

SELENIUM LOCATORS

By.id()
By.name()
By.linkText()
By.partialLinkText()
By.tagName()
By.className()
By.xpath()
By.cssSelector()

DROP DOWN

selectByindex(int index)
selectByVisibleText(String text)
selectByValue(String value)

ELEMENT VALIDATION

Returns boolean value

isEnabled()
isDisplayed()
isSelected()

HANDLE ALERTS

driver.switchTo().alert().getText()
driver.switchTo().alert().accept()
driver.switchTo().alert().dismiss()
driver.switchTo().alert().sendKeys()

Exceptions:

NoAlertPresentException - You did not expect an alert, so are not switch to it but you are working on the main page even before handling the alert.

UnhandledAlertException - You did not expect an alert, so did not switch but alert appeared unexpectedly

SELENIUM NAVIGATORS

Navigate to url :

driver.get()
driver.navigate().to()

Refresh page :

driver.navigate().refresh()

Navigate forwards in browser history :

driver.navigate().forward()

Navigate backward in browser history :

driver.navigate().back()

GET METHODS IN SELENIUM

getTitle() :

to retrieve the current title of the webpage

getCurrentUrl() :

to retrieve the current URL of the webpage

getPageSource() :

to retrieve the current page source of the webpage

getText() :

to retrieve the text of the specified web element

getAttribute() :

to retrieve the value specified in the attribute

FRAMES

driver.switchTo().frame(int frameNumber)
driver.switchTo().frame(String frameNameOrID)
driver.switchTo().frame(WebElement element)
driver.switchTo().defaultContent()
driver.switchTo().parentFrame()

Exceptions:

NoSuchFrameException - When no frame of (index/name/id) found

NoSuchElementException - When using frame as WebElement to switch

WINDOW HANDLING

getWindowHandle() - String is a return type
getWindowHandles() - Returns Set<String>
driver.switchTo().window(windowHandle)
driver.close()
driver.quit()

Exception:

NoSuchWindowException - When the window handle does not exist

TAKE SCREENSHOT

A Screenshot in Selenium Webdriver is used for bug analysis.

Syntax:

```
File src = driver.getScreenshotAs  
(OutputType.FILE);  
FileUtils.copyFile(src, new File(path));
```

MOUSE ACTIONS

click()
clickAndHold()
contextClick()
doubleClick()
dragAndDrop(source, target)
dragAndDropBy(source, xoffset, y-offset)
moveToElement(toElement)
moveByOffset(x-offset, y-offset)
release()

KEYBOARD ACTIONS

sendKeys()
keyUp()
keyDown()

IMPLICIT WAIT VS EXPLICIT WAIT

IMPLICIT WAIT

- This wait can be considered as element detection timeout. Once defined in a script, this wait will be set for all the Web Elements on a page.
- Selenium keeps polling to check whether that element is available to interact with or not.
`driver.manage().timeouts().implicitlyWait(90, TimeUnit.SECONDS);`

EXPLICIT WAIT

- This wait can be considered as conditional wait, and is applied to a particular Web Element with a condition

```
WebDriverWait wait = new WebDriverWait  
(Driver, 90);
```

FINDELEMENT VS FINDELEMENTS

FINDELEMENT

Returns a single WebElement

Syntax: WebElement findElement (By.locator())

Exception:

NoSuchElementException

FINDELEMENTS

Does not throw any exception

Returns an Empty List of WebElement Object

Syntax: java.util.List<WebElement>
findElements(By.locator())

Exception:

EmptyListofWebElementObject
