

---

## Action Chains

---

Handling special keyboard and mouse events are done using the **Advanced User Interactions API**.

It contains the **Action Chains** class which provides the methods to perform these actions.

Action Chains are a way to automate low level interactions such as mouse movements, mouse button actions, key press, and context menu interactions.

To use the action chains,

First we need to create the object of **Action Chains**, Which takes one parameter of type web driver.

**Syntax:**

```
ActionChains(webdriver.Driver)
```

Here, we need to pass the web driver instance to action chains class to pass the control from web driver to action chains, so that action chains class perform the actions like mouse over, right click, drag drop etc.

**IMPORTANT:**

- Whenever we use any methods of action class. We should use the perform method to generate the actions on the browser.

Generating user actions,

When you call methods for actions on the action class object. The action are stored in a queue in the action chains object. In order to fire the actions stored in the action class object we need to call perform () function. The events are fired in the order they are queued.

Following are the methods provided by the **Action Chains** to perform the mouse related actions.

```
action.move_to_element(to_element)
action.move_by_offset(xoffset, yoffset)
action.click(on_element)
action.click_and_hold(on_element)
action.context_click(on_element)
action.double_click(on_element)
action.drag_and_drop(source, target)
action.drag_and_drop_by_offset(source, xoffset, yoffset)
action.pause(seconds)
action.perform()
action.release(on_element)
action.send_keys()
action.send_keys_to_element(element)
```

## 1. Move to element

- Moves the mouse to the middle of the element.

Syntax:

MoveToElement (To Element)

- Parameter-> to element – element on which we want to move the cursor.

Write a script to move the cursor on element?

```
1 from selenium import webdriver
2 from selenium.webdriver.common.action_chains import ActionChains
3
4 driver = webdriver.Chrome()
5 driver.maximize_window()
6 driver.get("https://www.actimind.com/")
7 driver.implicitly_wait(30)
8
9 action = ActionChains(driver)
10 expertiseEle = driver.find_element_by_xpath("//a[contains(text(),'AREAS OF EXPERTISE')]")
11 action.move_to_element(expertiseEle)
12 action.perform()
13
14 driver.close()
```

## 2. Move by offset

- Moving the mouse to an offset from current mouse position.

Syntax:

move\_by\_offset (xOffset, yOffset)

Write a program to move the cursor by offset

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
import time

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

expertiseEle = driver.find_element_by_xpath("//a[contains(text(),'AREAS OF EXPERTISE')]")
action = ActionChains(driver)
loc = expertiseEle.location
x = loc.get('x')
y = loc.get('y')
action.move_by_offset(x, y).perform()

time.sleep(5)
driver.close()
```

**Composite Actions:**

- Performing multiple actions at the same time is called as composite actions.

### 3. Click

- Clicks on the element

Syntax: Click (On Element)

- Clicks an element

Write a script to click on element?

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//a[contains(text(),'AREAS OF EXPERTISE')]")
msOffice = driver.find_element_by_xpath("//a[contains(text(),'MS Office Plug-ins')]")
action.move_to_element(aboutCompany).click(msOffice).perform()

time.sleep(5)

driver.close()
```

### 4. Pause:

- Pause all inputs for the specified duration in seconds.
- We use this function, whenever we are performing multiple actions at a time, to make the action chain actions to happen with some time difference or else all the actions try to execute at a time and we end up not performing the actions properly.

```
1 action.pause("seconds")
```

Write a script to move the cursor with pause method?

```
1 from selenium import webdriver
2 from selenium.webdriver.common.action_chains import ActionChains
3
4 driver = webdriver.Chrome()
5 driver.maximize_window()
6 driver.get("http://www.istqb.in/")
7 driver.implicitly_wait(30)
8
9 action = ActionChains(driver)
10 foundation = driver.find_element_by_xpath("//span[text()='FOUNDATION']")
11 action.move_to_element(foundation).pause(1)
12 enrollment = driver.find_element_by_xpath("//span[text()='ENROLLMENT']")[1]
13 action.move_to_element(enrollment).pause(1)
14 corporateEnrollment = driver.find_element_by_xpath("//span[text()='CORPORATE ENROLLMENT']")[1]
15 action.move_to_element(corporateEnrollment).pause(1)
16 onlineEnrollment = driver.find_element_by_xpath("//span[text()='ONLINE ENROLLMENT']")
17 action.click(onlineEnrollment)
18 action.perform()
19
20 driver.close()
```

- In the above example it will make all the actions to wait for one second to perform the actions.

## 5. Context Click

- Performs the context click (right click) at the current mouse location.

Syntax:

Context\_click ()

Write a program to context click

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

action = ActionChains(driver)
action.context_click().perform()

time.sleep(5)

driver.close()
```

Write a script to right click on the element?

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//a[contains(text(),'ABOUT COMPANY')]")
action.context_click(aboutCompany).perform()

time.sleep(5)

driver.close()
```

## Handling the context menu bar

- When we right click on any element we get list of options which is called as context menu.
- To select the required option present in the context menu we use the shortcuts, IE, "t" to select open link in new tab, "w" to select link in new window.
- Since we cannot inspect the context menu, selenium cannot perform any operation on context menu.
- In order to handle the context menu we use **Keyboard module**.
- We type the shortcut using the **Keyboard**.

### Step 1: open the command prompt

C:\&gt;

C:\Windows\System32\cmd.exe

C:\&gt;

- Below scripts selects the “open link in new tab”

**Pramod K S**

## Write the script to right click on a link and open the link in new tab

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
import keyboard

driver = webdriver.Chrome()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://www.actimind.com/")

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//a[contains(text(),'ABOUT COMPANY')]")
action.context_click(aboutCompany)
action.perform()
keyboard.press("w")

driver.quit()
```

## 6. Send keys

- Sends the series of keystrokes onto element

Syntax: Send\_keys (on element key strokes)

## Write a program to open the link in new tab?

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//a[contains(text(),'ABOUT COMPANY')]")
action.context_click(aboutCompany).send_keys(Keys.CONTROL + Keys.ENTER).perform()

time.sleep(5)
|
driver.close()
```

## Write a script to launch the link in new window?

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://www.actimind.com/")
driver.implicitly_wait(30)

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//a[contains(text(),'ABOUT COMPANY')]")
action.context_click(aboutCompany).send_keys(Keys.SHIFT + Keys.CONTROL ).perform()

time.sleep(5)
driver.quit()
```

## 7. Click and hold

- Holds down the left mouse button on an element.

Write a script to click and hold the link?

```
import time

from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome()
driver.maximize_window()
driver.get("https://github.com/SeleniumHQ")
driver.implicitly_wait(30)

action = ActionChains(driver)
aboutCompany = driver.find_element_by_xpath("//span[contains(text(), 'www.seleniumhq.org')]")
action.click_and_hold(aboutCompany)
action.release(aboutCompany)
action.perform()

time.sleep(5)

driver.quit()
```

## 8. Double click

- Double-clicks an element.

Sample HTML:

```
<html>
<title>Qspiders</title>
<body>
  <div class="box">
    <span onclick="window.location='http://www.google.com'" style="color:blue;text-decoration: underline;">Double Click Here</span>
  </div>
</body>
</html>
```

Write a script to double on the element?

```
1 import time
2
3 from selenium import webdriver
4 from selenium.webdriver.common.action_chains import ActionChains
5
6 driver = webdriver.Chrome()
7 driver.maximize_window()
8 driver.get("file:///C:/Users/PriyaPramod/Desktop/HTML%20Pages/doubleclickq.html")
9 driver.implicitly_wait(30)
10
11 doubleClick = driver.find_element_by_xpath("//div[@class='box']/span")
12 actions = ActionChains(driver)
13 actions.double_click(doubleClick).perform()
14
15 time.sleep(5)
16
17 driver.close()
```

## 9. Drag and Drop

- Holds down the left mouse button on the source element, then moves to the target element and releases the mouse button.

### Write a script to drag and drop the element?

```

1 from selenium import webdriver
2 from selenium.webdriver.common.action_chains import ActionChains
3 import time
4
5 driver = webdriver.Chrome()
6 driver.maximize_window()
7 driver.get("https://demo.vtiger.com/")
8 driver.implicitly_wait(30)
9
10 driver.find_element_by_xpath("//button[text()='Sign in']").click()
11 source = driver.find_element_by_xpath("//b[contains(text(),'Total Amount by Sales stage')]")
12 target = driver.find_element_by_xpath("//b[contains(text(),'Top Opportunities')]")
13
14 action = ActionChains(driver)
15 action.drag_and_drop(source, target).perform()
16
17 time.sleep(5)
18
19 driver.close()

```

### Write a script to drag and drop the element using offset?

```

1 from selenium import webdriver
2 from selenium.webdriver.common.action_chains import ActionChains
3 import time
4
5 driver = webdriver.Chrome()
6 driver.maximize_window()
7 driver.get("https://demo.vtiger.com/")
8 driver.implicitly_wait(30)
9
10 webEle = driver.find_element_by_xpath("//button[text()='Sign in']")
11
12 driver.find_element_by_xpath("//button[text()='Sign in']").click()
13 source = driver.find_element_by_xpath("//b[contains(text(),'Total Amount by Sales stage')]")
14
15 action = ActionChains(driver)
16 action.drag_and_drop_by_offset(source, 450, 789).perform()
17
18 time.sleep(5)
19
20 driver.close()

```