

Datapalooza 2023

Web Scraping

Pontificia Universidad Católica de Chile

Clase 1: Introducción al Web Scraping

- Recomendaciones iniciales
- Introducción
- Cuestiones éticas y legales
- Introducción a HTML
- rvest
- Actividad
- RSelenium

Recomendaciones Iniciales

Informaciones

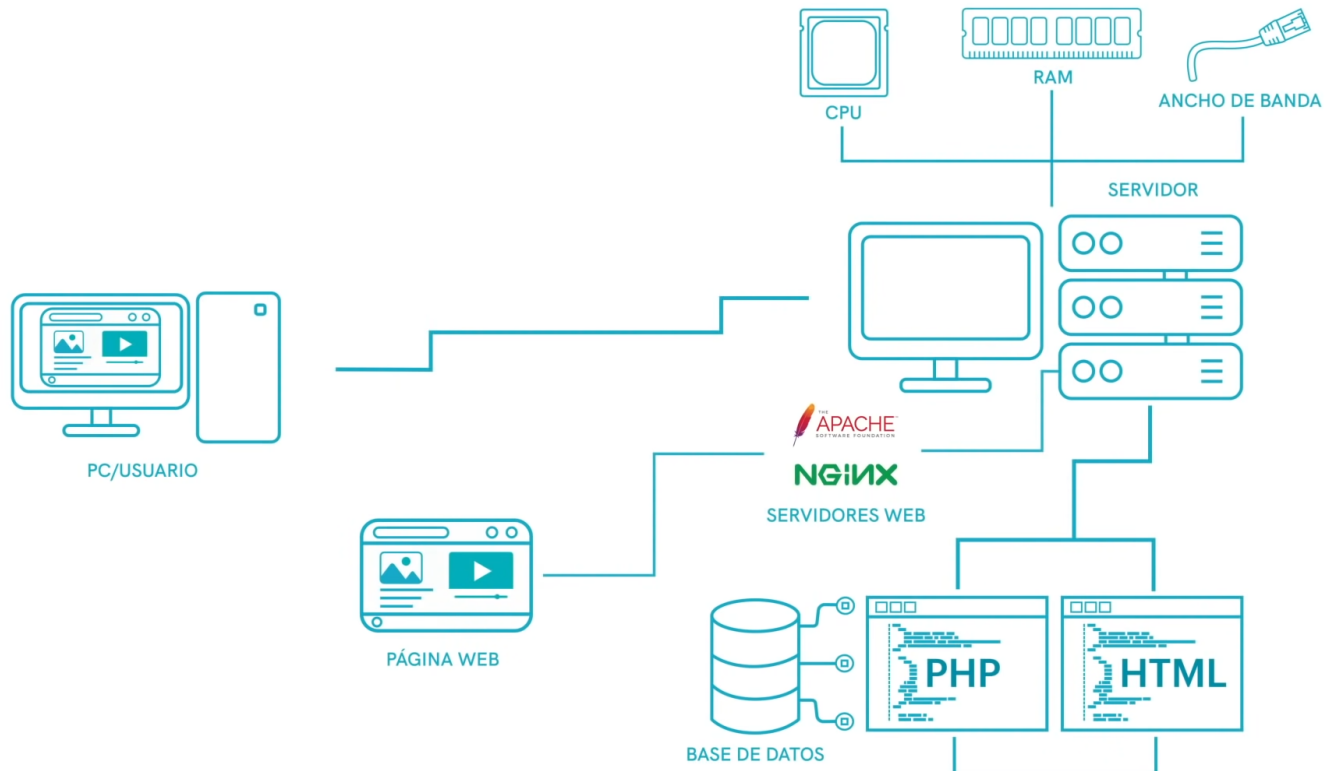
- Usaremos un script de R compartido para que puedas seguir la ejecución del código. Los scripts se compartirán en la siguiente **carpeta**.
- Pueden seguir al instagram oficial del diplomado UC en @DataScienceUC y etiquetar en Twitter con el hashtag #Datapalooza
- Pausa programada a las 11:30hrs de 15 minutos.



Introducción

¿Cómo funciona una página web?

- Realiza una consulta al servidor web
- El servidor determina la página web que quiere consultar
- Se envía la información en un formato HTML



¿Qué es Web Scraping?



Cuestiones éticas y legales

Problemas para el sitio WEB

Los propietarios de sitios WEB, justificadamente intentan protegerse de sistemas de web scraping, ya que estos pueden incurrir en problemas para ellos, por ejemplo:

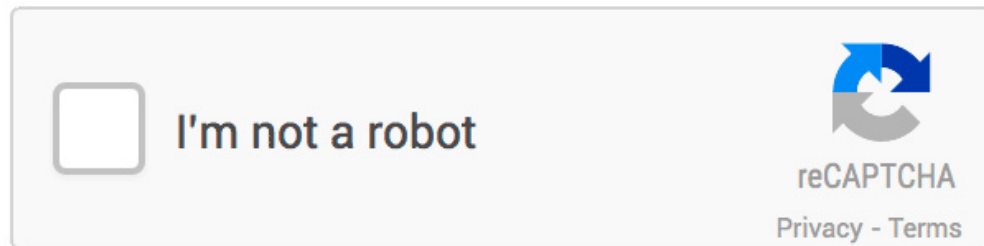
- El robo de información puede otorgar al 'ladrón' ventaja competitiva
- El sitio web puede fallar, debido a que el web scraping de forma masiva puede generar un ataque DDoS
- Entre otros



Métodos de prevención del web scraping

Algunos sistemas para protegerse de web scraping son:

- **crawler anti-scraping:** Estudia el número y frecuencia en que una misma IP realiza consultas al sitio web, en el momento en que se detecte un sistema extraño esta IP será bloqueada.
- **reCaptcha:** Sistema de detección de robot diseñado por Google.
- **User-Agent:** es un encabezado del sitio web para identificar como visita al usuario. Se usa información del sistema operativo, versión, tipo CPU, etc.



Directrices para un web scraping ético y responsable

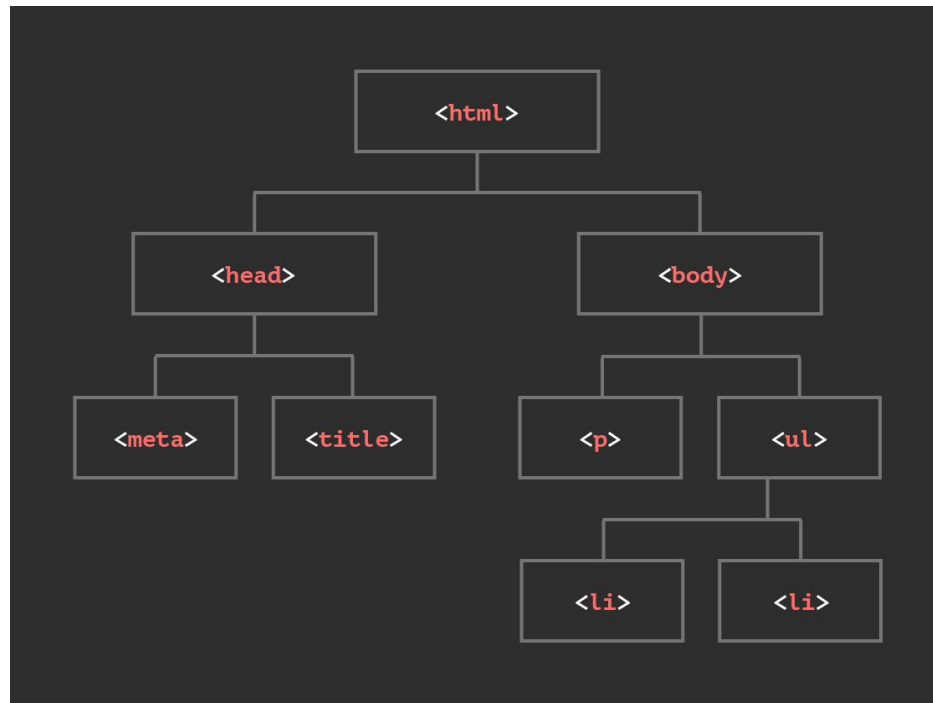
Algunas directrices para no ser un web scraper malvado son:

- Preferir las API sobre el Web Scraping siempre que sea posible
- Leer los terminos y condiciones para saber si la página web esta abierta a realizar web scraping o sobre el uso de los datos web scrapeados
 - Amazon
 - Wikipedia
- Establecer un tiempo entre consultas adecuado. El tiempo entre ejecución puede estar publicado en los archivos robot.txt de la páginas web, en caso contrario puede esperar al menos 10-15 segundos.

Introducción a HTML

¿Qué es el HTML?

- HTML es el lenguaje estándar para la creación de páginas web
- Describe la estructura de la página web
- Le indica al explorador que y como mostrar el contenido
- Las etiquetas de HTML indican el tipo de contenido, sea: párrafos, títulos, vínculos, etc.



¿Cómo funciona HTML?

```
<!doctype html>
<html>
  <body>
    <div class="article">
      <h2 class="title"> Title here </h2>
      <ul class="attribution">
        <li class="source"> Mizfa.com </li>
        <li class="pubdate"> 28 Aban </li>
      </ul>
      <p class="description">Content here ...</p>
    </div>
  </body>
</html>
```



Anatomía de HTML

Las etiquetas son bloques de contenido de la forma:


`<etiqueta> contenido </etiqueta>`

Estas pueden contener texto u otras etiquetas, creando un sistema de árbol. Además, las etiquetas contienen atributos los cuales agregan atributos a la etiqueta, como lo pueden ser: class, id, name, placeholder, etc.

Anatomía de un elemento de HTML

The diagram illustrates the structure of an HTML element: `<p class="saludo">Hola mundo!</p>`. Brackets are used to identify the different parts: the opening tag `<p` is labeled 'Etiqueta de apertura'; the attribute `class="saludo"` is labeled 'Un atributo y su valor'; the text `Hola mundo!` is labeled 'Contenido de texto encerrado'; and the closing tag `</p>` is labeled 'Etiqueta de cierre'.

Estructura de un HTML



```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <title>My first web page</title>
5     <link rel = "stylesheet"
6       type = "text/css"
7       href = "styles.css" />
8   </head>
9   <body>
10    <h1>Hello, world!</h1>
11    <script src="/js/script.js"></script>
12  </body>
13 </html>
```


Etiquetas básicas

- `<div>` división de información dentro del contenido.
- `<a>` para enlaces.
- `` para poner el texto en negrita.
- `
` para saltos de líneas.
- `<h1>` ... `<h6>` para títulos.
- `` para listas ordenadas, `` para listas desordenadas, `` para elementos dentro de una lista.
- `<p>` para párrafos.
- `` para estilos de una parte de texto.

Referencias

rvest para el web scraping

rvest



rvest es un paquete para realizar Web Scraping a página web de forma estática (sin interacción con esta) inspirado en [beautiful soup](#) y [RoboBrowser](#)

Las principales funciones de rvest son:

- **read_html**: Extrae la información HTML de la página consultada.
- **html_node/html_element**: Extrae la información HTML del primer mach realizado, la función recibe una consulta en formato css o xpath.
- **html_nodes/html_elements**: Extrae la información HTML de todas las coincidencias con la consulta realizada.
- **html_attr**: Extrae la información del atributo del node consultado.
- **html_table**: Extrae tablas de la página web consultada.
- **html_text2**: Extrae el text del node/página consultada.

Referencias:

- Selector css: [link](#)
- Selector xpath: [link](#)

Actividad 1

Actividad

Utilizando rvest extraiga la información de de los productos en promoción del día en la página web de **mercado libre**. Para ello se realizarán los siguientes pasos:

1. Extraer el número de páginas que tiene mercado libre en su sección de promoción del día.
2. Obtener los siguientes atributos de los productos de la pagina 1:
 - Nombre
 - Precio sin descuento
 - Precio con descuento
 - Envío gratis
 - Envío full
 - Vendedor
 - URL
3. Crear una función que extraiga la información de todos los productos en promoción del día.

RSelenium

RSelenium

Selenium Webdriver es un herramienta que nos permite automatizar pruebas de interfaz de usuario (UI), mediante una conexión realizada por Java.

Para establecer una conexión es requerido que se tenga instalado:

- Navegador web
- Webdriver correspondiente
- Java

Comandos útiles:

- `navigate()` inicia la página web seleccionada.
- `findElement()` busca un elemento xpath o css.
- `$getElementText()` extrae el texto del elemento seleccionado.
- `$getElementAttribute()` extrae un atributo del elemento seleccionado.
- `$clickElement()` realiza clic en el elemento seleccionado.
- `$selectTag()` retorna las opciones del elemento seleccionado, útil para revisar las opciones de un tag select
- `$sendKeysToElement(list('x'))` escribe en el elemento la palabra `x`, se utiliza en elementos input.
- `$getElementAttribute('innerHTML')[[1]]` extrae la información HTML del nodo hijo del elemento seleccionado, útil para combinar con `rvest` al realizar la extracción.