

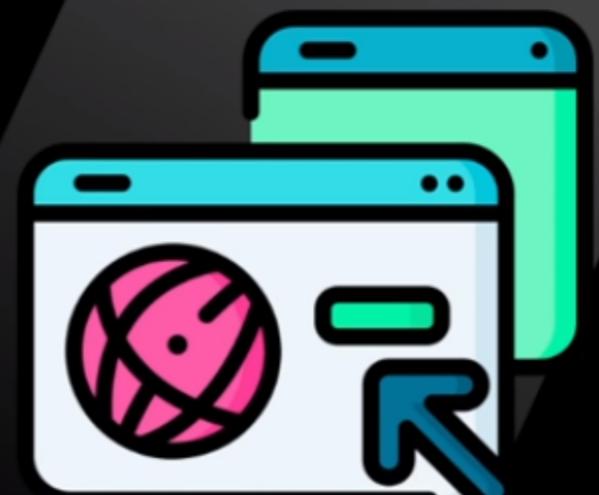
Locators

Tag Atributo Valor Atributo Valor

```
<input class="input_error form input" placeholder="Password" type="password"
data-test="password" id="password" name="password" autocorrect="off"
autocapitalize="none" value fdprocessedid="84hy06">
```

By.ID

```
browser.find_element(By.ID, "password")
```



Locators

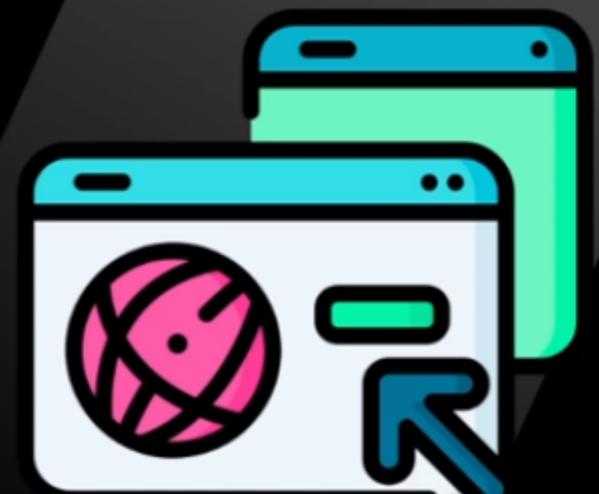
Tag Atributo Valor

Atributo Valor

```
<input class="input_error form_input" placeholder="Password" type="password"
data-test="password" id="password" name="password" autocorrect="off"
autocapitalize="none" value fdprocessedid="84hy06">
```

By.NAME

```
browser.find_element(By.NAME, "password")
```



Locators

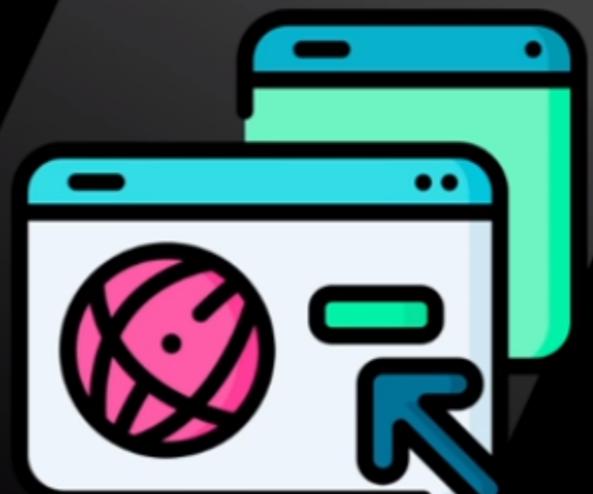
href: atributo de link

```
<a href="https://twitter.com/saucelabs" target="_blank" rel="noreferrer">Twitter</a>
```

Link text

By.LINK_TEXT

```
browser.find_element(By.LINK_TEXT, "Twitter")
```



Locators

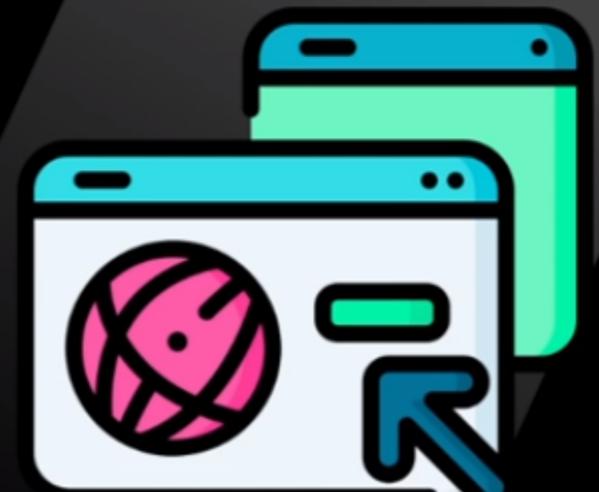
href: atributo de link

```
<a href="https://twitter.com/saucelabs" target="_blank" rel="noreferrer">Twitter</a>
```

Link text

By.PARTIAL_LINK_TEXT

```
browser.find_element(By.PARTIAL_LINK_TEXT, "Twi")
```



Locators

```
Tag      Atributo          Atributo          Valor  
↓       ↓                  ↓                  ↓  
<input class="input_error form_input" placeholder="Password" type="password"  
data-test="password" id="password" name="password" autocorrect="off"  
autocapitalize="none" value fdprocessedid="84hy06">
```

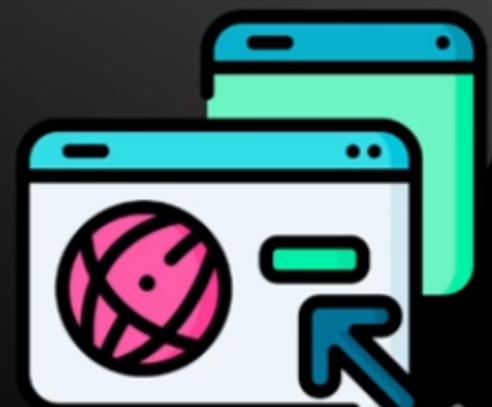
By.CLASS_NAME

Encontra a primeira ocorrência

```
browser.find_element(By.CLASS_NAME, "input_error form_input")
```

```
browser.find_elements(By.CLASS_NAME, "input_error form_input")
```

Retorna todas as ocorrências



Locators

```
Tag      Atributo  
↓       ↓  
<input class="input_error form_input" placeholder="Password" type="password"  
data-test="password" id="password" name="password" autocorrect="off"  
autocapitalize="none" value fdprocessedid="84hy06">
```

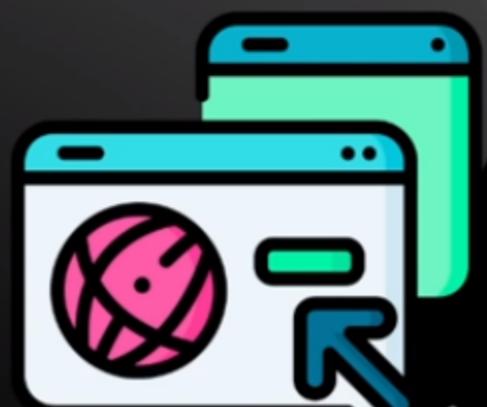
By.TAG_NAME

Encontra a primeira ocorrência

```
browser.find_element(By.TAG_NAME, "input")
```

Retorna todas as ocorrências

```
browser.find_elements(By.TAG_NAME, "input")
```



Locators

```
<input class="input_error form_input" placeholder="Password" type="password"
data-test="password" id="password" name="password" autocorrect="off"
autocapitalize="none" value fdprocessedid="84hy06">
```

Diagram illustrating the structure of the locator string:

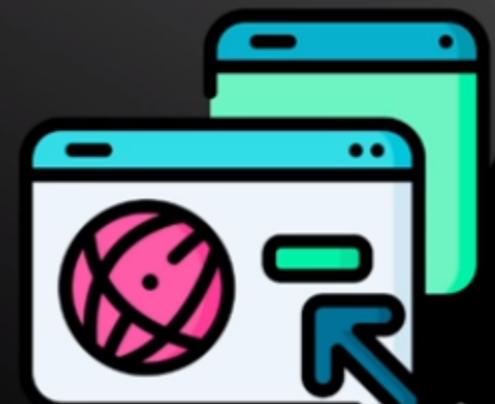
- Tag: Points to the opening tag <input>.
- Atributo: Points to the first attribute placeholder="Password".
- Atributo: Points to the second attribute data-test="password".
- Valor: Points to the value attribute value="fdprocessedid="84hy06".

By.CSS_SELECTOR

tag_name#id_value (tag é opcional)

```
browser.find_element(By.CSS_SELECTOR, "input#password")
```

```
browser.find_element(By.CSS_SELECTOR, "#password")
```



Locators

```
<input class="input_error form_input" placeholder="Password" type="password"  
data-test="password" id="password" name="password" autocorrect="off"  
autocapitalize="none" value fdprocessedid="84hy06">
```

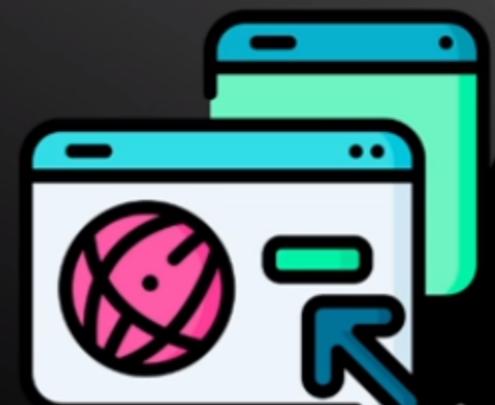
Tag Atributo Atributo Valor

By.CSS_SELECTOR

tag_name.class_value (tag é opcional)

```
browser.find_element(By.CSS_SELECTOR, "input.input_error")
```

```
browser.find_element(By.CSS_SELECTOR, ".input_error")
```



Locators

```
<input class="input_error form_input" placeholder="Password" type="password"  
data-test="password" id="password" name="password" autocorrect="off"  
autocapitalize="none" value fdprocessedid="84hy06">
```

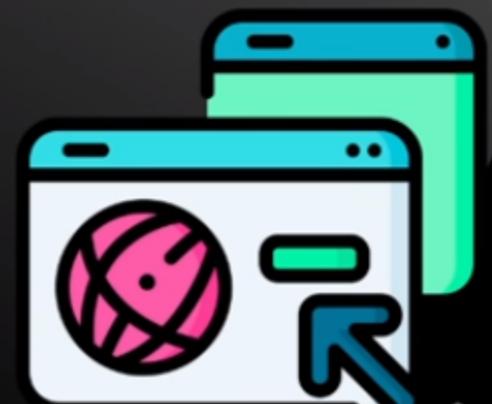
Tag Atributo Atributo Valor

By.CSS_SELECTOR

tag_name[attribute=value] (tag é opcional)

browser.find_element(By.CSS_SELECTOR, "input[type=password]")

browser.find_element(By.CSS_SELECTOR, "[type=password]")



Locators

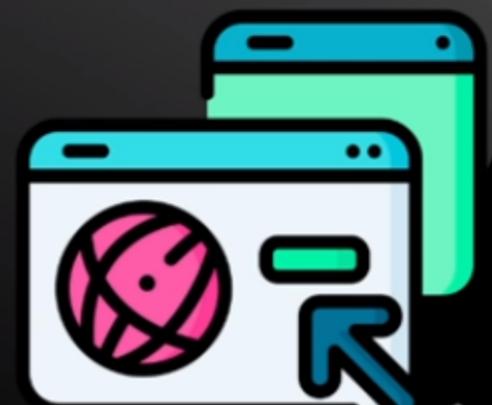
Tag Atributo Atributo Valor

```
<input class="input_error form_input" placeholder="Password" type="password"
data-test="password" id="password" name="password" autocorrect="off"
autocapitalize="none" value fdprocessedid="84hy06">
```

By.CSS_SELECTOR

tag_name.class_name[attribute=value]

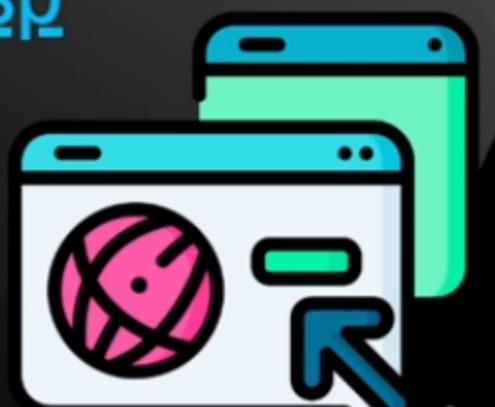
```
browser.find_element(By.CSS_SELECTOR, "input.input_error[type=password]")
```



XPath

- XPath é uma **sintaxe** para definir partes de um documento XML
- XPath pode ser usado para **navegar** por elementos e atributos em um documento XML
- XPath usa **expressões de caminho** para **navegar** em documentos XML
- XPath contém uma **biblioteca** de funções padrão
- XPath é usado para localizar elementos em uma página web por meio do DOM
- XPath é o endereço do elemento em uma página

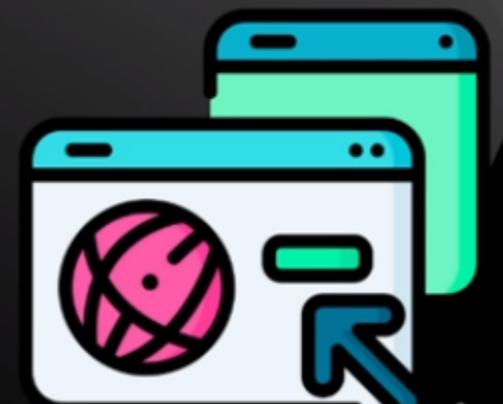
https://www.w3schools.com/xml/xpath_intro.asp



XPath

Absolute/Full XPath

Relative/Partial XPath



Início - Canva | Selenium Webdriver com Python | XPath Axes | Swag Labs

saucedemo.com

SWAGLABS

Arquivo Editar Exibir

FULL
/html/body/div[1]/div/div[2]/div[1]/div[1]/div/form/input

PARTIAL
//*[@id="login-button"]

Elements Console Sources Network Performance Memory > 1 ⋮

<input class="input_error form_input" placeholder="Username" type="text" data-
- er-name" name="user-name" autocorrect="off" autocapitalize="
- edid="awasia">

<input class="form_input" placeholder="Password" type="password" data-
- password" name="password" autocorrect="off"
- value fdprocessedid="c4m9xl">

</div></div> flex

<button data-test="login-button" id="login-button" value="Login" fdprocessedid="ztes26"> == \$0

div.login-box form input#login-button.submit-button.btn_action ...

DOM Breakpoints Properties Accessibility

:hov .cls +, Cancel

Ln 2, Col 14 | 290% | Windows (CRLF) | UTF-8

Hints for inactive CSS properties

Identify CSS styles that are entirely valid but have no
visible effect



Arquivo Editar Exibir



contains()
text()
and/or
|| I

Início - Canva | Selenium Webdriver com Python | XPath Axes | Swag Labs

saucedemo.com/inventory.html

SWAGLABS

PRODUCTS

NAME (A TO Z)

Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.



\$29.99

ADD TO CART

Sauce Labs Bike Light

A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.



\$9.99

ADD TO CART

Elements Console Sources Network

```
</div>
  ><div>...</div>
</div>
</div>
</div>
  ><div class="header_label">...</div>
  ><div id="shopping_cart_container" class="shopping_cart_container">
    ...<a class="shopping_cart_link"></a> == $0
  </div>
</div>
<div class="header_secondary_container" flex="flex">
  <span class="title">Products</span>
  <div class="peek"></div>
  <div class="right_component">
    <span class="select_container">
```

mary_header div#shopping_cart_container.shopping_cart_container a.shopping_cart_link

[@id="shopping_cart_container"]/a[@class="shopping_cart_link"] 1 of 1 Cancel

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +, ↗

element.style { }

Console What's New

Highlights from the Chrome 108 update

Hints for inactive CSS properties

Pesquisar

22:42 POR PTB2 dezembro udemy

COMANDOS DE NAVEGAÇÃO

browser.

get()

maximize_window()



refresh()

back()

forward()

close()

quit()



COMANDOS DA APLICAÇÃO

browser.

title

current_url

page_source



COMANDOS CONDICIONAIS



element.

`is_displayed()`

`is_enabled()`

`is_selected()`

Retorna True ou False



COMANDOS DE INTERAÇÃO COM ELEMENTOS

element.

click()

send_keys()

text

get_attribute()



COMANDOS DE ESPERA (WAIT)

element.



time.sleep()

implicit_wait

explicit_wait

