Important Commands:

OBJECT CREATION

```
chromedriver driver = new chromedriver();
```

WEBDRIVER

```
webdriver driver;

    driver = new chromedriver();
```

close

```
driver.close(); //close the tab
```

quit

```
driver.quit(); // quit the driver.
```

Findelement

```
driver.Findelement (By.xpath/name/ID/class/css/tagname/
                                                classname
```

Send values

```
driver.findelement (By.xpath(" ").sendKeys (" ");
```

title

```
string title = driver.gettitle(); //title of website
        if (title.equals(Expected))
```

text

```
string Loginbutton = driver.findelement (Byname.(" ").gettext();
sysout (loginbutton);
```

content verify:

```
string sc = driver.getpagesource();
    if (src.contains("gmail")
        }
    }.
```

Junit

@Before
@Test // unit testing Framework
@After // without Main method run code

Locators

name | ID / xpath | tagname | css | classname / Linkedlist

ENTER

Keys.ENTER // with sendkeys

Absolute xpath

→ From root node
→ '/' start
→ ▼

Relative xpath

→ selection from current node
→ '//' start
→ "// tagname [@attribute = 'value']"

Contains (xpath)

→ value of attribute changes dynamically
→ "//* [contains(@type, 'value')]"

//multiple webelements details check

tagname

```
List <webElement> li = driver. find Element (By. tagName
                                    ("a"));
```

Count links

```
li. size();
```

get links

```
string href = Link. get DomAttribute ("href");
```

Display

```
for (webElement link : li)
```

text of link (linkname)

```
string text = link. gettext ();
```

Sleep

```
Thread. sleep(50000);
// throws Interrupted Exception.
```

DropDow box

```
webElement day = driver. findElement (Bx. xpath (""));
Select DaySC = new select (day); → select class
Daysc. select By value ("08");
Day sc. select ByIndex (4);
Day sc. select Byvisible Text ("2006");
```

Count of Day/Month/year

```
List<WebElement> a = Daysc.getoptions();
sysout(a.size(););
```

whole Page Screenshot

```
File screenshot = ((TakesScreenshot)driver).getScreenshot
        as(OutputType.FIEE);

FileHandler.copy(screenshot, new File("D:\\screenshot\\
                                          img.png"));
```

throws IOException

FileHandler.copy → static method

((TakesScreenshot).driver) → interface

getScreenshotAs(OutputType.FILE) → method

WebElement Screenshot

```
WebElement screenshot = driver.findElement(By.xpath(" "));
File sc = screenshot.getScreenshotAs(OutputType.FILE
FileHandler.copy(sc, new File("D:\\screenshot\\img2.png");
```

Same Screenshot

```
FileHandler.copy(sc, new File("D:\\screenshot\\img.png"));
FileHandler.copy(sc, new File("./screenshot/ele.png"));
```

Folder in project

Alert

```
driver.switchto().alert().accept();
```

gettext

```
Alert a = driver.switchto().alert();
string text = a.get text();
```

Dismiss

```
a.dismiss();
```

Response code

```
URL obj = new URL(baseurl);
HttpURLconnection con = (HttpURLConnection) ob.
                        open Connection();
con.connect();
if (con.getRespons
```

TestNg

@Beforetest
@BeforeMethod
@Test
@AfterMethod
@Aftertest

① priority set
② Invocation count
③ enabled = false
④ dependsonMethod =
{("test1")}

parameter testing

```
public class Parameter {

        @Test
        @Parameter ("a")
      public void testP(string v){
              (v);
      }
```

xml

```
<suite>
        <test>
          <parameter>
           <classes>
              <class>
             </test>
```

grouping

```
@Test (groups ={"smoke", "sanity"})
```

Xml

```
<suite>
   <test>
     <groups>
        <run>
          <include name=" ">
          <exclude name=" ">
        </run>
```

Cross Browsing Test

```
Webdriver driver;
  @Parameters(" browser")
  @Beforetest
public void setup(string browser){
   if (browser.equalsIgnorecase("firefox")
       driver = new firefoxdriver();
```

@Test

driver.get("");

xml

```
<suite>
    <test>
        <parameter name="~~firefox~~ browser" value="firefox">
        <classes>
         <class>
          </test>
```

Action class

@Test

Actions act = new Actions(driver);

Right click

act.contextclick(~~rightclick~~ webelement).perform();

Double click

act.doubleclick (webelement).Perform();

Mouse-Hover

act.movetoElement(webelement).Perform();

Implicitwait

driver.manage().timeouts().implicitlywait(Duration.ofseconds(20));

Explicit wait

webelement.until(ExpectedConditions.visibilityofElementLocated(By.xpath("").click();

Drag&Drop

act.dragAndDrop(webEle1, webEle2).Perform();

ctrl + A/(select All)

act.KeyDown(element, Keys.control).sendKeys("a".Keys up(Keys.control));

ctrl + c (COPY)

act.KeyDown(element, Keys.CONTROL).sendkey("c").keys up(Keys.control);

ctrl + v (Paste)

act.KeyDown(~~action~~ element, Keys-CONTROL).sendKeys("v").Keysup(Keys.control);

PERFORM

act.Perform();