

- why selenium
- firefox plug in
  - facilitates recording & play back
  - can convert the steps into code you required

## Selenium webdriver

It is a main component used a lot.

It is best for complete web testing automation.

1. Selenium WebDriver was rebranded as part of Selenium 2.0
2. WebDriver is designed in a simple and better way as required for modern web applications
3. WebDriver uses browser native commands to drive the browser which is very robust.
4. Overcomes following main Selenium 1.0 or Selenium RC limitations.

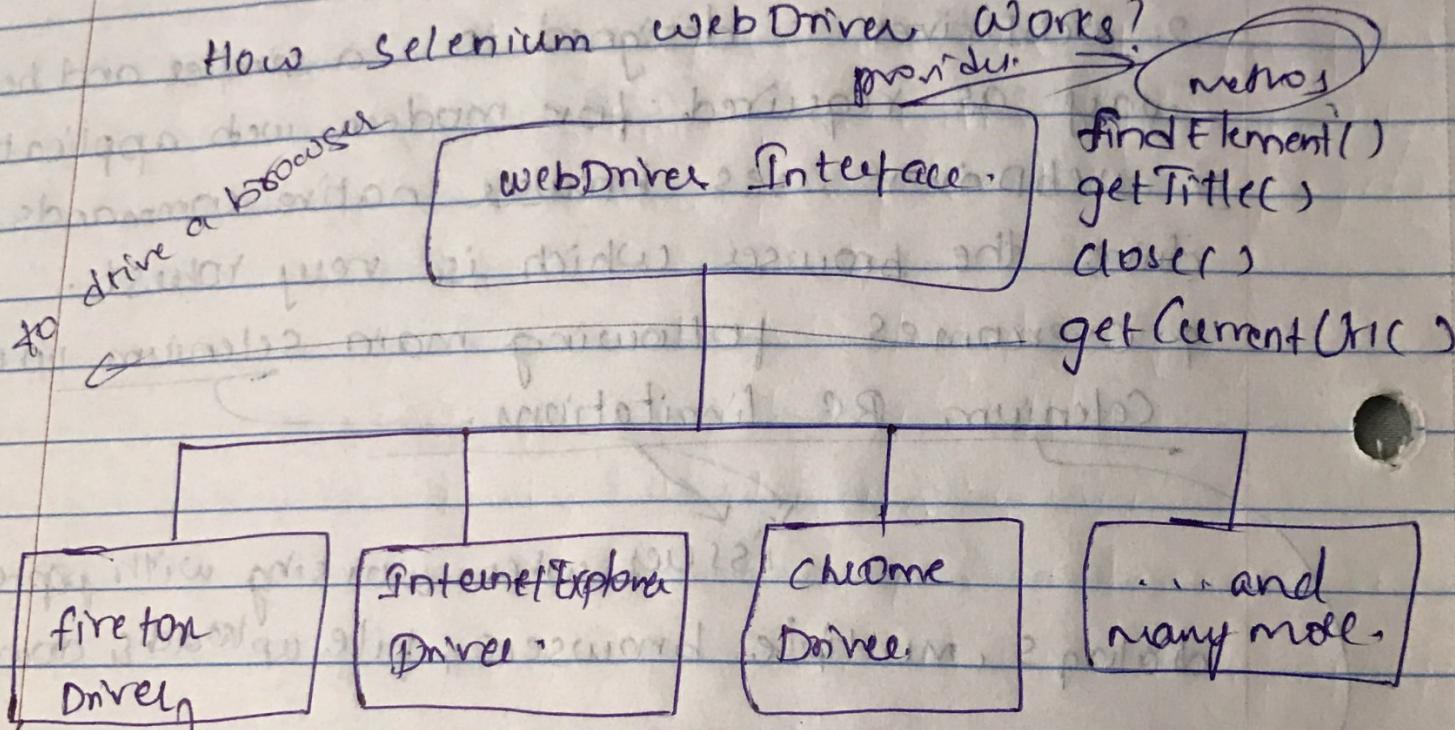
issues when working with pop-ups, dialogs, multiple browsers, file uploads & download

- reason for this is ~~selenium~~ Selenium RC uses Javascript to drive the browsers & different browsers work differently with javascript.
- WebDriver is more object oriented when compare to Selenium 1.0. you can work with web element windows, dialogs by creating objects for them
- It supports / runs old Selenium RC code so that you can still use your old selenium RC code instead of new writing it using

Selenium WebDriver is a tool for automation.

- you can use multiple programming languages like Java, C#, Ruby, Python with WebDriver maximum control.
- It supports multiple browsers and multiple operating systems.

How Selenium WebDriver Works?



Selenium WebDriver is an interface

(It's implementing the WebDriver interface that means it's providing methods)

and it defines the methods that we can't

use to drive a browser.

Selenium WebDriver interface provides methods to drive a browser.

Webdriver Interface it forces the implementing classes to provide the same methods.

When a class implements an interface, it provides the methods defined in that interface. With that, we can use same method names with different browsers.

Selenium WebDriver & different Drivers.

→ WebDriver is the name of the key interface against which tests should be written, but there are several implementations.

→ When you are writing a test we should create webdriver variable or webdriver object and use that object.

→ To see methods defined in webdriver interface and imp. implementations.

findElement (By by)

parameters:

by - the locating mechanism. using is

Returns :

The first matching element on the current page.

Throws:-

No such Element Exception - if no matching elements are found

write a simple java program.

Here we want to write Selenium webdriver test case that means we want to use selenium webdriver functionality, add Selenium webdriver functionality.

How do you add that you add that by adding the jar files. we have the jar files which provide Selenium webdriver functionality, so we need to add those jar files to our project. Let's download them.

selenium website → download page →

Here you can select jar file.

download client binding + selenium server.

There is no exe for selenium or there is no need to install it are only jar file for java, we add selenium server jar file to our project but for other language we need to add both client binding jar files and server selenium server jar files.

right click on project name → properties →  
java build path → libraries.

Here we need to add selenium webdriver library. Because selenium is external which is not built in our

System click on add external jars.



Select Scrapy file.



OK.

Now our project has atra section called referenced libraries



under this we can see lot of packages  
All these represent selenium functionalities, Selenium  
APT

Now lets go back to project and write  
Selenium WebDriver TestCase.

click on src folder, lets take website for test

indeed.com

New class

package name → com.indeed.lets.

Name → IndeedJobSearch.

In program -

- ① → create firefox driver to drive the browser
- ② → open indeed home page
- ③ → find what field & enter selenium
- ④ → " location " → London
- ⑤ → find FindJobs button and click it

⑥ // from jobsearch results page, get page title and jobs count message.

first write java code.

Create WebDriver & its implementation is of type Firefox. Let's write code for that.

① // create!

WebDriver driver = new FirefoxDriver();  
new Firefox browser window will be created.

add import statement  
Same for webDriver.

Now Firefox browser will be opened.

Now we have Driver of Firefox browser, we can interact with it & we can send all commands we want

② // open

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

③ // what find what

Thread.sleep(2000);  
driver.findElement(By.id("what")).sendKeys("Selenium");

④ // find locate

driver.findElement(By.id("where")).clear()  
Sendkeys("London")

You can slow the process by adding  
Thread.sleep command.

① // find And jobs.

driver.findElement(By.id("fj")).click();

⑥ // And job

driver.getTitle()

driver.findElement(By.id("SearchCount")).  
getText()

S.O.println(driver.getTitle());

S.O.println(driver.findElement(By.id("SearchCount")).  
getText());

driver.close(); → To close the browser.

}

b

But for Chrome ~~we~~ we have to download

~~selet~~ one from selenium downloads page  
Third party driver.

Chrome

↑ click

→ → windows

download.

& to get that into eclipse write a command

② Called System.setProperty("property", "path");

copy from error message ↗

copy path ↗

give two ↙  
back slashes

from epe ↗

Scanned by CamScanner

`System.setProperty ("webdriver.chrome.driver",  
"G:\Selenium Training\1 Selenium\1 Framework\15\01  
chromedriver.exe");`

for IE -

first make sure that in Settings  
zoom to set to 100% & internet options.

↓  
security

(I/L/T/S/R)

security level should be same for  
all the four

enable & disable but some  
for four -

Next, we need IE driver.exe same as chrome.

IE Driver level

32      64  
driver.

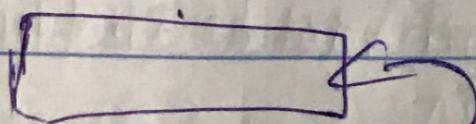
import java dos into our project

properties → gamma build path libraries

→ select selenium server standalone.

click on ~~add~~ click on jar adoc location & click on

Path:



selenium jar adoc location.

Chit Chatter

What are webElements?

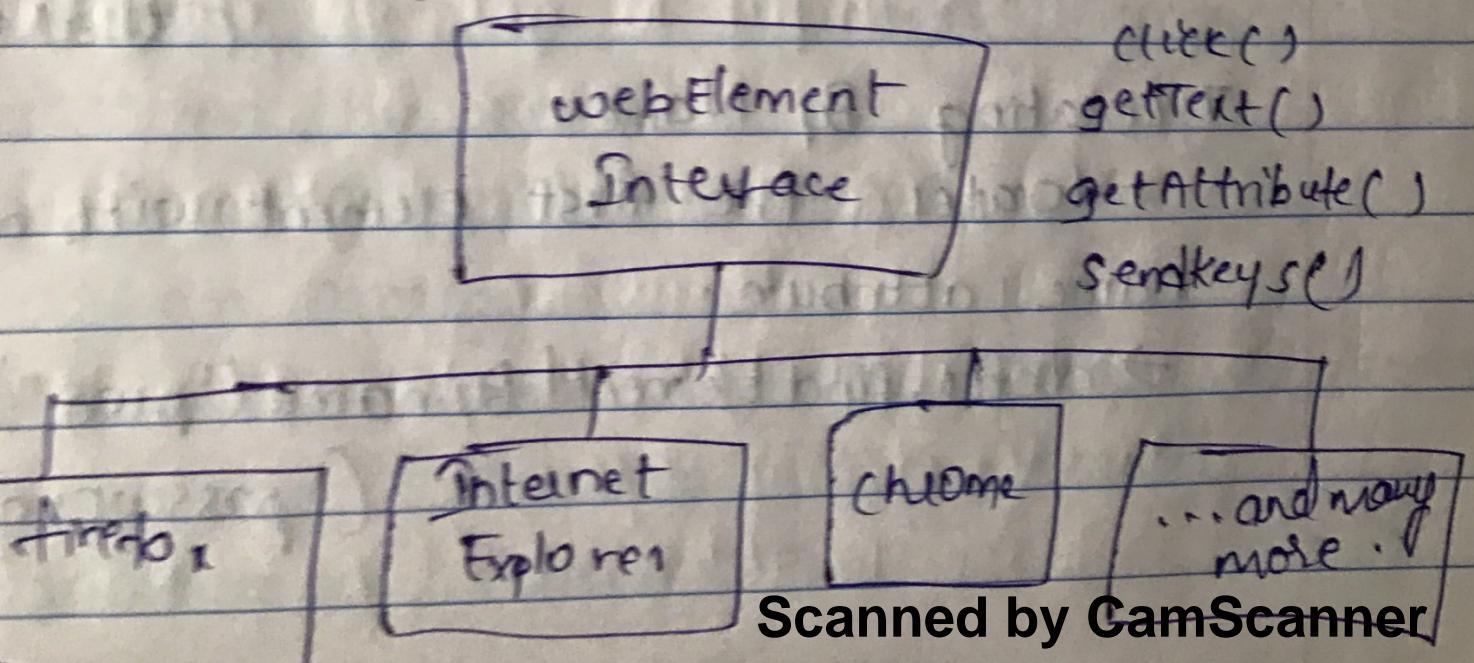
1. Selenium sees everything on the page as webElements ex:- textbox, button, link, dropdown
2. For simple ones or most of them, Web Element type is used.  
e.g:- textbox, button, link.

for all of these we have webElement interface to deal with.

3. For complex ones separate classes are used  
ex:- dropdown, alert/popup's, frame,

Select class is used for dropdown.

WebElement Interface



we can use same method names for web elements with different browsers.

### Identifying web elements in webpage.

1. for many selenium commands, a target is required.
2. Target identifies a web element uniquely in the content of web application.
3. Target consists of the locating strategy and format like `By.Locator Strategy Type("value")`.  
↓  
By class to locate web elements in Selenium WebDriver.
4. `findElement()` always finds the first web element using the given locating strategy.
5. `findElements()` returns all web elements matching the given locating strategy.
6. Locating By Id.
  - locates the first element with matching id attribute.
  - `driver.findElement(By.id("UserID")) .sendKeys ("UserName")`
7. Locating By Name.
  - locates the first element with matching name attribute.
  - `driver.findElement(By.name("pwd")) .sendKeys ("password")`

- locating By linkText
  - locates the first hyperlink with matching link text.
  - This is a simple method of locating a hyperlink in your web page by using the text of the link.
  - driver.findElement(By.linkText("Sign in")).click();

### //locating by Name

```
driver.findElement(By.name("q")).sendKeys("Selenium");
```

### //link text locating Strategy

```
driver.findElement(By.linkText("Post your CV"))  
    .click();
```

### → locating by partialLinkText

→ locates the first hyperlink which contains specified link text.

```
driver.findElement(By.partialLinkText("can't  
access")).click();
```

### → locating by xpath

→ locates the first element as specified by xpath strategy.

```
driver.findElement(By.xpath("//input[@place  
holder='Email']")).sendKeys("User Name");
```

### → locating by css selector

```
driver.findElement(By.cssSelector("css = input.  
input-submit")).click();
```

// locating by partial link text.

```
driver.findElement(By.partialLinkText("post job")).  
click()
```

// locating by XPath.

```
driver.findElement(By.xpath("//img[@title =  
'Indeed Job Search ']")).getAttribute("src");
```

```
s.println();
```

// locating by CSS selector.

```
s.println(driver.findElement(By.cssSelector("css  
input.input-submit")).getAttribute("value"));
```

→ locating By Tag Name:

→ useful to get all elements with a given tag.

```
driver.findElements(By.tagName("a")).size();
```

→ locating By className

→ useful to get all elements with a given class or display style.

```
driver.findElements(By.className("input-text")).size();
```

**Scanned by CamScanner**

In ~~oop~~ functions in a class are called methods.

```
package com.myStore.product;  
public class product
```

{  
    //  
    //  
    //

}

The area b/w the braces is called class body which contains all the code for program

→ fields to represent properties or state (non-static)

methods or functions for the behaviour

(non-static)

→ name

price,

SKUID

Syntax of field

Access level (static) Data Type Variable Name;

✓ ↗ right now not necessary.

ex- public String name;

public double price;

public String SKUID;

Creating methods or function.

Syntax - Access Level (static) Return DataType MethodName  
(<sup>if argument not many</sup> DataType arg1, DataType arg2... argN);

----- method body.  
return ReturnType values;

method name.

public void purchase (int quantity)

// logic or code to purchase the product.

System.out.println ("In purchase () method");  
for (Items: " + quantity);

Creating Objects

package

public class

public static void main (String [] args) {

product Iphone = new product();

product desktop= new product();

Iphone.name = "Iphone";

Iphone.price = 399.00;

Iphone.SkuID = "phone01";

System.out.println("iPhone.name");

Introduction to Automation & Web Testing.

1. why needed Automation
2. what is Automation Testing.
3. Advantages & Disadvantages of AT
4. " " of Manual Testing
5. Automation criteria (whether to automate or not)
6. focusing on Web Applications and Web Testing.
7. Automation for Web Application.

→ Why..?

1. software tests have to be repeated during development cycles to ensure quality.
2. Every time source code is changed slow tests should be performed.
3. ALSO, for each release of the SW it ~~can~~ may be tested on all supported operating systems and supported environments,
4. Manually repeating these tests is costly and time consuming. That's why we need automation.
5. Automation testing involves making manual testing automatic using a software/tool.

An effective automation testing involves following main areas:

- (i) Develop automation scripts.
- (ii) Execute automation scripts.
- (iii) Compare actual results with expected results.
- (iv) Control test execution.
- (v) Set-up/configure test pre-conditions and post-conditions.
- (vi) Report execution results.
- \*\*\* (vii) Automation framework.

Generally, test automation involves automating a manual testing process already in place.

Test cases ensure that user's

Order history is displayed correctly

Automation Testing Areas

1. Deploy latest build → Develop Automation Scripts
2. Make sure some users are there with orders in DB → Execute " "
3. Go to home page → Compare Actual results with expected results
4. Login as a user. → control test execution
5. Go to order history page → set-up/configure test pre-conditions & post conditions
6. View the order history table. → Report execution results
7. Repeat steps 3 to 6 for different users. → Automation framework.

## Advantages & disadvantage of Automation Testing

### Advantages

1. Saves time and cost
2. effective regression testing
3. Repeatable on different environments
4. Accuracy
5. Increases test coverage.
6. Comprehensive
7. Reusable
8. Automating daily tasks helps a lot.

### Disadvantages

1. It costs more to automate initially.
2. you can't automate everything like user acceptance testing, visual effects.

### 3. Automation tool limitations

## Adv & disadvantages Manual Testing

### Adv

### Disadv

1. Best if you just need to test 1 or 2 times
  2. Allows to perform more exploratory testing  

This is not possible in Automation testing because the steps are predefined
1. very time consuming
  2. Human mistakes

## Automation Testing vs Manual Testing

- Generally advantages of automation outweigh its disadvantages
- Manual testing advantages
- Automation testing is best over time

## Automation Criteria (Automate or Not)

- The general rule of thumb has always

B

1. for tests that need to be run one/two times or tests that are really expensive to automate - go for manual testing
2. For tests that need to be run multiple times or frequently (e.g. daily weekly) - go for automation testing
3. entire Automation vs manual testing criteria depends on which is worth in medium / long term in terms of time and cost

Is the test case automatable?	imp/frequency	Time/cost	Automate the test case
Yes	high or med	low	Yes
Yes	high or med	high	Yes
No	low	low	maybe No

Real world scenarios  
→ Applying automation criteria for  
following ecommerce testcases-

- i) A user registers for an account
- ii) A user purchases a product
- iii) A user deletes his account

Focusing on Web Application?

A web application is an application which is developed using a browser-supported language and accessed via a web browser over a network such as internet or intranet.

Web Applications are  
easy to maintain  
easily accessible

Getting more and more popular.

What is Web Testing?

- Web Testing is the name given to S/W testing that focuses on web applications. It is one of the fastest growing areas of S/W testing.
- Following are important web testing areas:
- Functionality Testing, Link Testing, Form/Fields Testing, Cookies Testing, DataBase Testing
- Usability Testing: Test for easy Navigation, Content/Grammar/Spell checking, Other

option like site search, sitemap, site help  
etc.

Interface Testing: website and application  
Server interface, Application server and  
Database Server interface.

Compatibility testing: Browser compatibility,  
Operating System Compatibility, mobile

Performance Testing: Load testing, Stress  
testing, Soak testing.

Security Testing:

We will test Invalid login and inputs,  
SQL injection, cross-site scripting, unauthorized  
access and penetration

This is very important for  
Bank and Financial firms

→ A number of commercial and open source  
tools are available for web testing automation  
like Selenium, QTP.

→ Selenium is possibly the most widely used  
open source solution for web testing automation

what is Selenium?

1. Selenium is a very powerful automation testing tool for web application.
2. Any web based tasks can also be automated.
3. Selenium is composed of following main components and each one has its own specific functionality

→ Selenium IDE

→ Selenium WebDriver

→ Selenium RC

→ Selenium Grid

4. Selenium will not work with desktop applications or windows applications.

Selenium IDE provides us to easily start with Selenium. It provides record and playback feature to quickly develop and run test <sup>web application</sup>.

Selenium WebDriver: It is best for full automation. It drives browser natively like on and user and provides multi language support, multi browser support and multi operating system support.

Selenium RC: Selenium RC is predecessor of Selenium WebDriver. RC drives browser using javascript and also provide multi language support, multi browser support and multi operating system support similar to WebDriver.

Selenium RC is officially duplicated and is not recommended to use for web automation.

Selenium Grid:- It allows distributed and parallel running of Selenium automation scripts.

Selenium Features:

- 1. Open source and free (All Selenium Components)
- 2. Record and playback feature (Selenium IDE)
- 3. Can run tests on multiple browsers (Selenium WebDriver)
- 4. Can run tests on multiple operating systems (Selenium WebDriver)
- 5. Supports multiple languages (Selenium WebDriver)
- 6. Can use with a number of testing frameworks (Selenium WebDriver)
- 7. Distributed and parallel running of tests (Selenium Grid).
- 8. Selenium automation is independent of the language in which you have developed your web application (All Selenium Components).
- 9. Add your custom functionality by using user extensions (All Selenium Components)
- 10. Customize / enhance source code to suite your requirements (All Selenium Components)

## Selenium IDE :-

1. Selenium IDE is very useful if you want to quickly create scripts to test your web application.
2. It provides an easy interface for developing / running / Debugging test scripts or test suites.
3. It is a Firefox add-on.
4. It provides record and playback functionality.
5. It can only run scripts on Firefox and it's not available for other browsers.
6. You can export or recorded tests in multiple languages to use with Selenium WebDrivers.

## Selenium WebDrivers :-

It is the main component that we use a lot. It's best and recommended for complete web testing automation.

- Selenium WebDrivers was released as part of Selenium 2.0.
- WebDrivers is designed in a simple and better way as required for modern web applications.
- WebDrivers uses browser native commands to drive the browser which is very robust.
- overcomes following main Selenium 1.0 or Selenium RC limitations.

- Issues when working with pop-ups, dialogs, multiple browsers, file upload and download.

- WebDriver is a more object-oriented when compared to Selenium 1.0, you can work with web elements, windows, dialogs by creating objects for them
- It supports old selenium RC code.
- You can use multiple programming languages like java, C#, ruby, python.
- It supports multiple browsers and multiple operating systems.

### Selenium RC

- Selenium RC is predecessor of Selenium WebDriver
  - RC drives browsers using JavaScript
  - It supports
    - multiple browsers
    - multiple operating systems
    - multiple languages
- Selenium RC is officially deprecated and it's not recommended to use for web automation.

### Selenium Grid

- Selenium Grid allows distributed and parallel running of test cases.

→ It helps in

- Running large test suites faster by breaking them down.

- Running test suites against multiple browsers, multiple versions of browser, and browsers running on different operating systems.

→ When you run Selenium Grid, it works as central hub and you can start multiple Selenium WebDriver nodes to register under it to execute tests distributed.

→ Selenium Grid is now part of Selenium Server  
Real world Scenarios.

#### Selenium IDE vs WebDriver vs Grid

As a automation tester, you want to record & play back test cases quickly	Yes	No	No
Run tests on multiple browsers	No	Yes	No
Run tests on multiple OS's	No	Yes	No
Do complete automation for your webapp	No	Yes	No
Distributed and parallel running of tests	No	No	Yes
Add your custom functionality	Yes	Yes	Yes
Customize/enhance source code to suit you.	Yes	Yes	Yes

Commands to assert / verify:

- Choosing between assert and verify commands depends on how you want to manage failures.
  - On assert failure: It will fail the test and abort the current test case execution.
  - On verify failure: It will fail the test and continue to run the rest of test case.
- use assert commands to make sure your test case is going in right direction.
- use verify commands to actually compare test results of output values.
- The best use of these commands is to logically group your test commands, and start each group with an "assert" followed by one or more "verify" test commands.

e.g.: - Make sure you are on correct page with assert and proceed with rest of tests using verify.

→ we have different assert commands like

- assertTitle → to assert the title of the current page.
- assertAttribute → " " " " attribute of an element
- assertElementPresent → check whether element present or not.

→ We have different verify commands like

- verifyTitle
- verifyAttribute

• verifyElementPresent

ex- go to yahoo website

Search Selenium means assert title

a selenium.

verify logo

verify

### Locating by xpath:-

1. xpath is a language used for locating nodes in XML documents.
2. one of the main reasons for using xpath is when you don't have a suitable id or name attribute for the element you want to locate.
3. xpath -Absolute - not recommended to use.

Syntax:

//html / body / tag [index] / tag2 [index]  
..... / tagN [index]

example:

//html / body / div[2] / div / div[2] / div / div / div  
/ fieldset / form / div[1] / input[1]

00 m.mys

TUnit is preferable than JUnit until JUnit is specified to use.

## Junit :-

- JUnit is a unit testing framework for java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit-test frameworks collectively known as XUnit, that originated with JUnit.
- It is heavily used by java developers for unit testing.
  - JUnit is available as a jar file from <http://junit.org/>
  - JUnit allows us to easily create repeatable tests, specify setup for tests. you can also generate reports on JUnit tests.

JUnit provides different Annotations useful in writing and running tests and apply these annotations to methods.

JUnit don't need main method.

## Annotations -

@Annotation name -

- Some are:-

@Test - Is used to specify a test.