Web Scraping en Python: Beautiful Soup y Proceso

El **web scraping** es una técnica que permite extraer datos e información de una web. En esta y las próximas sesiones haremos una pequeña iniciación al web scraping con Python, utilizando para ello la librería Beautiful Soup.

Lejos de lo que te puedas imaginar y de que pienses que el web scraping es cosa de hackers, lo cierto es que el web scraping permite, por ejemplo, llevar a cabo un análisis SEO de una web, comprobar enlaces rotos, generar el sitemap de una página, vigilar a la competencia o estar al tanto de cambios en una web. Esta técnica también puede utilizarse en las primeras fases de un proyecto de Big Data o Machine Learning, en los que datos e información juegan un papel importantísimo.

En esta primera mini-sesión únicamente vamos a hacer dos cosas:

- Presentar a Beautiful Soup la herramienta que, al igual que ElementTree nos sirvió con XML, nos va a ayudar a procesar ficheros HTML
- Presentar el proceso de Web Scraping.

Qué es Beautiful Soup

Beautiful Soup es una librería Python que permite extraer información de contenido en formato HTML o XML. Para usarla, es necesario especificar un **parser**, que es responsable de transformar un documento HTML o XML en un árbol complejo de objetos Python. Esto permite, por ejemplo, que podamos interactuar con los elementos de una página web como si estuviésemos utilizando las herramientas del desarrollador de un navegador.

A la hora de extraer información de una web, uno de los parsers más utilizado es el parser HTML de lxml. Y que será el que usemos. Si has seguido las instrucciones del WarmUp ya tendrás instalados tanto lxml como beautifulsoup4 y requests. Si no las tienes deberás instalarlas antes de continuar.

De hecho, ahora lo que vamos a hacer es importar las librerías como vamos a hacer normalmente (aunque no vayamos a usarlas más en esta sesión):

```
import lxml
import requests
from bs4 import BeautifulSoup
```

Pasos para hacer web scraping

Estos son los pasos generales cuando abordamos este tipo de proyectos:

1. **Identificar los elementos de la página de los que extraer la información**: Las páginas web son documentos estructurados formados por una jerarquía de elementos. El primer paso para extraer información es identificar correctamente el elemento o elementos que contienen la información deseada. Para ello, lo más fácil es abrir la página en un navegador e inspeccionar el elemento. Esto se consigue haciendo clic con el botón derecho sobre el elemento en cuestión y pulsando sobre la opción Inspeccionar o Inspeccionar elemento (depende del navegador). Quédate con toda la información disponible asociada al elemento (como la etiqueta, o los atributos id y/o class) ya que, posteriormente, te hará falta para utilizarla en Beautiful Soup.

2. **Descargar el contenido de la página**: Para ello, utiliza la librería requests. El contenido de la respuesta, el que contiene la página en HTML, será el que pasemos posteriormente a Beautiful Soup para generar el árbol de elementos y poder hacer consultas al mismo.

3. **Crear la «sopa»**: El contenido de la página obtenido en el paso anterior será el que utilicemos para crear la «sopa», esto es, **el árbol de objetos Python que representan al documento HTML**. Para ello, hay que crear un objeto de tipo BeautifulSoup, al cual se le pasa el texto en formato HTML y el identificador del parser a utilizar:

```
import requests
from bs4 import BeautifulSoup
r = requests.get('http://unapagina.xyz')
soup = BeautifulSoup(r.text, 'lxml')
```

4. **Buscar los elementos en la «sopa» y obtener la información deseada**: El último paso es hacer una búsqueda en el árbol y obtener los objetos que contienen la información y datos que necesitamos. En este sentido se va a parecer a lo que hacíamos con los ficheros XML, pero con un poco más de elegancia.