# Web Scraping: SPACENET WEBSITE

## Title:

Web Scraping Project Report: Web Scraping Extracting Headphone Data from an Online Store Using Python.
Website: spacenet.tn

## Introduction:

In our web scraping project, we focused on extracting data from a website that specializes in selling headphones which is called spacenet.tn.

## Objectif:

The objective of the project is to extract relevant information such as headphone names, prices, specifications, and customer reviews from the website.

## Methodology:

To accomplish this task, we leveraged the power of Python as my primary tool. Using Python's web scraping libraries, such as Beautiful Soup and requests, We crafted a program that navigated through the website's pages and retrieved relevant information about the available headphones. We utilized the website's HTML structure to identify and extract key details such as headphone references, types, brands, color , availability and the prices. Python's flexibility and robust libraries allowed us to handle dynamic web pages and overcome challenges like handling pagination and dealing with different data formats. By employing Python's data manipulation capabilities, I organized the scraped data into a structured format, such as Excel, for further analysis or integration with other applications. This web scraping project enabled us to gain hands-on experience in data extraction, web parsing, and utilizing Python as a powerful tool for web scraping tasks.

## CODING

The purpose of this phase of the project: collecting the necessary data for the final project phase. Chosen Product: Wireless Headphones Scraped Website: **spacenet.tn**

## *1) Importing the necessary modules:*

```
import re # Import the 're' module, which provides support for regular_
import math as mt # module for mathematical functions
import pandas as pd # module for data manipulation and analysis
import requests # module for making HTTP requests
from bs4 import BeautifulSoup # module for parsing HTML documents
```

Now that we have imported the necessary modules, we can start using them in our code to perform various tasks, such as web scraping, data analysis, and more

## *2) Creating a list of URLs to scrape:*

First, we create a list of URLs for the search term ‘Écouteurs’ by looping through the page numbers 1 to 20 and appending the URL to the list. Next, we create a list of URLs for the search term ‘casques’ by looping through the page numbers 1 to 20 and appending the URL to the list. Finally, we create a list of URLs for the category ‘earbuds’ by looping through the page numbers 1 to 20 and appending the URL to the list.

```
#SPACENET
URLS = []
for i in range(1,20):
    url1 = 'https://spacenet.tn/328-casque-ecouteurs?type=ecouteurs&page=' + str(i)
    URLS.append(url1)
for i in range(1, 20):
    url2 = 'https://spacenet.tn/328-casque-ecouteurs?type=casques&page=' + str(i)
    URLS.append(url2)
for i in range(1, 20):
    url3 = 'https://spacenet.tn/328-casque-ecouteurs?type=earbuds&page=' + str(i)
    URLS.append(url3)
print(URLS)
```

## *3) Scrapping phase:*

```
# Initializing empty lists to store the extracted data
list_ref=[]
list_price=[]
list_name=[]
list_avail=[]
list_feat=[]

# Looping through the list of URLs and extracting data from each page
for u in URLS:
    response = requests.get(u) # sending an HTTP request to the URL
    if response.ok: # ensuring that the link is scrappable (status code = 200)
        soup = BeautifulSoup(response.text,"html.parser")
        title = soup.find('title') # extracting the page title
        lists = soup.find_all('section', class_='active_grid')
        for i in lists:
            r=i.find("div", class_="product-reference").text
            if r not in list_ref:
                p = i.find("span", class_="price").text
                n = i.find('h2', class_='product_name').text
                a = i.find("div", class_="product-quantities").text
                # adding the extracted data to their respective lists
                list_ref.append(r)
                list_price.append(p)
                list_name.append(n)
                list_avail.append(a)
```

Now we have extracted the data from all the URLs in the list, and stored it in separate lists.

## *4)Extracting more detailed characteristics from the features and product name lists:*

```
# Classifing each product
list_type = [] # Create an empty list to store the results
ln = len(list_name)
for i in range(ln):
    x = list_name[i].split()
    print(x)
    if x[0] in ["Casque", "Écouteur"]:
        list_type.append("Casque Bluetooth")
    else:
        list_type.append("EarBuds")
```

```
: # Extracting brand name of each Product
list_brand = [] # Create an empty list to store the results
ln = len(list_name)
for i in range(ln):
    x = list_name[i].split()
    if "Anker" in x:
        j = x.index("Anker")
        list_brand.append(x[j+1])
    elif "Apple" in x:
        j = x.index("Apple")
        list_brand.append(x[j+1])
    elif "Awei" in x:
        list_brand.append("Awei")
    elif "Belkin" in x:
        list_brand.append("Belkin")
    elif "BOROFONE" in x:
        j = x.index("BOROFONE")
        list_brand.append(x[j+1] + " " + x[j+2] + " " + x[j+3])
    elif "Celly" in x:
        j = x.index("Celly")
        list_brand.append(x[j] + " " + x[j+1] + " " + x[j+2] + " " + x[j+3])
    elif "Discovery" in x:
        j = x.index("Discovery")
        list_brand.append(x[j] + " " + x[j+1] + " " + x[j+2] + " " + x[j+3])
    elif "Haino Teko" in x:
        j = x.index("Haino Teko")
```

```
    elif "hama" in x:
        j = x.index("hama")
        list_brand.append(x[j] + " " + x[j+1] + " " + x[j+2])
    elif "Havit" in x:
        j = x.index("Havit")
        list_brand.append(x[j] + " " + x[j+1] + " " + x[j+2])
    elif "HAYLOU" in x:
        print('HAYLOU')
        j = x.index("HAYLOU")
        list_brand.append(x[j] + " " + x[j+1])
    elif "Hoco" in x:
        j = x.index("Hoco")
        list_brand.append(x[j] + " " + x[j+1])
    elif "Huawei" in x:
        print('huawei')
        j = x.index("Huawei")
        list_brand.append(x[j] + " " + x[j+1])
    elif "Iconix" in x:
        list_brand.append("Iconix")
    elif "ideus" in x:
        j = x.index("ideus")
        list_brand.append(x[j+1])
    elif "INFINIX" in x:
        list_brand.append("INFINIX")
    elif "Inkax" in x:
        list_brand.append("Inkax")
```

```
            list_brand.append("Inkax")
        elif "IPLUS" in x:
            list_brand.append("IPLUS")
        elif "Jabra" in x:
            list_brand.append("Jabra")
        elif "JBL" in x:
            list_brand.append("JBL")
        elif "Joyroom" in x:
            list_brand.append("Joyroom")
        elif "KSIX" in x:
            list_brand.append("KSIX")
        elif "Ledwood" in x:
            list_brand.append("Ledwood")
        elif "Lenovo" in x:
            list_brand.append("Lenovo")
        elif "XIAOMI" in x:
            j = x.index("XIAOMI")
            list_brand.append(x[j+2])
        elif "Nokia" in x:
            j = x.index("Nokia")
            list_brand.append(x[j+2])
        elif "OPPO" in x:
            j = x.index("OPPO")
            list_brand.append(x[j+2])
        elif "PHILIPS" in x:
            j = x.index("PHILIPS")
```

```
        elif "Samsung" in x:
            j = x.index("Samsung")
            list_brand.append(x[j+1])
        elif x[1] == "Bluetooth":
            list_brand.append(x[2])
        else:
            list_brand.append(x[1])
```

```
# Extracting the color of each product
list_colo = [] # Create an empty list to store the results
for e in list_name:
    x = e.split("-")
    if len(x) == 2:
        list_colo.append(x[1])
        print(list_colo)
    else:
        z = x[0].split()
        list_colo.append(z[-1])
# Some products do not have an indicated color.
for i in range(len(list_colo)):
    if list_colo[i] in ["Lipstick", "HP-03", "mémoire", "Pro(MXY72LL-A)"]:
        list_colo[i] = None
```

## 5)Creating Our Dataframe

```
# Create a dictionary
d={'Reference':list_ref,'type':list_type,'Brand': list_brand,'Color': list_colo,'Availablity':list_avail,'Price(DT)':list_price}
# Convert the dictionary to a pandas DataFrame
df = pd.DataFrame(d)
```

## 6)Exporting The Data Frame

```
df.to_excel("PROJECT.xlsx",index=False)
# Save the DataFrame as excel file
```

# Final result:

| Reference | type | Brand | Color | Availablity | Price(DT) |
|---|---|---|---|---|---|
| Réf : M61 | EarBuds | Hoco Avec | Noir | En stock | 3,500 TND |
| Réf : ZBW4354TY | Casque Bluetooth | intra-auriculaire | intra | En stock | 17,500 TND |
| Réf : BM32-WH | EarBuds | 1.2M | Blanc | En stock | 39,000 TND |
| Réf : BE-336RD | Casque Bluetooth | sans | 336 Rouge & Vert Fluo | En stock | 12,500 TND |
| Réf : 98824 | Casque Bluetooth | JBL | Beige | En arrivage | 275,000 TND |
| Réf : LP40-BK | Casque Bluetooth | Lenovo | Noir | En stock | 39,000 TND |
| Réf : BXATWS02 | Casque Bluetooth | Sans | KSIX BXATWS02 | En stock | 75,000 TND |
| Réf : BXTW02 | EarBuds | Ksix | BXTW02 | En stock | 95,000 TND |
| Réf : A3927011 | EarBuds | SoundCore | Noir | En stock | 199,000 TND |
| Réf : T0004K | EarBuds | Huawei FREEBUDS | Rouge | Sur commande | 299,000 TND |
| Réf : 97954 | EarBuds | JBL | Blanc | En stock | 475,000 TND |