Driver Initialization Basics:
Firefox WebDriver driver = new FirefoxDriver();
Chrome WebDriver driver = new ChromeDriver();
Internet Explorer WebDriver driver = new SafariDriver();
Safari Driver WebDriver driver = new InternetExplorerDriver();

Selenium Locators:
by ID
driver.findElement(By.id("q")).sendKeys("Se lenium 3");
by Name
driver.findElement(By.name("q")).sendKeys ("Selenium 3");
by Xpath
driver.findElement(By.xpath("//input[@id='q'])).sendKeys("Selenium 3");
Hyperlinksby Link Text
driver.FindElement(By.LinkText("edit this page")).Click();
Locating by DOM
dom =document.getElementById('signinForm')
by CSS
driver.FindElement(By.CssSelector("#rightbar> .menu >li:nth-of-type(2) > h4"));
Locating by ClassName
driver.findElement(By.className("profileheader"));
TagName
driver.findElement(By.tagName("select")).C lick();
by LinkText
driver.findElement(By.linkText("NextP age")).click();
PartialLinkText
driver.findElement(By.partialLinkText(" NextP")).click();

Xpath text/sublings/parent
driver.findElement(By.xpath("//ul[@class='responsive-tabs__list']/li[1]/following-sibling::li[2]")).click();

System.out.println(driver.findElement(By.xpath(".//*[@id='tablist1-tab2']/parent::ul")).getAttribute("role"));

//

JUNIT
@Test: test method to run with public void return type
@After: method to run after test method
@AfterClass: method to run before test method
@Before: method to run before test method

@BeforeClass: method to run once before any of the test methods in the class have been executed
@Ignore: This annotation is used to ignore a method

TestNG:
@test: This annotation marks a class or method as a part of a test
@BeforeSuite: This annotation makes sure that the method only run once before all the tests have run
@AfterSuite: This annotation makes sure that the method runs once after the execution of all the tests
@BeforeTest: This annotation will make sure that the method marked with this annotation runs before the first method annotated with @test
@AfterTest: This annotation will make sure that the method marked with this annotation runs after all the methods annotated with @test execute all the classes inside <test> tag in the testng.xml file.
@BeforeGroups : A method annotated with this annotation will run before all the first test methods run in that specific group
@AfterGroups: A method annotated with this annotation will run after all the test methods run in that specific group
@BeforeClass: A method annotated with this annotation will run only once per class and before all the first test methods run
@AfterClass: A method annotated with this annotation will run only once per class and after all the test methods run
@BeforeMethod: A method annotated with this annotation will run before every @test annotated method
@AfterMethod: A method annotated with this annotation will run after every @test annotated method

Windows:

```
String handle=driver.getWindowHandle();
Set<String> handles = getWindowHandles();
driver.switchTo().window(handle);
```

How to switch to newly created window

```
 String curWindow=driver.getWindowHandle();
```

Get all window handles

```
Set<String> handles = getWindowHandles();
For(string handle: handles){
If (!handle.equals(curWindow))
{driver.switchTo().window(handle);
}
```

```
}
```

Basic Operations:

```
(Mouse)Click:
driver.findElement(By.xpath("//input[@value='OBJECT NAME']")).click();
// Print the Title,URL,PageSource of the Page:
String pagetitle = driver.getTitle();
or
driver.getTitle();
driver.getPageSource
driver.getCurrentUrl();
System.out.print(pagetitle);
// navigate between the URLs.
driver.get(“http://newexample.com”)
driver.navigate().to(“http://newexample.com”)
driver.navigate().refresh()
driver.navigate().forward()
driver.navigate().back()
```

Advanced Operations:

```
// Handle Alert
AlertAlertpopup = driver.switchTo().alert();
// Disable a Field
getElementsByName('<ObjectID>') [0].setAttribute('disabled', '')
//Enable a Field :
getElementsByName('<ObjectID>') [0].removeAttribute('disabled');
// Screenshot :
File src=          ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(src,new File("C:\\Users\\rahul\\screenshot.png"));*/
// Implicit Wait:
manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
// Explicit Wait:
WebDriverWait wait = new WebDriverWait(driver, 20);
// Fluent Wait
Wait wait = new FluentWait(driver)
// Sleep :
Thread.Sleep(10);
// Handle JavaScript pop-ups
// Maximize window
driver.manage().window().maximize();
//Delete Cookies
driver.manage().deleteAllCookies();
driver.manage().deleteCookieNamed("sessionKey");
```

```
//Close every associated window.
driver.quit();
// mouse action
Actions actions = new Actions(driver);
WebElement mouseHover = driver.findElement(By.xpath("//div[@id='mainmenu1']/div"));
actions.moveToElement(mouseHover);
actions.perform();
// Drag and Drop
WebElement sourceLocator driver.findElement(By.xpath("//*[@id='image1']/a"));
WebElement destinationLocator = driver.findElement(By.xpath("//*[@id='stage']/li"));
Actions actions=new Actions(driver);
actions.dragAndDrop(sourceLocator, destinationLocator).build().perform();
// Close single window.
driver.close();
// closes all the windows
driver.quit();
// getConnection
DriverManager.getConnection(URL, "username", "password" )
```

DropDown

isEnabled() This method checks if an element is enabled. Returns true if enabled, else false for disabled.
isSelected() This method checks if element is selected (radio button, checkbox, and so on). It returns true if selected, else false for deselected
isDisplayed() This method checks if element is displayed.
In this recipe, we will use some of these methods to check the status and handle possible errors.

```
Handle Dynamic dropdowns with Webdriver API
Select dropdown=new
Select(driver.findElement(By.xpath(".//*[@id='ctl00_mainContent_ddl_Adult']")));
dropdown.selectByIndex(4);
dropdown.selectByVisibleText("9 Adults");
dropdown.selectByValue("8");*/

 Dropdown looping UI
   /*int i=1;
while(i<5)
{
driver.findElement(By.id("hrefIncAdt")).click();//4 times
i++;
}*/
   System.out.println(driver.findElement(By.id("divpaxinfo")).getText());
```

```
for(int i=1;i<5;i++)
{
driver.findElement(By.id("hrefIncAdt")).click();
}
driver.findElement(By.id("btnclosepaxoption")).click();
Assert.assertEquals(driver.findElement(By.id("divpaxinfo")).getText(), "5 Adult");
System.out.println(driver.findElement(By.id("divpaxinfo")).getText(
AutoSuggestive dropdowns
WebElement source=driver.findElement(By.id("hp-widget__sfrom"));
                source.sendKeys("MUM");
                source.sendKeys(Keys.ARROW_DOWN);
                source.sendKeys(Keys.ENTER);
```

Frames:

```
// Using Frame Index:
driver.switchTO().frame(1);
// Using Name of Frame:
driver.switchTo().frame("name")
// Using web Element Object:
driver.switchTO().frame(element);
// Get back to main document:
driver.switchTO().defaultContent();
```

Alerts:

```
Capture the alert message:
Driver.switchTO().alert.getText();
Click on the 'OK' button of the alert:
driver.switchTO().alert.accept();
Click on the 'Cancel' button of the alert:
driver.switchTO().alert.dismiss();
Send some data to alert box:
driver.switchTO().alert.sendKeys("Text");
System.out.println(driver.switchTo().alert().getText());
```

Print links

```
count of links on the page.
Count of footer section-
System.out.println(driver.findElements(By.tagName("a")).size());
footer
WebElement footerdriver=driver.findElement(By.id("gf-BIG"));
System.out.println(footerdriver.findElements(By.tagName("a")).size());
coloumdriver
WebElement coloumndriver=footerdriver.findElement(By.xpath("//table/tbody/tr/td[1]/ul"));
System.out.println(coloumndriver.findElements(By.tagName("a")).size());
```

```
//SSl certificates

//Desired capabilities=
//general chrome profile
DesiredCapabilities ch=DesiredCapabilities.chrome();
//ch.acceptInsecureCerts();
ch.setCapability(CapabilityType.ACCEPT_INSECURE_CERTS, true);
ch.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);

//Belows to your local browser
ChromeOptions c= new ChromeOptions();
c.merge(ch);
System.setProperty("webdriver.chrome.driver", "");
WebDriver driver=new ChromeDriver(c);
```

Selenium Grid:

Start hub:

```
java-jar selenium-server- standalone-x.xx.x.jar-role hub
```

Start node:

```
java-jar selenium-server- standalone-x.xx.x.jar-role node-hub
http://localhost:4444/grid/register
```

Server:

```
http://localhost:4444/grid/console
```