WEB SCRAPING

Cómo extraer datos estructurados de una web

Hirikilabs, Tabakalera, Donostia 2018

wiki.montera34.com

¿SCRAPING?

Técnica consistente en extraer datos de una web de forma **automatizada**.

Un **scraper** entra en una web, selecciona unos **datos concretos**, y los copia en otro sitio.

A menudo se nombra también como **crawler**, **araña**, o **bot**.

Google, Facebook, Twitter y otros muchos utilizan estas técnicas.

CÓMO FUNCIONA INTERNET

(Un resumen)

INTERNET BÁSICO

1. Se realiza una **petición** de información desde un dispositivo (cliente)

2. La web (server) **interpreta la petición** y manda una **respuesta**

3. El dispositivo **interpreta la respuesta** y “pinta” la página web

PETICIÓN HTML

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

RESPUESTA HTML

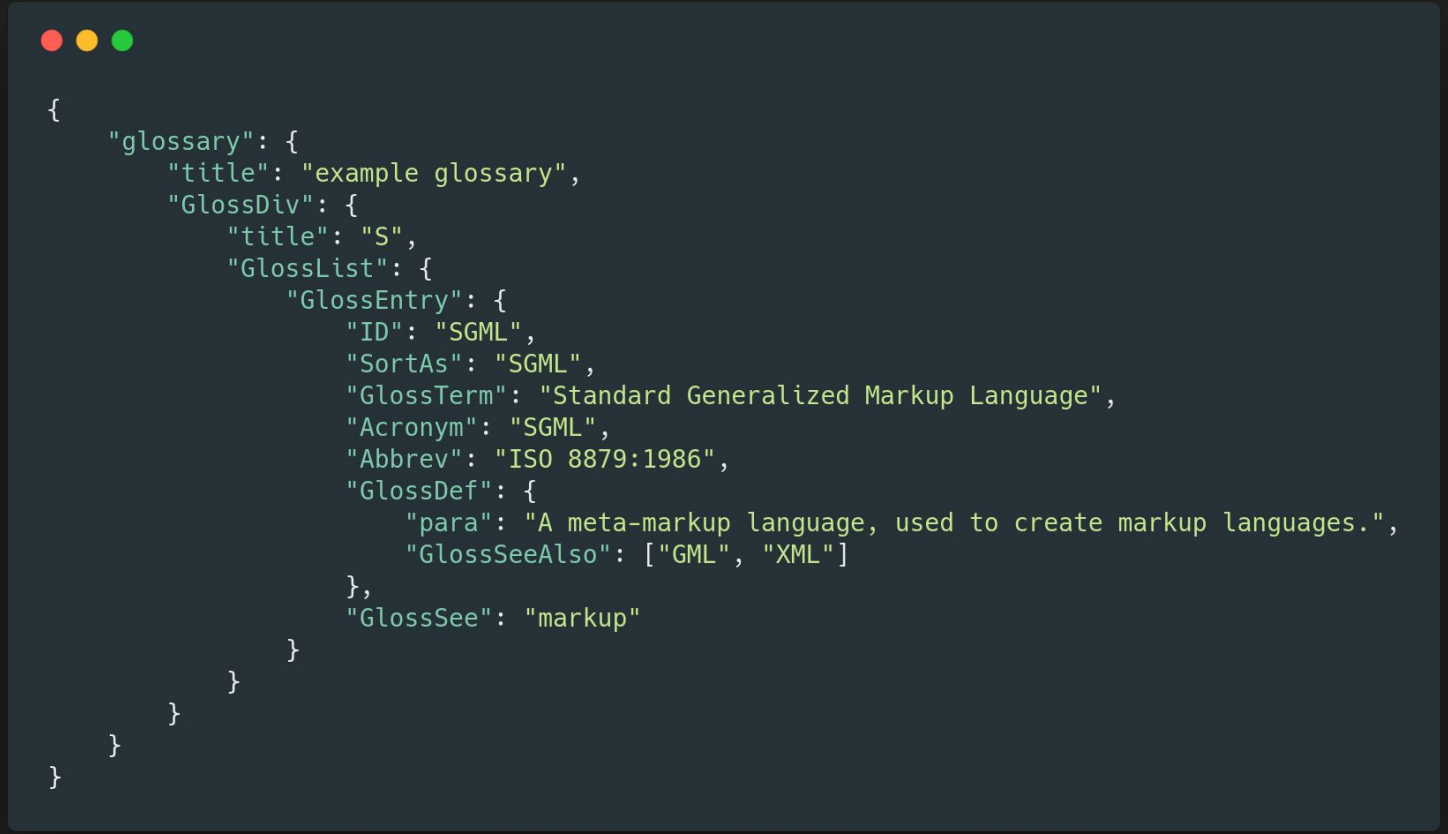
```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

HTML

```
<html>
 <head>
  <title>this is the title</title>
 </head>
 <body>
  <h1>My Heading</h1>
  <p>This is the first paragraph of text.</p>
  <p>This is the second paragraph of text.</p>
  <p>A link: <a href="http://www.simplehtmlguide.com">html guide </a></p>
 </body>
</html>
```

JSON



```
{
    "glossary": {
        "title": "example glossary",
        "GlossDiv": {
            "title": "S",
            "GlossList": {
                "GlossEntry": {
                    "ID": "SGML",
                    "SortAs": "SGML",
                    "GlossTerm": "Standard Generalized Markup Language",
                    "Acronym": "SGML",
                    "Abbrev": "ISO 8879:1986",
                    "GlossDef": {
                        "para": "A meta-markup language, used to create markup languages.",
                        "GlossSeeAlso": ["GML", "XML"]
                    },
                    "GlossSee": "markup"
                }
            }
        }
    }
}
```

SCRAPING CON PYTHON

¿PYTHON?

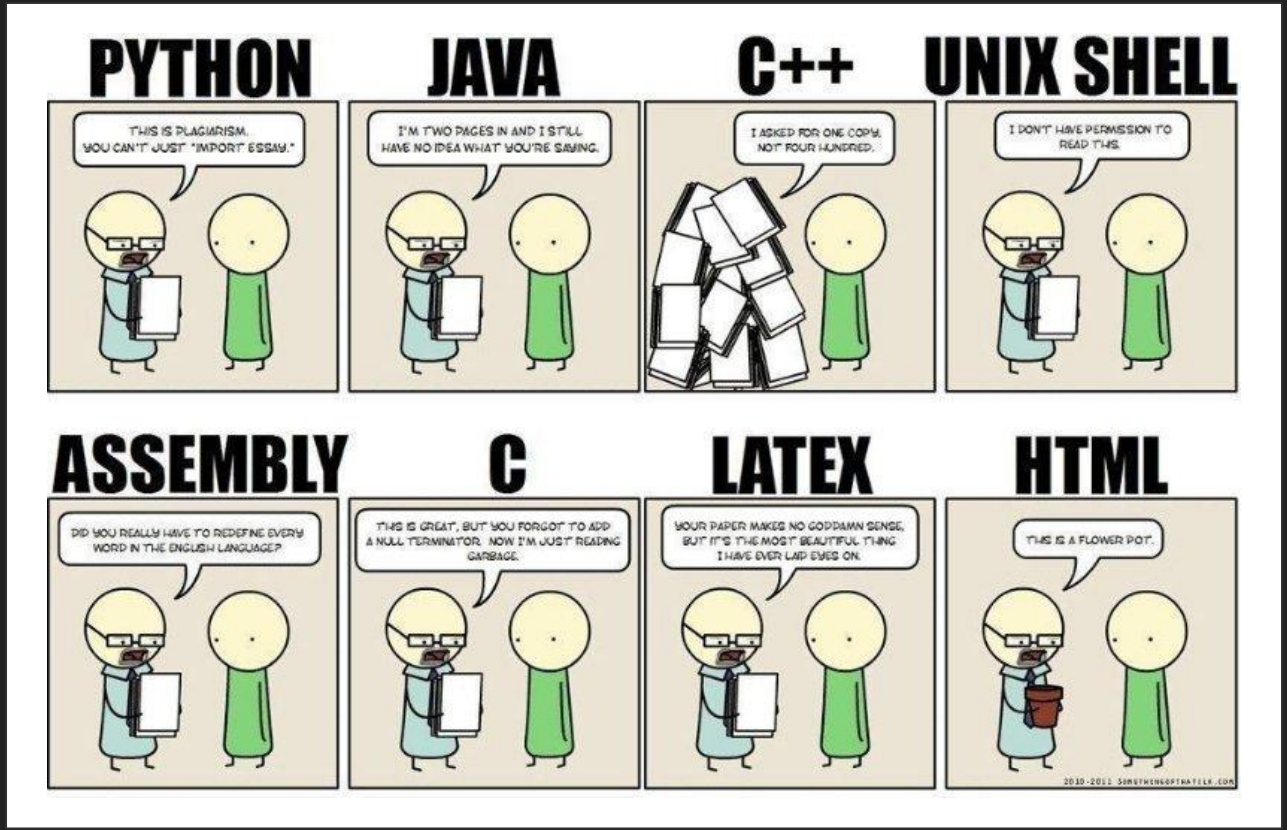
Lenguaje de programación hecho para favorecer el **código legible**.

Soporta orientación a objetos, programación imperativa y programación funcional.

Es un lenguaje **multiplataforma**, interpretado, y **software libre**.

Muy usado en **Data Science** y **Ciberseguridad**.

¿PYTHON?


PYTHON
THIS IS PLAGIARISM. YOU CAN'T JUST "IMPORT ESSAY."
JAVA
I'M TWO PAGES IN AND I STILL HAVE NO IDEA WHAT YOU'RE SAYING.
C++
I ASKED FOR ONE COPY, NOT FOUR HUNDRED.
UNIX SHELL
I DON'T HAVE PERMISSION TO READ THIS.
ASSEMBLY
DID YOU REALLY HAVE TO REDEFINE EVERY WORD IN THE ENGLISH LANGUAGE?
C
THIS IS GREAT, BUT YOU FORGOT TO ADD A NULL TERMINATOR. NOW I'M JUST READING GARBAGE.
LATEX
YOUR PAPER MAKES NO GODDAMN SENSE, BUT IT'S THE MOST BEAUTIFUL THING I HAVE EVER LAID EYES ON.
HTML
THIS IS A FLOWER POT.
2010-2011 SOMETHINGOFTHATILK.COM

MÓDULOS DE PYTHON

(PARA SCRAPING)

Urllib: Hace peticiones HTTP

BeautifulSoup: Interpretar respuestas HTTP

Json: Interpreta documentos (y respuestas) JSON

Selenium: Controlar un navegador web

Scrapy: Framework preparado para scrapers

OTRAS HERRAMIENTAS

(PARA SCRAPING)

POSTMAN: Hace peticiones HTTP y muestra su respuesta.

PhantomJS: Framework JavaScript para scraping.

CÓMO HACER SCRAPING

1. ANÁLISIS

1. ¿Dónde y cómo se encuentra la **información** que queremos obtener?
2. ¿Qué estructura de **URLs** utiliza?
3. ¿Emplea algún **captcha** o método similar?
4. ¿Tiene alguna **API**?
5. ¿Utiliza algún tipo de **indexación**?

2. DIVIDE & CONQUER

1. Un scraper para cada **velocidad**
2. Guardar **URLs de los índices**
3. No llegar nunca al **límite** de la query
4. Preparar el código para **interrupciones**

3. HIDE & SEEK

1. Scrapear sin prisa pero sin pausa
2. Atacar desde **IPs dinámicas**
3. Cambiar aleatoriamente de **headers**
4. Utilizar **proxies**
5. Aleatorizar **tiempos**
6. Cambiar **patrones de navegación**