Day4 Locators
getText getAttribute.

How many total locators we have?

Selenium has total of 8 locators.

- id
- name
- className
- linkText
- partialLinkText
- tagName
- cssSelector
- xpath

cssSelector

- cssSelector is one of 8 locators of Selenium.
- It allows users/us to create custom locators using attributes and values.
- It does not limit us using only class, id, name attributes.
- We can use ANY attribute and their values.
- It has its own syntax that we must follow to create cssSelector.
- Using cssSelector we can go from parent to child element.
- To go from parent to child we use “>” sign.

  syntax: tagName[attribute='value'] > childTagName
- Using cssSelector we cannot go from child to parent.

Why do we need to move from parent to child?

Ex: Locating "Forgot password" header from
http://practice.cydeo.com/forgot_password

- ** **
  <h2>Forgot Password</h2>
- div[class='example'] > h2

- Sometimes the web element we are trying to locate does not have a unique attribute/value.
- In this scenario, we can locate one of the parents that has a unique attribute value and move down to child web element we are trying to locate.
- We CANNOT go from CHILD TO PARENT using cssSelector.

There are two different types of syntaxes for cssSelector.

#1.syntax: tagName[attribute='value’]

 TESLA

1- locate above link with cssSelector using id:
a[id='uh7’]

2- locate above link with cssSelector using name:
a[name='bb95’]

3- locate above link with cssSelector using href:
a[href='https://www.tesla.com’]

NOTE: If you want to be less specific, you don't have to pass tagName with this locator.

[attribute='value’]
[id='uh7’]
[name='bb95']

#2.syntax: Second syntax is limited to use with "id" and "class" only.

- tagName#idValue
- tagName.classValue

---> stands for id attribute

. ---> stands for class attribute

 TESLA CYBERTRUCK

tagName.classValue --> a.ff58
tagName#idValue --> a#bb22

XPATH LOCATOR

- xpath is one of 8 locators of Selenium
- xpath allows us to create custom locators using provided attributes and their values
- we can also use the text of the provided web element to create
- XPATH has 2 different types

***Interview question: What is the difference between absolute xpath and relative xpath?

ABSOLUTE XPATH

- Absolute xpath starts with single slash "/”
- It starts looking in html from the root/parent element : html element
- It starts from html tag, and it goes down 1 by 1 until we reach to the web element we are looking for
- This is not good way of locating a web element.
- It will break with any minimal change in the html code.

/html/body/table/tbody/tr[2]/td/div/div/form/div[4]/button

RELATIVE XPATH

- Relative xpath starts with double slash "//”
- "//" means you can start from anywhere in the HTML code
- Since we are allowed to start from anywhere in the HTML code, relative xpath is very dependable
- We will use relative xpath, not absolute xpath
- The only time your relative xpath is breaking (not working) is when/if the developer is specifically changing the attribute value we used

MAIN SYNTAX: //tagName[@attribute='value']

Ex:

** TESLA CYBERTRUCK **

- //tagName[@attribute='value’]
- locate above <a> tag using relative xpath locator with different attributes.

  using name attribute : //a[@name='uh68’]

  using class attribute : //a[@class='ff58’]

  using id attribute : //a[@id='bb22’]

- We are NOT limited with id, name, class, or href attributes.
- We can use any custom attribute and their value with XPATH locator.

COMMONLY USED XPATH SYNTAXES:

1. //tagName[@attribute='value’]
2. //tagName[contains(@attribute, 'value’)]
3. //tagName[.='text'] same as //tagName[text()='text’]
   . --> stands for text in xpath
4. //*[@attribute='value’]

- * --> is used when we do not want to search by a tagName.
- If we want to be less specific, we pass *, and it will only match and return whatever attribute and value is provided.

HOW TO GO FROM CHILD TO PARENT and PARENT TO CHILD IN XPATH?

- We go from parent to child using "/“
- We use "/.." this goes from child to parent
- We use ”/following-sibling::tagname” this goes to the next sibling.
- We use “/preceding-sibling::tagname” this goes to previous sibling.

CSS vs XPATH

- Css is slightly faster than xpath on IE, other browsers it is negligible.
- Css is easier to read and write

1. css cannot locate using text of web elements
- xpath://*[.="Don't Click!"]
- css: NA

2. It cannot find from matches base on index (different parent)
- xpath:--> (xpathFormula)[indexNumber]
- css: NA

3. child to parent NA
- //button/../ --> goes back to parent
- CSS : NA

- xpath can do more than css, but it is little bit complex

.findElement(By.locator("STRING"))

- Finds and returns a single web element.
- It accepts locator parameter as String.
- Return type is WebElement comes from Selenium library.
- It will throw NoSuchElementException if it cannot somehow find given web element.

**What happens when exception is thrown in Selenium code execution? Is it going to execute the rest of the lines
- Once it throws the exception, the rest of the lines will not be executed UNLESS we handle it.

- When do we have NoSuchElementException?
  #1- Wrong locator provided
  #2- Synchronization: when browser driver and browser are not on the same page, driver will try to find a web element that is not loaded yet. If this happens, it will throw NoSuchElementException.

.click();

- It clicks to the given web element.
- it doesn’t accept any argument
- syntax: driver.findElement(locator).click();

.sendKeys();

- It will pass the provided string into given WebElement.
- It accepts String argument.
- We can pass Keys. commands into sendKeys() method as well.
- **Keys.ENTER**:
- This piece of code will imitate user pressing ENTER from keyboard.

.getText();

- It will get the content from in between the opening tag and closing tag
- Return type is String
- it doesn’t accept any argument

--> We cannot say driver.getText();
syntax: driver.findElement(locator).getText(); --> it will return the text of given web element

--> Ex: TESLA CYBERTRUCK

syntax: driver.findElement(By.name("uh68")).getText() --> TESLA CYBERTRUCK

.getAttribute();

- It will accept an attribute and return its value.
- It accepts a String argument
- Return type is String

--> Ex: TESLA CYBERTRUCK

- syntax: driver.findElement(By.name("uh68")).getAttribute("href") --> https://tesla.com
- syntax: driver.findElement(By.name("uh68")).getAttribute("name") --> uh68
- syntax: driver.findElement(By.name("uh68")).getAttribute("class") --> ff58
- syntax: driver.findElement(By.name("uh68")).getAttribute("id") --> bb22

.isDisplayed():

- It returns boolean value on a given web element.
- If web element is displayed, it will return "true"
  If web element is not displayed, it will return "false”
- It does not accept any argument.
  - syntax: driver.findElement(locator).isDisplayed(); --> true, if displayed
  - syntax: driver.findElement(locator).isDisplayed(); --> false, if not displayed