

Ejercitación-2-NB

Nahuel Bargas

Tabla de contenidos

1 Punto 1	1
1.1 Consigna	1
1.2 Respuesta	2
1.2.1 a)	2
1.2.2 b)	5
2 Punto 2	6
2.1 Consigna	6
2.2 Respuesta	6
3 Punto 3	7
3.1 Consigna	7
3.2 Respuesta	7
4 Punto 4	7
4.1 Consigna	7
4.2 Respuesta	7

1 Punto 1

1.1 Consigna

1. Hemos preparado un formulario para que puedas poner en práctica los contenidos del segundo encuentro 1. Realiza las siguientes actividades con Python y Selenium:
 - a. Desarrolla un script que liste en pantalla (por consola) los elementos HTML que posee y los valores de cada uno de sus atributos.

- b. Desarrolla un script que emule la interacción con todos los elementos HTML que posea para luego presionar el botón 'Ingresar'.

1.2 Respuesta

1.2.1 a)

```
from bs4 import BeautifulSoup
import requests
import pandas as pd

# Dirección web formulario

formulario = 'https://jumafernandez.github.io/soco-web_scraping/data/encuentro-02/formulari

# Consulta al formulario

response = requests.get(formulario)

html = BeautifulSoup(response.text, 'html.parser')

# Mostrar todos los elementos HTML

for elemento_html in html.find_all():
    print(elemento_html.name)
```

```
html
head
meta
meta
title
style
body
div
h1
form
label
input
label
input
```

```
div
label
input
label
select
option
option
option
option
option
option
option
button
h2
div
script
```

```
# Mostrar atributos

nombres_atributos = []
valor_atr = []

for elemento_html in html.find_all():
    nombres_atributos += list(elemento_html.attrs)
    valor_atr += list(elemento_html.attrs.values())

data = {'Atributo': nombres_atributos,
        'Valor': valor_atr
        }

df = pd.DataFrame(data)

print(df.to_string(index=False))
```

Atributo	Valor
lang	es
charset	UTF-8
name	viewport
content	width=device-width, initial-scale=1.0

class	[container]
id	formulario
action	#
target	_self
for	email
type	email
id	email
name	email
placeholder	Ingrese su correo electrónico
required	
for	passwd
type	password
id	passwd
name	passwd
placeholder	Ingrese su contraseña
required	
onkeypress	submitOnEnter(event)
class	[checkbox-container]
for	recordar
class	[checkbox-label]
type	checkbox
id	recordar
name	recordar
class	[checkbox-custom]
for	pais
id	pais
name	pais
value	argentina
value	brasil
value	chile
value	colombia
value	mexico
value	peru
value	uruguay
type	submit
id	tituloContenido
style	display: none;
id	contenidoMostrado

1.2.2 b)

```
# Librerías
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import NoSuchElementException

# Configuración del WebDriver para Chrome
options = webdriver.ChromeOptions()

# maximizar el tamaño de la pantalla del webdriver:
options.add_argument("--start-maximized")

# Creación del WebDriver de Chrome
driver = webdriver.Chrome(options=options)

# Navegación en el formulario
driver.get(formulario)

## Correo
ingreso_correo = driver.find_element(By.ID, "email")
ingreso_correo.send_keys("nahuelbargas@hotmail.com")

## Password
ingreso_contraseña = driver.find_element(By.ID, "passwd")
ingreso_contraseña.send_keys("prueba")

## Botón que recuerda el ingreso
boton_recordar = driver.find_element(By.ID, "recordar")
boton_recordar.click()

## Botón que selecciona el país
seleccionar_pais = driver.find_element(By.ID, "pais")
seleccionar_pais_boton = seleccionar_pais.find_element(By.XPATH,
                                                         "//*[@option[contains(text(), 'Uruguay')]]")
```

```

seleccionar_pais_boton.click()

## Botón para ingresar nuestras credenciales
ingresar_credenciales = driver.find_element(By.CSS_SELECTOR, 'button')
ingresar_credenciales.click()

## Mostrar en consola la información que se ingresó en el formulario

contenido_mostrado = driver.find_element(By.ID, "contenidoMostrado")

texto_ingresado = contenido_mostrado.find_elements(By.TAG_NAME, 'p')
texto_ingresado_lista = []

for texto in texto_ingresado:
    texto_ingresado_lista.append(texto.text)

for texto in texto_ingresado_lista:
    print(texto)

```

Correo Electrónico: nahuelbargas@hotmail.com

Contraseña: prueba

Recordar Usuario: Sí

País: uruguay

2 Punto 2

2.1 Consigna

2. Retoma el sitio web que analizaste con la ejercitación del encuentro pasado. Desarrolla un script que te permita scrapearlos precios de los productos de los bienes que posee publicados.

2.2 Respuesta

Se desarrolló el script [easy.py](#) el cual descarga los precios de algunas secciones seleccionadas y genera un archivo en [excel](#).

A su vez, se crearon otros scripts para realizar un paso más, la posibilidad del ingreso de ubicación, ya que la disponibilidad de los productos difiere. Se agregaron funciones al scrap de

precios para ingresar un partido y cerrar pestañas emergentes, `pj`, las de suscripción. El script para probar la descarga de precios por partido es `easy_main.py`, que utiliza las funciones para ingresar a la página, ingresar el partido y descargar precios del script `easy_funciones.py` y que también guarda los resultados de la descarga en un [archivo de excel](#).

3 Punto 3

3.1 Consigna

3. Preparará una muy breve presentación para explicar en 10' al inicio del siguiente encuentro los siguientes aspectos:
 - a. ¿Qué características del sitio web hicieron complejo el scraping?
 - b. Muestra las cuestiones más interesantes del script que permite el scraping.
 - c. ¿El mismo está terminado? En caso contrario, expone cuáles fueron las características del sitio web que no pudiste sortear para el desarrollo del scraping.

3.2 Respuesta

Presentación oral

4 Punto 4

4.1 Consigna

Por último, documenta todo tu trabajo en un repositorio GitHub y comparte el enlace al equipo docente por Slack hasta el día previo al siguiente encuentro.

4.2 Respuesta

Documentado usando [Quarto](#)

Todos los archivos de esta ejercitación se encuentran en sección dentro del [repositorio](#).

Recomiendo ver este documento en su versión en [html](#).