WEB SCRAPING

¿Qué es el web scraping?

Es extraer información de internet, un sitio web, a la cual luego le podemos dar diversos usos.

Esto es altamente usado en aplicaciones como trivago donde se compara el precio de hoteles, o en empresas de e-comerce que quieren ver el precio que ofrecen vs. el precio que ofrece su competencia.

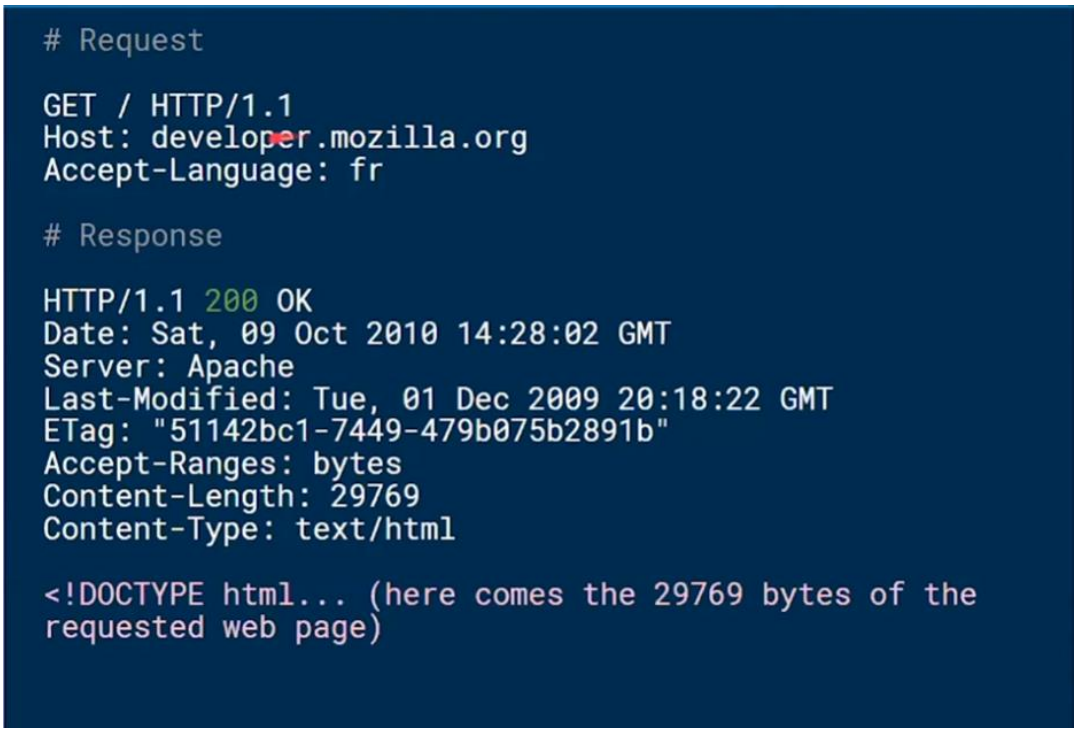
Google es otra de estas empresas que realiza constantemente web scraping para saber que contiene una pagina web y poder mostrarlo en las búsquedas.

Para esto haremos uso de la librería 'request' que nos permite controlar HTTP, y la librería 'BeautifulSoup' que es para extraer información de un documento HTML.

HTTP:

Como definición rápida son las reglas con las cuales dos sistemas se comunican en internet.

Protocolo con el cual una computadora se conecta con un servidor, la computadora hace una petición y el servidor responde a esta.



Explicación request:

```
GET / HTTP/1.1
```

Traer / protocolo / versión protocolo

```
Host: developer.mozilla.org
```

Página web de donde extraer info: facebook.com

```
Accept-Language: fr
```

Lenguaje de la info: spanish

Explicación response:

```
HTTP/1.1 200 OK
```

Protocolo / versión protocolo / código de respuesta

```
Date: Sat, 09 Oct 2010 14:28:02 GMT
```

Fecha de la respuesta: ---

```
Server: Apache
```

Servidor que respondió: ---

```
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
```

Ultima fecha en la que se modificó esta respuesta: ---

```
ETag: "51142bc1-7449-479b075b2891b"
```

Cache: ---

```
Accept-Ranges: bytes
```

En que vamos a aceptar los bits de información recibidos: ---

```
Content-Length: 29769
```

Cuantos bytes de información: ---

```
Content-Type: text/html
```

Tipo de dato que contiene la respuesta del servidor: ---

HTML:

Leguaje que permite definir la estructura de una página web.

```html
<!DOCTYPE HTML>
<HTML>

<HEAD>

<TITLE> EDITORES DE TEXTO WYSIWG</TITLE>

<META name="description" content="ejemplos de editores de texto WYSIWG">
<META name="keywords" content="editores de texto WYSIWG">

</HEAD>

<BODY>

</BODY>

</HTML>

```

<script> Código script ejecutable; JavaScript.

<inframe> Pagina web dentro de otra página web.

<meta> Información de los metadatos.

robots.txt:

Este archivo tiene los limites que permite un sitio web de web scraping; lo podremos encontrar en la raíz de un sitio web.

Este archivo es importante ya que establece un límite a la información que los programadores pueden o no tocar.

Sitio web / robots.txt

```
User-Agent: *
Allow: /conf/*
Allow: /conf-og/*
Disallow: /*/*/concepto/*/*/material/
Disallow: /login/facebook/
Disallow: /login/twitter/
Disallow: /*/*/live/
Disallow: /*/*/%7B%7Burl%20absolute=/
Disallow: /*/*/add_contribution/
Disallow: /mi-suscripcion/
Disallow: /r/
Disallow: /clases/*/nuevos_materiales/
Disallow: /kit-ui/
Disallow: /ui/
Disallow: /sfotipy/
Disallow: /streaming/*
Disallow: /payments/*
Disallow: /*/add_review/
Disallow: /*/save/
Disallow: /adquirir/*
Disallow: /comentario/
Disallow: /comment/
Disallow: /comprar/
Disallow: /precios/*/
Disallow: /yearly-stats-share/
Disallow: /courses/
Disallow: /historias/
Disallow: /becas-fb/
Disallow: /testimonios/

Sitemap: https://platzi.com/sitemap.xml
```

Explicación robots.txt:

```
User-Agent: *
```

Identificación de una computadora al hacer una petición a una web: se permite a cualquier computadora.

```
Allow: /conf/*
```

Que directorios puede visitar el user-agent: ---

```
Disallow: /*/*/concepto/*/*/material/
```

Directorios que no puede visitar mi user agent: ---

Xpath: Tipos de nodos

Xpath es un lenguaje que nos permite movernos entre nodos para poder extraer la información deseada.

Un nodo es una etiqueta con su contenido.

Xpath: Expresiones

$x(‘/’) –> Selección de todo el documento.

$x(‘/html’) –> Selección del nodo html.

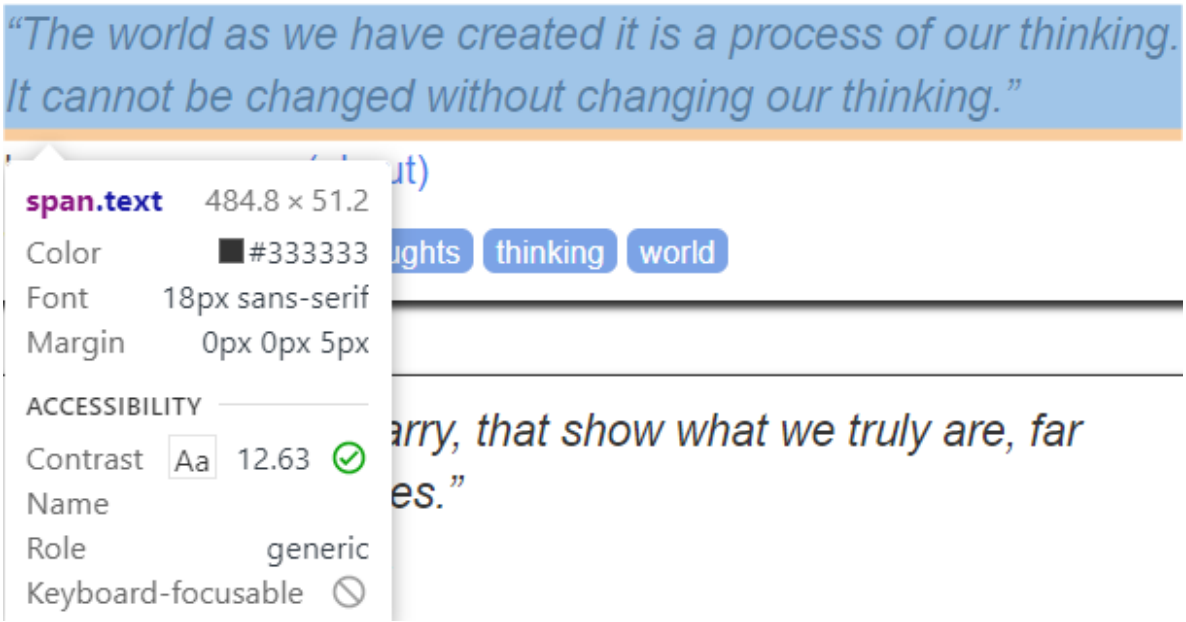
$x(‘//h1’) –> Saltar camino y selección de nodo h1.

$x(‘//h1/a/text()’) –> Texto que contiene el nodo h1.

$x(‘//span/..’) –> Acceder a todos los nodos padre de span.

$x(‘//span/@class’) –> Traer todos los atributos tipo class de todos los span.

Xpath: Predicados



Al inspeccionar la página, concretamente la información que queremos obtener se ve que se encuentra en el folder de ‘span’ y la clase ‘text’.

$x(‘//span[@class = ‘text’]/text()’)

Xpath: Wilcards

$x(‘//span/*’) –> Traer todos los nodos que se encuentran después de span.

$x(‘//*’) –> Traer todos los nodos y todos los atributos de los nodos.

Xpath: In-text search

$x('//small[@class = "author" and starts-with(.,"A")]/text()') –> Búsqueda nombre autores que empiezan por ‘A’.

$x('//small[@class = "author" and contains(.,"Ro")]/text()') –> Búsqueda nombre autores que contengan las letras ‘Ro’.

$x('//small[@class = "author" and ends-with(.,"t")]/text()') –> Búsqueda nombre autores que terminen por ‘t’.

$x('//small[@class = "author" and ,matches(.,"A.*n")]/text()') –> Búsqueda nombre autores que empiezan por ‘A’ y terminan por la letra ‘n’.

Xpath: Axes

$x(‘/html/body/div/.’) o $x(‘/html/body/div/self::div’) –> Traer al nodo en sí mismo

$x(‘/html/body/div/child::div’) –> Trae a los nodos hijos del div

$x(‘/html/body/div/descendant::div’) –> Trae a todos los nodos de los niveles inferiores del div

$x(‘/html/body/div/descendant-or-self::div’) –> Trae a todos los nodos de los niveles inferiores del div y al div

*Cuando haya múltiples encabezados con el mismo nombre se puede recurrir a [n], n siendo el número del encabezado que se desea. *

*Para poder imprimir los atributos de un nodo se usa ‘x=> x.values’ *