Web Element Operations in selenium python

Everything on the website is a web element. We usually work with elements like text bar, button, menus, sub-menus, radio button, headers, footer, links, plain texts, forms so on.

When we do perform tests manually, we tend to click elements and will observe the behaviors, and we will set the values, and we will observe the conditions, and we will have our own checkpoints.

Selenium also provides the capability to test these kinds of actions; we can also retrieve the values from the elements to check colors and sizes as selenium cannot perform visual testing.

Click in selenium python bindings

We can perform a clicking operation using selenium python, before performing the click operation we have to find the elements using locators uniquely.

The height and width of the element must be more than 0px to click the element in selenium python. If the element is not enabled and if you perform a click on it, selenium throws InvalidElementStateException.

The below program performs a click on the 'About' link on the google page at the bottom

```
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
```

```python
driver = webdriver.Chrome(executable_path=r'D:/PATH/chromedriver.exe');
# open website
driver.get("https://google.com")
driver.find_element_by_link_text("About").click()
```

Sendkeys in selenium python

sendKeys method sets text into an editable element (text bar, text area, button) without replacing the previously available content.

If you use more than one **sendkeys** on a field then selenium appends all of them; selenium will not replace the existing value when we use sendKeys.

```python
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Chrome(executable_path=r'D:/PATH/chromedriver.exe');
# open website
driver.get("https://google.com")
driver.find_element_by_name("q").send_keys("Tech")
```

Clear an element in selenium python

If we want to set values to a text field, we have to clear the field and set the value using sendkeys.

clear() method will help the user to clear the field on the webpage, if we do not clear a field then also selenium will not throw any exception, but it just appends the new value with the old value.

```python
# open website
driver.get("https://google.com")
driver.find_element_by_name("q").send_keys("jony")
# clear the field
driver.find_element_by_name("q").clear()
# set the value
driver.find_element_by_name("q").send_keys("john")
```

Get Text of an Element in Python Selenium

text variable present in the driver will fetch the values of the text, not only the current element text but also the text of other elements inside that element.

```
# open website
driver.get("https://google.com")
# get the text
print("Text retrieved : "+ driver.find_element_by_link_text("About").text)
Output : Text retrieved : About
```

Get Attribute of an Element in selenium python.

get_attibute() method fetches the value of an attribute, in HTML code whatever is present on the left side of '=' is an attribute, the value on the right side is an Attribute value

```
# open website
driver.get("https://google.com")
# get the title attribute
print("Title of searchbar : "+ driver.find_element_by_id("q").get_attribute("title"))

Output : Title of searchbar : Search
```

Insight about get Attributes:

Below are few insights from my experience about the get attribute.

- Selenium returns the attribute present at the time of the query if the attribute value changes after the query then selenium will not return the new value.
- getAttibute() returns blank as a value if the attribute is not set to any value (exception for boolean values).
- getAttibute() returns 'true' in case if you are searching for a boolean value, and the value is not set.
- Few Boolean attribute examples: checked, readonly, required, multiple, complete so on.
- getAttibute() returns 'null' if there no such attribute

Get CSS Value in selenium python

value_of_css_property method in selenium python fetches the value of a CSS property of a web element in selenium python bindings, and we have to pass CSS property, which selenium binding have to fetch.

Whenever you want to retrieve any CSS property, you should check the property name in the Computed tab and pass the value to the method.

```
# open website
driver.get("https://google.com")
# get the css value
cssValue = driver.find_element_by_name("q").value_of_css_property("font-size")
print("font size searchbar : "+ cssValue)
```

Output

```
Output : font size searchbar : 16px
```

CSS values depend on the browser; you may not get the same values for all the browser, below are few such examples

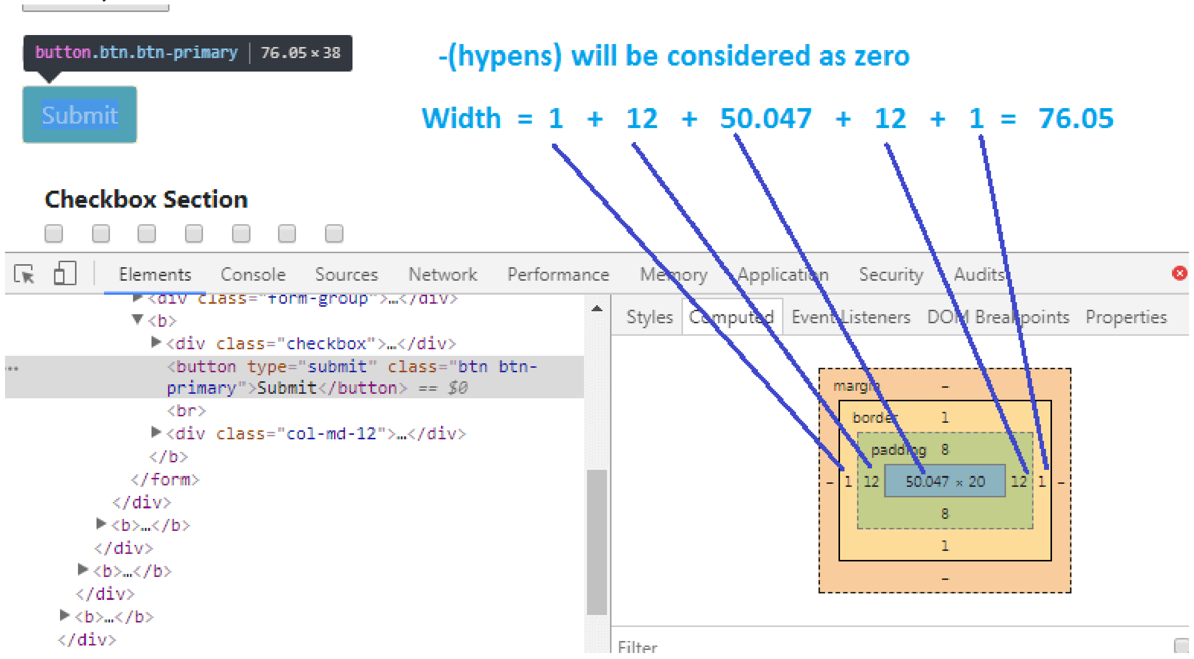
Few browsers may show 1px and others may show 1 px [there is space indifference]

Few browser may show rgba(0, 123, 255, 1) and other may show rgb(0, 123, 255) [rgb and rgba difference].

et Size of an Element in selenium python.

size property present in selenium determines the size of an element, size consists of two values width and height which are the sum of respective attributes.

Width = margin-left + margin-right + padding-left + padding-right + actual width;



-(hypens) will be considered as zero

Width = 1 + 12 + 50.047 + 12 + 1 = 76.05

```
<div class="form-group">…</div>
▼<b>
  ▶<div class="checkbox">…</div>
  <button type="submit" class="btn btn-primary">Submit</button> == $0
  <br>
  ▶<div class="col-md-12">…</div>
</b>
</form>
</div>
▶<b>…</b>
</div>
▶<b>…</b>
</div>
▶<b>…</b>
</div>
```

Height = margin-top + margin-bottom + padding-top + padding-bottom + actual height;

```
# open website
driver.get("https://google.com")
# get the css value
sizeOfElement = driver.find_element_by_name("q").size
print("Width of the searchbar : ", sizeOfElement["width"])
print("Height of the searchbar : ", sizeOfElement["height"])
Output :
Width of the searchbar :  403
Height of the searchbar :  34
```

is_displayed in selenium python

is_displayed() method in selenium python verifies and returns a boolean value based on the state of the element, whether it is displayed or not.

If there is no such attribute as hidden, selenium binding considers that the element is displayed (visit Boolean Attribute for details) and returns true.

```
# open website
driver.get("https://google.com")
# get the css value
isDisplayed = driver.find_element_by_name("q").is_displayed()
print("Is Searchbar Displayed : ", isDisplayed)
```

Output

```
Output : Is Searchbar Displayed :  True
```

is_enabled in selenium python

is_enabled() method in selenium python verifies and returns a boolean value based on the state of the element, whether it is enabled or not.

If there is no such attribute as enabled, python selenium considers that the element is enabled (visit Boolean Attribute for details) and returns true.

```
# open website
driver.get("https://google.com")
# element enabled
isEnabled = driver.find_element_by_name("q").is_enabled()
print("Is searchbar enabled : ", isEnabled)
Output : Is searchbar enabled :  True
```

is_selected in selenium python

is_selected() verifies if an element is selected or not. is_selected() method returns a boolean value, true if the element is selected and false if it is not.

It is widely used on checkboxes, radio buttons, and dropdowns in selenium bindings.

```
# open website
driver.get("https://google.com")
# get the css value
```

```
isSelected = driver.find_element_by_xpath("//input[@id='selected']").is_selected()
print("Is checkbox selected : ", isSelected)
```

The output

```
Output : Is checkbox selected : ", False
```

Location of the element in selenium python

location property in selenium bindings fetches the x and y coordinates of the element. This property returns the dictionary as a return value.

```
driver.get("https://google.com")
# get location
locate = driver.find_element_by_name("q").location
print("x value: ", locate["x"])
print("y value: ", locate["y"])
```

the output

```
Output :
x value:  290
y value:  323
```

-----END-----

Muntasir Abdullah Mizan
muntasir.abdullah01@gmail.com