39. Handling or Performing Keyboard & Scroll Wheel Actions

We can use the **`ActionChains`** class with **`Keys`** methods to handle **keyboard actions** and **`scrollWheel`** methods for **scrolling actions.**

Keyboard Actions in Selenium

- ➔ **key_down()** ⇒ Simulates pressing a key down, typically used to combine with other key actions.
  - ◆ **Syntax:** actions.key_down(Keys key).perform()
- ➔ **key_up()** ⇒ Simulates releasing a key that was pressed down.
  - ◆ **Syntax:** actions.key_up(Keys key).perform()
- ➔ **send_keys()** ⇒ Sends a sequence of keystrokes to the currently focused element or a specified element.
  - ◆ **Syntax:** actions.send_keys(WebElement target, String keys).perform()

KeyboardActionsDemo.py

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

Initialize ActionChains

```
actions = ActionChains(driver)
```

Input box 1

```
inputbox1 = driver.find_element(By.XPATH, "//input[@id='input1']")
actions.send_keys_to_element(inputbox1, "Welcome").perform()  # Send text to inputbox1
```

Select all text (Ctrl+A)

```
actions.key_down(Keys.CONTROL).send_keys("a").key_up(Keys.CONTROL).perform()
```

Copy text (Ctrl+C)

```
actions.key_down(Keys.CONTROL).send_keys("c").key_up(Keys.CONTROL).perform()
```

Press Tab key twice to move focus

```
actions.send_keys(Keys.TAB).perform()
actions.send_keys(Keys.TAB).perform()
```

Input box 2 (Paste text with Ctrl+V)

```
actions.key_down(Keys.CONTROL).send_keys("v").key_up(Keys.CONTROL).perform()
```

Press Tab key twice to move focus

```
actions.send_keys(Keys.TAB).perform()
actions.send_keys(Keys.TAB).perform()
```

Input box 3 (Paste text with Ctrl+V)

```
actions.key_down(Keys.CONTROL).send_keys("v").key_up(Keys.CONTROL).perform()
```

```
driver.quit()
```

Scroll Wheel Actions in Selenium

- ➔ **scroll_to_element()** ⇒ Scrolls the page until the specified element is in view.
  - ◆ **Syntax:** actions.scroll_to_elemen(WebElement target).perform()
- ➔ **scroll_by_amount()** ⇒ Scrolls the page by a specific number of pixels horizontally andvertically.
  - ◆ **Syntax:** actions.scroll_by_amount(xOffset, yOffset).perform()
- ➔ **scroll_from_origin()** ⇒ Scrolls from the current location by a specific amount.
  - ◆ **Syntax:** actions.scroll_from_origin(scroll_origin: ScrollOrigin, xOffset, yOffset).perform()

ScrollWheelActionsDemo.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver import ActionChains
from selenium.webdriver.common.actions.wheel_input import ScrollOrigin
opt = webdriver.ChromeOptions()
opt.add_experimental_option("detach",True)
opt.add_argument("--start-maximized")
driver = webdriver.Chrome(options = opt)
driver.implicitly_wait(10)
driver.get("https://demowebshop.tricentis.com/")
```

Initialize ActionChains

```
actions = ActionChains(driver)
```

Scroll to element

```
footer = driver.find_element(By.XPATH, "//div[@class='footer-disclaimer']")
actions.scroll_to_element(footer).perform()
```

Scroll by amount

```
actions.scroll_by_amount(0, 200).perform()  # Scrolls down by 200 pixels
```

Alternative way to scroll by amount, using footer element position

```
deltaY = footer.rect['y']
actions.scroll_by_amount(0, deltaY).perform()
```

Scroll from origin

```
origin = ScrollOrigin.from_element(footer)
actions.scroll_from_origin(origin, 0, -200).perform()  # Scrolls up by 200 pixels from the footer
driver.quit()
```

ScrollingInsideTheDropDownDemo.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver import ActionChains
opt = webdriver.ChromeOptions()
opt.add_experimental_option("detach",True)
opt.add_argument("--start-maximized")
driver = webdriver.Chrome(options = opt)
```

```
driver.implicitly_wait(10)
driver.get("https://testautomationpractice.blogspot.com/")
                    Click on the combobox to open the dropdown
driver.find_element(By.XPATH, "//input[@id='comboBox']").click()
                    Initialize ActionChains
actions = ActionChains(driver)
                    Find the dropdown item to scroll to
item = driver.find_element(By.XPATH, "//div[normalize-space()='Item 100']")
                    Scroll to the item inside the dropdown
actions.scroll_to_element(item).perform()
                    Print the text of the item
print(item.text)
driver.quit()
```

ScrollingInsideTheTable.py

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver import ActionChains
opt = webdriver.ChromeOptions()
opt.add_experimental_option("detach",True)
opt.add_argument("--start-maximized")
driver = webdriver.Chrome(options = opt)
driver.implicitly_wait(10)
driver.get("https://datatables.net/examples/basic_init/scroll_xy.html")
                    Initialize ActionChains
actions = ActionChains(driver)
time.sleep(3)
                    Vertical scrolling: Scroll to the element with last name 'Kelly'
lastname_element = driver.find_element(By.XPATH, "//td[normalize-space()='Kelly']")
actions.scroll_to_element(lastname_element).perform()
time.sleep(3)
                    Horizontal scrolling: Scroll to the element with email 'c.kelly@datatables.net'
email_element = driver.find_element(By.XPATH, "//td[normalize-space()='c.kelly@datatables.net']")
actions.scroll_to_element(email_element).perform()
driver.quit()
```