Gishnavi Kolli

Web Scraper Using Python

OVERVIEW

A fundamental project that gives you a better understanding of working with Python. Scrapping Data from Justdial using different python packages.

Problem Statement

Scrap data of Armstrong Ceiling Tile Dealers in a particular city and their information along with their phone numbers and addresses using python in less than 40 lines of code and export it as a CSV file format.

Introduction:

Web scraping is the process of collecting structured web data in an automated fashion. It's also called web data extraction. Some of the main use cases of web scraping include price monitoring, price intelligence, news monitoring, lead generation and market research among many others.

The web scraping process

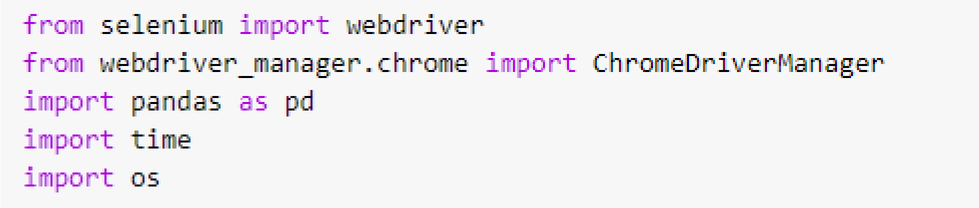
1. Identify target website
2. Collect URLs of the pages where you want to extract data from
3. Make a request to these URLs to get the HTML of the page
4. Use locators to find the data in the HTML
5. Save the data in a JSON or CSV {In this case csv} file or some other structured format.

Software Requirements

1. Programming Language: Python
2. Environment: Jupyter Notebooks
3. Database: CSV (export type)
4. Operation System: Windows XP or above
5. Libraries Used: Selenium, Pandas, webdriver_manager, time, OS.

Creating the Scraper

Open a New Notebook and import the required libraires



```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
import time
import os
```

Introduction to used Packages

Selenium

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language.

[**Installation:** pip install Selenium

Usage: import selenium]

WebDriverManager

WebDriverManager is a library which allows to automate the management of the drivers (e.g. *chromedriver*, *geckodriver*, etc.) required by Selenium WebDriver.

[**Installation:** pip install webdrivermanager

Usage: import webdrivermanager]

pandas

pandas python-based data analysis toolkit. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array. This makes pandas a trusted ally in data science and machine learning.

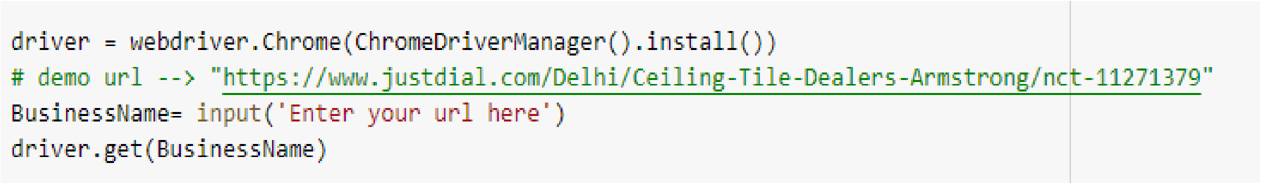
[**Installation:** pip install pandas

Usage: import pandas as pd]

- Numpy is required before installing pandas

Time and os modules are inbuilt which helps in interacting with system features of operating systems

Declaring URL & post forwarding a variable

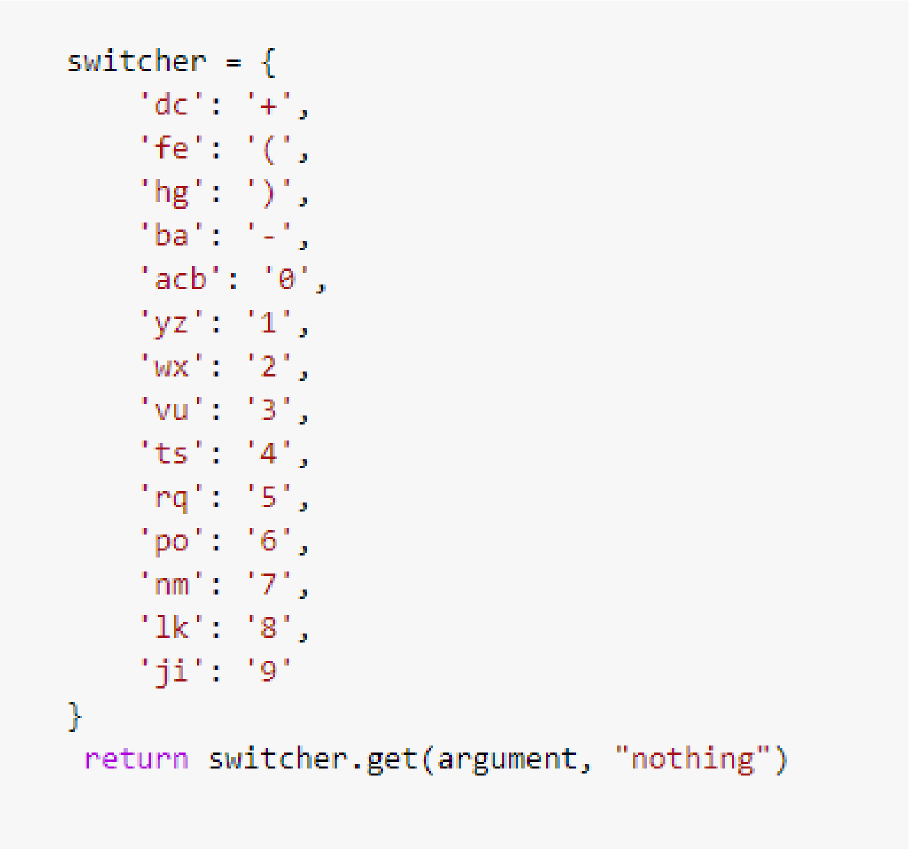


```
driver = webdriver.Chrome(ChromeDriverManager().install())
# demo url --> "https://www.justdial.com/Delhi/Ceiling-Tile-Dealers-Armstrong/nct-11271379"
BusinessName= input('Enter your url here')
driver.get(BusinessName)
```

Webdriver.chrome detects the system’s chrome version and installs the suitable Webdriver.

The Next Step is to give URL from where the data is to be scrapped.

As the Justdial developer has use encoding for displaying mobile number We have to decode it and write the class name for corresponding number for example 0 has class name “**mobilesv icon-acb**”

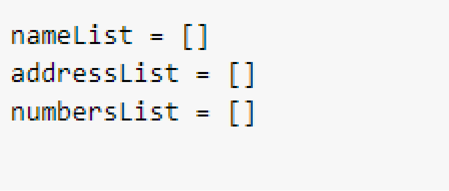


```
switcher = {
    'dc': '+',
    'fe': '(',
    'hg': ')',
    'ba': '-',
    'acb': '0',
    'yz': '1',
    'wx': '2',
    'vu': '3',
    'ts': '4',
    'rq': '5',
    'po': '6',
    'nm': '7',
    'lk': '8',
    'ji': '9'
}
 return switcher.get(argument, "nothing")
```

Unlike every other programming language, Python does not have a switch or case statement. Instead, we use dictionary mapping.

get () method of dictionary data type returns value of passed argument if it is present in dictionary otherwise second argument will be assigned as default value of passed argument

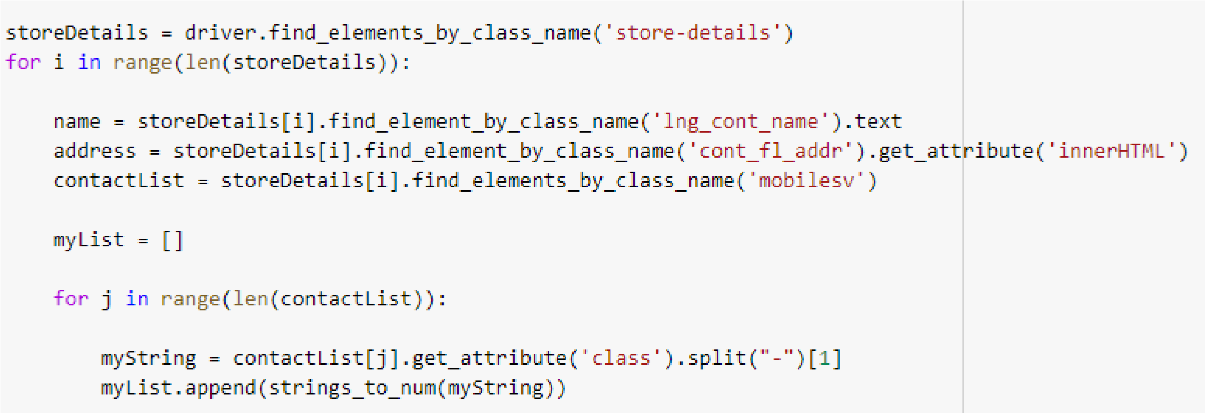
Declare variables



```
nameList = []
addressList = []
numbersList = []
```

We use nameList variable to store the data which we generate for name of the business, Similarly, for addressList and numberList

Main Function Process – Attaching Classes to Declared Variables



```
storeDetails = driver.find_elements_by_class_name('store-details')
for i in range(len(storeDetails)):

    name = storeDetails[i].find_element_by_class_name('lng_cont_name').text
    address = storeDetails[i].find_element_by_class_name('cont_fl_addr').get_attribute('innerHTML')
    contactList = storeDetails[i].find_elements_by_class_name('mobilesv')

    myList = []

    for j in range(len(contactList)):

        myString = contactList[j].get_attribute('class').split("-")[1]
        myList.append(strings_to_num(myString))
```

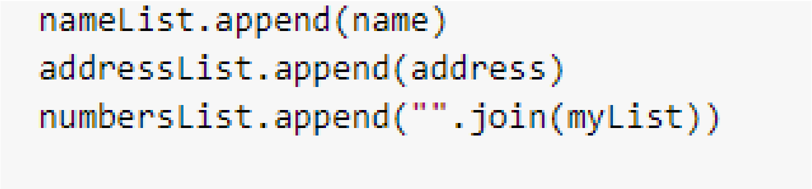
The main Logic of the scrapper lies in this part, where we have to find the class of the field which we want to extract and match its corresponding class and store it in a variable.

***find_element_by_class_name :** the first element with the matching class attribute name will be returned. [Selenium class]

*.**text:** corresponding text will be returned

*.**get_attribute**: Is used to get attributes of an element, such as getting href attribute of anchor tag. This method will first try to return the value of a property with the given name. If a property with that name doesn't exist, it returns the value of the attribute with the same name. If there's no attribute with that name, none is returned.

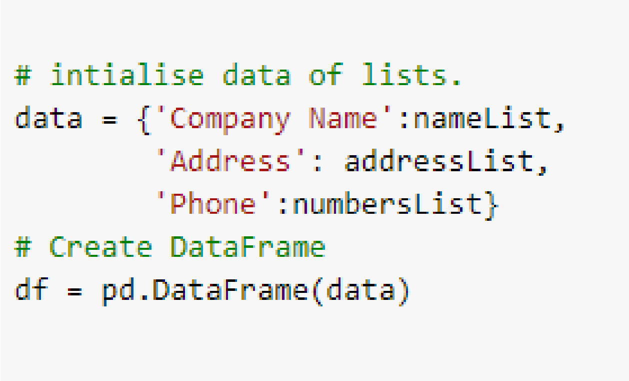
Appending the results to the declared variables



```
nameList.append(name)
addressList.append(address)
numbersList.append("".join(myList))
```

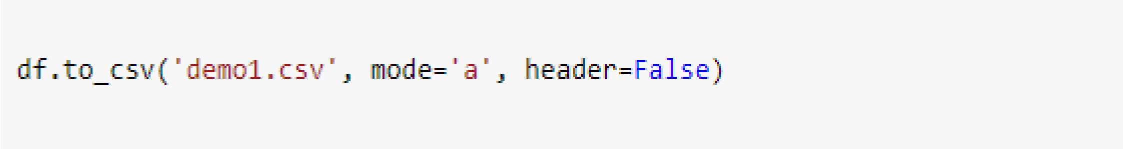
The results we get from the main functions will be appended to the variables declared by us.

Combining various variables into a single dictionary & data framing the Dictionary using Pandas



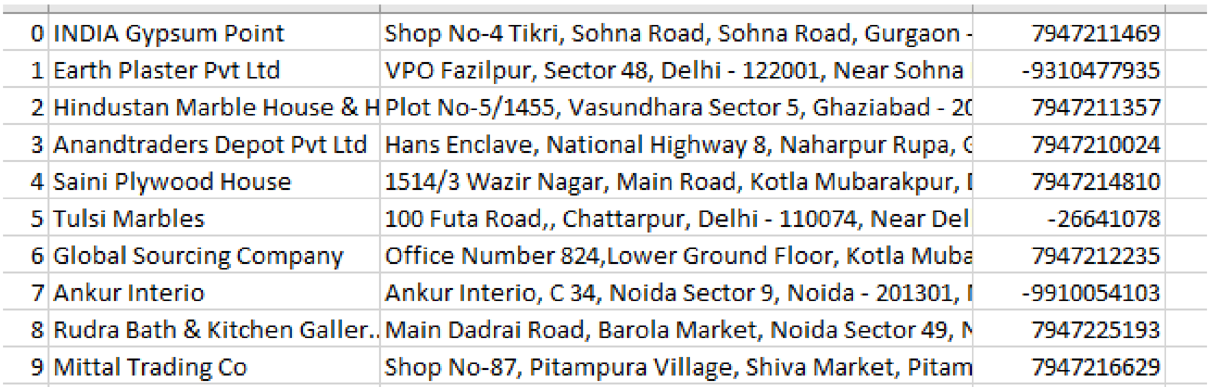
```
# intialise data of lists.
data = {'Company Name':nameList,
        'Address': addressList,
        'Phone':numbersList}
# Create DataFrame
df = pd.DataFrame(data)
```

Converting the Data frames into CSV File



```
df.to_csv('demo1.csv', mode='a', header=False)
```

A Glimpse of the CSV File



0	INDIA Gypsum Point	Shop No-4 Tikri, Sohna Road, Sohna Road, Gurgaon -	7947211469
1	Earth Plaster Pvt Ltd	VPO Fazilpur, Sector 48, Delhi - 122001, Near Sohna	-9310477935
2	Hindustan Marble House & H	Plot No-5/1455, Vasundhara Sector 5, Ghaziabad - 20	7947211357
3	Anandtraders Depot Pvt Ltd	Hans Enclave, National Highway 8, Naharpur Rupa, G	7947210024
4	Saini Plywood House	1514/3 Wazir Nagar, Main Road, Kotla Mubarakpur, I	7947214810
5	Tulsi Marbles	100 Futa Road,, Chattarpur, Delhi - 110074, Near Del	-26641078
6	Global Sourcing Company	Office Number 824,Lower Ground Floor, Kotla Muba	7947212235
7	Ankur Interio	Ankur Interio, C 34, Noida Sector 9, Noida - 201301, I	-9910054103
8	Rudra Bath & Kitchen Galler..	Main Dadrai Road, Barola Market, Noida Sector 49, N	7947225193
9	Mittal Trading Co	Shop No-87, Pitampura Village, Shiva Market, Pitam	7947216629

Conclusion

Therefore, we have successfully scraped the Data of Armstrong Ceiling Tile Dealers along with their mobile numbers, addresses & URLs using Python