**Chrome Driver**

1. System. setProperty("webdriver.chrome.driver", "path of the exe file\\chromedriver.exe"

**IE driver**

1. System. setProperty("webdriver.IE.driver", "path of the exe file\\chromedriver.exe"

**Find element**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  2  3  4  5  6 | WebElement element = driver.findElement(By.linkText("Partial Link Test"));  element.clear();    //Or can be identified as  WebElement element = driver.findElement(By.partialLinkText("Partial");  element.clear();  **Select Class for drop down**   |  | | --- | | select the first operator using "select by value" |  |  |  | | --- | --- | | 46 | Select selectByValue = newSelect(driver.findElement(By.id("SelectID\_One"))); |  |  |  | | --- | --- | | 47 | selectByValue.selectByValue("greenvalue"); |  |  |  | | --- | --- | | 48 | Thread.sleep(5000); |  |  |  | | --- | --- | | 49 |  |  |  |  | | --- | --- | | 50 | // select the second dropdown using "select by visible text" |  |  |  | | --- | --- | | 51 | Select selectByVisibleText = new Select (driver.findElement(By.id("SelectID\_Two"))); |  |  |  | | --- | --- | | 52 | selectByVisibleText.selectByVisibleText("Lime"); |  |  |  | | --- | --- | | 53 | Thread.sleep(5000); |  |  |  | | --- | --- | | 54 |  |  |  |  | | --- | --- | | 55 | // select the third dropdown using "select by index" |  |  |  | | --- | --- | | 56 | Select selectByIndex = newSelect(driver.findElement(By.id("SelectID\_Three"))); |  |  |  | | --- | --- | | 57 | selectByIndex.selectByIndex(2); |  |  |  | | --- | --- | | 58 | Thread.sleep(5000); | | **Radio button and check boxes** |  |   WebElement radio1 = driver.findElement(By.id("vfb-7-1"));  WebElement radio2 = driver.findElement(By.id("vfb-7-2"));    //Radio Button1 is selected  **Tabs and windows**  String Parent\_Window = driver.getWindowHandle();    // Switching from parent window to child window  for (String Child\_Window : driver.getWindowHandles())  {  driver.switchTo().window(Child\_Window)  **Frames in Selenium**  / prints the total number of frames  driver.switchTo().frame(0); // Switching the Outer Frame  System.out.println (driver.findElement(By.xpath("xpath of the outer element ")).getText());  //Printing the text in outer frame  size = driver.findElements(By.tagName("iframe")).size();  // prints the total number of frames inside outer frame    System.out.println("Total Frames --" + size);  driver.switchTo().frame(0); // Switching to innerframe  System.out.println(driver.findElement(By.xpath("xpath of the inner element ")).getText());    //Printing the text in inner frame  driver.switchTo().defaultContent();  **Key strokes handling in selenium**  WebElement webElement = driver.findElementByXPath("");  webElement.sendKeys(Keys.TAB);  webElement.sendKeys(Keys.ENTER);  **Mouse movements in selenium**  Actions action = new Actions(driver);    action.moveToElement(element).build().perform();    driver.findElement(By.linkText("iPads")).click()  **Desired Cpability**  DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();    capabilities.setCapability(CapabilityType.BROWSER\_NAME, "IE");  capabilities.setCapability(InternetExplorerDriver.  INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true);  **Firefox profiles**  ProfilesIni profile = new ProfilesIni();    FirefoxProfile myprofile = profile.getProfile("profileToolsQA");    WebDriver driver = new FirefoxDriver(myprofile); |
| **Testng** | **@beforesuite**  **@beforemethod**  **@test(priority=1,Dependsonmethods{(add)}**  **@aftermethod**  **@Aftersuite**  **@Data provider**  **Object array**  **#table: depends on key value player. Here key is Column names** |

**Testng listeners: overridden metheods- like triggers**

**1.onfailure**

**2.on Skip**

GIT and GIT hub:

* Git local repository
* Git init
* Git add \*.txt
* Git Comit –m” my frst commit”- commit to local repository
* GIt hub repository- master branch
* Push to the master branch
* Pull requests
* Branch creation

Maven:

Dependencies are maintained in Pom.xml and the jars are downloaded to m2 folders in cdrive users.

Validate

Compile

Package

Maven test.

Jenkins