⛳ Web Scraping Template

Import Libraries

In [1]:
```
import requests # for getting web contents
from bs4 import BeautifulSoup # for scraping web contents
import pandas as pd # for data analysis
```

⛳ URL

For web scrapping it's important to have url.

In [3]:
```
# link of web page that you want to scrap data
URL = ''

# get web data
page = requests.get(URL)

# parse web data
soup = BeautifulSoup(page.content, "html.parser")
```

In [ ]:
```
# find the table
# our trageted table is last

# getting the table head because it may contains headings (column names)
html_thead = soup.find_all('thead')[-1]

#getting all the rows in table head
html_tr = [tr for tr in html_thead.find_all('tr')]

# list to store all table headings
headings = []

# loop through table head
for tr in html_tr:
    # getting all th
    th = tr.find_all(['th'])
    # storing all th value in row and removing white space
    row = [i.text.strip() for i in th]
    # append headings
    headings.append(row)

# print heading
print(headings)
```

In [ ]:
```
# getting the table body
html_tbody = soup.find_all('tbody')[-1]

#getting all the rows in table body
html_text = [tr for tr in html_tbody.find_all('tr')]

# list to store all content
content = []
```

```
# loop through table body
for tr in html_text:
    # getting all th, td
    th = tr.find_all(['th','td'])
    # storing all th value in row and removing white space
    row = [i.text.strip() for i in th]
    # append content
    content.append(row)

# print content
print(content)
```

In [ ]:
```
# save contents in a dataframe
data = pd.DataFrame(content[:], columns=headings[0])
```

Data Analysis

Look at Example Records

In [ ]:
```
# check few top rows of data
data.head()
```

Descriptive Statistics

In [ ]:
```
data.describe()
```

Data Cleaning

Rename Column Name

In [ ]:
```
# rename column name if required
data = data.rename(columns={'First Column Name':'New Name', 'Second Column Name':'New Name
```

Remove unwanted symbols (like % and thousand comma from integer)

In [ ]:
```
# remove extra characters from columns
data['column name'] = data['column name'].str.replace('%','')
data['column name'] = data['column name'].str.replace(',','')
```

Save Data into CSV

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

In [ ]:
```
This template helps you to scrap data from any website and convert data to CSV file
```

```
and once you have CSV file save file in your local
directry or cloud and  than you can do data cleaning or Analysing same as any CSV files.
```

This template helps you to scrap data from any website and convert data to CSV file

And once you have CSV file save file in your local

directry or cloud and than you can do data cleaning

or Analysing same as any CSV files.

Good Luck eveybody!!!! 🙂

Please let me know if you like this templates or not.