To perform operations on elements 🡪

1. get() – Opens a specific URL

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

1. getTitle() - Returns the browser title

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

String browserTitle = driver.getTitle();

System.out.println(browserTitle);

1. getCurrentUrl() - Returns the current url of the browser

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

String browserURL = driver.getCurrentUrl();

System.out.println(browserURL);

Browser Navigation Methods 🡪

1. navigate().to() - loads a new page in browser window

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

driver.navigate().to(“http://www.rediff.com”);

1. navigate().back() - Moves a single item back in the browser history

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

Thread.sleep(3000);

driver.navigate().to(“http://www.rediff.com”);

driver.navigate().back();

1. navigate().forward() - Moves a single item forward in the browser history

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

Thread.sleep(3000);

driver.navigate().to(“http://www.rediff.com”);

Thread.sleep(3000);

driver.navigate().back();

Thread.sleep(3000);

driver.navigate().forward();

1. navigate().refresh() - Refreshes the current web page

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

driver.navigate().refresh();

1. close() - Closes the focused browser

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

driver.close();

1. quite() - closes all browsers that opened by webdriver during execution

WebDriver driver = new ChromeDriver();

driver.get(“https://google.co.in/”);

driver.findElement(By.linkTest(“Sign In”)).click();

driver.quite();

1. findElement() - finds first element within webpage using given locator

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.ElementLocator(“value”));

1. sendKeys() - Enters a value in edit box

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.id(“Email”)).sendKeys(“India12”);

1. clear() - Clears the value from edit box

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.id(“Email”)).clear();

1. click() - Clicks an element like button, link, radio button, select/unselect/check box

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.name(“Google Search”)).click();

1. isDisplayed() - Checks if the element is displayed or not in current webpage & returns Boolean vaue

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

boolean a = driver.findElement(By.id(“next”)).isDisplayed();

System.out.println(a);

OR

System.out.println(driver.findElement(By.id(“next”)).isDisplayed());

1. isEnabled() - Checks if element is in enabled state or not & returns Boolean vaue

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

boolean a = driver.findElement(By.id(“next”)).isEnabled();

System.out.println(a);

1. isSelected() - Checks if element is selected or not

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.xpath(“html/body/input[2]”)).click();

System.out.println(driver.findElement(By.xpath(“html/body/input[2]”))).isSelected());

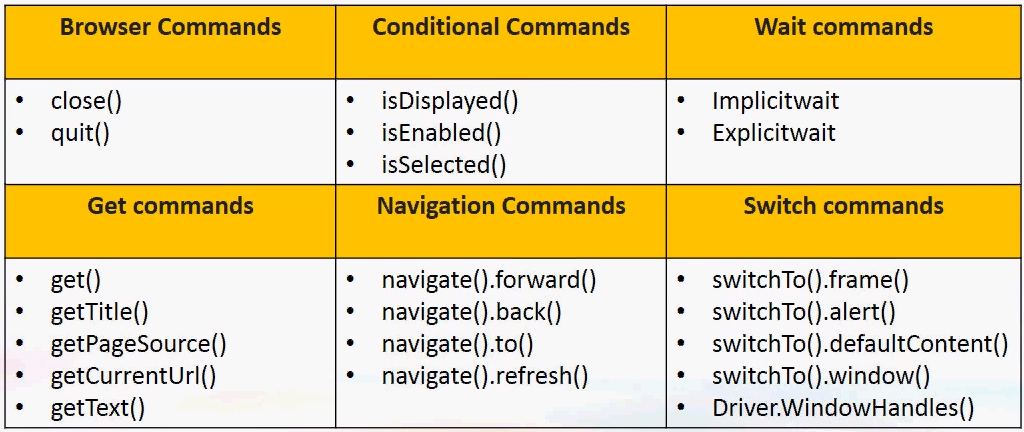
1. getText() - Returns specified element’s text value

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

String pageHeader = driver.findElement(By.tagName(“h1”)).getText();

System.out.printlm(pageHeader);



**Element Locators**

* Element locators are HTML properties of web elements.
* It is address that identifies a web element uniquely in a web page.
* Selenium supports 8 types of element locators.
* Xpath is unique but takes time, ID (also unique) is faster in execution

Different locators in Selenium –

* id,
* name,
* className,
* tagName,
* linkText,
* partialLinkText,
* cssSelector,
* xpath

1. id

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.id(“identifierId”)).sendKeys(“abds@gmail.com”)

OR

WebElement Email = driver.findElement(By.id(“identifierId”));

Email. sendKeys(“abds@gmail.com”);

Here, findElement - webDriver Command

By - predefined class

Id - Element locator

1. name

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(by.name(“username”)).sendKeys(“admin”);

1. className

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.className(“gb\_P”)).click();

1. tagName

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.tagName(“input”)).sendKeys(“xyz@gmail.com”);

1. linkText

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.linkText(“Gmail”)).click();

1. patialLinkText

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.partialLinkText(“ma”)).click();

1. xpath

WebDriver driver = new ChromeDriver();

driver.get(“https://www.gmail.com/”);

driver.findElement(By.xpath(“//input[@id=’login-signin’]”)).click();

1. cssSelector

WebDriver driver = new ChromeDriver();

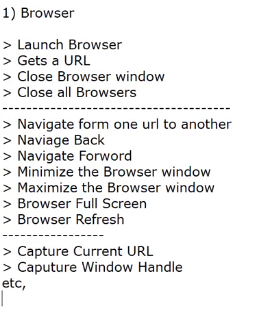
driver.get(“https://www.gmail.com/”);

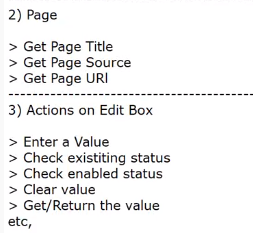
driver.findElement(By.cssSelector(“#login-signin”)).click();

**Different Web Elements in Selenium and Actions on them**

* Browser
* Page
* Edit Box
* Text Box
* Link
* Button
* Image, Image Button, Image Link
* Text Area, Error Message, PopUp Window
* Checkbox
* Radio Button
* Dropdown box
* List box
* Combo box
* List box
* Combo box
* Web table / HTML table
* Frame

Actions 🡪





**Wait 🡪**

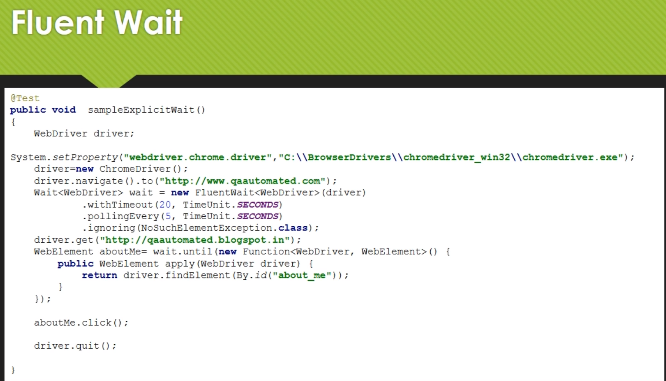
Implicit Wait – driver.manage().timeouts().implicitlyWait(3000, TimeUnit.SECONDS);



WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.ElementToBeClickable(By.name(“abcd”)))





**Frames** 🡪 Frames are area / section under webpage. By using frames, user can divide webpage in different PODs.

Frames are something that are built over your application. Selenium can not handle frames directly. We need to have some code to switch from browser to frame.

driver.switchTo().frame(id)

We can pass dram id / name / frame element

iFrames are used to show information from other webpage.

Handling frame using web element

WebElement frame = driver.findElement(By.xpath(“//[@id’Single’]/iframe”));

Driver.switchTo().frame(frame);

driver.findElement(By.xpath(“html/body/section/div/div/div/input”)).sendKeys(“Selenium”)

To come out of frame to home page

driver.switchTo().defaultContent();

**Handling Windows**

Set <String> s = driver.getWindowHandles(); //will return set of window IDs

It = s.iterator();

It.next(); // 0th index of window

It.next(); // 1th index of window

String windowid3rd = It.next();

driver.switchTo.window(windowid3rd);

getWindowHandles() 🡪 returns the details of all the windows that are open in browser

getWindowHandle() 🡪 returns the details of particular window

**xpath**

xpath syntax if id is given 🡪 //tagname[@id=’value’]

xpath syntax if css is given 🡪 //tagname[id=value]

Regular expressions for xpath. If id is changing dynamically (say after refreshing the page it becomes u012, u056, u035), then

Contains regular expression for xpath - //tagname[contains(@id,’u0’)]

CSS regular expression for xpath - tagname[id\*=’u0’]

Xpath syntax is always as 🡪 //tagname[@Attribute=’Value’]

Tagname like div, input etc

Attribute like id

Absolute xpath – direct way to find element. Disadvantage is small change in page will fail xpath. It starts with root node html.

e.g. - /html/body/div[1]/section/div[1]/div

Relative xpath – It starts from middle of DOM structure. It starts with double slash (//)

e.g. - //input[@id =’email’]

Functions of xpath 🡪

1. Contains () 🡪 //\*[contains(@type,’sub’)]

e.g. //img[contains(@src,’newone’)]

1. Starts-with () 🡪

e.g. //img[starts-with(@alt,’shop’)]

1. Text () 🡪

e.g. //\*[text()=’Orders’]

Combining two functions together 🡪

//input[@name=’username’] | //input[@id=’login-username’]

**Handling Windows / Tabs 🡪**