Automatizacion de pruebas con selenium y python



**1.-Ejemplo:** Automatizar la búsqueda de una página usando Firefox como navegador.

Explicación del Código:

* importa el módulo webdriver y permite automatizar la interacción con con navegadores web:

from selenium import webdriver

* importa la clase keys que permite simular la pulsacion de teclas en un teclado:

from selenium.webdriver.common.keys import Keys

* importa la calse by que se usa para localizar elementos de una página web mediante diferentes estrategias:

from selenium.webdriver.common.by import By

**Inicia el WebDriver de Firefox:**

* Se crea una instancia del navegador Firefox para interactuar con la web.
* Asegúrate de tener GeckoDriver instalado para que funcione.

driver = webdriver.Firefox()

Abre la página web "http://www.python.org" en el navegador.

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

**Verifica que el título de la página contenga "Python".**

* Si "Python" no está en el título, la prueba fallará.

assert "Python" in driver.title

**Encuentra la barra de búsqueda en la página utilizando By.NAME.**

* Busca el elemento HTML con name="q", que es el campo de búsqueda en Python.org.

elem = driver.find\_element(By.NAME, "q")

* Limpia el campo de búsqueda, en caso de que tenga texto previo.

elem.clear()

* Escribe "pycon" en la barra de búsqueda.

elem.send\_keys("pycon")

* Simula presionar ENTER (Keys.RETURN) para enviar la búsqueda.

elem.send\_keys(Keys.RETURN)

**Verifica que la página de resultados no muestre "No results found."**

* Si "No results found." aparece en el código fuente de la página (driver.page\_source), significa que **no hubo resultados**, y la prueba fallará.

assert "No results found." not in driver.page\_source

* Cierra el navegador una vez que termina la prueba.

driver.close()

**Codigo:**

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.by import By

driver = webdriver.Firefox()

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

assert "Python" in driver.title

elem = driver.find\_element(By.NAME, "q")

elem.clear()

elem.send\_keys("pycon")

elem.send\_keys(Keys.RETURN)

assert "No results found." not in driver.page\_source

driver.close()

**2.-Ejemplo:** Automatizar la búsqueda de una página con unittest.

**Importaciones de las bibliotecas necesarias**:

* unittest: Framework de pruebas de Python.
* webdriver: Para controlar el navegador.
* Keys: Para simular teclas del teclado.
* By: Para encontrar elementos en la página web.

import unittest

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.by import By

**Creación de la clase de prueba que se hereda de unittest.TestCase.**

* Todas las pruebas dentro de esta clase deben comenzar con test\_.

class PythonOrgSearch(unittest.TestCase):

**Configuración del entorno de prueba** antes de ejecutar cada test.

* Abre el navegador Firefox (webdriver.Firefox()).

    def setUp(self):

        self.driver = webdriver.Firefox()

* Abre la página web "http://www.python.org".

    def test\_search\_in\_python\_org(self):

        driver = self.driver

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

**Verifica que el título de la página contiene "Python"**.

* Si el título no contiene "Python", la prueba fallará.

        self.assertIn("Python", driver.title)

* Encuentra la barra de búsqueda usando By.NAME.

        elem = driver.find\_element(By.NAME, "q")

* Escribe "pycon" en la barra de búsqueda y presiona ENTER (Keys.RETURN).

        elem.send\_keys("pycon")

        elem.send\_keys(Keys.RETURN)

**Verifica que los resultados no están vacíos**.

* Si "No results found." aparece en el código fuente de la página (driver.page\_source), la prueba fallará.

        self.assertNotIn("No results found.", driver.page\_source)

* Cierra el navegador después de cada prueba.

    def tearDown(self):

        self.driver.close()

**Ejecuta la prueba si el script se ejecuta directamente**.

* unittest.main() ejecuta todas las pruebas dentro de la clase PythonOrgSearch.

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

**Codigo:**

**import** **unittest**

**from** **selenium** **import** webdriver

**from** **selenium.webdriver.common.keys** **import** Keys

**from** **selenium.webdriver.common.by** **import** By

**class** **PythonOrgSearch**(unittest.TestCase):

**def** setUp(self):

self.driver = webdriver.Firefox()

**def** test\_search\_in\_python\_org(self):

driver = self.driver

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

self.assertIn("Python", driver.title)

elem = driver.find\_element(By.NAME, "q")

elem.send\_keys("pycon")

elem.send\_keys(Keys.RETURN)

self.assertNotIn("No results found.", driver.page\_source)

**def** tearDown(self):

self.driver.close()

**if** \_\_name\_\_ == "\_\_main\_\_":

unittest.main()