web scrapping

In [1]:
```
!pip install requests beautifulsoup4
```

```
Requirement already satisfied: requests in c:\users\pooja reddy\anaconda3\lib\si
te-packages (2.31.0)
Requirement already satisfied: beautifulsoup4 in c:\users\pooja reddy\anaconda3
\lib\site-packages (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\pooja reddy
\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\pooja reddy\anaconda3\li
b\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\pooja reddy\anacon
da3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\pooja reddy\anacon
da3\lib\site-packages (from requests) (2023.7.22)
Requirement already satisfied: soupsieve>1.2 in c:\users\pooja reddy\anaconda3\l
ib\site-packages (from beautifulsoup4) (2.4)
```

In [2]:
```
import requests
from bs4 import BeautifulSoup
url = "http://quotes.toscrape.com/"
response = requests.get(url)
if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')
    quotes = soup.find_all("div", class_="quote")
    for i, quote in enumerate(quotes[:5]):
        text = quote.find("span", class_="text").text
        author = quote.find("small", class_="author").text
        tags = [tag.text for tag in quote.find_all("a", class_="tag")]
        print(f"{i+1}. \"{text}\" - {author}")
        print(f"Tags:{','.join(tags)} \n")
    else:
        print(f"Failed to retrieve the webpage.statuscode:{response.status_code}"
```

```
1. "“The world as we have created it is a process of our thinking. It cannot be
changed without changing our thinking.”" - Albert Einstein
Tags:change,deep-thoughts,thinking,world

2. "“It is our choices, Harry, that show what we truly are, far more than our ab
ilities.”" - J.K. Rowling
Tags:abilities,choices

3. "“There are only two ways to live your life. One is as though nothing is a mi
racle. The other is as though everything is a miracle.”" - Albert Einstein
Tags:inspirational,life,live,miracle,miracles

4. "“The person, be it gentleman or lady, who has not pleasure in a good novel,
must be intolerably stupid.”" - Jane Austen
Tags:aliteracy,books,classic,humor

5. "“Imperfection is beauty, madness is genius and it's better to be absolutely
ridiculous than absolutely boring.”" - Marilyn Monroe
Tags:be-yourself,inspirational

Failed to retrieve the webpage.statuscode:200
```

In [5]:
```
import requests
from bs4 import BeautifulSoup
city = "india/hyderabad"
url = f"https://www.timeanddate.com/weather/{city}"
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')
temp = soup.find("div", class_="h2").text.strip() if soup.find("div", class_="h2"
desc = soup.find("p").text.strip() if soup.find("p") else "N/A"
print(f"Current Weather in Hyderabad:{temp}|{desc}")
```

```
Current Weather in Hyderabad:28 °C|Haze.
```

In [6]:
```
import requests
from bs4 import BeautifulSoup
# Product search URL (Example: iPhone)
search_url = "https://www.amazon.in/s?k=iphone&crid=PQVCJSNISAH4&sprefix=iphone%2
# Headers (Mimic a browser)
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
# Send GET request
response = requests.get(search_url, headers=headers)
soup = BeautifulSoup(response.text, "html.parser")
# Extract first product name & price
product = soup.select_one("span.a-size-medium")
price = soup.select_one("span.a-price-whole")
# Display product details
if product and price:
    print(f"Product: {product.text.strip()}")
    print(f"Price: ${price.text.strip()}")
else:
    print("Could not find product details.")
```

```
Could not find product details.
```

In [8]:
```
import requests
from bs4 import BeautifulSoup
# Wikipedia page URL
url = "https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_popula
# Send GET request
response = requests.get(url,headers=headers)
soup = BeautifulSoup(response.text, "html.parser")
# Find the table
table = soup.find("table", class_="wikitable")
# Extract the first 5 countries and their population
for row in table.find_all("tr")[1:6]:  # Skip the header row
    columns = row.find_all("td")
    country = columns[1].text.strip()
    population = columns[2].text.strip()
    print(f"{country}: {population}")
```

```
World: 8,119,000,000
China: 1,408,280,000
1,402,737,000: 17.2%
United States: 340,110,988
Indonesia: 282,477,584
```

In [12]:
```
from IPython.display import display, HTML
```

In [13]:
```
display(HTML("""<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
"""))
```

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

In [ ]: