtech, B Capital, Baron, others Series F \n",

"21 Titan Capital, others Seed \n",

"22 Accel Partners Series A \n",

"23 PayU Series A \n",

"\n",

" Amount(in USD) \n",

"0 1,80,00,000 \n",

"1 6,172,258.50 \n",

"2 30,000,000 \n",

"3 15,000,000 \n",

"4 16,000,000 \n",

"5 15,000,000 \n",

"6 100,000,000 \n",

"7 4,000,000 \n",

"8 2,500,000 \n",

"9 250,000,000 \n",

"10 2,747,045.20 \n",

"11 1,50,00,000 \n",

"12 75,000,000 \n",

"13 4,773,958 \n",

"14 55,000,000 \n",

"15 15,000,000 \n",

"16 18,000,000 \n",

"17 250,000,000 \n",

"18 140,000,000 \n",

"19 8,000,000 \n",

"20 460,000,000 \n",

"21 300,000,000 \n",

"22 11,000,000 \n",

"23 27,500,000 "

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Empty lists\n",

"fund\_dict = {}\n",

"fund\_dict['Date'] = []\n",

"fund\_dict['Startup Name'] = []\n",

"fund\_dict['Industry/Vertical'] = []\n",

"fund\_dict['Sub-Vertical'] = []\n",

"fund\_dict['Location'] = []\n",

"fund\_dict['Investor'] = []\n",

"fund\_dict['Investment Type'] = []\n",

"fund\_dict['Amount(in USD)'] = []\n",

"\n",

"\n",

"for i in range(54,57):\n",

" # Date\n",

" dt = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[2]'.format(i))\n",

" for d in dt:\n",

" fund\_dict['Date'].append(d.text)\n",

"\n",

" # Startup Name\n",

" sn = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[3]'.format(i))\n",

" for n in sn:\n",

" fund\_dict['Startup Name'].append(n.text)\n",

" \n",

" # Industry/Vertical\n",

" ind = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[4]'.format(i))\n",

" for n in ind:\n",

" fund\_dict['Industry/Vertical'].append(n.text)\n",

" \n",

" # Sub-Vertical\n",

" sv = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[5]'.format(i))\n",

" for s in sv:\n",

" fund\_dict['Sub-Vertical'].append(s.text)\n",

"\n",

" # Location\n",

" loc = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[6]'.format(i))\n",

" for l in loc:\n",

" fund\_dict['Location'].append(l.text)\n",

" \n",

" # Investor\n",

" inv = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[7]'.format(i))\n",

" for n in inv:\n",

" fund\_dict['Investor'].append(n.text)\n",

" \n",

" # Investment Type\n",

" invt = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[8]'.format(i))\n",

" for n in invt:\n",

" fund\_dict['Investment Type'].append(n.text)\n",

" \n",

" # Amount\n",

" amt = driver.find\_elements\_by\_xpath('//table[@id=\"tablepress-{}\"]/tbody/tr/td[9]'.format(i))\n",

" for a in amt:\n",

" fund\_dict['Amount(in USD)'].append(a.text)\n",

" \n",

"fund\_df = pd.DataFrame(fund\_dict)\n",

"fund\_df"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q7. Write a program to scrap all the available details of top 10 gaming laptops from digit.in."

]

},

{

"cell\_type": "code",

"execution\_count": 71,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Name</th>\n",

" <th>price</th>\n",

" <th>OS</th>\n",

" <th>Display</th>\n",

" <th>HDD</th>\n",

" <th>RAM</th>\n",

" <th>processor</th>\n",

" <th>weight</th>\n",

" <th>Dimension</th>\n",

" <th>Graphical processor</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>ALIENWARE AREA 51M R2</td>\n",

" <td>N/A</td>\n",

" <td>WINDOWS 10 HOME</td>\n",

" <td>17.3\" (1920 X 1080)</td>\n",

" <td>1 TB SSD</td>\n",

" <td>16 GBGB DDR4</td>\n",

" <td>10TH GENERATION INTELÂ® COREâ„¢ I7-10700 | 2.90 GHZ</td>\n",

" <td>4.1</td>\n",

" <td>27.65 x 402.6 x 319.14</td>\n",

" <td>IntelÂ® UHD Graphics 630</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>ALIENWARE M15 R3</td>\n",

" <td>â‚¹341990</td>\n",

" <td>WINDOWS 10 HOME</td>\n",

" <td>15.6\" (3840 X 2160)</td>\n",

" <td>1 TB SSD</td>\n",

" <td>16 GBGB DDR4</td>\n",

" <td>10TH GENERATION INTELÂ® COREâ„¢ I9-10980HK | NA</td>\n",

" <td>NA</td>\n",

" <td>NA</td>\n",

" <td>NA</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>ASUS ROG STRIX SCAR 15</td>\n",

" <td>N/A</td>\n",

" <td>WINDOWS 10 HOME</td>\n",

" <td>15.6\" (1920 X 1080)</td>\n",

" <td>1 TB SSD</td>\n",

" <td>16 GBGB DDR4</td>\n",

" <td>AMD RYZENâ„¢ 9 5900HX | 3.3 GHZ</td>\n",

" <td>2.30</td>\n",

" <td>35.4 x 25.9 x 2.26</td>\n",

" <td>NVIDIAÂ® GeForce RTXâ„¢ 3070</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>ASUS ROG ZEPHYRUS G14</td>\n",

" <td>â‚¹164990</td>\n",

" <td>WINDOWS 10 HOME</td>\n",

" <td>14\" (1920 X 1080)</td>\n",

" <td>1 TB SSD</td>\n",

" <td>16 GBGB DDR4</td>\n",

" <td>AMD 3RD GENERATION RYZEN 9 | 3.3 GHZ</td>\n",

" <td>1.65</td>\n",

" <td>32.5 x 22.1 x 1.8</td>\n",

" <td>NVIDIA GeForce RTX 2060</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>LENOVO LEGION 5I</td>\n",

" <td>â‚¹71990</td>\n",

" <td>WINDOWS 10 PRO</td>\n",

" <td>15.6\" (1920 X 1080)</td>\n",

" <td>1 TB SSD</td>\n",

" <td>16 GBGB DDR4</td>\n",

" <td>10TH GENERATION INTELÂ® COREâ„¢ I5-10300H | 2.50 GHZ</td>\n",

" <td>2.3</td>\n",

" <td>363.06 x 259.61 x 23.57</td>\n",

" <td>NVIDIAÂ® GeForceÂ® GTX 1650 4GB</td>\n",

" </tr>\n",

" <tr>\n",

" <th>5</th>\n",

" <td>ASUS ROG ZEPHYRUS DUO 15</td>\n",

" <td>â‚¹199990</td>\n",

" <td>WINDOWS 10</td>\n",

" <td>15.6\" (3840 X 1100)</td>\n",

" <td>512 GB SSD</td>\n",

" <td>4 GBGB DDR4</td>\n",

" <td>INTEL CORE I7 10TH GEN 10875H | NA</td>\n",

" <td>2.4</td>\n",

" <td>268.30 x 360.00 x 20.90</td>\n",

" <td>NVIDIA GeForce RTX 2070 Max-Q</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6</th>\n",

" <td>ACER ASPIRE 7 GAMING</td>\n",

" <td>â‚¹56990</td>\n",

" <td>WINDOWS 10 HOME</td>\n",

" <td>15.6\" (1920 X 1080)</td>\n",

" <td>512 GB SSD</td>\n",

" <td>8 GBGB DDR4</td>\n",

" <td>AMD RYZENâ„¢ 5-5500U HEXA-CORE | NA</td>\n",

" <td>2.15</td>\n",

" <td>2.29 x 36.3 x 25.4</td>\n",

" <td>NVIDIAÂ® GeForceÂ® GTX 1650</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Name price OS Display \\\n",

"0 ALIENWARE AREA 51M R2 N/A WINDOWS 10 HOME 17.3\" (1920 X 1080) \n",

"1 ALIENWARE M15 R3 â‚¹341990 WINDOWS 10 HOME 15.6\" (3840 X 2160) \n",

"2 ASUS ROG STRIX SCAR 15 N/A WINDOWS 10 HOME 15.6\" (1920 X 1080) \n",

"3 ASUS ROG ZEPHYRUS G14 â‚¹164990 WINDOWS 10 HOME 14\" (1920 X 1080) \n",

"4 LENOVO LEGION 5I â‚¹71990 WINDOWS 10 PRO 15.6\" (1920 X 1080) \n",

"5 ASUS ROG ZEPHYRUS DUO 15 â‚¹199990 WINDOWS 10 15.6\" (3840 X 1100) \n",

"6 ACER ASPIRE 7 GAMING â‚¹56990 WINDOWS 10 HOME 15.6\" (1920 X 1080) \n",

"\n",

" HDD RAM \\\n",

"0 1 TB SSD 16 GBGB DDR4 \n",

"1 1 TB SSD 16 GBGB DDR4 \n",

"2 1 TB SSD 16 GBGB DDR4 \n",

"3 1 TB SSD 16 GBGB DDR4 \n",

"4 1 TB SSD 16 GBGB DDR4 \n",

"5 512 GB SSD 4 GBGB DDR4 \n",

"6 512 GB SSD 8 GBGB DDR4 \n",

"\n",

" processor weight \\\n",

"0 10TH GENERATION INTELÂ® COREâ„¢ I7-10700 | 2.90 GHZ 4.1 \n",

"1 10TH GENERATION INTELÂ® COREâ„¢ I9-10980HK | NA NA \n",

"2 AMD RYZENâ„¢ 9 5900HX | 3.3 GHZ 2.30 \n",

"3 AMD 3RD GENERATION RYZEN 9 | 3.3 GHZ 1.65 \n",

"4 10TH GENERATION INTELÂ® COREâ„¢ I5-10300H | 2.50 GHZ 2.3 \n",

"5 INTEL CORE I7 10TH GEN 10875H | NA 2.4 \n",

"6 AMD RYZENâ„¢ 5-5500U HEXA-CORE | NA 2.15 \n",

"\n",

" Dimension Graphical processor \n",

"0 27.65 x 402.6 x 319.14 IntelÂ® UHD Graphics 630 \n",

"1 NA NA \n",

"2 35.4 x 25.9 x 2.26 NVIDIAÂ® GeForce RTXâ„¢ 3070 \n",

"3 32.5 x 22.1 x 1.8 NVIDIA GeForce RTX 2060 \n",

"4 363.06 x 259.61 x 23.57 NVIDIAÂ® GeForceÂ® GTX 1650 4GB \n",

"5 268.30 x 360.00 x 20.90 NVIDIA GeForce RTX 2070 Max-Q \n",

"6 2.29 x 36.3 x 25.4 NVIDIAÂ® GeForceÂ® GTX 1650 "

]

},

"execution\_count": 71,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(2)\n",

"\n",

"# opening www.digit.in\n",

"url = \"https://www.digit.in/\"\n",

"driver.get(url)\n",

"time.sleep(2)\n",

"\n",

"#clicking on top 10 option \n",

"top\_10=driver.find\_element\_by\_xpath(\"/html/body/div[1]/div[2]/div[4]/ul/li[4]/a\")\n",

"top\_10.click()\n",

"\n",

"time.sleep(2)\n",

"#clicking on laptops option\n",

"laptops=driver.find\_element\_by\_xpath(\"/html/body/div[3]/div/div/div[2]/div[5]/div[1]/div/button[2]\")\n",

"laptops.click()\n",

"\n",

"time.sleep(2)\n",

"#best gaming laptops link\n",

"best\_gaming=driver.find\_element\_by\_xpath(\"//div[@id='laptops']//div[3]//a\")\n",

"driver.get(best\_gaming.get\_attribute('href'))\n",

"\n",

"#intialising lists\n",

"name = []\n",

"Price = []\n",

"OS = []\n",

"display = []\n",

"processor = []\n",

"HDD = []\n",

"RAM = []\n",

"weight = []\n",

"dimension = []\n",

"GPU = []\n",

"\n",

"time.sleep(1)\n",

"#scraping names\n",

"names=driver.find\_elements\_by\_xpath(\"//div[@class='right-container']/div/a/h3\")\n",

"for i in names:\n",

" name.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"#scraping operating system\n",

"os=driver.find\_elements\_by\_xpath(\"//div[@class='product-detail']/div/ul/li[1]/div/div\")\n",

"for i in os:\n",

" OS.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"#scraping display\n",

"displays=driver.find\_elements\_by\_xpath(\"//div[@class='product-detail']/div/ul/li[2]/div/div\")\n",

"for i in displays:\n",

" display.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"#scraping processor\n",

"processors=driver.find\_elements\_by\_xpath(\"//div[@class='product-detail']/div/ul/li[3]/div/div\")\n",

"for i in processors:\n",

" processor.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"# scraping memory\n",

"memories=driver.find\_elements\_by\_xpath(\"//div[@class='Spcs-details'][1]/table/tbody/tr[6]/td[3]\")# extrat HDD and RAM form xpath\n",

"for i in memories:\n",

" HDD.append(i.text.split(\"/\")[0])\n",

" RAM.append(i.text.split(\"/\")[1])\n",

"\n",

"time.sleep(1)\n",

"# scraping weight\n",

"weights=driver.find\_elements\_by\_xpath(\"//div[@class='Spcs-details'][1]/table/tbody/tr[7]/td[3]\")# extrat weight form xpath\n",

"for i in weights:\n",

" weight.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"# scraping dimension\n",

"dimension=[]\n",

"dimensions=driver.find\_elements\_by\_xpath(\"//div[@class='Spcs-details'][1]/table/tbody/tr[8]/td[3]\") \n",

"for i in dimensions:\n",

" dimension.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"# scraping graphical processor\n",

"GPUs=driver.find\_elements\_by\_xpath(\"//div[@class='Spcs-details'][1]/table/tbody/tr[9]/td[3]\") \n",

"for i in GPUs:\n",

" GPU.append(i.text)\n",

"\n",

"time.sleep(1)\n",

"\n",

"#scraping price\n",

"price=driver.find\_elements\_by\_xpath(\"//table[@id='summtable']//tr//td[3]\")\n",

"for i in price:\n",

" Price.append(i.text)\n",

"\n",

"#Make data frame\n",

"df=pd.DataFrame({\"Name\":name,\n",

" \"price\":Price,\n",

" \"OS\":OS,\n",

" \"Display\":display,\n",

" \"HDD\":HDD,\n",

" \"RAM\":RAM,\n",

" \"processor\":processor,\n",

" \"weight\":weight,\n",

" \"Dimension\":dimension,\n",

" \"Graphical processor\":GPU})\n",

"df"

]

},

{

"cell\_type": "code",

"execution\_count": 73,

"metadata": {},

"outputs": [],

"source": [

"df.to\_csv(\"Gaming laptops\_digit.csv\")"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q8- Write a python program to scrape the details for all billionaires from www.forbes.com. Details to be scrapped: â€œRankâ€, â€œNameâ€, â€œNet worthâ€, â€œAgeâ€, â€œCitizenshipâ€, â€œSourceâ€, â€œIndustryâ€."

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Rank</th>\n",

" <th>Name</th>\n",

" <th>Net Worth</th>\n",

" <th>Age</th>\n",

" <th>Country of Citizenship</th>\n",

" <th>Source</th>\n",

" <th>Industry</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1.</td>\n",

" <td>Jeff Bezos</td>\n",

" <td>$177 B</td>\n",

" <td>57</td>\n",

" <td>United States</td>\n",

" <td>Amazon</td>\n",

" <td>Technology</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2.</td>\n",

" <td>Elon Musk</td>\n",

" <td>$151 B</td>\n",

" <td>49</td>\n",

" <td>United States</td>\n",

" <td>Tesla, SpaceX</td>\n",

" <td>Automotive</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>3.</td>\n",

" <td>Bernard Arnault &amp; family</td>\n",

" <td>$150 B</td>\n",

" <td>72</td>\n",

" <td>France</td>\n",

" <td>LVMH</td>\n",

" <td>Fashion &amp; Retail</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>4.</td>\n",

" <td>Bill Gates</td>\n",

" <td>$124 B</td>\n",

" <td>65</td>\n",

" <td>United States</td>\n",

" <td>Microsoft</td>\n",

" <td>Technology</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>5.</td>\n",

" <td>Mark Zuckerberg</td>\n",

" <td>$97 B</td>\n",

" <td>36</td>\n",

" <td>United States</td>\n",

" <td>Facebook</td>\n",

" <td>Technology</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2750</th>\n",

" <td>2674.</td>\n",

" <td>Daniel Yong Zhang</td>\n",

" <td>$1 B</td>\n",

" <td>49</td>\n",

" <td>China</td>\n",

" <td>e-commerce</td>\n",

" <td>Technology</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2751</th>\n",

" <td>2674.</td>\n",

" <td>Zhang Yuqiang</td>\n",

" <td>$1 B</td>\n",

" <td>65</td>\n",

" <td>China</td>\n",

" <td>Fiberglass</td>\n",

" <td>Manufacturing</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2752</th>\n",

" <td>2674.</td>\n",

" <td>Zhao Meiguang</td>\n",

" <td>$1 B</td>\n",

" <td>58</td>\n",

" <td>China</td>\n",

" <td>gold mining</td>\n",

" <td>Metals &amp; Mining</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2753</th>\n",

" <td>2674.</td>\n",

" <td>Zhong Naixiong</td>\n",

" <td>$1 B</td>\n",

" <td>58</td>\n",

" <td>China</td>\n",

" <td>conglomerate</td>\n",

" <td>Diversified</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2754</th>\n",

" <td>2674.</td>\n",

" <td>Zhou Wei family</td>\n",

" <td>$1 B</td>\n",

" <td>54</td>\n",

" <td>China</td>\n",

" <td>Software</td>\n",

" <td>Technology</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>2755 rows Ã— 7 columns</p>\n",

"</div>"

],

"text/plain": [

" Rank Name Net Worth Age Country of Citizenship \\\n",

"0 1. Jeff Bezos $177 B 57 United States \n",

"1 2. Elon Musk $151 B 49 United States \n",

"2 3. Bernard Arnault & family $150 B 72 France \n",

"3 4. Bill Gates $124 B 65 United States \n",

"4 5. Mark Zuckerberg $97 B 36 United States \n",

"... ... ... ... .. ... \n",

"2750 2674. Daniel Yong Zhang $1 B 49 China \n",

"2751 2674. Zhang Yuqiang $1 B 65 China \n",

"2752 2674. Zhao Meiguang $1 B 58 China \n",

"2753 2674. Zhong Naixiong $1 B 58 China \n",

"2754 2674. Zhou Wei family $1 B 54 China \n",

"\n",

" Source Industry \n",

"0 Amazon Technology \n",

"1 Tesla, SpaceX Automotive \n",

"2 LVMH Fashion & Retail \n",

"3 Microsoft Technology \n",

"4 Facebook Technology \n",

"... ... ... \n",

"2750 e-commerce Technology \n",

"2751 Fiberglass Manufacturing \n",

"2752 gold mining Metals & Mining \n",

"2753 conglomerate Diversified \n",

"2754 Software Technology \n",

"\n",

"[2755 rows x 7 columns]"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(2)\n",

"\n",

"# Opening the forbes.com\n",

"url = \"https://www.forbes.com/?sh=69e6b8c92254\"\n",

"driver.get(url)\n",

"time.sleep(2)\n",

"\n",

"#clicking the explore button\n",

"button = driver.find\_element\_by\_xpath(\"//button[@class='icon--hamburger']\")\n",

"button.click()\n",

"time.sleep(1)\n",

"\n",

"#select billioners \n",

"billioners = driver.find\_element\_by\_xpath(\"/html/body/div[1]/header/nav/div[3]/ul/li[1]\")\n",

"billioners.click()\n",

"time.sleep(1)\n",

"\n",

"#select world billioners \n",

"world\_billioners= driver.find\_element\_by\_xpath(\"/html/body/div[1]/header/nav/div[3]/ul/li[1]/div[2]/ul/li[2]/a\")\n",

"world\_billioners.click()\n",

"time.sleep(1)\n",

"\n",

"\n",

"#make empty listes\n",

"Rank = [] \n",

"Person\_Name = [] \n",

"total\_net\_worth = [] \n",

"Age = []\n",

"country\_of\_citizenship = [] \n",

"Source = []\n",

"industry = []\n",

"\n",

"\n",

"while(True):\n",

" #scraping rank of billionaire\n",

" rank= driver.find\_elements\_by\_xpath(\"//div[@class='rank']\")\n",

" for i in rank:\n",

" Rank.append(i.text)\n",

" time.sleep(1)\n",

" \n",

" #scraping name of billionaire\n",

" name= driver.find\_elements\_by\_xpath(\"//div[@class='personName']//div\")\n",

" for i in name:\n",

" Person\_Name.append(i.text)\n",

" time.sleep(1)\n",

" \n",

" #scraping Age of billionaire\n",

" age= driver.find\_elements\_by\_xpath(\"//div[@class='age']//div\")\n",

" for i in age:\n",

" Age.append(i.text) \n",

" time.sleep(1)\n",

" \n",

" #scraping citizenship of billionaire \n",

" citizenship= driver.find\_elements\_by\_xpath(\"//div[@class='countryOfCitizenship']\")\n",

" for i in citizenship:\n",

" country\_of\_citizenship.append(i.text)\n",

" time.sleep(1)\n",

" \n",

" #scraping source of income\n",

" source= driver.find\_elements\_by\_xpath(\"//div[@class='source']\")\n",

" for i in source:\n",

" Source.append(i.text) \n",

" time.sleep(1)\n",

" \n",

" #scraping Age of billionaire\n",

" industries= driver.find\_elements\_by\_xpath(\"//div[@class='category']//div\")\n",

" for i in industries:\n",

" industry.append(i.text)\n",

" \n",

" #scraping net\_worth of billionaire\n",

" net\_worth= driver.find\_elements\_by\_xpath(\"//div[@class='netWorth']//div[1]\")\n",

" for i in net\_worth:\n",

" total\_net\_worth.append(i.text)\n",

" time.sleep(1) \n",

" \n",

" try:\n",

" next\_button = driver.find\_element\_by\_xpath(\"//button[@class='pagination-btn pagination-btn--next ']\")\n",

" next\_button.click()\n",

" except:\n",

" break \n",

"\n",

" \n",

"Net\_Worth = [] \n",

"for i in range(0,len(total\_net\_worth),2):\n",

" Net\_Worth.append(total\_net\_worth[i])\n",

" \n",

"\n",

"time.sleep(2) \n",

"#creating dataframe\n",

"df=pd.DataFrame({'Rank':Rank,\n",

" 'Name':Person\_Name,\n",

" 'Net Worth':Net\_Worth,\n",

" 'Age':Age,\n",

" 'Country of Citizenship':country\_of\_citizenship,\n",

" 'Source':Source,\n",

" 'Industry':industry})\n",

"df"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\* Q9. Write a program to extract at least 500 Comments, Comment upvote and time when comment was posted from any YouTube Video."

]

},

{

"cell\_type": "code",

"execution\_count": 86,

"metadata": {},

"outputs": [],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\")\n",

"time.sleep(2)\n",

"\n",

"# Opening the youtube.com\n",

"url = \"https://www.youtube.com/\"\n",

"driver.get(url)\n",

"time.sleep(2)\n",

"\n",

"#finding element for search bar\n",

"search\_bar = driver.find\_element\_by\_id('search')\n",

"search\_bar.send\_keys(\"GOT\") #entering Video name\n",

"time.sleep(1)\n",

"\n",

"#clicking on search button\n",

"search\_btn = driver.find\_element\_by\_id(\"search-icon-legacy\") \n",

"search\_btn.click()\n",

"time.sleep(1)\n",

"\n",

"#clicking on first video\n",

"link\_click = driver.find\_element\_by\_xpath(\"//yt-formatted-string[@class ='style-scope ytd-video-renderer']\")\n",

"link\_click.click()"

]

},

{

"cell\_type": "code",

"execution\_count": 88,

"metadata": {},

"outputs": [],

"source": [

"# 1000 time we scroll down by 10000 in order to generate more Comments\n",

"for \_ in range(1000):\n",

" driver.execute\_script(\"window.scrollBy(0,10000)\")"

]

},

{

"cell\_type": "code",

"execution\_count": 90,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Comments</th>\n",

" <th>Time</th>\n",

" <th>Likes</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Its actually so crazy to me that that 5 second...</td>\n",

" <td>10 months ago</td>\n",

" <td>3.8K</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>After watching S8 i just wish that hodor shoul...</td>\n",

" <td>5 months ago</td>\n",

" <td>567</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Itâ€™s been a year and a half, and am still wait...</td>\n",

" <td>9 months ago</td>\n",

" <td>606</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>After 10,000 years the Night King made it past...</td>\n",

" <td>9 months ago</td>\n",

" <td>314</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Who else just finished GOT's every seasons! (d...</td>\n",

" <td>10 months ago (edited)</td>\n",

" <td>1.8K</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>535</th>\n",

" <td>\"If u still think this has a happy ending, u h...</td>\n",

" <td>2 years ago</td>\n",

" <td>179</td>\n",

" </tr>\n",

" <tr>\n",

" <th>536</th>\n",

" <td>Who all are disappointed when daenarys ðŸ”¥Targar...</td>\n",

" <td>1 week ago</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>537</th>\n",

" <td>And the noobest writer award goes to GOT season 8</td>\n",

" <td>9 months ago</td>\n",

" <td></td>\n",

" </tr>\n",

" <tr>\n",

" <th>538</th>\n",

" <td>DC fans got their remake. What about us?</td>\n",

" <td>2 months ago</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>539</th>\n",

" <td>âž¡ï¸ hotslut.live/1330nlilovedollxxx ...</td>\n",

" <td>3 months ago</td>\n",

" <td>6</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>540 rows Ã— 3 columns</p>\n",

"</div>"

],

"text/plain": [

" Comments \\\n",

"0 Its actually so crazy to me that that 5 second... \n",

"1 After watching S8 i just wish that hodor shoul... \n",

"2 Itâ€™s been a year and a half, and am still wait... \n",

"3 After 10,000 years the Night King made it past... \n",

"4 Who else just finished GOT's every seasons! (d... \n",

".. ... \n",

"535 \"If u still think this has a happy ending, u h... \n",

"536 Who all are disappointed when daenarys ðŸ”¥Targar... \n",

"537 And the noobest writer award goes to GOT season 8 \n",

"538 DC fans got their remake. What about us? \n",

"539 âž¡ï¸ hotslut.live/1330nlilovedollxxx ... \n",

"\n",

" Time Likes \n",

"0 10 months ago 3.8K \n",

"1 5 months ago 567 \n",

"2 9 months ago 606 \n",

"3 9 months ago 314 \n",

"4 10 months ago (edited) 1.8K \n",

".. ... ... \n",

"535 2 years ago 179 \n",

"536 1 week ago 1 \n",

"537 9 months ago \n",

"538 2 months ago 1 \n",

"539 3 months ago 6 \n",

"\n",

"[540 rows x 3 columns]"

]

},

"execution\_count": 90,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"#make empty lists\n",

"comments = []\n",

"comment\_time = []\n",

"Time = []\n",

"Likes = []\n",

"No\_of\_Likes = []\n",

"\n",

"#scrape comments\n",

"cm = driver.find\_elements\_by\_id(\"content-text\")\n",

"for i in cm:\n",

" if i.text is None:\n",

" comments.append(\"--\")\n",

" else:\n",

" comments.append(i.text)\n",

"time.sleep(5)\n",

"\n",

"\n",

"# scrape time when comment was posted\n",

"tm = driver.find\_elements\_by\_xpath(\"//a[contains(text(),'ago')]\")\n",

"for i in tm:\n",

" Time.append(i.text)\n",

"\n",

"for i in range(0,len(Time),2):\n",

" comment\_time.append(Time[i])\n",

"time.sleep(5)\n",

" \n",

"# scrape the comment likes\n",

"like = driver.find\_elements\_by\_xpath(\"//span[@class='style-scope ytd-comment-action-buttons-renderer']\")\n",

"for i in like:\n",

" Likes.append(i.text)\n",

" \n",

"for i in range(1,len(Likes),2):\n",

" No\_of\_Likes.append(Likes[i])\n",

" \n",

" \n",

"time.sleep(2)\n",

"#creating dataframe\n",

"df=pd.DataFrame({'Comments':comments,\n",

" 'Time':comment\_time,\n",

" 'Likes':No\_of\_Likes})\n",

"df"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"Q10. Write a python program to scrape a data for all available Hostels from https://www.hostelworld.com/ in â€œLondonâ€ 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"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"metadata": {},

"outputs": [],

"source": [

"# Activating the chrome browser\n",

"driver=webdriver.Chrome(\"chromedriver.exe\") \n",

"time.sleep(3)\n",

"\n",

"# Opening the homepage of hostelworld.com\n",

"url = \"https://www.hostelworld.com/\"\n",

"driver.get(url)"

]

},

{

"cell\_type": "code",

"execution\_count": 76,

"metadata": {},

"outputs": [],

"source": [

"#locating the location search bar\n",

"search\_loc = driver.find\_element\_by\_id('search-input-field')\n",

"# write Lonodn in search bar\n",

"search\_loc.send\_keys(\"London\")\n",

"time.sleep(2)\n",

"\n",

"#select london\n",

"london = driver.find\_element\_by\_xpath('/html/body/div[1]/div/div/div[1]/div[1]/div/div[2]/div[4]/div/div[2]/div/div[1]/div/div/ul/li[2]/div')\n",

"london.click()\n",

"time.sleep(2)\n",

"\n",

"\n",

"# do click in search button\n",

"search\_btn = driver.find\_element\_by\_id('search-button')\n",

"search\_btn.click()\n"

]

},

{

"cell\_type": "code",

"execution\_count": 77,

"metadata": {},

"outputs": [],

"source": [

"# Make empty lists\n",

"Hostel\_Name = []\n",

"Distance = []\n",

"overall\_review = []\n",

"total\_reviews = []\n",

"facilities = []\n",

"price = []\n",

"Rating = []\n",

"property\_description = []"

]

},

{

"cell\_type": "code",

"execution\_count": 78,

"metadata": {},

"outputs": [],

"source": [

"while(True):\n",

" \n",

" # Hostel name\n",

" names = driver.find\_elements\_by\_xpath('//h2[@class=\"title title-6\"]')\n",

" for name in names:\n",

" Hostel\_Name.append(name.text)\n",

" time.sleep(2)\n",

" \n",

" # Distance from city centre\n",

" dis = driver.find\_elements\_by\_xpath('//span[@class=\"description\"]')\n",

" for d in dis:\n",

" Distance.append(d.text)\n",

" time.sleep(2)\n",

" \n",

" # Overall Review \n",

" review = driver.find\_elements\_by\_xpath('//div[@class=\"keyword\"]//span')\n",

" for r in review:\n",

" overall\_review.append(r.text)\n",

" time.sleep(2)\n",

" \n",

" # Total No of reviews \n",

" t\_review = driver.find\_elements\_by\_xpath('//div[@class=\"reviews\"]')\n",

" for tr in t\_review:\n",

" total\_reviews.append(tr.text)\n",

" time.sleep(2)\n",

" \n",

" # facilities\n",

" service = driver.find\_elements\_by\_xpath('//div[@class=\"facilities-label facilities\"]')\n",

" for s in service:\n",

" facilities.append(s.text)\n",

" time.sleep(2)\n",

" \n",

" # Prices \n",

" prices = driver.find\_elements\_by\_xpath('//div[@class=\"price-col\"]')\n",

" for p in prices:\n",

" price.append(p.text)\n",

" time.sleep(2) \n",

" \n",

" try:\n",

" next\_button = driver.find\_element\_by\_xpath('//div[@class=\"pagination-item pagination-next\"]')\n",

" next\_button.click()\n",

" except:\n",

" break\n",

" \n",

"time.sleep(4) \n",

" \n",

"# Separate Privates\_From price and Dorms\_From price\n",

"Privates\_From = []\n",

"for i in range(0,len(price),2):\n",

" Privates\_From.append(price[i])\n",

"time.sleep(2)\n",

"\n",

"Dorms\_From = []\n",

"for i in range(1,len(price),2):\n",

" Dorms\_From.append(price[i]) "

]

},

{

"cell\_type": "code",

"execution\_count": 80,

"metadata": {},

"outputs": [],

"source": [

"# scrape Hostels URL\n",

"hostel\_url = []\n",

"\n",

"while(True):\n",

" urls = driver.find\_elements\_by\_xpath('//h2[@class=\"title title-6\"]//a')\n",

" for url in urls:\n",

" hostel\_url.append(url.get\_attribute(\"href\"))\n",

" time.sleep(2) \n",

" \n",

" try:\n",

" next\_button = driver.find\_element\_by\_xpath('//div[@class=\"pagination-item pagination-next\"]')\n",

" next\_button.click()\n",

" except:\n",

" break\n",

"\n",

"\n",

" \n",

"Rate = []\n",

"for page in hostel\_url:\n",

" driver.get(page)\n",

" \n",

" # Rating\n",

" try:\n",

" ratings = driver.find\_element\_by\_xpath('/html/body/div[1]/div/div/div[1]/section/div[6]/div/div[1]/div[1]/div[1]')\n",

" Rate.append(ratings.text)\n",

" except NoSuchElementException:\n",

" Rate.append(\"No Rating\") \n",

" time.sleep(2)\n",

" \n",

" \n",

" # Property Description\n",

" try:\n",

" pd = driver.find\_element\_by\_xpath('/html/body/div[1]/div/div/div[1]/section/div[6]/div/div[2]/div[2]/div/div[2]')\n",

" property\_description.append(pd.text)\n",

" except NoSuchElementException:\n",

" property\_description.append(\"No Description\") \n",

"\n",

" \n",

"time.sleep(2) \n",

"# remove extra data from Rating \n",

"all\_text = []\n",

"for i in Rate:\n",

" all\_text.append(i.split())\n",

"time.sleep(2)\n",

"\n",

"for i in all\_text:\n",

" Rating.append(i[0]) "

]

},

{

"cell\_type": "code",

"execution\_count": 84,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Hostel Name</th>\n",

" <th>Distance from city centre</th>\n",

" <th>Overall Review</th>\n",

" <th>Total Reviews</th>\n",

" <th>Facilities</th>\n",

" <th>Privates From Price</th>\n",

" <th>Dorms From Price</th>\n",

" <th>Rating</th>\n",

" <th>Property Description</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>St Christopher's Village</td>\n",

" <td>Hostel - 1.8km from city centre</td>\n",

" <td>Fabulous</td>\n",

" <td>10814 Total Reviews</td>\n",

" <td>Free WiFi\\nFollows Covid-19 sanitation guidance</td>\n",

" <td>Privates From\\nRs3391</td>\n",

" <td>Dorms From\\nRs1360</td>\n",

" <td>8.9</td>\n",

" <td>COVID 19 Policy Update.\\nIn response to Corona...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Generator London</td>\n",

" <td>Hostel - 3km from city centre</td>\n",

" <td>Very Good</td>\n",

" <td>6729 Total Reviews</td>\n",

" <td>Free WiFi\\nFollows Covid-19 sanitation guidance</td>\n",

" <td>Privates From\\nRs8323</td>\n",

" <td>Dorms From\\nRs1987</td>\n",

" <td>7.5</td>\n",

" <td>Generator London is a design hotel-hostel loca...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Safestay London Kensington Holland Park</td>\n",

" <td>Hostel - 5.9km from city centre</td>\n",

" <td>Very Good</td>\n",

" <td>1070 Total Reviews</td>\n",

" <td>Free WiFi</td>\n",

" <td>No Privates Available</td>\n",

" <td>Dorms From\\nRs1017</td>\n",

" <td>7.9</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>PubLove @ The Crown, Battersea</td>\n",

" <td>Hostel - 4.7km from city centre</td>\n",

" <td>Very Good</td>\n",

" <td>204 Total Reviews</td>\n",

" <td>Free WiFi\\nFollows Covid-19 sanitation guidance</td>\n",

" <td>No Privates Available</td>\n",

" <td>Dorms From\\nRs1315</td>\n",

" <td>7.9</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>Leman House</td>\n",

" <td>Hostel - 3.6km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>13 Total Reviews</td>\n",

" <td>Free WiFi</td>\n",

" <td>Privates From\\nRs5857</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>83</th>\n",

" <td>The Dover</td>\n",

" <td>Hotel - 1.9km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>4 Total Reviews</td>\n",

" <td>Free WiFi\\nFree Breakfast</td>\n",

" <td>Privates From\\nRs7193</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>84</th>\n",

" <td>Park Hotel Essex</td>\n",

" <td>Hotel - 24.1km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>108 Total Reviews</td>\n",

" <td>Free Breakfast\\nFollows Covid-19 sanitation gu...</td>\n",

" <td>Privates From\\nRs3597</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>85</th>\n",

" <td>Cranbrook Hotel</td>\n",

" <td>Hotel - 14.8km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>58 Total Reviews</td>\n",

" <td>Free Breakfast\\nFollows Covid-19 sanitation gu...</td>\n",

" <td>Privates From\\nRs3597</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>86</th>\n",

" <td>St. Athans</td>\n",

" <td>Bed and Breakfast - 2.9km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>234 Total Reviews</td>\n",

" <td>Free WiFi\\nFollows Covid-19 sanitation guidance</td>\n",

" <td>Privates From\\nRs3877</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" <tr>\n",

" <th>87</th>\n",

" <td>Aron Guest House</td>\n",

" <td>Bed and Breakfast - 13.1km from city centre</td>\n",

" <td>No Rating</td>\n",

" <td>26 Total Reviews</td>\n",

" <td>Free WiFi</td>\n",

" <td>Privates From\\nRs6679</td>\n",

" <td>No Dorms Available</td>\n",

" <td>No</td>\n",

" <td>No Description</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>88 rows Ã— 9 columns</p>\n",

"</div>"

],

"text/plain": [

" Hostel Name \\\n",

"0 St Christopher's Village \n",

"1 Generator London \n",

"2 Safestay London Kensington Holland Park \n",

"3 PubLove @ The Crown, Battersea \n",

"4 Leman House \n",

".. ... \n",

"83 The Dover \n",

"84 Park Hotel Essex \n",

"85 Cranbrook Hotel \n",

"86 St. Athans \n",

"87 Aron Guest House \n",

"\n",

" Distance from city centre Overall Review \\\n",

"0 Hostel - 1.8km from city centre Fabulous \n",

"1 Hostel - 3km from city centre Very Good \n",

"2 Hostel - 5.9km from city centre Very Good \n",

"3 Hostel - 4.7km from city centre Very Good \n",

"4 Hostel - 3.6km from city centre No Rating \n",

".. ... ... \n",

"83 Hotel - 1.9km from city centre No Rating \n",

"84 Hotel - 24.1km from city centre No Rating \n",

"85 Hotel - 14.8km from city centre No Rating \n",

"86 Bed and Breakfast - 2.9km from city centre No Rating \n",

"87 Bed and Breakfast - 13.1km from city centre No Rating \n",

"\n",

" Total Reviews Facilities \\\n",

"0 10814 Total Reviews Free WiFi\\nFollows Covid-19 sanitation guidance \n",

"1 6729 Total Reviews Free WiFi\\nFollows Covid-19 sanitation guidance \n",

"2 1070 Total Reviews Free WiFi \n",

"3 204 Total Reviews Free WiFi\\nFollows Covid-19 sanitation guidance \n",

"4 13 Total Reviews Free WiFi \n",

".. ... ... \n",

"83 4 Total Reviews Free WiFi\\nFree Breakfast \n",

"84 108 Total Reviews Free Breakfast\\nFollows Covid-19 sanitation gu... \n",

"85 58 Total Reviews Free Breakfast\\nFollows Covid-19 sanitation gu... \n",

"86 234 Total Reviews Free WiFi\\nFollows Covid-19 sanitation guidance \n",

"87 26 Total Reviews Free WiFi \n",

"\n",

" Privates From Price Dorms From Price Rating \\\n",

"0 Privates From\\nRs3391 Dorms From\\nRs1360 8.9 \n",

"1 Privates From\\nRs8323 Dorms From\\nRs1987 7.5 \n",

"2 No Privates Available Dorms From\\nRs1017 7.9 \n",

"3 No Privates Available Dorms From\\nRs1315 7.9 \n",

"4 Privates From\\nRs5857 No Dorms Available No \n",

".. ... ... ... \n",

"83 Privates From\\nRs7193 No Dorms Available No \n",

"84 Privates From\\nRs3597 No Dorms Available No \n",

"85 Privates From\\nRs3597 No Dorms Available No \n",

"86 Privates From\\nRs3877 No Dorms Available No \n",

"87 Privates From\\nRs6679 No Dorms Available No \n",

"\n",

" Property Description \n",

"0 COVID 19 Policy Update.\\nIn response to Corona... \n",

"1 Generator London is a design hotel-hostel loca... \n",

"2 No Description \n",

"3 No Description \n",

"4 No Description \n",

".. ... \n",

"83 No Description \n",

"84 No Description \n",

"85 No Description \n",

"86 No Description \n",

"87 No Description \n",

"\n",

"[88 rows x 9 columns]"

]

},

"execution\_count": 84,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"#creating dataframe\n",

"df = pd.DataFrame({'Hostel Name':Hostel\_Name,\n",

" 'Distance from city centre':Distance,\n",

" 'Overall Review':overall\_review[3:],\n",

" 'Total Reviews':total\_reviews,\n",

" 'Facilities':facilities,\n",

" 'Privates From Price':Privates\_From,\n",

" 'Dorms From Price':Dorms\_From,\n",

" 'Rating':Rating,\n",

" 'Property Description':property\_description})\n",

"df"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.7"

}

},

"nbformat": 4,

"nbformat\_minor": 4

}