LA ELABORACIÓN DE NUESTRO WEB SCRAPER

Gaona Amaya Silvia Alexa, León Morales Guadalupe.

Descargando las librerías necesarias y el web driver

1. El primer paso es descargar anaconda, esto facilitará la instalación de las librerías necesarias.
2. Instalar desde la consola de anaconda:
    - **--user selenium==3.141.0**
    - **pandas**
    - **urllib**
    - **matplotlib**
3. Descargar Google Chrome
4. Descargar un web driver adecuado para la versión de Chrome que tenemos en la computadora.

Comenzando el Proyecto: Definiendo la función

1. Para poder visualizar el código de una forma más ordenada, usaremos Jupyter Notebook, para poder utilizar JupyterLab
2. Una vez que estemos en JupyterLab, importamos las siguientes librerías:

```
import pandas as pd
from urllib.request import urlopen
import urllib.request
import requests
import time
from multiprocessing import Process, Queue, Pool
import threading
import sys
import re
import numpy as np
from datetime import date
from datetime import datetime
```

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import pandasql as ps
from IPython.display import display,HTML
import matplotlib.pyplot as plt
```

3. Definimos una función para comenzar con nuestro primer sitio web, que recibirá como argumento el producto que queremos buscar.
4. Asignamos la ruta donde se encuentra el web driver:

```
path ="C:\webscraperlya\chromedriver.exe"
```

5. Creamos la variable producto y la igualamos al argumento que recibirá la función.
6. Para el siguiente paso necesitamos abrir el sitio web donde buscaremos los productos y dentro de la barra de búsqueda indagaremos hasta encontrar cómo es que el sitio web busca el producto, copiaremos la url y reemplazaremos el nombre del producto por la variable producto.

```
url="https://bellisima.mx/search?type=product&options%5Bprefix%5D=last&options%5Bunavailable_products%5D=last&q="+producto+"&type=product&options%5Bprefix%5D=last&options%5Bunavailable_products%5D=last"
```

7. Usando el driver, abriremos la url

```
driver.get(url)
```

8. A continuación, “dormiremos” el equipo, esto para ganar tiempo mientras que el sitio termina de cargar.

```
time.sleep(15)
```

9. Mediante la función de inspeccionar la página web, buscaremos la clase que muestra toda la información de los productos, para poder usar la función de encontrar ítems mediante el nombre de la clase.

```
productos=driver.find_elements_by_class_name("product-item.boost-pfs-action-list-enabled")
```

10. Crearemos una lista donde se irán almacenando los nombres de los productos, pero usaremos una excepción, esto con la finalidad de que, si algún producto no tiene nombre, no afecte la longitud de nuestra lista.

    El proceso para conseguirlo será buscando la clase que contenga el nombre del producto.

```
lista_nombres=[]
    for i in range(0,len(productos)):
        try:
            lista_nombres.append(productos[i].find_elements_by_class_name("product-item-meta__title")[0].text)
        except:
            lista_nombres.append(np.nan)
```

11. El siguiente paso es análogo, pues será la asignación de los precios en una lista también.

12. Podemos repetir este proceso para cada uno de los datos que queramos obtener del producto (algo muy importante de mencionar es que estos deben ser datos visibles).

13. Una vez que tengamos todas las listas con la información que obtuvimos del sitio web, crearemos el dataframe.

    Para esto, asignaremos cada característica del producto en columnas del dataframe, y luego, en cada columna, añadiremos las listas con la información.

```
df_bellisima =pd.DataFrame(columns=["NOMBRE","PRECIO_1","PRECIO_2","MARCA","FECHA","AUTOSERVICIO"])
df_bellisima["NOMBRE"] = lista_nombres
df_bellisima["PRECIO_1"] = precio_1
df_bellisima["PRECIO_2"] = precio_2
df_bellisima["MARCA"] = lista_nombres
df_bellisima["FECHA"] = (date.today())
df_bellisima["AUTOSERVICIO"] = "BELLÍSIMA"
```

    En la penúltima línea (donde asignamos la fecha) utilizamos la función date.today() para que de forma automática se añada la fecha en que se realizó la consulta.

    Y en la última línea, igualamos toda la columna del dataframe al nombre del sitio web.

14. A partir de aquí, lo que hicimos fue limpiar la información del dataframe para mejorar su formato.

15. El último paso será definir el return de la función.

```
return df_bellisima
```

Segunda parte: Probando la función

1. Definiremos el número de variables según el número de productos que queramos buscar, esto con la finalidad de agilizar el cambio de productos, sin tener que modificar todo el código.

```
Producto_1="labial"
Producto_2="sombras"
Producto_3="rubor"
Producto_4="corrector"
Producto_5="brocha"
```

2. Asignamos el dataframe de cada producto a una variable.

```
df_be1=scrap_bellisima(Producto_1)
df_be2=scrap_bellisima(Producto_2)
df_be3=scrap_bellisima(Producto_3)
df_be4=scrap_bellisima(Producto_4)
df_be5=scrap_bellisima(Producto_5)
```

3. Una vez que verificamos que no haya errores al correr el código, podemos continuar con la creación de los dataframes definitivos.

Creando los dataframes definitivos

1. Concatenamos todos los dataframe del primer sitio en uno solo.
2. Reseteamos los índices del dataframe de Bellísima(el primer sitio).
3. Convertimos el dataframe en un archivo .csv para poder comprobar en un formato visual más agradable que se creó correctamente.

Creando el dataframe con la información de todos los sitios

1. En un solo dataframe concatenamos el dataframe de los tres sitios.
2. Reseteamos los índices.
3. Creamos un archivo .csv con la base de datos final.

Realizando nuestras consultas y creando gráficas

1. Utilizando pandasql, podremos realizar todas nuestras consultas como si estuviéramos usando SQL.
2. Aprovechando el paso anterior, podemos obtener el promedio de precios de cada producto en cada uno de los sitios web.
3. Para poder tener un acceso más fácil y rápido, introducimos todos los promedios en una sola base de datos.
4. Una vez que tenemos el dataframe de los promedios, será fácil y rápido crear las gráficas usando matploblib, donde comparamos cada uno de los productos en cada sitio y finalmente creamos una gráfica donde comparamos los precios de cada producto en todos los sitios web.
5. A continuación añadimos unos ejemplos de los resultados finales.