# THIS CODE WILL SCRAPE THE NAME, PRICE AND SELLER FROM MANY PRODUCTS AND DO A PRICE COMPARISON OF THEM(LOWEST PRICE, HIGHEST PRICE, ETC).

In [ ]:

```

```

## Importing Libraries:

In [1]:

```
import time
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import requests
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from matplotlib import pyplot as plt
from selenium.common.exceptions import NoSuchElementException
from sklearn import linear_model
import sklearn.model_selection as ms
import sklearn.linear_model as lm
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

## Searching the product on web based on its name :

In [2]:

```
produto = input("What do you want to buy ?")

# Create a Chrome WebDriver instance
driver = webdriver.Edge()

# Navigate to the website
website_url = 'https://shopping.google.com/?nord=1'
driver.get(website_url)

# Find the search bar element by its HTML attribute (e.g., id or name)
search_bar = driver.find_element(By.ID,'REsRA')  # Replace with the actual ID

# Clear any existing text in the search bar (optional)
search_bar.clear()

# Send the text you want to enter
search_text = produto
search_bar.send_keys(search_text)

# Simulate pressing the Enter key to perform the search
search_bar.send_keys(Keys.RETURN)
```

```
What do you want to buy ?ergometric bike
```

## Scrapping data from all tabs :

In [3]:

```
# List to store the HTML contents from tabs
html_contents = []

# Loop to colect the content from all tabs
while True:
    # Colect the HTML content from the current tab
    html_content = driver.page_source
    html_contents.append(html_content)

    try:
        # Try to find and click in the next tab button
        next_tab_button = driver.find_element(By.XPATH,"""//*[@id="pnnext"]/span[2]""")
        next_tab_button.click()
    except NoSuchElementException:
        # If theres no more next tab button, break the loop
        break

# Close the driver
#driver.quit()

# Concatenates all the colected HTML contents in a single string
combined_html = '\n'.join(html_contents)

# Creates o Beautiful Soup object with the combined HTML content
soup = BeautifulSoup(combined_html, 'html.parser')
```

In [4]:

```
soup
```

### Searching for product name, price and seller in the HTML file :

In [6]:

```
def empty_row():
    print("                                                                               ")
    return

dados_produtos = []

produtos = soup.findAll("div", attrs = {"class" : "sh-dgr__content"})

for produto in produtos:


    # Product Title :

    product_title = produto.find("h3", attrs = {"class" : "tAxDx"})
    product_title.text


    # Product Price :

    product_price = produto.find("span", attrs = {"class" : "a8Pemb OFFNJ"})
    product_price.text


    # Seller :

    product_seller = produto.find("div", attrs = {"class" : "aULzUe IuHnof"})
    product_seller.text

    empty_row()



    print("Title:", product_title.text)
    print("Price:", product_price.text)
    print("Seller:", product_seller.text)

    dados_produtos.append([product_title.text, product_price.text, product_seller.text])





produtos_df = pd.DataFrame(dados_produtos, columns = ["Product", "Price", "Seller"])
```

```
Title: Bicicleta Ergométrica Spinning Ergometric Importada Com Nf Blend Shoop
Price: R$ 1.999,97
Seller: Mercado Livre

Title: Bike Pelegrin PEL-2311 Spinning Racing, Profissional
Price: R$ 1.403,47
Seller: Netshoes

Title: Spinning Bike Oneal Tp1000
Price: R$ 3.561,84
Seller: Magazine Luiza

Title: Bicicleta Spinning Schwinn Ic3 / Ic7 / 700IC
Price: R$ 6.300,00
Seller: Amazon.com.br - Seller

Title: Bicicleta Ergométrica Magnética Cycle C5 Indoor Movement
Price: R$ 4.427,28
```

## Visualizing data :

In [38]:
```
produtos_df
```

Out[38]:

| | Product | Price | Seller |
|---|---|---|---|
| **0** | Bicicleta Ergométrica Vertical Gallant Flow GB... | R$ 398,50 | Magazine Luiza |
| **1** | Bicicleta Ergométrica Vertical Gallant Trainer... | R$ 529,00 | Amazon.com.br - Seller |
| **2** | Contrate Montador de Bicicleta Ergométrica | R$ 199,00 | Outlet das Fabricas |
| **3** | Bicicleta Ergométrica Gallant Elite X Spinning... | R$ 1.267,20 | Netshoes |
| **4** | Bicicleta Ergométrica Dream Concept 550, 6 Fun... | R$ 523,26 | Carrefour |
| **...** | ... | ... | ... |
| **430** | Tensor Bicicleta Ergométrica BH-3800 Polimet | R$ 75,45 | Mercado Livre |
| **431** | Bicicleta Ergométrica Horizontal Podiumfit H90... | R$ 1.791,00 | PodiumFit |
| **432** | Bicicleta Ergométrica Exercício Perna Mini Bik... | R$ 284,00 | Shoptime |
| **433** | Sensor De Velocidade Esteira E Bicicleta Ergom... | R$ 61,20 | Mercado Livre |
| **434** | Tensor Bicicleta Ergométrica Nitro 4300 Polimet | R$ 32,80 | Mercado Livre |

435 rows × 3 columns

## Transforming columns :

In [39]:
```
# Removing the R$ :

produtos_df["Price"] = produtos_df["Price"].apply(lambda x: x.replace('R$', ''))
```

In [40]:
```
# Transforming the price column and converting to int :

produtos_df["Price"] = produtos_df["Price"].str.replace('.', '').str.split(',').str[0].astype(int)
```

```
C:\Users\amade\AppData\Local\Temp\ipykernel_13804\1948588733.py:1: FutureWarning: The default value of regex will change from True to False in a future
version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
  produtos_df["Price"] = produtos_df["Price"].str.replace('.', '').str.split(',').str[0].astype(int)
```

In [41]:
```
produtos_df.describe()

# Most expensive product
# avg product price
# cheapest product
```

Out[41]:

| | Price |
|---|---|
| **count** | 435.000000 |
| **mean** | 2650.767816 |
| **std** | 2821.410891 |
| **min** | 21.000000 |
| **25%** | 547.500000 |
| **50%** | 1670.000000 |
| **75%** | 3977.000000 |
| **max** | 19999.000000 |

## Seller that has the lowest price :

In [42]:

```
ind_min = produtos_df["Price"].idxmin()

min_price_seller = produtos_df["Seller"].loc[ind_min]

min_price_product = produtos_df["Product"].loc[ind_min]

print("The Seller", min_price_seller, "has the lowest price product :", min_price_product, "of R$",produtos_df["Price"].min(), "reais !!!!")
pular_linha()
```

```
The Seller Magazine Luiza has the lowest price product : Cinta Carga Freio Bicicleta Ergometrica Dream Fitness EX500 - Ansantos of R$ 21 reais !!!!
```

## Seller with the highest price :

In [43]:

```
ind_max = produtos_df["Price"].idxmax()

max_price_seller = produtos_df["Seller"].loc[ind_max]

max_price_product = produtos_df["Product"].loc[ind_max]

print("The Seller", max_price_seller, "has the biggest price product :", max_price_product, "of R$",produtos_df["Price"].max(), "reais !!!!")
pular_linha()
```

```
The Seller Facer has the biggest price product : Bicicleta Ergométrica Horizontal (recumbent Bike) First Ahead Sports of R$ 19999 reais !!!!
```

In [ ]: