Perform webscraping using the demonstrated methods on a website of your choice.

In [2]:
```
from requests import get
from bs4 import BeautifulSoup
import pandas as pd
```

In [3]:
```
response = get('https://www.antutu.com/en/ranking/rank1.htm')
```

In [4]:
```
html_soup = BeautifulSoup(response.text, 'html.parser')
num = html_soup.find_all('div', class_ = 'm_l fl')
print(type(num))
print(len(num))
```

```
<class 'bs4.element.ResultSet'>
1
```

In [5]:
```
phones = html_soup.find_all('ul')

data = []
for phone in phones[1:]:
    items = phone.find_all('li')
    if len(items) >= 6:
        Phone = items[0].text.strip()
        CPU = items[1].text.strip()
        GPU = items[2].text.strip()
        MEM = items[3].text.strip()
        UX = items[4].text.strip()
        SCORE = items[5].text.strip()
        data.append([Phone, CPU, GPU, MEM, UX, SCORE])

df = pd.DataFrame(data, columns=['Phone', 'CPU', 'GPU', 'MEM', 'UX', 'SCORE'])
df.head(10)
```

Out[5]:

	Phone	CPU	GPU	MEM	UX	SCORE
0	Device	CPU□	GPU□	MEM□	UX□	Total Score□
1	1iQOO 13 (S-8 Elite 16+512)	586291	1183870	492574	432974	2695709
2	2Red Magic 10 Pro (S-8 Elite 12+256)	593626	1193547	476871	387160	2651204
3	3Mi 15 Ultra (S-8 Elite 16+512)	542576	1036853	474641	423893	2477963
4	4vivo X200 Pro (M-9400 12+256)	580620	1089373	394090	412650	2476733
5	5Mi 15 (S-8 Elite 12+512)	550938	1038053	446477	359902	2395370
6	6vivo X200 Pro mini (M-9400 12+256)	541412	1033210	375355	392217	2342194
7	7Galaxy S25 Ultra (S-8 Elite OC 12+512)	562773	880816	427168	345515	2216272
8	8Galaxy S25+ (S-8 Elite OC 12+512)	518255	884552	449756	341128	2193691
9	9OnePlus 13 (S-8 Elite 16+512)	420683	1016797	428637	322146	2188263

In [6]:

```
df = df.reset_index(drop=True)
```

In [7]:

```
df = df.drop(index=0)
```

In [8]:

```
df.head(10)
```

Out[8]:

	Phone	CPU	GPU	MEM	UX	SCORE
1	1iQOO 13 (S-8 Elite 16+512)	586291	1183870	492574	432974	2695709
2	2Red Magic 10 Pro (S-8 Elite 12+256)	593626	1193547	476871	387160	2651204
3	3Mi 15 Ultra (S-8 Elite 16+512)	542576	1036853	474641	423893	2477963
4	4vivo X200 Pro (M-9400 12+256)	580620	1089373	394090	412650	2476733
5	5Mi 15 (S-8 Elite 12+512)	550938	1038053	446477	359902	2395370
6	6vivo X200 Pro mini (M-9400 12+256)	541412	1033210	375355	392217	2342194
7	7Galaxy S25 Ultra (S-8 Elite OC 12+512)	562773	880816	427168	345515	2216272
8	8Galaxy S25+ (S-8 Elite OC 12+512)	518255	884552	449756	341128	2193691
9	9OnePlus 13 (S-8 Elite 16+512)	420683	1016797	428637	322146	2188263
10	10realme Neo7 (M-9300+ 12+256)	514038	820169	369862	369570	2073639

In [9]:

```
df = df.drop(columns = ["MEM", "UX"])
df.head(10)
```

Out[9]:

	Phone	CPU	GPU	SCORE
1	1iQOO 13 (S-8 Elite 16+512)	586291	1183870	2695709
2	2Red Magic 10 Pro (S-8 Elite 12+256)	593626	1193547	2651204
3	3Mi 15 Ultra (S-8 Elite 16+512)	542576	1036853	2477963
4	4vivo X200 Pro (M-9400 12+256)	580620	1089373	2476733
5	5Mi 15 (S-8 Elite 12+512)	550938	1038053	2395370
6	6vivo X200 Pro mini (M-9400 12+256)	541412	1033210	2342194
7	7Galaxy S25 Ultra (S-8 Elite OC 12+512)	562773	880816	2216272
8	8Galaxy S25+ (S-8 Elite OC 12+512)	518255	884552	2193691
9	9OnePlus 13 (S-8 Elite 16+512)	420683	1016797	2188263
10	10realme Neo7 (M-9300+ 12+256)	514038	820169	2073639

In [10]:

```
df.dtypes
```

Out[10]:

```
Phone    object
CPU      object
GPU      object
SCORE    object
dtype: object
```

In [27]:

```
df[["CPU", "GPU", "SCORE"]] = df[["CPU", "GPU", "SCORE"]].apply(pd.to_numeric, erro
```

In [29]:

```
df.dtypes
```

Out[29]:

```
Phone     object
CPU      float64
GPU      float64
SCORE    float64
dtype: object
```

In [31]:

```
df.to_csv("Phone_Rankings.csv", index = False)
```

In [ ]:

```

```