

Automação com Selenium usando Python parte 6

COMANDOS DE INTERAÇÃO COM ELEMENTOS

element.
click()
send_keys()
text
get_attribute()

Vamos aprender a usar os comandos acima, para isso crie um novo arquivo de nome `Teste_iteracao_elementos.py` e copie as primeiras linhas de código do arquivo anterior, ficara assim:

```
import time

from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
servico = Service(ChromeDriverManager().install())
browser = webdriver.Chrome(service=servico) # aqui está
instanciando o chrome
browser.get("https://saucedemo.com")
```

Vamos acrescentar o código responsável por fazer login na página www.saucedemo.com, segue o código comentado abaixo:

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
servico = Service(ChromeDriverManager().install())
browser = webdriver.Chrome(service=servico) # aqui está
instanciando o chrome
```

```

browser.get("https://saucedemo.com")

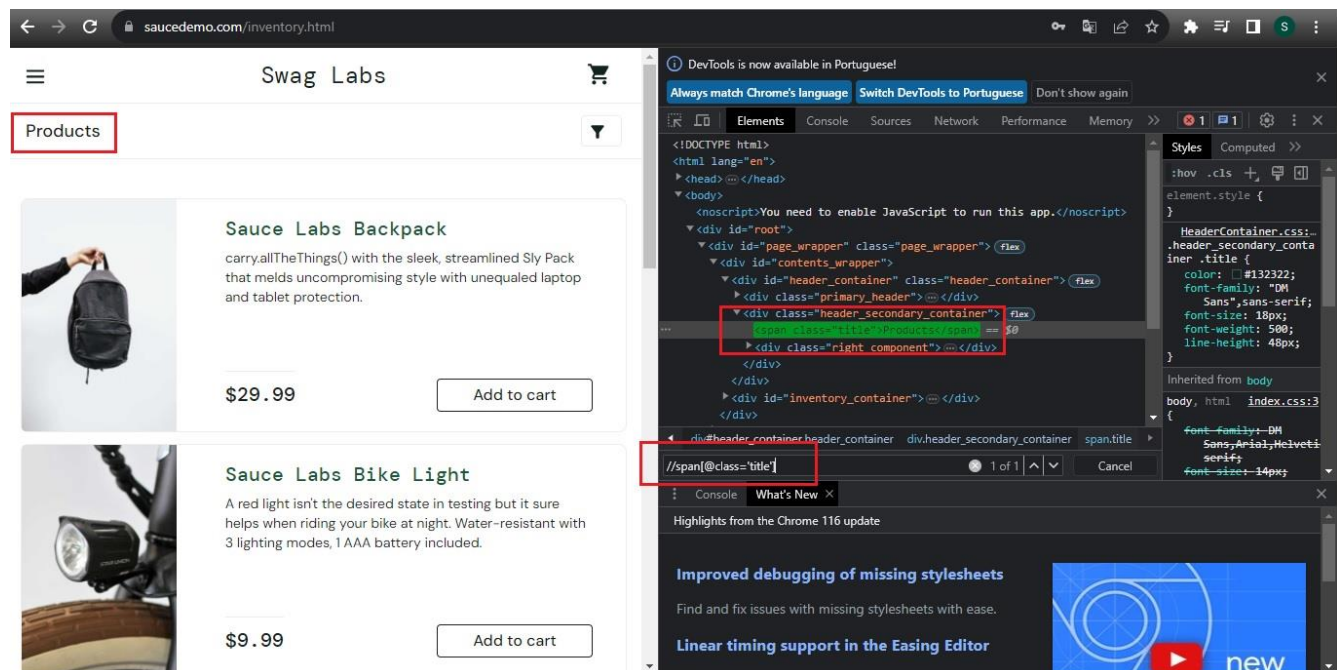
username = browser.find_element(By.ID, "user-name")
#variável username recebe o elemento user-name
password = browser.find_element(By.ID, "password")
#variável password recebe o elemento password
btn_login = browser.find_element(By.ID, 'login-button') #a
variável btn_login recebe o elemento do botão de login

username.send_keys("standard_user") #o campo será
preenchido com o nome do usuário
password.send_keys("secret_sauce") #o campo será preenchido
com a senha
btn_login.click() # clique no botão de login
time.sleep(5)

```

Salve e execute o código até aqui, ele tem que fazer login na página com sucesso.

Próximo passo é mapear um texto após o login para comprovar que o mesmo foi efetuado com sucesso. Escolhemos a palavra “Products” que está no canto superior esquerdo da página. Observe a imagem abaixo, é um elemento do tipo texto que não tem muita coisa nele, então temos que criar um xpath pra ele.



Use o CTRL + F para mapear o elemento de texto, abaixo o xpath dele:

`//span[@class='title']`

O xpath foi criado a partir das características do elemento encontradas no código como mostra a imagem acima.

Abaixo o código responsável por fazer login e na sequência validar a mensagem de texto da tela, a palavra Products:

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
servico = Service(ChromeDriverManager().install())
browser = webdriver.Chrome(service=servico) # aqui está
instanciando o chrome

browser.get("https://saucedemo.com")

username = browser.find_element(By.ID, "user-name")
#variável username recebe o elemento user-name
password = browser.find_element(By.ID, "password")
#variável password recebe o elemento password
btn_login = browser.find_element(By.ID, 'login-button') #a
variável btn_login recebe o elemento do botão de login

#send_keys
username.send_keys("standard_user") #o campo será
preenchido com o nome do usuário
password.send_keys("secret_sauce") #o campo será preenchido
com a senha

#.click()
btn_login.click() # clique no botão de login

#text
products_title = browser.find_element(By.XPATH,
"//span[@class='title']") #a variável products_title recebe
o elemento
print(products_title.text) # vai printar no terminal a
palavra Products que está na página
assert products_title.text == 'Products' #asserção
```

Usando o comando `get.attribute()` podemos mapear uma imagem na tela, nesse caso a primeira bolsa que aparece na página, vamos ter que mapear usando xpath, ficara assim:

`(//img[@class='inventory_item_img'])[1]` - uma imagem, que possui uma classe de nome `inventory_item_img`, primeira ocorrência [1].

O código final ficara conforme mostrado abaixo e comentado:

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
servico = Service(ChromeDriverManager().install())
browser = webdriver.Chrome(service=servico) # aqui está
instanciando o chrome

browser.get("https://saucedemo.com")

username = browser.find_element(By.ID, "user-name")
#variável username recebe o elemento user-name
password = browser.find_element(By.ID, "password")
#variável password recebe o elemento password
btn_login = browser.find_element(By.ID, 'login-button') #a
variável btn_login recebe o elemento do botão de login

#send_keys
username.send_keys("standard_user") #o campo será
preenchido com o nome do usuário
password.send_keys("secret_sauce") #o campo será preenchido
com a senha

#.click()
btn_login.click() # clique no botão de login

#text
products_title = browser.find_element(By.XPATH,
"//span[@class='title']") #a variável products_title recebe
o elemento
print(products_title.text) # vai printar no terminal a
palavra Products que está na página
assert products_title.text == 'Products' #asserção

#get_attribute()
img_backpack = browser.find_element(By.XPATH,
"(/img[@class='inventory_item_img'])[1]") #variável recebe
o elemento
print(img_backpack.get_attribute('alt')) #vai printar no
terminal o conteúdo do atributo alt lá no código da página
assert img_backpack.get_attribute('alt') == 'Sauce Labs
Backpack' #asserção, Sauce Labs Backpack é o texto que está
dentro do atributo alt
```

