What are different ways of locating elements in selenium?

To locate an element by its id, we use

<div id="name">

|  |  |
| --- | --- |
|  | driver.findElement(By.id("name"));  By **Class Name:**  Here, the “Class” refers to the attribute on the DOM element which is related to a CSS property  <span class="even">  driver.findElement(By.className("even"));  By **Tag Name:**  The tag name of the DOM Element  <img src="./logo.png">  driver.findElement(By.tagName("img"));  By **Name:**  Find the input element with matching name attribute  <input name="color" value="blue">  driver.findElement(By.linkText("Software Testing"));  driver.findElement(By.name("color"));  By **Link Text:**  Find the link element with matching visible text  <ahref="http://www.testingexcellence.com/">SoftwareTesting</a>  driver.findElement(By.linkText("Software Testing"));  By **Partial Link Text:**  Find the link element with partial matching visible text  <a href="http://www.testingexcellence.com/">Softwar Testing</a>  driver.findElement(By.partialLinkText("Testing"));  different types of waits or synchronization in selenium web driver?  It is a mechanism which involves more than one components to work parallel with Each other.  Generally in Test Automation, we have two components **1. Application Under Test** **2. Test Automation Tool.**  Synchronization can be classified into two categories:  **1. Unconditional** **2. Conditional Synchronization**  **Unconditional :** In this we just specify timeout value only. We will make the tool to wait until certain amount of time and then proceed further.  *Examples: Wait() and [Thread.Sleep();](https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html" \t "_blank)*  **Conditional Synchronization:**  We specify a condition along with timeout value, so that tool waits to check for the condition and then come out if nothing happens.  **Implicit Wait.**  An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available.  driver.manage.TimeOuts.implicitwait(6,Timeunit.SECONDS);  WebDriver driver = **new** FirefoxDriver(); driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS); driver.**get**("[http://www.google.com"](http://www.google.com/));  **Explicit Wait:**  We need to define a wait statement for certain condition to be satisfied until the specified timeout period. If the Webdriver finds the element within the timeout period the code will get executed.  /\*Explicit **wait** **for** **state** dropdown field\*/     WebDriverWait **wait** = new WebDriverWait(driver, 10);     **wait**.**until**(ExpectedConditions.visibilityOfElementLocated(By.id("statedropdown")));  **Fluent Wait:**  Using Fluent Wait we can define the maximum amount of time to wait for a condition, as well as the frequency with which to check for the condition  Wait<WebDriver> wait = new FluentWait<WebDriver>(driver) //Wait for the condition        .withTimeout(30, TimeUnit.SECONDS)          // which to check for the condition with interval of 5 seconds.        .pollingEvery(5, TimeUnit.SECONDS)      //Which will ignore the NoSuchElementException        .ignoring(NoSuchElementException.class);  how to lanuch webpage using chrome driver?  System.setProperty("webdriver.chrome.driver","C://chromedriver.exe");  WebDriver driver=new ChromeDriver();  driver.get("http://www.google.co.in");  what is desired capabilities in selenium webdriver?  difference between driver.close and driver.quit methods?   WebDriver.Close() This method is used to close the current open window. It closes the current open window on which driver has focus on.  2) WebDriver.Quit() This method is used to destroy the instance of WebDriver. It closes all Browser Windows associated with that driver and safely ends the session. WebDriver.Quit() calls Dispose.  3) WebDriver.Dispose() This method closes all Browser windows and safely ends the session  how to handle multiple windows in selenium web driver?  **import**java.util.List; **import**org.junit.After; **import**org.junit.Before; **import**org.junit.Test; **import**org.openqa.selenium.By; **import**org.openqa.selenium.WebDriver; **import**org.openqa.selenium.WebElement; **import**org.openqa.selenium.firefox.FirefoxDriver; **public class**MultipleWindowsHandle {  WebDriver driver;        @Before        **public void**setup() **throws**Exception {        driver=**new**FirefoxDriver();        String URL="http://www.seleniummaster.com";         driver.get(URL);        driver.manage().window().maximize();        }        @Test        **public void**test() **throws**Exception {         // Opening site        driver.findElement(By.xpath("//img[@alt='SeleniumMasterLogo']")).click();        // Storing parent window reference into a String Variable        String Parent\_Window = driver.getWindowHandle();           // Switching from parent window to child window         **for**(String Child\_Window : driver.getWindowHandles())        {        driver.switchTo().window(Child\_Window);        // Performing actions on child window        driver.findElement(By.id("dropdown\_txt")).click();        List  dropdownitems=driver.findElements(By.xpath("//div[@id='DropDownitems']//div"));        **int**dropdownitems\_Size=dropdownitems.size();        System.out.println("Dropdown item size is:"+dropdownitems\_Size);        ((WebElement) dropdownitems.get(1)).click();        driver.findElement(By.xpath("//\*[@id='anotherItemDiv']")).click();        }        //Switching back to Parent Window        driver.switchTo().window(Parent\_Window);        //Performing some actions on Parent Window        driver.findElement(By.className("btn\_style")).click();        }         @After         **public void**close() {         driver.quit();         }         } |