what are limitations in selenium webdriver  
ex: theory only

Some limitations of Selenium Automation tool are as follows:

* It does not support non web-based applications, it only supports web based applications.
* Its and open source tool so in case of any technical issues you need to rely on the selenium community forums to get your issue resolved.
* You need to know at least one of the supported language very well in order to automate your application successfully.
* No inbuilt reporting capability so you need plugins like JUnit and TestNG for test reports.
* Lot of challenges with IE browser.

installing/configure selenium  
ex: write down steps

* **Install Java on your computer.**
* **Install Eclipses IDE**
* **Download the Selenium Java Client Driver**
* **Create a New project in Eclipse called Test Project**
* Now Right-click on test project and select Properties.  
  On the Properties window, click on “Java Build Path”.  
  Click on the Libraries tab, and then click “Add External JARs.”  
  Navigate to location where you saved the extracted contents of “selenium-2.37.0.zip” Add all the JAR files inside and outside the “libs” folder. Your Properties window after adding all the jar file.
* Finally, click OK and we have completed importing Selenium libraries into our project.
* Now we can write the code for selenium code using java language

what are different ways of locating elements in selenium  
ex : write code for all locators  
Locating Elements with Selenium WebDriver, findElement () method returns and WebElement and findElements () returns a list of WebElements.

1. By **ID:** driver.findElement(By.id("element id"));
2. By **CLASS:** driver.findElement(By.className("element class"));
3. By **NAME:** driver.findElement(By.name("element name"));
4. By **TAGNAME:** driver.findElement(By.tagName("element html tag name"));
5. By **CSS Selector**: driver.findElement(By.cssSelector("css selector"));
6. By **Link**: driver.findElement(By.link("link text"));
7. By **XPath**: driver.findElement(By.xpath("xpath expression"));

which is fastest way to identify elements in web page?  
ex: theory only

In my opinion, the most accurate way is XPath as you can use XPath's to describe the exact position of an element within the DOM, however there are some instances where CSS locators work better than XPaths.Using selenium's ID locator is the most simple, but unless the element you are looking for has an ID not always useful.

what is absolute path and relative path in xpath  
ex: write code  
**Absolute Xpath**: It uses Complete path from the Root Element to the desire element.

**Relative Xpath**: You can simply start by referencing the element you want and go from there.

Always Relative Xpaths are preferred as they are not the complete paths from the Root element. (//html//body). Because in future any of the webelement when added/Removed then Absolute Xpath changes. So Always use Relative Xpaths in your Automation.

**package** Sample.practise;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** Sample {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

WebDriver driver = **new** FirefoxDriver();

// open URL

driver.get("http://www.facebook.com");

// Absolute Xpath

driver.findElement(By.*xpath*(".//\*[@id='u\_0\_1']"));

//Relative Xpath

driver.findElement(By.*xpath*("//input = [@id ='u\_0\_1']"));

}

}

different types of waits or synchronization in selenium webdriver  
ex: write code

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.

Explicit wait is mostly used when we need to Wait for a specific content/attribute change after performing any action

**package** Sample.practise;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public** **class** SyncInSelenium {

**public** **static** **void** main(String[] args) {

// Creating the Driver object

WebDriver selenium ;

selenium = **new** FirefoxDriver();

//Using Implicit wait for 20 seconds,which is applicable to all the controls in application

selenium.manage().timeouts().implicitlyWait(20, TimeUnit.***SECONDS***);

//Opening the application page

selenium.navigate().to("http://www.facebook.com");

//Using Explicit wait , Expected Condition is search box to be visible

WebDriverWait wait = **new** WebDriverWait(selenium, 10);

wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*("u\_0\_1")));

//Fill the value into the search

selenium.findElement(By.*id*("u\_0\_1")).sendKeys("Wait");

}

}  
  
how to save screen shots using selenium webdriver  
Ex: write code

**package** Sample.practise;

**import** java.io.File;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** com.sun.jna.platform.FileUtils;

**public** **class** ScreenShots {

**public** **static** **void** main(String[] args) {

WebDriver driver = **new** FirefoxDriver();

driver.get("http://www.google.com/");

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

// Now you can do whatever you need to do with it, for example copy somewhere

FileUtils.copyFile(scrFile, **new** File("c:\\tmp\\screenshot.png"));

}

}  
  
how to handle multiple windows in selenium webdriver  
Ex: write code

**package** Sample.practise;

**import** java.util.List;

**import** java.util.Set;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** MultipleWindowsHandle {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String[] args) {

*driver* = **new** FirefoxDriver();

*driver*.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

*driver*.get("http://toolsqa.com/automation-practice-switch-windows/");

String handle= *driver*.getWindowHandle();

System.***out***.println(handle);

*driver*.findElement(By.*name*("New Message Window")).click();

Set handles = *driver*.getWindowHandles();

System.***out***.println(handles);

**for** (String handle1 : *driver*.getWindowHandles()) {

System.***out***.println(handle1);

*driver*.switchTo().window(handle1);

}

*driver*.close();

*driver*.quit();

}

}  
  
how to lanuch webpage using chrome driver  
Ex: write code

**package** Sample.practise;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** ChromeDriverLaunch {

**public** **static** **void** main(String[] args){

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\SUPRAJA\\Downloads\\chromedriver.exe");

WebDriver driver=**new** ChromeDriver();

driver.get("http://google.com");

System.***out***.println(driver.getCurrentUrl());

driver.close();

}

}  
what is desired capabilities in selenium webdriver  
Ex: write code  
  
how to set language while opening website   
Ex: write code  
  
how to handle windows based popups (upload and dropdown)  
Ex: write code  
  
write code to verify any application login page is working or not  
(u should write code to use textbox, button click events)  
Ex: write code

**package** Sample.practise;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** Login {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

FirefoxDriver driver = **new** FirefoxDriver();

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

driver.findElement(By.*id*("Email")).sendKeys("email id here");

driver.findElement(By.*id*("next")).click();

Thread.*sleep*(10000);

driver.findElement(By.*id*("Passwd")).sendKeys("password");

driver.findElement(By.*id*("signIn")).click();

driver.close();

}

}

how to select items from dropdown/select box  
Ex: write code  
  
how to know if checkbox is checked or not in webpage  
ex: write code

**package** Sample.practise;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**public** **class** CheckBox {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

FirefoxDriver driver = **new** FirefoxDriver();

//get url

driver.get("https://login.yahoo.com/config/login?.src=fpctx&.intl=us&.lang=en-US&.done=https%3A%2F%2Fwww.yahoo.com");

driver.findElement(By.*id*("persistent")).isEnabled();

**boolean** enabled = **false**;

**if**(!enabled)

{

System.***out***.println("Already checked");

}

**else**

{

driver.findElement(By.*id*("persistent")).click();

System.***out***.println("checked now");

}

driver.close();

}

}  
  
tell me code to pass values from parent window to child window  
Ex: write code  
  
write code to find out if all links are working or not  
Ex: write code  
  
write code on how to use javascriptexecutor?  
Ex: write code  
  
difference between assert and verify?  
Ex: write code  
  
difference between driver.close and driver.quit methods?  
Ex: write code  
  
common exceptions in selenium?  
Ex: write code  
  
how to handle Ajax calls in selenium?  
Ex: write code  
  
we have webtable, need to click on second row from table.  
when we click on child will be populated. first column in primary column for each row.  
tell me steps to verify child form has proper data or not  
Ex:   
  
How to assign the value to textbox other than sendkeys method?