Flipkart mi Phones Web Scrapping

- Scrapping Elements Are:
- 1. Name : Name of the Product
- 1. Price : Price of the product
- 1. Stars : Stars out of 5
- 1. Ratings: In Thousands
- 1. Reviews: In Hundreds
- 1. Ram and Rom : -
- 1. Battery: -
- 1. Camera: -

Import Essential Libraries

In [1]:
```
import requests
import re
```

Reading the Web Page into Python

In [2]:
```
page = requests.get("https://www.flipkart.com/search?q=mi+mobiles&sid=tyy%2C4io&as=on&as-show=on&otracker=AS_QueryStore_OrganicAutoSuggest_1_4_na_na_ps&otrack
```

In [3]:
```
page
```

Out[3]:
```
<Response [200]>
```

In [4]:
```
page.text[:500]
```

Out[4]:
```
'<!doctype html><html lang="en"><head><link href="https://rukminim1.flixcart.com" rel="preconnect"/><link rel="stylesheet" href="//static-assets-web.flixcart.
com/www/linchpin/fk-cp-zion/css/app.chunk.fdabe1.css"/><meta http-equiv="Content-type" content="text/html; charset=utf-8"/><meta http-equiv="X-UA-Compatible"
content="IE=Edge"/><meta property="fb:page_id" content="102988293558"/><meta property="fb:admins" content="658873552,624500995,100000233612389"/><meta name="r
obots" content="noodp"/><li'
```

Parsing the Html using BeautifulSoup

In [5]:
```
from bs4 import BeautifulSoup as bs
```

In [6]:
```
# Get the page content and parsing the html with BeautifulSoup
soup = bs(page.content, "html.parser")
```

In [10]:
```
# Class Types
title = soup.title
print(type(soup))
print(type(title.string))
```

```
<class 'bs4.BeautifulSoup'>
<class 'bs4.element.NavigableString'>
```

In [899…
```
# Just Title Of the Page with the help of title keyword
print(soup.title)
```

```
<title>Mi Mobiles- Buy Products Online at Best Price in India - All Categories | Flipkart.com</title>
```

Scrapping The Data

- Collecting the records with the help of tags and attributes of tags

Price

In [164…
```
# Extracting The prices
Price = soup.find_all("div" , attrs={"class":"_30jeq3 _1_WHN1"})
```

In [165…
```
Price[0].get_text()
```

Out[165…
```
'₹9,085'
```

In [262…
```
# Adding the Product Price in the list
Prices = []
for i in range(0,len(Price)):
    Prices.append(Price[i].get_text().replace("₹",'').strip())
```

In [263…
```
len(Prices)
```

Out[263…
```
24
```

Stars

In [168…
```
#Extracting The Product Stars
Star = soup.find_all("div", attrs={"class":"_3LWZlK"})
```

In [169…
```
Star[0].get_text()
```

Out[169…
```
'4.2'
```

In [397…
```
# Adding the Product Stars in the List
Stars = []
for i in range(0, len(Prices)):
    Stars.append(Star[i].get_text().strip())
```

In [398…
```
len(Stars)
```

Out[398…
```
24
```

In [399…
```
Stars
```

Out[399…
```
['4.2',
 '4.2',
 '4.3',
 '4.3',
 '4.3',
 '4.3',
 '4.3',
 '4.2',
 '4.4',
 '4.2',
 '4.4',
 '4.2',
 '4.2',
 '4.3',
 '4.3',
 '4.4',
 '4.4',
 '4.4',
 '4.4',
 '4.5',
 '4.4',
 '4.2',
 '4.4']
```

Name

In [122…
```
# Extracting the Product Name
Name = soup.find_all("div", attrs={"class":"_4rR01T"})
```

In [123…
```
Name[0].get_text()
```

Out[123…
```
'Redmi 9 (Sky Blue, 64 GB)'
```

In [267…
```
# Adding the Product Name in the list
Names = []
for i in range(0, len(Name)):
    Names.append(Name[i].get_text().strip())
```

In [268…
```
len(Names)
```

Out[268…
```
24
```

In [269…
```
Names[4]
```

Out[269…
```
'Redmi 9A (Midnight Black, 32 GB)'
```

Ratings

In [127…
```
# Extracting The Product Ratings
Rating = soup.find_all("span" , {"class": re.compile("_2_R_DZ")})
Rating[0].text
```

Out[127…
```
'14,288 Ratings\xa0&\xa0814 Reviews'
```

In [278…
```
# Adding the Product Ratings in the List
Ratings = []
for i in soup.find_all("span" , {"class": re.compile("_2_R_DZ")}):
    Ratings.append(i.text[:6].strip())      # Slicing the first 6 index
```

In [279…
```
len(Ratings)
```

Out[279…
```
24
```

In [309…
```
type(Ratings)
```

Out[309…
```
list
```

Reviews

In [181…
```
# Extracting The Product Reviews in to the String
Reviews = ''
for i in soup.find_all("span" , {"class": re.compile("_2_R_DZ")}):
    Reviews += (i.text.split("&")[1])
```

In [182…
```
Reviews
```

Out[182…
```
'\xa0814 Reviews\xa0814 Reviews\xa0481 Reviews\xa0353 Reviews\xa0481 Reviews\xa0481 Reviews\xa0353 Reviews\xa0926 Reviews\xa067,010 Reviews\xa0814 Reviews\xa0
100 Reviews\xa0814 Reviews\xa0926 Reviews\xa060 Reviews\xa060 Reviews\xa067,010 Reviews\xa03 Reviews\xa01,588 Reviews\xa0100 Reviews\xa01,588 Reviews\xa05,399
Reviews\xa03,784 Reviews\xa0926 Reviews\xa03,784 Reviews'
```

In [183…
```
reviewss = Reviews.split() # Spliting the Reviews in the list so that we can iterate throw every element init
                           # and extracting the \xa
```

In [184…
```
# Iterating throw split list and removing the \xa characters
og_string = ''
for i in reviewss:
    og_string += i
```

In [185…
```
og_string
```

Out[185…
```
'814Reviews814Reviews481Reviews353Reviews481Reviews481Reviews353Reviews926Reviews67,010Reviews814Reviews100Reviews814Reviews926Reviews60Reviews67,010
Reviews3Reviews1,588Reviews100Reviews1,588Reviews5,399Reviews3,784Reviews926Reviews3,784Reviews'
```

In [186…
```
# With the help of Regular Expression and Character Classes Gettin only Numeric digit from previous String
ro = re.compile(r"\d+") # Character Class \d+
fall = ro.findall(og_string)
print(fall)
```

```
['814', '814', '481', '353', '481', '481', '353', '926', '67', '010', '814', '100', '814', '926', '60', '60', '67', '010', '3', '1', '588', '100', '1', '588',
'5', '399', '3', '784', '926', '3', '784']
```

    - Here there are many values for reviews so, we will be deleting some of them to get the len of reviews = 24

In [192…
```
# With the Help of Np.delete , Eliminating the unwanted Reviews
import numpy as np
i = [9 ,10, 15, 17, 18, 19, 20] # Eliminating the unwanted reviews with the help of their Index number
list_of_Reviews = np.delete(fall,i).tolist()
```

In [193…
```
len(list_of_Reviews)
```

Out[193…
```
24
```

Ram

In [138…
```
# Extracting The Product Ram and Rom
Ram = soup.find_all("li", attrs={"class":"rgWa7D"})
Ram[0].get_text()
```

Out[138…
```
'4 GB RAM | 64 GB ROM'
```

In [139…
```
# Adding the Product Ram in the list
RAM = []
for i in range(0,len(Ram)):
    RAM.append(Ram[i].get_text())
```

In [140…
```
print(RAM[0]) # 0th index element
print("The lenth of the RAM is:--",len(RAM[0]))
print()
print(RAM[17]) # 17th index element
print("The Length of the RAM is:--",len(RAM[17]))
```

```
4 GB RAM | 64 GB ROM
The lenth of the RAM is:-- 20

3 GB RAM | 32 GB ROM | Expandable Upto 512 GB
The Length of the RAM is:-- 45
```

In [141…
```
RAMS = []
for i in RAM:
    if(len(i)==20 or len(i)==21 or len(i)==45 or len(i)==46):
# Iterating throw the len of the Product if the len of their name is 20,21,45,46
        RAMS.append(i)
```

In [142…
```
len(RAMS)
```

Out[142…
```
24
```

Camera

In [143…
```
# Extracting The Product Camera
CAM = soup.find_all("li" , attrs={"class":"rgWa7D"})
CAM[1].get_text()
```

Out[143…
```
'16.59 cm (6.53 inch) HD+ Display'
```

In [144…
```
list_cam = []
for i in range(0, len(CAM)):
            list_cam.append(CAM[i].get_text())
```

In [145…
```
print(list_cam[2]) # 2nd index element
print("The length of Camera Name is:--",len(list_cam[2]))
print()
print(list_cam[14]) # 14th index element
print("The length is:--",len(list_cam[14]))
print()
print(list_cam[59]) # 59th index element
print("The length is:--",len(list_cam[59]))
```

```
13MP + 2MP | 5MP Front Camera
The length of Camera Name is:-- 29

13MP Rear Camera
The length is:-- 16

108MP + 8MP + 2MP + 2MP | 16MP Front Camera
The length is:-- 43
```

In [288…
```
CAM1 = []
for i in list_cam:
    if(len(i)==29 or len(i)==16 or len(i)==43 or len(i)==54 or len(i)==37 or len(i)==42):
# Iterating throw the len of the Product if the len of their name is 29,16,43,54,37,42
        CAM1.append(i)
```

- There are other elements whose len is same as len of Camera name... so, Eliminating the unwanted elements from the list

In [293…
```
# Eliminating unwanted list with the help of their index number
unwanted_list = [1,3,4,6,7,8,9,11,12,13,15,16,17,20,22,23,25,27,29,31,32,34,36,38,39,41,42,44,45,47,48]
list_of_mi_cams = np.delete(CAM1,unwanted_list).tolist()
```

In [392…
```
len(list_of_mi_cams)
```

Out[392…
```
22
```

In [295…
```
Cam = {"CAMERA":list_of_mi_cams}
Cam1 = pd.DataFrame(Cam)
Cam1
```

Out[295…

	CAMERA
0	13MP + 2MP | 5MP Front Camera
1	13MP + 2MP | 5MP Front Camera
2	13MP Rear Camera
3	13MP Rear Camera
4	13MP Rear Camera
5	13MP + 2MP | 8MP Front Camera
6	12MP + 2MP | 8MP Front Camera
7	13MP + 8MP | 5MP Front Camera
8	108MP + 8MP + 2MP + 2MP | 16MP Front Camera
9	13MP + 8MP | 5MP Front Camera
10	13MP + 2MP | 8MP Front Camera
11	108MP + 8MP + 2MP + 2MP | 16MP Front Camera
12	108MP + 8MP + 2MP + 2MP | 16MP Front Camera
13	12MP + 2MP | 8MP Front Camera
14	48MP Rear Camera
15	48MP + 8MP + 5MP + 2MP | 16MP Front Camera
16	108MP + 8MP + 2MP + 2MP | 16MP Front Camera
17	48MP + 8MP + 5MP + 2MP | 16MP Front Camera
18	48MP + 13MP + 8MP | 20MP Front Camera
19	48MP + 8MP + 2MP + 2MP | 13MP Front Camera
20	13MP + 2MP | 8MP Front Camera
21	48MP + 8MP + 2MP + 2MP | 13MP Front Camera

Battery

In [151…
```
# Extracting The Product Battery
Bat = soup.find_all("li", attrs={"class":"rgWa7D"})
Bat[3].get_text()
```

Out[151…
```
'5000 mAh Battery'
```

In [152…
```
list_bat = []
for i in range(0, len(Bat)):
    list_bat.append(Bat[i].get_text())
```

In [153…
```
print(list_bat[48])
print("The length is:--",len(list_bat[48]))
```

```
5000 mAh Battery
The length is:-- 16
```

In [154…
```
print(list_bat[72])
print("The length is:--",len(list_bat[72]))
```

```
5000 mAh Lithium Polymer Battery
The length is:-- 32
```

In [155…
```
Battery = []
for i in list_bat:
    if(len(i)==16 or len(i)==32):
# Iterating throw the len of the Product if the len of their name is 16,32
        Battery.append(i)
```

In [156…
```
len(Battery)
```

Out[156…
```
33
```

In [315…
```
# Eliminating the unwanted elemets from the list
unwanted_list = [0,2,4,6,8,10,12,16,19,21,25]
list_of_battery = np.delete(Battery, unwanted_list).tolist()
```

In [316…
```
len(list_of_battery)
```

Out[316…
```
22
```

In [356…
```
Batt = " "
for i in list_of_battery:
    Batt += i
```

In [357…
```
" ".join(str1.split(","))
```

Out[357…
```
' 5000 mAh Battery5000 mAh Battery5000 mAh Battery5000 mAh Battery5000 mAh Battery5000 mAh Battery5000 mAh Battery5000 mAh Lithium Polymer Battery5000 mAh Bat
tery5000 mAh Battery4820 mAh Battery5000 mAh Battery4820 mAh Battery4820 mAh Battery5000 mAh Battery4520 mAh Battery5020 mAh Battery4820 mAh Battery5020 mAh B
attery4000 mAh Lithium Polymer Battery5000 mAh Lithium Polymer Battery4000 mAh Lithium Polymer Battery'
```

In [358…
```
ro = re.compile(r"\d+") # Character Class \d+
final_battery = ro.findall(str1)
print(final_battery)
```

```
['5000', '5000', '5000', '5000', '5000', '5000', '5000', '5000', '5000', '5000', '4820', '5000', '4820', '4820', '5000', '4520', '5020', '4820', '5020', '400
0', '5000', '4000']
```

In [360…
```
len(final_battery)
```

Out[360…
```
22
```

In [364…
```
data = {"Battery_in_mAh":final_battery}
BAT = pd.DataFrame(data)
BAT.head(2)
```

Out[364…

	Battery_in_mAh
0	5000
1	5000

Tabular Data Structure

In [463…
```
import pandas as pd
```

In [464…
```
data = {"PName":Names,
        "Ram_Rom":RAMS,
        "Stars":Stars,
        "Reviews":list_of_Reviews,
        "Ratings":Ratings,
        "Prices":Prices}
df = pd.DataFrame(data)
```

In [465…
```
df.head()
```

Out[465…

	PName	Ram_Rom	Stars	Reviews	Ratings	Prices
0	Redmi 9 (Sky Blue, 64 GB)	4 GB RAM | 64 GB ROM	4.2	814	14,288	9,085
1	Redmi 9 (Carbon Black, 64 GB)	4 GB RAM | 64 GB ROM	4.2	814	14,288	9,277
2	Redmi 9A (SeaBlue, 32 GB)	2 GB RAM | 32 GB ROM	4.3	481	8,481	7,339
3	Redmi 9A (Sea Blue, 32 GB)	3 GB RAM | 32 GB ROM | Expandable Upto 512 GB	4.3	353	5,698	7,844
4	Redmi 9A (Midnight Black, 32 GB)	2 GB RAM | 32 GB ROM	4.3	481	8,481	7,318

In [466…
```
# Concatenating Two DATAFRAME
frames = [df, Cam1, BAT]
mi = pd.concat(frames, axis="columns")
mi.head()
```

Out[466…

	PName	Ram_Rom	Stars	Reviews	Ratings	Prices	CAMERA	Battery_in_mAh
0	Redmi 9 (Sky Blue, 64 GB)	4 GB RAM | 64 GB ROM	4.2	814	14,288	9,085	13MP + 2MP | 5MP Front Camera	5000
1	Redmi 9 (Carbon Black, 64 GB)	4 GB RAM | 64 GB ROM	4.2	814	14,288	9,277	13MP + 2MP | 5MP Front Camera	5000
2	Redmi 9A (SeaBlue, 32 GB)	2 GB RAM | 32 GB ROM	4.3	481	8,481	7,339	13MP Rear Camera	5000
3	Redmi 9A (Sea Blue, 32 GB)	3 GB RAM | 32 GB ROM | Expandable Upto 512 GB	4.3	353	5,698	7,844	13MP Rear Camera	5000
4	Redmi 9A (Midnight Black, 32 GB)	2 GB RAM | 32 GB ROM	4.3	481	8,481	7,318	13MP Rear Camera	5000

In [467…
```
# Start index from 1 in DataFrame
mi.index += 1
```

In [468…
```
mi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 1 to 24
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   PName           24 non-null     object
 1   Ram_Rom         24 non-null     object
 2   Stars           24 non-null     object
 3   Reviews         24 non-null     object
 4   Ratings         24 non-null     object
 5   Prices          24 non-null     object
 6   CAMERA          22 non-null     object
 7   Battery_in_mAh  22 non-null     object
dtypes: object(8)
memory usage: 1.6+ KB
```

In [469…
```
# Assigning Value from the Original Source
mi.iloc[22:, 7] = 4000
```

In [470…
```
mi.tail()
```

Out[470…

	PName	Ram_Rom	Stars	Reviews	Ratings	Prices	CAMERA	Battery_in_mAh
20	Redmi Note 9 Pro (Glacier White, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	4.4	3	22,238	14,795	48MP + 8MP + 2MP + 2MP | 13MP Front Camera	4000
21	Redmi K20 (Carbon Black, 64 GB)	6 GB RAM | 64 GB ROM	4.5	784	50,659	22,999	13MP + 2MP | 8MP Front Camera	5000
22	Redmi Note 8 (Neptune Blue, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	4.4	926	52,089	13,999	48MP + 8MP + 2MP + 2MP | 13MP Front Camera	4000
23	Redmi 8A Dual (Sea Blue, 32 GB)	3 GB RAM | 32 GB ROM | Expandable Upto 512 GB	4.2	3	14,077	8,999	NaN	4000
24	Redmi Note 8 (Space Black, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	4.4	784	52,089	13,999	NaN	4000

In [471…
```
mi.isna().sum()
```

Out[471…
```
PName             0
Ram_Rom           0
Stars             0
Reviews           0
Ratings           0
Prices            0
CAMERA            2
Battery_in_mAh    0
dtype: int64
```

In [472…
```
# Reassigning The Colmns
mi = mi.reindex(columns=["PName","Ram_Rom","CAMERA","Stars","Reviews","Ratings","Prices","Battery_in_mAh"])
mi.head(2)
```

Out[472…

	PName	Ram_Rom	CAMERA	Stars	Reviews	Ratings	Prices	Battery_in_mAh
1	Redmi 9 (Sky Blue, 64 GB)	4 GB RAM | 64 GB ROM	13MP + 2MP | 5MP Front Camera	4.2	814	14,288	9,085	5000
2	Redmi 9 (Carbon Black, 64 GB)	4 GB RAM | 64 GB ROM	13MP + 2MP | 5MP Front Camera	4.2	814	14,288	9,277	5000

In [ ]: