

## Selenium Locators

<b>Locating by ID</b>	<code>driver.findElement(By.id("q")).sendKeys("Selenium 3");</code>
<b>Locating by Name</b>	<code>driver.findElement(By.name("q")).sendKeys("Selenium 3");</code>
<b>Locating by Xpath</b>	<code>driver.findElement(By.xpath("//input[@id='q']")).sendKeys("Selenium 3");</code>
<b>Locating Hyperlinks by Link Text</b>	<code>driver.FindElement(By.LinkText("edit this page")).Click();</code>
<b>Locating by DOM</b>	<code>dom =document.getElementById('signinForm')</code>
<b>Locating by CSS</b>	<code>driver.FindElement(By.CssSelector("#rightbar &gt; .menu &gt; li:nth-of-type(2) &gt; h4"));</code>
<b>Locating by ClassName</b>	<code>driver.findElement(By.className("profile-header"));</code>
<b>Locating by TagName</b>	<code>driver.findElement(By.tagName("select")).Click();</code>
<b>Locating by LinkText</b>	<code>driver.findElement(By.linkText("NextPage")).click();</code>
<b>Locating by PartialLinkText</b>	<code>driver.findElement(By.partialLinkText("NextP")).click();</code>

## Handle Alerts/ Pop-ups

<code>driver.switchTO().alert.getText()</code>	to retrieve the alert message
<code>driver.switchTO().alert.accept()</code>	to accept the alert box
<code>driver.switchTO().alert.dismiss()</code>	to cancel the alert box
<code>driver.switchTO().alert.sendKeys("Text")</code>	to send data to the alert box

## Annotations

TestNG	@BeforeSuite @AfterSuite @BeforeTest @AfterTest @BeforeGroups @AfterGroups @BeforeClass @AfterClass @BeforeMethod @AfterMethod
JUnit	@After @AfterClass @Before @BeforeClass @Ignore @Test

## Handle multiple windows and tabs

<code>getWindowHandle()</code>	used to retrieve the handle of the current page (a unique identifier)
<code>getWindowHandles()</code>	used to retrieve a set of handles of the all the pages available
<code>driver.switchTo().window("windowName/handle")</code>	switch to a window
<code>driver.close()</code>	closes the current browser window

```
driver.quit();
```

This method Closes all windows opened by the WebDriver.

## Practice Script

Launch Webpage	<code>driver.get("www.webdriverinselenium.com");</code>
Click Button	<code>driver.findElement(By.id("submit")).click();</code>
Store Text	<code>String txtDropdown = driver.findElement(By.ID("select")).getText();</code>
Enter Text	<code>driver.findElement(By.xpath("//input[@name='FirstName034']")).sendKeys("Shaheryar");</code>
Handle Alert	<code>Alert Alertpopup = driver.switchTo().alert();</code>
(Mouse)Click	<code>driver.findElement(By.xpath("//input[@value='OBJECT NAME']")).click();</code>
Compare Text	<code>Assert.assertTrue(chkbox12.isSelected());</code>
Disable a Field	<code>driver.getElementsByName('&lt;ObjectID&gt;')[0].setAttribute('disabled', "");</code>
Enable a Field	<code>driver.getElementsByName('&lt;ObjectID&gt;')[0].removeAttribute('disabled');</code>
Screenshot	<code>File snapshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);</code> <code>FileUtils.copyFile(snapshot, new File("C:\\\\screenshot.jpg"));</code>
Print the Title of the Page	<code>String pagetitle = driver.getTitle();</code> <code>System.out.print(pagetitle);</code>
Implicit Wait	<code>driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);</code>
Explicit Wait	<code>WebDriverWait wait = new WebDriverWait(driver, 20);</code> <code>wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath("//div[@id='&lt;Object ID&gt;']"), "Enter Your Personal Details"));</code>
Fluent Wait	<code>Wait wait = new FluentWait(driver)</code> <code>.withTimeout(20, SECONDS)</code>
Sleep	<code>Thread.Sleep(10);</code>

## Difference between Implicit & Explicit Wait

The **Implicit Wait in Selenium** is used to tell the web driver to wait for a certain amount of time before it throws a "No Such Element Exception". The default setting is 0. Once we set the time, the web driver will wait for the element for that time before throwing an exception.

The **Explicit Wait in Selenium** is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or maximum time exceeded before throwing "ElementNotVisibleException" exception. It is an intelligent kind of wait, but it can be applied only for specified elements. It gives better options than implicit wait as it waits for dynamically loaded Ajax elements.