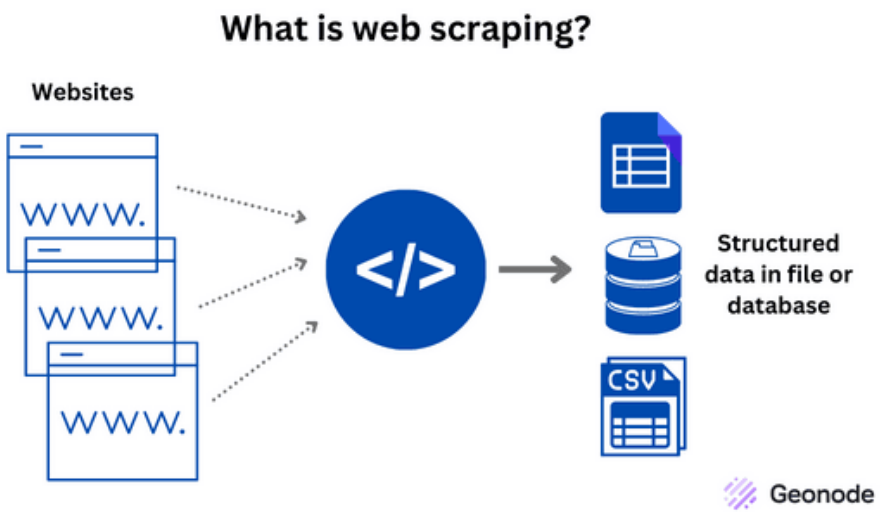

WEB SCRAPING

¿Qué es el Web Scraping?



Web Scraping:

Extracción Automatizada de Datos

Es el proceso de extraer información valiosa de páginas web de forma automática y estructurada

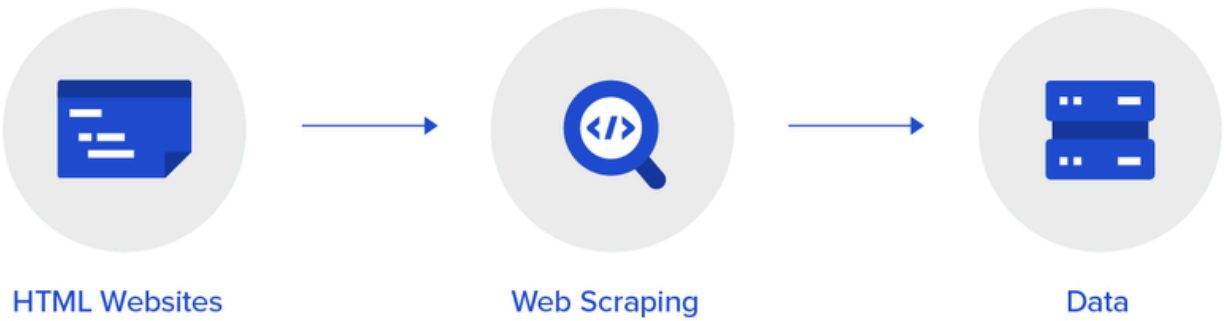
¿Qué es el Web Scraping?

Automatización
Proceso que extrae información de páginas web de forma automática.

Estructuración
Convierte contenido visual en datos estructurados y utilizables.

Alternativa
Solución cuando los sitios web no ofrecen APIs para acceder a sus datos.

¿Qué es el Web Scraping?



Aplicaciones Prácticas

Datos Públicos
Extracción de información de interés público desde sitios gubernamentales o educativos.

Monitorización
Seguimiento de precios, noticias y productos en tiempo real.

Análisis
Entrenamiento de modelos, investigación académica y análisis de mercado.

PROS Y CONTRAS DEL WEBSCRAPING

- VIOLAR LOS TÉRMINOS DE USO
- COMPLEJIDAD
- PÁGINAS CAMBIAN
- MANTENIMIENTO Y VALIDACIÓN

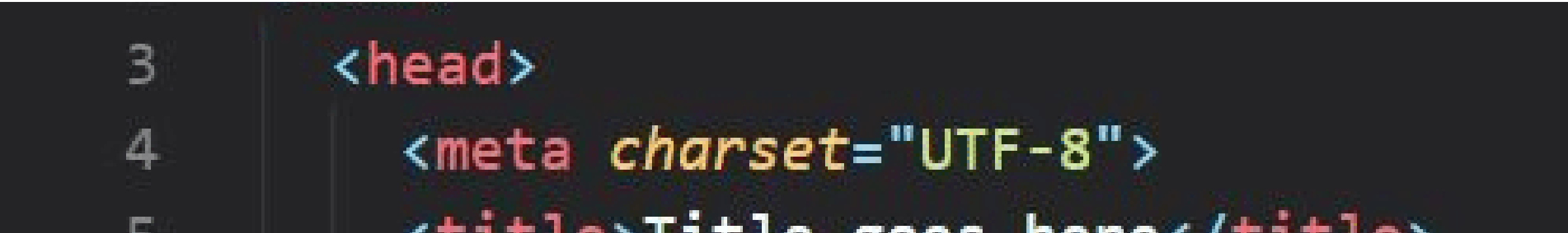
- ANÁLISIS DE COMPETENCIA
- ESCALABLE
- NO REQUIERE ACUERDOS
- AHORRA TIEMPO

HTML: El Lenguaje de la Web

Definición
HyperText Markup Language es el lenguaje base para estructurar páginas web.

Elementos
Define títulos, párrafos, tablas, imágenes y otros componentes visuales.

Jerarquía
Organiza el contenido de forma jerárquica formando el DOM (Document Object Model).

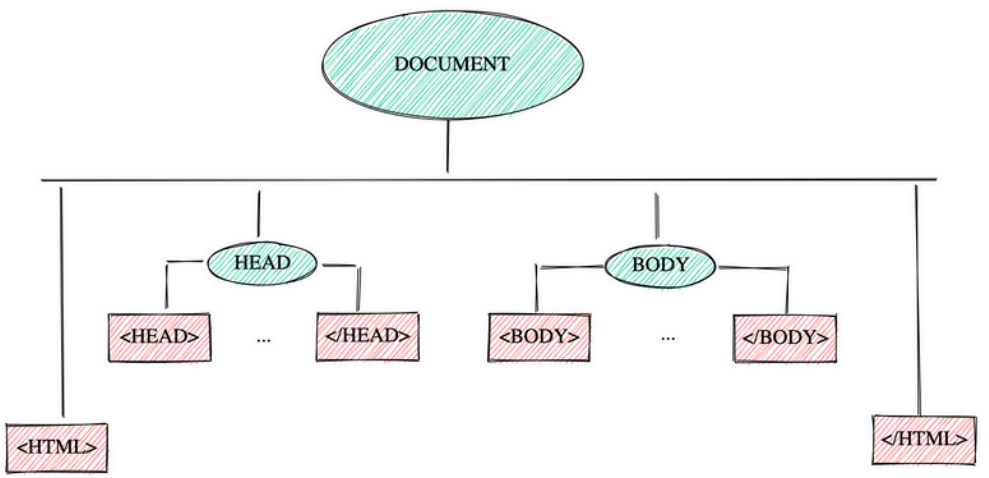


ESTRUCTURA

Título Principal

Este es un párrafo.

Enlace



Declaración

Define el tipo de documento.

Estructura

Organiza el contenido en secciones.

Elementos

Etiquetas que definen cada componente.

Elementos Básicos - TAGS

`<html>`: Etiqueta raíz que contiene todo el contenido de la página web.

`<head>`: Sección que contiene metainformación de la página, como el título, los estilos, los scripts, etc.

`<body>`: Sección que contiene el contenido visible de la página.

`<h1>` a `<h6>`: Etiquetas de encabezado, que indican la importancia de un texto dentro de la estructura de la página.

`<p>`: Etiqueta de párrafo, que define un bloque de texto.

`<a>`: Etiqueta de enlace, que permite crear hipervínculos para navegar a otras páginas o recursos.

`<img>`: Etiqueta de imagen, que permite incrustar imágenes en la página.

`<ul>`: Etiqueta de lista no ordenada.

`<ol>`: Etiqueta de lista ordenada.

`<li>`: Etiqueta de elemento de lista, que se usa dentro de `<ul>` o `<ol>`.

`<div>`: Etiqueta de división, que permite agrupar contenido para aplicar estilos o lógica a través de CSS o JavaScript.

Los elementos básicos de HTML son las **etiquetas** (tags) que definen la estructura y el contenido de una página web. No son únicos.
Estas etiquetas, como <h1>, <p>, <a>, <img>, <ul>, etc., ayudan a organizar la información en diferentes tipos de contenido.

Estructura del árbol

Etiqueta de apertura: Indica el inicio del elemento (ej: `<p>`).

Contenido: El texto o datos que aparecerán en la página (ej: "Hola mundo").

Etiqueta de cierre: Indica el final del elemento (ej: `</p>`).

Atributos (opcionales): Proporcionan información adicional sobre el elemento, como `class`, `id`, `src`, etc.

```
<!DOCTYPE html>
<html>
<head>
    <title>Título de la página</title>
</head>
<body>
    <h1>Título principal</h1>
    <p>Este es un párrafo de texto.</p>
    <a href="https://www.ejemplo.com">Enlace a una página</a>
    <img src="imagen.jpg" alt="Descripción de la imagen">
</body>
</html>
```

Un documento HTML sigue una estructura en forma de árbol: los elementos pueden tener otros elementos "dentro" - es decir, entre su etiqueta de **apertura y cierre**.

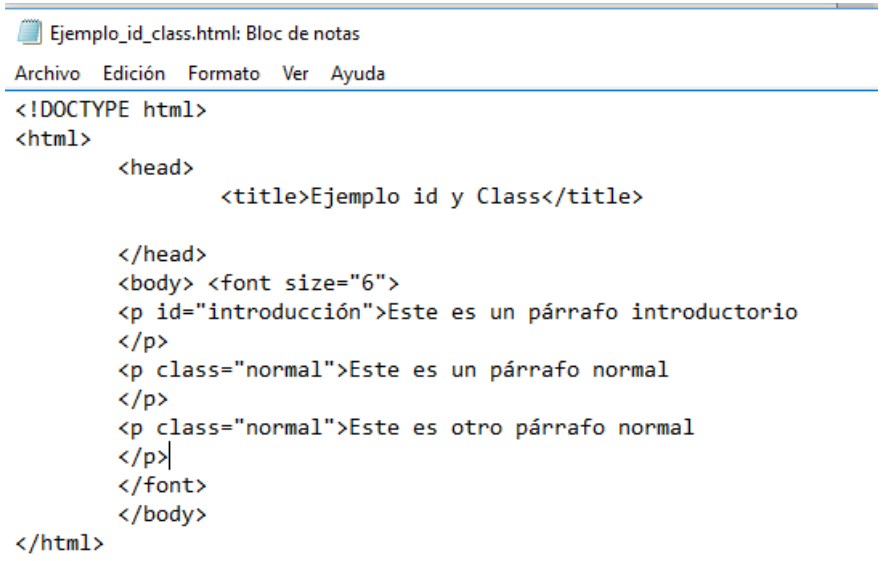
Cuando están "dentro" de un elemento diremos que son hijos de ese elemento.

Atributos

Hay dos atributos que merecen especial atención:

- El atributo id: a menos que el creador del sitio haya roto las convenciones básicas, los id suelen ser únicos. Eso los convierte en los mejores atributos para localizar datos en una página. Si descubres que la información que quieres recolectar está en un elemento con id, tu trabajo será MUY FÁCIL. Pero malas noticias: eso no ocurre muy a menudo.

- El atributo class: se usa frecuentemente para aplicar estilos a múltiples elementos.



El DOM: Estructura Jerárquica

Árbol de elementos

Representa la estructura del HTML de forma jerárquica.

Navegación

Permite recorrer y acceder a los elementos del documento.

Manipulación

Facilita la extracción precisa de información.

Selectores

Acceso por etiquetas, clases, ID o atributos.

```
<html>
    <head>
        <title> </title>
    </head>
    <body>
        Here your content will come.
    </body>
</html>
```

LINK

EJEMPLOS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>World of wines</title>
</head>
<body>
    <div id="container">
        <nav>
            <ul>
                <li><a href="#">France</a></li>
                <li><a href="#">Italy</a></li>
                <li><a href="#">Spain</a></li>
                <li><a href="#">California</a></li>
                <li><a href="#">South-Africa</a></li>
            </ul>
        </nav>
    </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <title>Page Title</title>
</head>

<body>
    <h2>Heading Content</h2>
    <p>Paragraph Content</p>
</body>

</html>
```

https://www.w3schools.com/tags/tag_doctype.asp

Ejercicio

```
* html_doc = """
<!DOCTYPE html>
<html>
<head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>
<p class="story">...</p>
</body>
</html>
"""
```

Proceso del Web Scraping

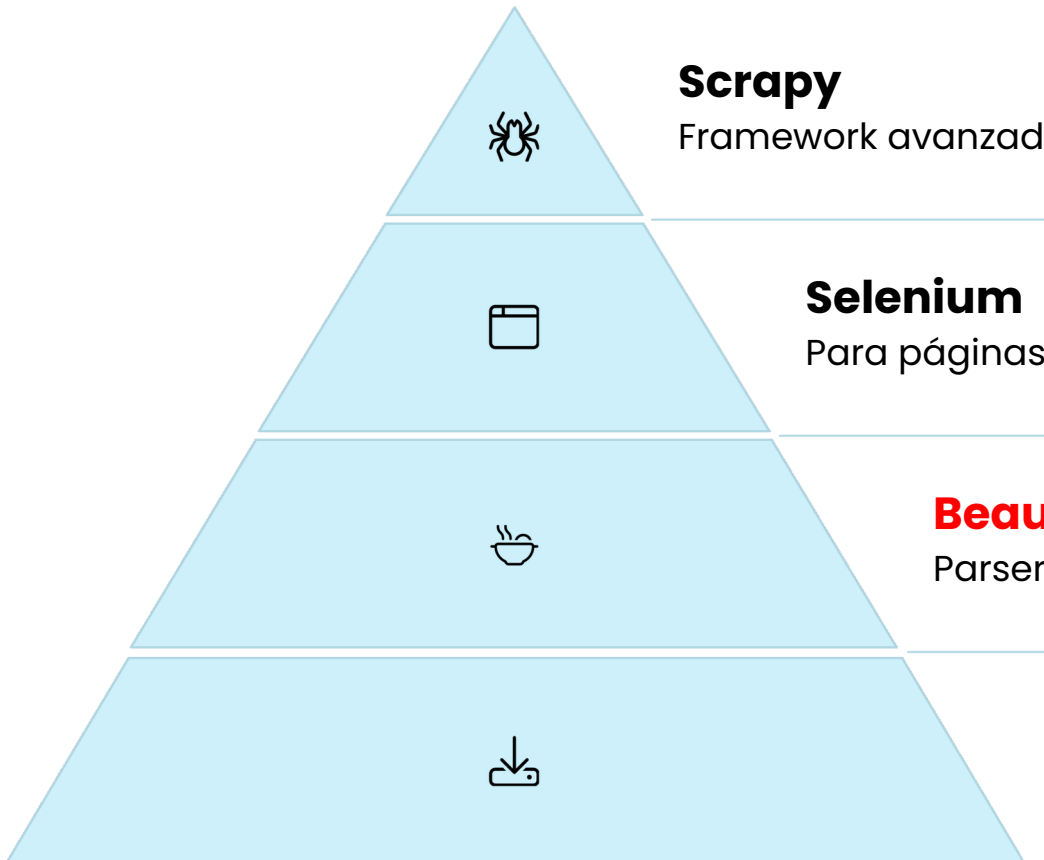
Solicitud HTTP
Enviar petición al servidor web usando bibliotecas como requests.

Parseo HTML
Analizar el código HTML recibido con herramientas como BeautifulSoup.

Extracción
Identificar y extraer los datos deseados mediante selectores.

Almacenamiento
Guardar la información en formatos como CSV, JSON u otros.

Herramientas Esenciales



Scrapy
Framework avanzado para scraping a gran escala

Selenium
Para páginas con contenido dinámico

BeautifulSoup y lxml
Parsers de HTML

Requests
Biblioteca para solicitudes HTTP

Librerías

Beautiful Soup

- User friendly
- Popular (lots of tutorials stackoverflow posts...)
- Slow

Scrapy

- Fast
- Requires no dependencies
- Not user friendly (set up pains)

Selenium

- Versatile (also works for testing)
- Can scrape JavaScript content
- Difficult
- Slow

Selección de Elementos HTML

Método	Ejemplo	Descripción
find()	soup.find('h1')	Encuentra el primer elemento que coincida
find_all()	soup.find_all('div', class_='item')	Encuentra todos los elementos que coincidan
select()	soup.select('div#main > ul > li')	Usa selectores CSS para encontrar elementos

Ejemplo Práctico de Código

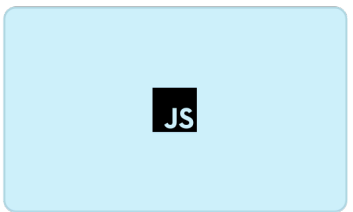
Importar bibliotecas
Cargar las herramientas necesarias.

Obtener página
Descargar el contenido HTML.

Extraer datos
Buscar y procesar la información.

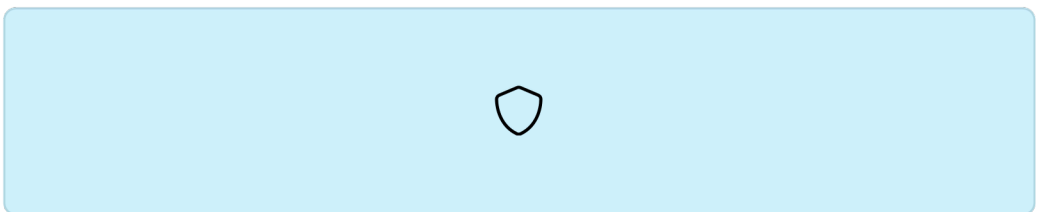
```
import requests from bs4 import BeautifulSoup

url = 'https://example.com'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

titulos = soup.find_all('h2')
for titulo in titulos:
    print(titulo.text)
```

Desafíos



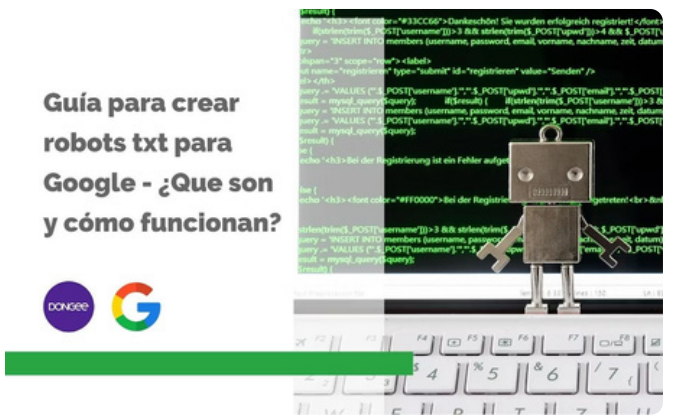
JavaScript dinámico
Contenido generado dinámicamente requiere Selenium.

Cambios estructurales
Modificaciones en el HTML pueden romper el scraper.



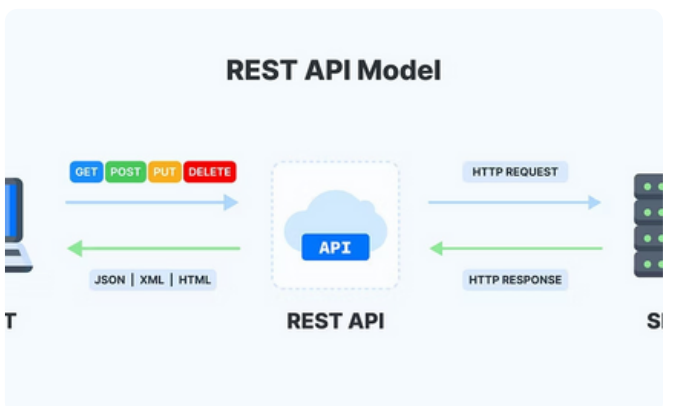
Limitaciones y bloqueos
Restricciones de velocidad o bloqueos por IP.

Ética y Alternativas



Consideraciones Éticas

Revisar robots.txt, respetar términos de uso y tiempos de espera. No recopilar datos personales.



APIs Oficiales

Interfaces programáticas oficiales que proporcionan acceso estructurado a los datos.

Datos Abiertos

Fuentes como Open Data, feeds RSS o JSON que ofrecen información ya estructurada.

Reglas de oro para recordar

- La gran mayoría de las etiquetas deben abrirse (<etiqueta>) y cerrarse (</etiqueta>), con la información del elemento (como un título o texto) ubicada entre las etiquetas.

- Al usar múltiples etiquetas, deben cerrarse en el orden inverso al que fueron abiertas. Por ejemplo:

```
<strong><em>¡Esto es realmente importante!</em></strong>
```

CHROME INSPECTOR

Ctrl+Shift+C shortcut para Windows
Command+Shift+C para Mac