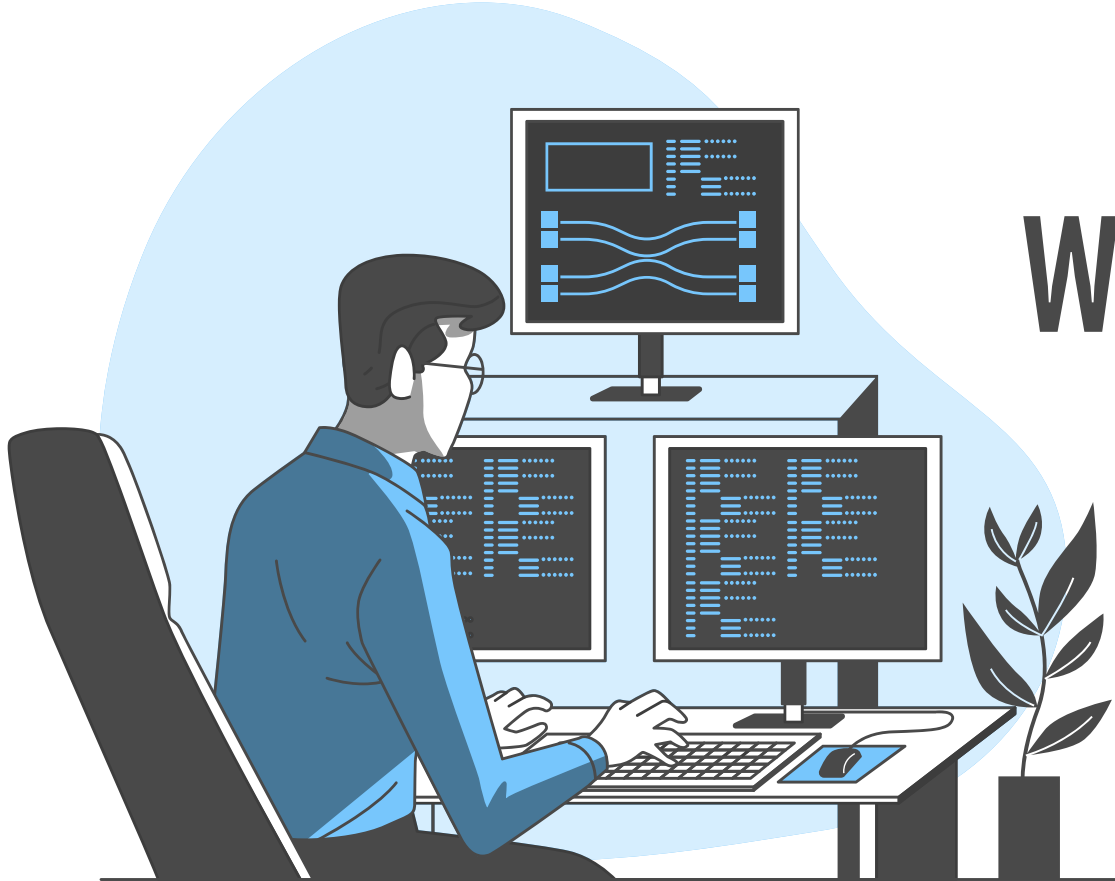
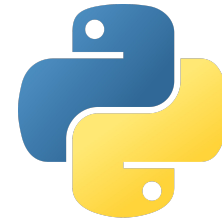


# Web Scrapping



# ¿Qué es el Web Scraping?

El web scraping es un proceso automatizado que extrae datos de sitios web. Se utilizan programas o bots para acceder al código HTML de las páginas y extraer información específica, que luego se puede almacenar en una base de datos o utilizar de otras formas. En resumen, es una forma de recopilar datos de internet de manera automática.



# Conceptos Básicos

## **Análisis de HTML:**

Los sitios web se crean con HTML, y los web scrapers necesitan comprender esta estructura para encontrar los datos que necesitan.

## **Solicitudes HTTP:**

Bibliotecas como *Requests* permiten que tu código Python envíe solicitudes a sitios web y reciba su contenido HTML.

## **Extracción de datos:**

Una vez que tengas el HTML, utiliza herramientas como *BeautifulSoup* para navegar y extraer los datos específicos que te interesan.




# Bibliotecas Clave



**Requests:** Simplifica la creación de solicitudes HTTP para obtener páginas web.

**BeautifulSoup:** Analiza HTML y XML, lo que facilita la navegación y la extracción de datos.

**Selenium:** Se utiliza para automatizar navegadores e interactuar con contenido dinámico (contenido cargado con JavaScript).



**Scrapy:** Un framework para crear raspadores web más complejos, especialmente para la extracción de datos a gran escala.



# Consideraciones

## **Legalidad:**

Consulte siempre el archivo robots.txt y las condiciones de servicio de un sitio web antes de realizar el scraping, ya que algunos sitios pueden restringirlo o prohibirlo.

## **Contenido dinámico:**

Si un sitio web utiliza JavaScript para cargar contenido, es posible que necesite Selenium para interactuar con la página y renderizar el contenido antes del scraping.



## **Limitación de velocidad:**

Tenga en cuenta la carga del servidor del sitio web y evite realizar demasiadas solicitudes en poco tiempo.

# ¿Qué es Requests?

Cuando haces scraping, lo primero que necesitas es **acceder al contenido HTML** de la página. Para ello utilizamos `requests`.

`requests` es una **librería de Python** que te permite **hacer peticiones HTTP** (como lo hace tu navegador cuando visitas una página), pero desde tu código. Es muy útil para:

- Acceder a páginas web.
- Descargar contenido (HTML, JSON, imágenes...).
- Interactuar con APIs.
- Obtener el código fuente de un sitio para luego analizarlo con `BeautifulSoup`.

# Requests – Ejemplo Sencillo

```
import requests

url = "https://example.com"
response = requests.get(url)

print(response.status_code)  # Muestra el código de respuesta (200 = OK)
print(response.text)        # Muestra el contenido HTML de la página
```

# ¿Qué es BeautifulSoup?

**Beautiful Soup** como una biblioteca de Python para extraer datos de archivos HTML y XML. Funciona con tu analizador (parser) favorito para proporcionar formas idiomáticas de navegar, buscar y modificar el árbol de análisis. Comúnmente ahorra a los programadores horas o días de trabajo.

## ¿Qué significa esto en términos sencillos?

- **Beautiful Soup** es una librería de Python.
- Sirve para leer el contenido de una página web (HTML o XML) y extraer datos específicos.
- Permite **navegar por el código HTML** como si estuvieras usando etiquetas y atributos de manera organizada.
- Es muy útil para **automatizar la recolección de información** de páginas web (por ejemplo, precios, titulares de noticias, descripciones de productos...).
- Funciona junto con un "parser", como `html.parser` o `lxml`, que se encarga de interpretar el HTML.



# Instalaciones

**Para instalar BeautifulSoup simplemente abre tu terminal y ejecuta:**

```
> pip install beautifulsoup4
```

**Para instalar Requests simplemente abre tu terminal y ejecuta:**

```
> pip install requests
```

# Beautiful Soup – Ejemplo Sencillo

```
from bs4 import BeautifulSoup

html = "<html><body><h1>Hola Mundo</h1></body></html>"
soup = BeautifulSoup(html, 'html.parser')

print(soup.h1.text)  # Resultado: Hola Mundo
```

Python



```
import requests
from bs4 import BeautifulSoup

# 1. Fetch the HTML
url = "http://example.com"
response = requests.get(url)
html_content = response.text

# 2. Parse the HTML
soup = BeautifulSoup(html_content, 'html.parser')

# 3. Extract the title (example)
title = soup.find('title').text
print(f"The title of the page is: {title}")

# 4. Extract all links (example)
for link in soup.find_all('a'):
    print(link.get('href'))
```

# Beautiful Soup – Ejemplos

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

# 1. Buscar por etiqueta y clase

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

```
descripcion = soup.find('p', class_='descripcion')
print(descripcion.text)
```

## 2. Obtener un atributo

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

```
enlace = soup.find('a', class_='enlace')
print(enlace['href']) # /link1
```

### 3. Buscar todos los elementos con cierto atributo

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

items = soup.find_all('li', attrs={'data-id': True})
for item in items:
    print(item['data-id'], item.text)

soup = BeautifulSoup(html, 'html.parser')
```

## 4. Usar selectores CSS (**select**, **select\_one**)

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

```
# Seleccionar por ID
titulo = soup.select_one('#titulo-principal').text

# Todos los <li> dentro del #lista
elementos_lista = soup.select('#lista li')
for el in elementos_lista:
    print(el.text)
```



## 5. Buscar por contenido de texto (usando función)

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')

dato_edad = soup.find('span', string=lambda x: 'Edad' in x)
print(dato_edad.text)
```

## 6. ¿Cómo acceder al segundo `<div>` de verdad?

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

```
segundo_div = soup.find_all('div')[1] # div con class="info"
```

```
segundo_span = segundo_div.find_all('span')[1]
print(segundo_span.text) # Resultado: "Edad: 30"
```

## 6. ¿Cómo acceder al segundo `<div>` de verdad?

```
html = """
<html>
  <head><title>Mi Página</title></head>
  <body>
    <div class="contenedor">
      <h1 id="titulo-principal">Bienvenido</h1>
      <p class="descripcion">Este es un ejemplo con <a href="/link1" class="enlace">un enlace</a>.
      <ul id="lista">
        <li data-id="1">Elemento 1</li>
        <li data-id="2" class="especial">Elemento 2</li>
        <li data-id="3">Elemento 3</li>
      </ul>
      <div class="info">
        <span class="dato">Nombre: Juan</span>
        <span class="dato">Edad: 30</span>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')

print(soup.find_all('div')[1].find_all('span')[1].text) # Edad: 30
```