Web Scrapping con Python y Selenium

José Miguel Amaya Camacho
Python Piura
miguel.amaya99@gmail.com
www.pythonpiura.wordpress.com
https://www.facebook.com/pythonpiura/

¿Qué es Web Scraping?

- **Técnica utilizada para extraer información de sitios web de manera automática mediante el uso de software.**
- **Simula la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.**

Cuestiones Legales

- **Puede ir en contra de los términos de uso de algunos sitios webs. El cumplimiento de estos términos no está totalmente claro.**
- **El grado de protección de estos contenidos aún no está establecido, y dependerá del tipo de acceso realizado, de la cantidad de información recopilada y del grado en el que afecten estos factores al propietario del sitio web.**

Medidas para Detener a los Scrapers

- **Bloquear la dirección IP.**
- **Deshabilitar cualquier interfaz de programación de aplicaciones que el sitio web pudiera estar brindando.**
- **Añadir un captcha u otro sistema de verificación manual al sitio web.**
- **Incrementar el uso de JavaScript y AJAX.**

- **Servicios comerciales antibots: algunas empresas ofrecen servicios antibots y antiscraping.**

¿Es robo de información?

- **No lo es, ya que los datos se captan de la misma manera que lo haría cualquier ser humano, por medio del internet – solo que se utilizan herramientas de automatización que se encargan de hacerlo mucho más rápida y eficientemente que copiar y pegar a mano.**

Selenium

- **Es una suite de herramientas para automatizar la navegación web.**
- **Se usa para el testing de aplicaciones web.**
- **Permite automatizar acciones de usuario.**
- **Si juntamos todo esto, tenemos una herramienta perfecta para el webscraping.**

Disponible

- **Python**
- **Java**
- **C#**
- **Ruby**
- **Etc...**

Instalación

pip install selenium

Si por algún motivo tuviesemos algún error o ya estuviese instalado selenium y se necesita actualizarlo haremos lo siguiente:

pip install selenium --upgrade

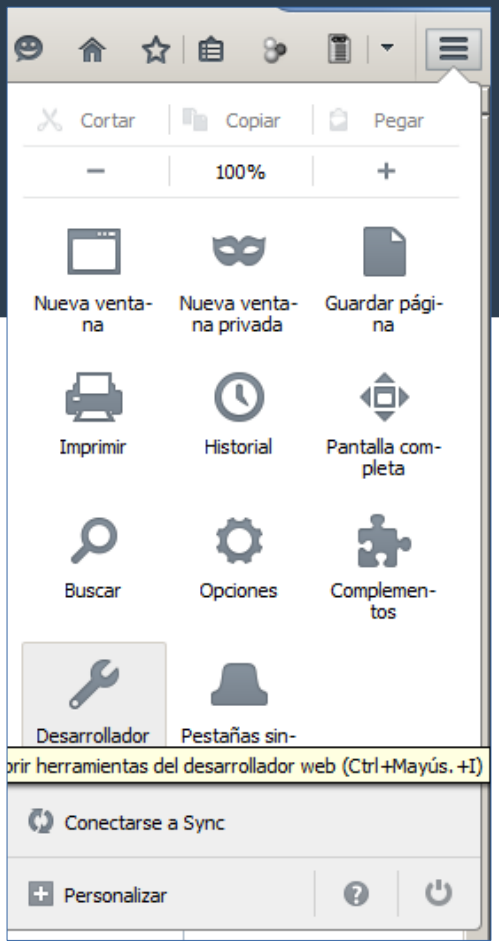
Un ejemplo

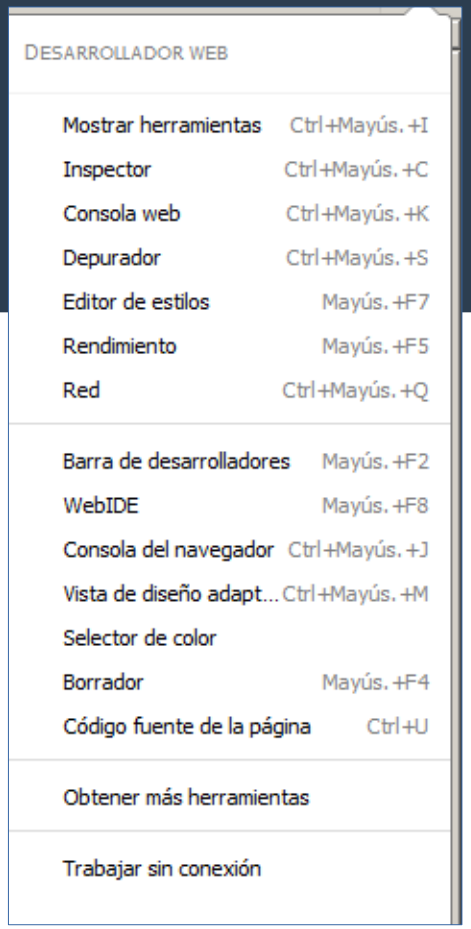
```
#Importamos el módulo webdriver que nos permite interactuar con nuestro navegador

from selenium import webdriver

#Importamos la clase Keys que provee las teclas RETURN, F1, ALT, etc de nuestro teclado

from selenium.webdriver.common.keys import Keys

#Creamos una instancia de Firefox webdriver para poder usar el navegador firefox en esta prueba

driver = webdriver.Firefox()

#El método get permite navegar hacia un enlace determinado en este caso la página de Python.org

driver.get("http://www.python.org")
```

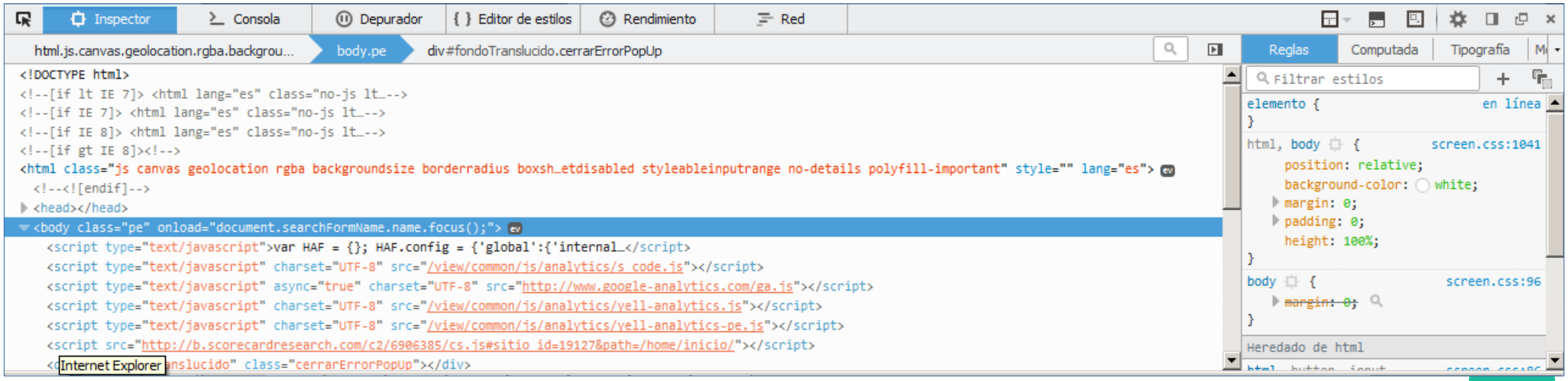
```
#En esta línea confirmaremos si el título tiene la palabra Python
assert "Python" in driver.title
#WebDriver ofrece varias maneras de encontrar los elementos de una página web
#En este ejemplo queremos encontrar el elemento que tenga como atributo name la letra "q"(es la barra de busqueda de la página)
elem = driver.find_element_by_name("q")
#Como ya localizamos el elemento y sabemos que es una caja de texto, vamos a escribir en ella la palabra "selenium"
elem.send_keys("selenium")
#Y finalmente enviaremos la busqueda pulsando la tecla RETURN
elem.send_keys(Keys.RETURN)
#Cerramos el navegador
driver.close()
```

Forma de Trabajo

- **http://www.paginasblancas.pe/**
- **Conocer la estructura HTML de una página web es el primer paso para extraer y usar los datos.**
- **Una buena opción para hacerlo son las herramientas para desarrolladores web que nos proporcionan navegadores como Firefox o Chrome**


Cortar
Copiar
Pegar
100%
Nueva venta-na
Nueva venta-na privada
Guardar pági-na
Imprimir
Historial
Pantalla com-pleta
Buscar
Opciones
Complemen-tos
Desarrollador
Pestañas sin-
rir herramientas del desarrollador web (Ctrl+Mayús. +I)
Conectarse a Sync
Personalizar


DESARROLLADOR WEB
Mostrar herramientas Ctrl+Mayús. +I
Inspector Ctrl+Mayús. +C
Consola web Ctrl+Mayús. +K
Depurador Ctrl+Mayús. +S
Editor de estilos Mayús. +F7
Rendimiento Mayús. +F5
Red Ctrl+Mayús. +Q
Barra de desarrolladores Mayús. +F2
WebIDE Mayús. +F8
Consola del navegador Ctrl+Mayús. +J
Vista de diseño adapt... Ctrl+Mayús. +M
Selector de color
Borrador Mayús. +F4
Código fuente de la página Ctrl+U
Obtener más herramientas
Trabajar sin conexión


Inspector
Consola
Depurador
Editor de estilos
Rendimiento
Red
html.js.canvas.geolocation.rgba.backgrou...
body.pe
div#fondoTranslucido.cerrarErrorPopUp
Reglas
Computada
Tipografía
Filtrar estilos
<!DOCTYPE html>
<!--[if lt IE 7]> <html lang="es" class="no-js lt…-->
<!--[if IE 7]> <html lang="es" class="no-js lt…-->
<!--[if IE 8]> <html lang="es" class="no-js lt…-->
<!--[if gt IE 8]><!-->
<html class="js canvas geolocation rgba backgroundsize borderradius boxsh…etdisabled styleableinputrange no-details polyfill-important" style="" lang="es">
<!--<![endif]-->
<head></head>
<body class="pe" onload="document.searchFormName.name.focus();">
<script type="text/javascript">var HAF = {}; HAF.config = {'global':{'internal…</script>
<script type="text/javascript" charset="UTF-8" src="/view/common/js/analytics/s_code.js"></script>
<script type="text/javascript" async="true" charset="UTF-8" src="http://www.google-analytics.com/ga.js"></script>
<script type="text/javascript" charset="UTF-8" src="/view/common/js/analytics/yell-analytics.js"></script>
<script type="text/javascript" charset="UTF-8" src="/view/common/js/analytics/yell-analytics-pe.js"></script>
<script src="http://b.scorecardresearch.com/c2/6906385/cs.js#sitio_id=19127&path=/home/inicio/"></script>
Internet Explorer
elemento { en línea
}
html, body { screen.css:1041
position: relative;
background-color: white;
margin: 0;
padding: 0;
height: 100%;
}
body { screen.css:96
margin: 0;
}
Heredado de html

Funcionamiento Página Objetivo

- **En este caso la interfaz principal es una ventana de búsquedas donde se debe ingresar la cadena a buscar en una caja de texto y presionar un botón que dice “Buscar” para enviar la consulta, si hay datos coincidentes con el texto ingresado, la página nos mostrará una lista de resultados que contienen la razón social o nombre, la dirección y el teléfono.**

Resultados


Universidad De Lima en Perú | 102 resultados
Refinar por: Localidades
Acciones Multiples
Seleccionar Accion
Universidad De Lima
Avenida Javier Prado, 46 , Santiago de Surco - Lima
http://www.ulima.edu.pe
Universidad De Lima
Ver Telefono
Consulta
Compartir
Universidad De Lima
Avenida Javier Prado Este - Cdra. 46 s/n - Monterrico , Santiago de Surco - Lima
http://www.ulima.edu.pe
Universidad De Lima
Ver Telefono
Consulta
Compartir

Identificando Elementos

- **Empezamos identificando la caja de texto donde se envían los datos a consultar:**

```
<div class="span-19 prepend-1">
    <input id="nName" class="m-search--searchbox-input m-search--searchbox-buscar user-success"
    type="text" autocomplete="off" x-webkit-speech="" placeholder="¿A quién buscas?" tabindex="1"
    value="" name="name"></input>
    <input id="nLocality" class="m-search--searchbox-input m-search--searchbox-donde" type="text"
```

- **Y el botón que ejecuta la consulta:**

```
<div class="span-3 append-1 last">
  <button id="btnSrchName" class="m-button--search" type="submit" tabindex="3">Buscar</button>
</div>
```

- **Veamos el script :-)**

¿Y si nos cruzamos con un Captcha?

- **Captcha o CAPTCHA son las siglas de Completely Automated Public Turing test to tell Computers and Humans Apart (prueba de Turing completamente automática y pública para diferenciar ordenadores de humanos).**
- **Se trata de una prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano.**

Debilidades de un Captcha

Hay algunas aproximaciones a cómo se puede romper un CAPTCHA:

- **Usando humanos para reconocerlos.**
- **Explotando bugs en la implementación que permitan a un atacante saltarse el reconocimiento.**
- **Con software de reconocimiento óptico de caracteres(OCR).**

¿Qué es OCR?

- **Proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de una imagen símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos.**

Tesseract

- **Es un motor OCR libre.**
- **Desarrollado originalmente por HP como software propietario entre 1985 y 1995.**
- **Liberado como código abierto en el 2005 por HP y la Universidad de Nevada.**
- **Es desarrollado actualmente por Google y distribuido bajo la licencia Apache, versión 2.0.**
- **Considerado como uno de los motores OCR libres con mayor precisión.**

Descarga e Instalación

- **Para Windows**

  http://en.osdn.jp/projects/sfnet_tesseract-ocr-alt/downloads/tesseract-ocr-setup-3.02.02.exe/

- **En Ubuntu**

  sudo apt-get install tesseract-ocr

pytesseract

- **Herramienta OCR para python, invoca al programa tesseract, por defecto trabaja con tiff y bmp pero si se integra con PIL(Python Image Library) puede reconocer una variedad de formatos.**
- **Instalación:**

  pip install pytesseract

Pil/Pillow

- **Python Imaging Library (PIL) es una librería gratuita que permite la edición de imágenes directamente desde Python.**
- **Soporta una variedad de formatos, incluídos los más utilizados como GIF, JPEG y PNG. Una gran parte del código está escrito en C, por cuestiones de rendimiento.**

- **PIL soporta únicamente hasta la versión 2.7 de Python.**
- **Por lo que se ha desarrollado Pillow, una bifuración más “amigable”, que pretende mantener una librería estable y que se adapte a las nuevas teconologías (Python 3.x).**

Instalación

- **sudo apt-get install libjpeg-dev**
- **sudo apt-get install python-dev python-setuptools**
- **sudo apt-get install libtiff5-dev libjpeg8-dev zlib1g-dev libfreetype6-dev liblcms2-dev libwebp-dev tcl8.6-dev tk8.6-dev python-tk**
- **pip install pillow**

Ejemplo

- **Veamos otro script :-)**

- **Ahora nuestro objetivo será……**

Una entidad estatal.

Muchas gracias por su atención.