APEXAIQ®

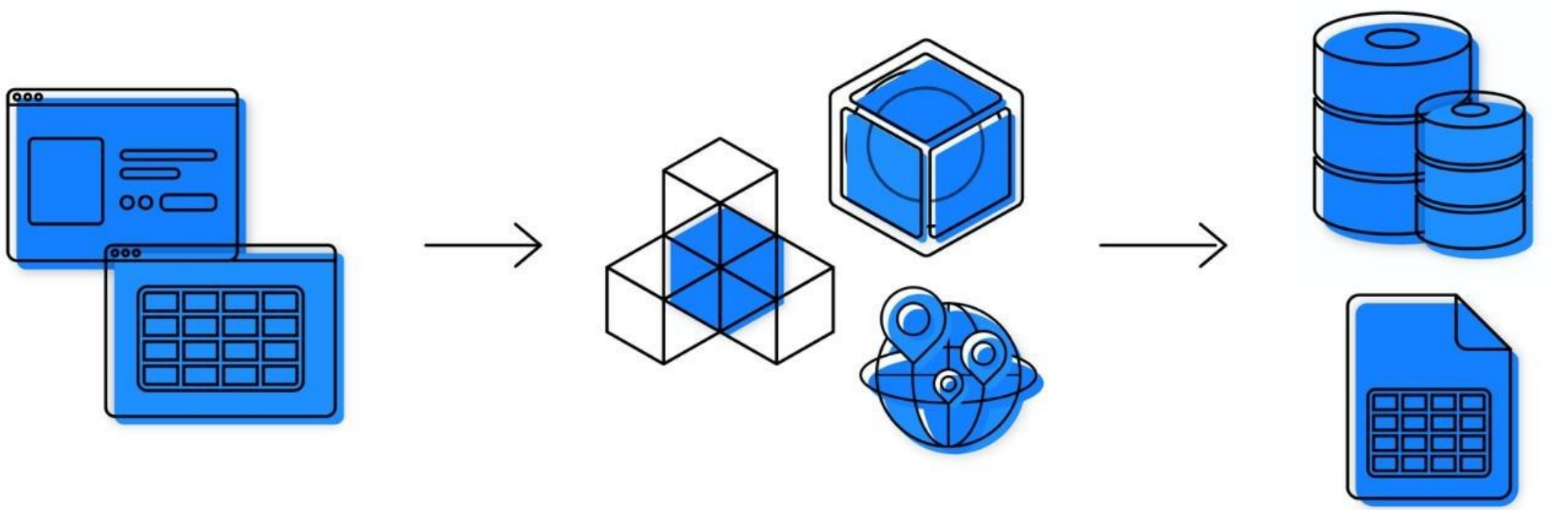
Web-Scraping

What is Web Scraping?

- Automatic method to obtain large amount of data from websites.
- Converting unstructured HTML or other data to structured data format.

For instance, we can visit website like Amazon, Flipkart it has huge amount of data and if we want to extract it then web scraping is used.

APEXAIQ®



Websites

Scraping Platform

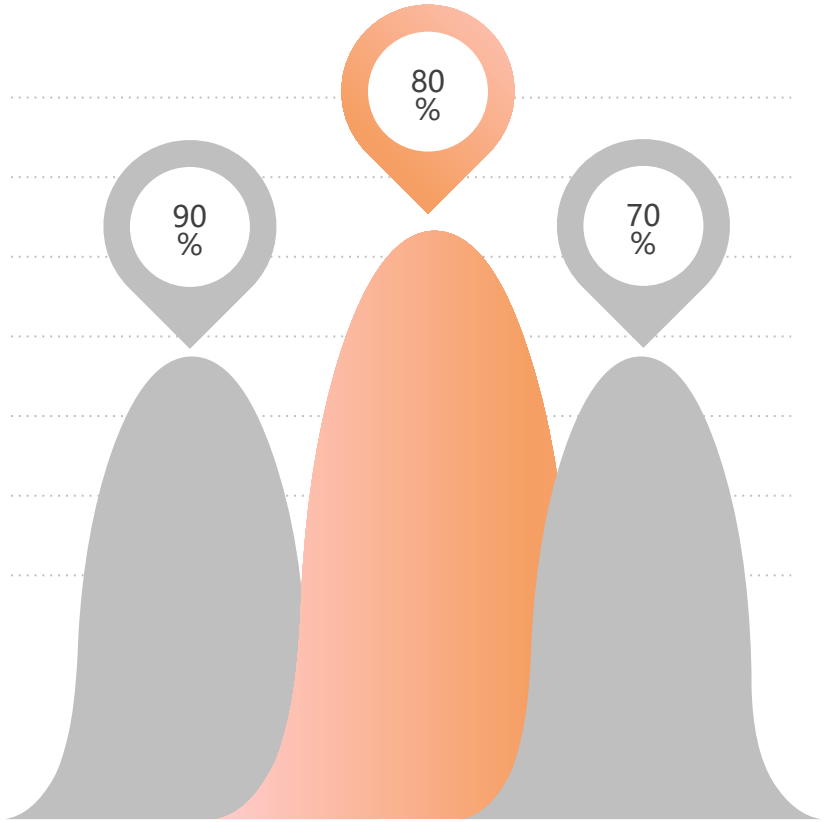
Structured Data

APEXAIQ®

Key-Points

- Web Scraping can be performed using python libraries such as beautifulsoup, Selenium, framework such as Scrapy.
- The extracted data can be used for making analysis,making predictions and can be used in other websites.
- Used in Price Monitoring,Market Research,News Monitoring,Sentimental Analysis,Email Marketing.

APEXAIQ®



Why Scraping?

- Company can scrape their product data and competing products as well to see how it impacts their pricing strategies and fix to optimal pricing.
- Helps in analysing consumer trends and understanding which direction the company should move in future.
- Here helps to extract the versions,eol dates,year of release,name,ip addresses,host names.

APEXAIQ®

Selenium WebDriver Feature

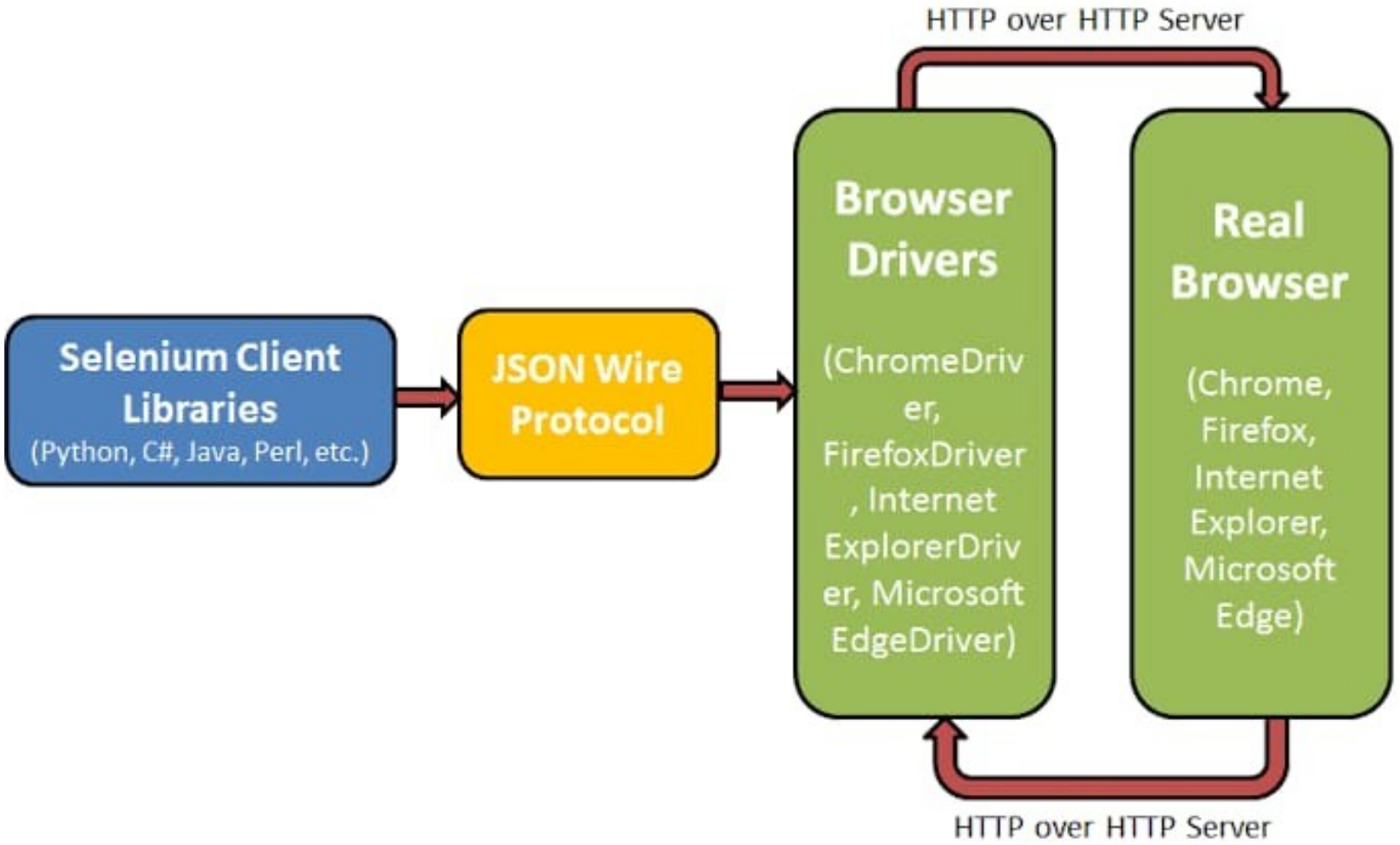
1. Multi-Brower Compatibility
2. Multiple Language Support
3. Speed & performance
4. Community Support
5. Open Source & Protable

6. Work on Different OS
7. Add- ons & Reusability
8. Simple Commands
9. Reduced test Execution time
10. No server installation

InterviewBit

- Selenium is a powerful tool for controlling web browsers through programs and performing browser automation.
- It is functional for all browsers, works on all major OS and its scripts are written in various languages in Python, Java.
- Compatible with various languages ,platforms, browsers
- An important component of Selenium is a **webdriver** which is similar to a API, it is nothing but a module which contains classes functions, methods.

Selenium Architecture



Selenium Architecture

Selenium Client Library : Selenium Developers have developed libraries for various languages.

JSON Wire Protocol : Used to make interaction between the client and server.
Acts as the intermediate between client and server and converts the request generated by client in a format which is understood by server and same for the response.

Browser driver: Enables a secure connection with the browser without revealing its internal logic.

Key Points

- Install Selenium→ pip install selenium
- Selenium requires driver to interface with the chosen browser.
- Different types of drivers available in Selenium WebDriver are:

  1.ChromeDriver

  2.FirefoxDriver

  3.InternetExplorerDriver

  4.EdgeDriver

  5.RemoteWebDriver

Key Points

- Install a webdriver compatible with our chrome version .
- Import libraries services, by and ChromeDriverManager from selenium.
- Create driver by providing the appropriate browser path
- Open the website by driver.get(“url”)
- Identify the tags and find elements by specifying their XPATH
- XPATH→contains path of the element situated at the web page
- Syntax:
  //tag_name[@attribute=’value’]

Note: you may need to restart the kernel to use updated packages.

In [ ]:

```
#imported webdriver from selenium
from selenium import webdriver

#selenium.webdriver module provides all the WebDriver implementations.
from selenium.webdriver.chrome.service import Service

#The By class is used to locate elements within a document.
from selenium.webdriver.common.by import By

import pandas
import time
import re

#Keys class provide keys in the keyboard like RETURN, F1, ALT etc.
from selenium.webdriver.common.keys import Keys
from webdriver_manager.chrome import ChromeDriverManager

options=webdriver.ChromeOptions()

# It allows users to run automated scripts in headless mode, meaning that the browser window wouldn't be visible.
options.add_argument("--headless")
options.add_argument("--window-size=1920,1080")

#the instance of Chrome WebDriver is created.
driver=webdriver.Chrome()

#The driver.get method will navigate to a page given by the URL.
driver.get("https://www.scrapethissite.com/pages/simple/")
```

Countries of the World: A Simple Example 250 items

A single page that lists information about all the countries in the world. Good for those just get started with web scraping. Practice looking for patterns in the HTML that will allow you to extract information about each country. Then, build a simple web scraper that makes a request to this page, parses the HTML and prints out each country's name.

There are 4 video lessons that show you how to scrape this page.

Data via http://peric.github.io/GetCountries/

Andorra
Capital: Andorra la Vella
Population: 84000
Area (km²): 468.0

Antigua and Barbuda
Capital: St. John's
Population: 86754
Area (km²): 443.0

Armenia
Capital: Yerevan
Population: 2968000
Area (km²): 29800.0

Argentina

United Arab Emirates
Capital: Abu Dhabi
Population: 4975593
Area (km²): 82880.0

Anguilla
Capital: The Valley
Population: 13254
Area (km²): 102.0

Angola
Capital: Luanda
Population: 13068161
Area (km²): 1246700.0

American Samoa

Afghanistan
Capital: Kabul
Population: 29121286
Area (km²): 647500.0

Albania
Capital: Tirana
Population: 2986952
Area (km²): 28748.0

Antarctica
Capital: None
Population: 0
Area (km²): 1.4E7

Austria

jupyter Apexa_IQ_ppt Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

Run Code

In [8]:
```
#get the population for the country
population_list=driver.find_elements(By.CLASS_NAME,'country-population')

#parse the data

populations=[]
for population in population_list:
    #get the text data
    temp=population.text
    populations.append(temp)
populations
```

Out[8]:
```
['84000',
 '4975593',
 '29121286',
 '86754',
 '13254',
 '2986952',
 '2968000',
 '13068161',
 '0',
 '41343201',
 '57881',
 '8205000',
 '21515754',
 '71566',
 '26711',
 '8303512',
 '4590000',
 '285653',
 '156118464',
```

In [6]:
```
#get country names
#XPath is the language used for locating nodes in an XML document.
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [6]:
```
#get country names
#XPath is the language used for locating nodes in an XML document.
country_list=driver.find_elements( By.XPATH,"//h3[@class='country-name']")

#parse the data
countries=[]
for country in country_list:
    #get the text data
    temp=country.text
    countries.append(temp)
countries
```

Out[6]:
```
['Andorra',
 'United Arab Emirates',
 'Afghanistan',
 'Antigua and Barbuda',
 'Anguilla',
 'Albania',
 'Armenia',
 'Angola',
 'Antarctica',
 'Argentina',
 'American Samoa',
 'Austria',
 'Australia',
 'Aruba',
 'Åland',
 'Azerbaijan',
 'Bosnia and Herzegovina',
 'Barbados',
 'Bangladesh',
```

In [10]:
```
data=pandas.DataFrame()
data['Country Names']=countries
data['Population']=populations
data
```

Out[10]:

	Country Names	Population
0	Andorra	84000
1	United Arab Emirates	4975593
2	Afghanistan	29121286
3	Antigua and Barbuda	86754
4	Anguilla	13254
...	...	...
245	Yemen	23495361
246	Mayotte	159042
247	South Africa	49000000
248	Zambia	13460305
249	Zimbabwe	11651858

250 rows × 2 columns

In [ ]:

0	Andorra	84000
1	United Arab Emirates	4975593
2	Afghanistan	29121286
3	Antigua and Barbuda	86754
4	Anguilla	13254
...	...	...
245	Yemen	23495361
246	Mayotte	159042
247	South Africa	49000000
248	Zambia	13460305
249	Zimbabwe	11651858

250 rows × 2 columns

In [12]:
```
#save the data
data.to_csv('countries.csv',index=False)

driver.quit()
```

In [ ]:

Microsoft Excel (Product Activation Failed) - countries

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Country Names	Population																
2	Andorra	84000																
3	United Arab Emirates	4975593																
4	Afghanistan	29121286																
5	Antigua and Barbuda	86754																
6	Anguilla	13254																
7	Albania	2986952																
8	Armenia	2968000																
9	Angola	13068161																
10	Antarctica	0																
11	Argentina	41343201																
12	American Samoa	57881																
13	Austria	8205000																
14	Australia	21515754																
15	Aruba	71566																
16	Ã…land	26711																
17	Azerbaijan	8303512																
18	Bosnia and Herzegovina	4590000																
19	Barbados	285653																
20	Bangladesh	156118464																
21	Belgium	10403000																
22	Burkina Faso	16241811																
23	Bulgaria	7148785																
24	Bahrain	738004																
25	Burundi	9863117																
26	Benin	9056010																
27	Saint BarthÃ©lemy	8450																
28	Bermuda	65365																
29	Brunei	395027																
30	Bolivia	9947418																
31	Bonaire	18012																
32	Brazil	201103330																
33	Bahamas	301790																

countries

Type here to search

ENG 20:29 15-06-2023

In [19]:
```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import pandas
import time
import re
from selenium.webdriver.common.keys import Keys
from webdriver_manager.chrome import ChromeDriverManager

options=webdriver.ChromeOptions()
options.add_argument("--headless")
options.add_argument("--window-size=1920,1080")

driver=webdriver.Chrome()
```

In [10]:
```
driver.get("https://winscp.net/eng/docs/history")

time.sleep(5)

main_tag=driver.find_element(By.XPATH,"//section[@class='col-md-9']")
```

In [18]:
```
version_elements=main_tag.find_elements(By.XPATH,".//h2")
date_elements=main_tag.find_elements(By.XPATH,".//time")


# for versions in version_elements:
#     print(versions.text)


# for date in date_elements:
#     print(date.text)
```

In [18]:
```
version_elements=main_tag.find_elements(By.XPATH,".//h2")
date_elements=main_tag.find_elements(By.XPATH,".//time")


# for versions in version_elements:
#     print(versions.text)


# for date in date_elements:
#     print(date.text)


list_of_dict=[]

for version,date in zip(version_elements,date_elements):
#     print(version.text,date.text)

    new_row={
        "version":version.text,
        "releaseDate":date.text
    }
    list_of_dict.append(new_row)
#     print(new_row)
# print(list_of_dict)

dataframe=pandas.DataFrame(list_of_dict)
print(dataframe)

dataframe.to_csv("winscpnew",index=False)


# driver.close()
```

version releaseDate

```
# driver.close()
```

	version	releaseDate
0	6.2 (not released yet)	2023-06-09
1	6.1.1 (not released yet)	2023-05-30
2	6.1	2023-05-23
3	6.0.2 RC	2023-04-18
4	6.0.1 beta	2023-03-07
5	6.0 beta	2023-02-08
6	5.21.8	2023-04-11
7	5.21.7	2023-01-23
8	5.21.6	2022-11-28
9	5.21.5	2022-10-06
10	5.21.4	2022-09-12
11	5.21.3	2022-09-06
12	5.21.2	2022-08-08
13	5.21.1	2022-06-24
14	5.21	2022-06-15
15	5.20.4 RC	2022-06-08

In [ ]:

In [ ]:

APEXAIQ®

THANKS