25. Handle Web Elements (Input/Text Box, Radio Buttons, Checkboxes and Links)

Input Box or Text Box

➔ A textbox or input box is a field in a user interface where users can type and enter text or data.

HandleTextBox.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
options = webdriver.ChromeOptions()
options.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=options)
driver.implicitly_wait(10) # implicit wait
driver.get("https://testautomationpractice.blogspot.com/")
driver.maximize_window()
```

Input box/Text Box /Text Area

```
text_box = driver.find_element(By.XPATH, "//input[@id='name']")
print("Is text box displayed? ", text_box.is_displayed())
print("Is text box enabled? ", text_box.is_enabled())
print("Is text box mandatory field? ", text_box.get_attribute("required") is not None)
print("Placeholder: ", text_box.get_attribute("placeholder"))
print("Max size: ", text_box.get_attribute("maxlength"))
text_box.clear()  # Clear the text box
text_box.send_keys("John Canedy")
print("Entered value: ", text_box.get_attribute("value"))
driver.quit()
```

Radio Button

➔ A radio button is a GUI element that allows users to select one option from a set of mutually exclusive options.

HandleRadioButton.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
options = webdriver.ChromeOptions()
options.add_experimental_option("detach",True)
options.add_argument("--start-maximized")
driver = webdriver.Chrome(options=options)
driver.implicitly_wait(10) # implicit wait
driver.get("https://testautomationpractice.blogspot.com/")
```

Radio Button

```
male_rd = driver.find_element(By.XPATH, "//input[@id='male']")
female_rd = driver.find_element(By.XPATH, "//input[@id='female']")
print("Before selection.....")
print("Select status of the male radio button:", male_rd.is_selected())
print("Select status of the female radio button:", female_rd.is_selected())
```

```
male_rd.click()
print("After selecting male radio button.....")
print("Select status of the male radio button:", male_rd.is_selected())
print("Select status of the female radio button:", female_rd.is_selected())
driver.quit()
```

Checkbox or Tick box or Selection box

➔ A check box or tick box or selection box, is an input field provided to the user on a webpage forms which allows them to select one or more options from a predefined list.

checkbox.py

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
options = webdriver.ChromeOptions()
options.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=options)
driver.implicitly_wait(10) # implicit wait
driver.get("https://testautomationpractice.blogspot.com/")
driver.maximize_window()
```

select specific checkbox

```
driver.find_element(By.XPATH,"//input[@id='monday']").click()
```

select all the checkboxes

```
checkboxes=driver.find_elements(By.XPATH,"//input[@type='checkbox' and contains(@id,'day')]")
print(len(checkboxes)) # 7
```

Appraoch 1	Appraoch 2
`for i in range(len(checkboxes)):` `  checkboxes[i].click()`	`for checkbox in checkboxes:` `  checkbox.click()`

clearing all the checkboxes

```
time.sleep(3)
for checkbox in checkboxes:
  if checkbox.is_selected():
    checkbox.click()
```

select multiple checkboxes by choice

```
for checkbox in checkboxes:
  weekname=checkbox.get_attribute('id')
  if weekname=='monday' or weekname=='sunday':
    checkbox.click()
```

select last 2 checkboxes

totalnumberofelements - 2 = starting index ⇒ range(5,7) → 5,6

```
for i in range(len(checkboxes)-2,len(checkboxes)):
  checkboxes[i].click()
```

select first 2 checkboxes

```
for i in range(len(checkboxes)):
  if i<2:
    checkboxes[i].click()
driver.quit()
```

Link

- ➔ Link is a web element when we click on it it will open another page.
- ➔ **Internal Link:** When you click on a link, it takes you to another part of the same webpage.
- ➔ **External Link:** Clicking on this link will lead you to a different webpage.
- ➔ **Broken Link or Dead Links or Link Rots :** This is a link on a webpage that doesn't lead to any other page. Clicking on it won't take you anywhere. Developers sometimes keep these for future use. When there are errors greater than 400 or more, they are considered error links or broken links.
- ➔ **Reasons for broken links**
  - ◆ A simple reason for broken links is that the webpage or resource originally linked to no longer exists or has been moved to a different location.
  - ◆ This can happen due to website updates, content reorganization, or the removal of the linked content.
  - ◆ As a result, when users click on the link, they encounter an error message instead of being directed to the intended page or resource.

brokenlinks.py

Install requests package through File → Settings → Project Intrepreter → requests

```
import requests
from selenium import webdriver
from selenium.webdriver.common.by import By
options = webdriver.ChromeOptions()
options.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=options)
driver.get("http://www.deadlinkcity.com/")
driver.maximize_window()
driver.implicitly_wait(10) # implicit wait
allLinks=driver.find_elements(By.TAG_NAME,'a')  or  allLinks=driver.find_elements(By.XPATH,'//a')
print("total number of links:",len(allLinks))
count=0
for link in allLinks:
  print(link.text)
  url=link.get_attribute('href')
  try:
    res=requests.head(url)
  except:
    None
  if res.status_code>=400:
```

```
        print(url,"  is broken link")
        count+=1
    else:
        print(url,"   is valid link")
print("Total number of broken links:",count)
driver.quit()
```

Note

1. When we are working with client and server there are some exceptions return by default .
2. To avoid those exceptions we have put in try and except block.

text and get_attribute('name of attribute') methods

- ➔ **text** → This method always returns the inner text of the web element.If no inner text returns nothing.
- ➔ **get_attribute()** → This method returns the value of an attribute or property of a web element. It first checks for a value of property with the given name, then for an attribute value. If neither exists, it returns `None`.

Aspect	text	get_attribute("attribute name")
Purpose	Gets the visible / inner text of an element	Gets the value of the specified attribute of an element
Common Use Case	Capturing text from non-input elements like ,	Fetching value from form fields (<input>, <textarea>)
Handles Hidden Text	No	Yes, can retrieve attribute values from hidden elements
Example Element	Hello World (returns "Hello World")	<input value="UserInput"> (returns "UserInput")

TextVsGetAttribute.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
options = webdriver.ChromeOptions()
options.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=options)
driver.get("https://demowebshop.tricentis.com/")
driver.maximize_window()
driver.implicitly_wait(10)
searchbox = driver.find_element(By.XPATH, "//input[@id='small-searchterms']")
searchbox.send_keys("T-shirts")
```

text

```
print("result of text:",searchbox.text)  # printed nothing
```

get_attribute

```
print("result of get_attribute():",searchbox.get_attribute('class')) #search-box-text ui-autocomplete
print("result of get_attribute():",searchbox.get_attribute('value')) #T-shirts
driver.close()
```