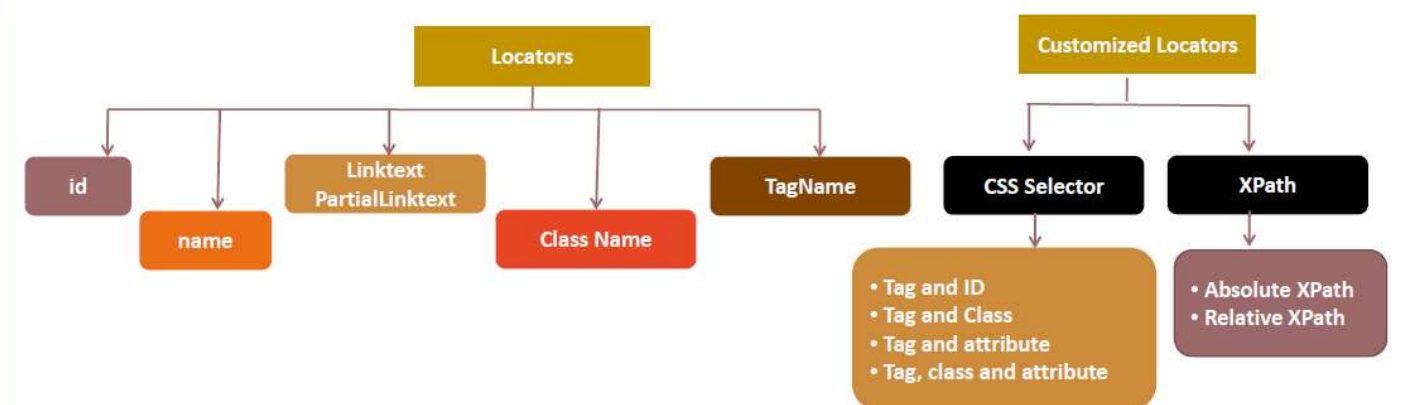


19. Basic Locators - ID, NAME, LINK TEXT, PARTIAL LINK TEXT, TAGNAME, CLASSNAME

Locators are addresses of an elements. In automation code we use locators to

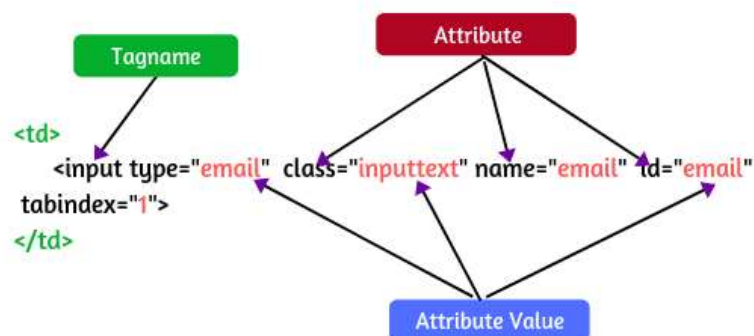
- Identify web elements uniquely on web page
- Perform actions on those web elements

Types of Locators



- Every webpage is designed by using HTML,CSS and JS.
- Webelement details can be found on HTML code of webpage.
- **Normal Locators** or **Basic Locators** or **Locators** can be directly found on HTML code of webpage
- Customized Locators can't directly be found in HTML.We have to write our own locator or we can use some tools to generate CSS Selector and XPath.

Understanding HTML



- Every element has a **Tagname** in the HTML line which has a certain **Attributes/Properties** and every **Attribute/Property** has a certain value.

Common HTML tags used for different types of web elements

Element	Code
Paragraph	<p>This is a paragraph.</p>
Heading	<h1>This is a heading</h1>
Image	
Link (Anchor)	Click here
Button	<button type="button">Click Me</button>
Input Box (Text Field)	<input type="text" name="username" placeholder="Enter your name">
Checkbox	<input type="checkbox" name="subscribe" value="newsletter"> Subscribe
Radio Button (Male)	<input type="radio" name="gender" value="male"> Male
Radio Button (Female)	<input type="radio" name="gender" value="female"> Female
Drop-down List (Select Box)	<select name="cars"> <option value="volvo">Volvo</option> <option value="saab">Saab</option> </select>
Text Area	<textarea name="message" rows="4" cols="50">Enter your message here...</textarea>

Locators or Normal Locators or Basic Locators

Locator Type	Example Code
ID	<code>element = driver.find_element(By.ID, "username")</code>
Name	<code>element = driver.find_element(By.NAME, "password")</code>
LinkText	<code>element = driver.find_element(By.LINK_TEXT, "About Us")</code>
PartialLinkText	<code>element = driver.find_element(By.PARTIAL_LINK_TEXT, "Contact")</code>
TagName	<code>element = driver.find_element(By.TAG_NAME, "button")</code>
ClassName	<code>element = driver.find_element(By.CLASS_NAME, "btn-primary")</code>

Note

- **LINK_TEXT** and **PARTIAL_LINK_TEXT** Locators are used to identify Links.
- Part of **Link Text** is called **Partial Link Text**. Normally we perform **click action** on Link.
- **ID**, **NAME**, **LINK_TEXT**, **PARTIAL_LINK_TEXT** are used to identify a single web element because most of the time these attributes are unique.
- If we want to **find more than one Web element (Object) or multiple Web elements** we go with **CLASS NAME**, **TAGNAME**. We have to use **find_elements** method instead of **find_element** method.

find_element() Vs find_elements()

Scenario 1: Locator matches a single web element

- **find_element(locator)** → Returns the first matching element as a **WebElement**.
- **find_elements(locator)** → Returns a **list of WebElement** containing the single matching element.

Scenario 2: Locator matches multiple web elements

- **find_element(locator)** → Returns the first matching element as a **WebElement**.
- **find_elements(locator)** → Returns a **list of WebElement** containing all matching elements.

Scenario 3: Locator does not match any elements

- **find_element(locator)** → Throws a **NoSuchElementException**.
- **find_elements(locator)** → Returns an **empty list []** of size 0 without throwing an exception.

[LocatorsDemo.py](#)

```
from selenium import webdriver
```

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

```
opt = webdriver.ChromeOptions()
```

```
opt.add_experimental_option("detach", True)
```

```
driver = webdriver.Chrome(options=opt)
```

```
driver.get("http://www.automationpractice.pl/index.php")
```

```
driver.maximize_window()
```

```
driver.implicitly_wait(10)
```

Locate element by 'ID'

```
search_box = driver.find_element(By.ID, "search_query_top")
```

```
search_box.send_keys("T-shirts")
```

Locate element by 'Name' and click (single step)

```
driver.find_element(By.NAME, "submit_search").click()
```

Locate element by 'LinkText' and click (preferable for links)

```
driver.find_element(By.LINK_TEXT, "Printed Chiffon Dress").click()
```

'PartialLinkText'

```
driver.find_element(By.PARTIAL_LINK_TEXT, "Chiffon Dress").click()
```

Locate multiple elements by 'ClassName'

```
slider_count = len(driver.find_elements(By.CLASS_NAME, "homeslider-container"))
```

```
print("Number of sliders:", slider_count)
```

Locate multiple elements by 'TagName'

Find and print the number of links on the page

```
links = driver.find_elements(By.TAG_NAME, "a")
```

```
print("Number of links on the page:", len(links)) # Example output: 88
```

Find and print the number of images on the page

```
images = driver.find_elements(By.TAG_NAME, "img")
```

```
print("Number of images on the page:", len(images)) # Example output: 22
```

Compare find_element() vs find_elements()

Case 1: find_element() - Throws NoSuchElementException if the locator is incorrect or not found

try:

```
element = driver.find_element(By.ID, "xyz")
```

except Exception as e:

```
print("Element not found using find_element:", str(e))
```

Case 2: find_elements() - Returns an empty list if the locator is incorrect or not found

```
elements = driver.find_elements(By.ID, "xyz")
```

```
print("Number of elements found using find_elements:", len(elements)) #output: 0
```

```
driver.quit()
```

Lab Assignment

1. Open the application URL: <https://demowebshop.tricentis.com/>
2. Write code to:
 - a. Find the total number of links on the homepage.
 - b. Find the total number of images on the homepage.
 - c. Click on any product link using either linkText or partialLinkText.