

ex- go to yahoo website  
Search selenium mean 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 / tag1 [index] / tag2 [index]  
... / tagN [index]

### example:

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

Tarata 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 creating 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

Annotatons have special meaning with Junit

### Test level annotations:-

@Before - is used to specify pre test setup or actions for each test.

@After - is used to specify post test setup or actions for each test.

@Ignore - is used to ignore or skip a test.

### Test class level annotations:-

@Before class - is used to specify pre test setup or actions before any of your tests.

@After Class - is used to specify post test setup or actions after all your tests.

```
ex:- package com.my.junit;
```

```
public class JunitTest@case {
```

```
@Test
```

```
public void testHomePage()
```

```
{
```

```
System.out.println("In testHomePage() method");
```

```
}
```

```
}
```

O/P:- In testHomePage() method.

since

But JUnit programs doesn't use main method, running this type of program is different

go to run → run as → Junit test

on Alt+Shift+F9, T

The methods are executed based on the annotations they have. Only the methods with annotations are executed.

\* Methods for before class and after class must be 'static'

we have similar Annotations in TestNG also

```
// assertEquals ("MessageOnFailure", "comparisons or conditions");
// assertTrue (MessageOnFailure, comparisons or conditions);
    final String is true or false.
    en- assertTrue ("Error : Title is wrong", "my site title");
        equals ("my site title");
    }
    fail (MessageOnFailure);
    en- if (!("my site title".equals ("my site title"))
    {
        fail ("Error : Title is wrong");
    }
    en- assertEquals ("error : title is okay", "my site title",
        "my site title");
```

37:25

driver.findElement(By.id("what")) .sendKeys("kkm")

To create JUnit test suite we use these lines of code:

@RunWith(Suite.class)

runnername → nose suite

@SuiteClasses({

sunjava → JUnit → junit  
testsuite

IndeedSiteTests.class,

IndeedSiteTests2.class,

JUnitTestcases.class;

dept.

)

Open name →

public class RegressionSuite {

g

### TestNG:

1. TestNG is a testing framework for the Java programming language inspired from JUnit & Nunit.
2. TestNG has additional features and functionalities that make it more powerful and easier to use compared to JUnit.
3. TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc - but JUnit is designed to perform only unit testing.
4. TestNG is available as a jar file from <http://testng.org>.

### TestNG's main features:

1. Allow us to easily create repeatable tests and specify setup for tests using annotations.
2. provides HTML reports but Junit doesn't.
3. Supports parameterized and Data-Driven testing. Junit also supports parameterized and data-driven testing but TestNG is more efficient and easy to use.
4. Distributed and parallel running of tests.

### Differences between Junit & TestNG

1. Both the frameworks are very similar in functionality and once you know one, it's easy to learn and work with other ~~one~~.
2. There are jobs based on both Junit and TestNG but in Selenium we use any one framework. TestNG is preferable until and unless any specification is given to use Junit. TestNG is more powerful.
3. TestNG provides almost everything Junit provides. In addition to that TestNG:
  - (i) is designed to cover all categories of tests. It provides more annotations and configuration around annotations.
  - (ii) provides html reports.

- (iii) Efficient and easy parameterized and data-driven testing.
- (iv) supports XML configuration, using XML doc we can specify TestNG Suite.
- (v) distributed & parallel running of the tests.

TestNG.org → download → ~~for eclipse-4.6~~

~~Next~~

Help → ~~Install~~ → install new features

TestNG select → OK.

Now, create new java project right click on project name → properties → Java build path → libraries → Add library → TestNG  
Select → Next → finish → OK.

Now TestNG referenced library is added.

No need to use main method in TestNG annotations will take care of it.

Annotations:-

- @Test - is used to specify a Test Method Test Method level annotations.
- @BeforeMethod - is used to specify setup actions before each test.
- @AfterMethod - is used to specify tear down actions after each test.

Test class level annotations -

@Before Class - is used to specify setup actions before all tests in a class.

@After Class - is " " " teardown,  
" after " " " "

Test class group level annotations:

@Before Test - is used to specify setup actions before all tests from a group of classes.

@After Test - " " " teardown

" after " " " "

Test Suite level annotations:-

@Before Suite - is used to specify setup actions before the test suite begins.

@After Suite: " " " teardown  
after " " " ends.

→ After refreshing we can see html reports.  
order of execution :

test suite setup site.

test class group setup DB

test class starts selenium.

test Method setup -

test method homepage  
test method teardown.

test method setup 1

test method teardown 1

STOP

- ① Assert.assertEquals ("My Site Title", "my site title", "ERROR : home page title is wrong");
- ② Assert.assertTrue ("mysite Title ".equals("My site Title"), "ERROR ..");
- ③ if (!("My Site Title ".equals("My Site Title")))

{  
    Assert.fail ("ERROR : Home page title is wrong");  
}

If assertion is failed the statements after that are not executed - Most of the time assertions are used at the end.

→ Each test is independent.

→ @Test(enabled = false) → is used to ignore a test we can't see the ignored test in console window or in TestNG results but we can find it in html reports

`@Test(enabled = false){  
 TO skip a method within the test code use  
 * throw new SkipException("Test is skipped in executable");  
}`

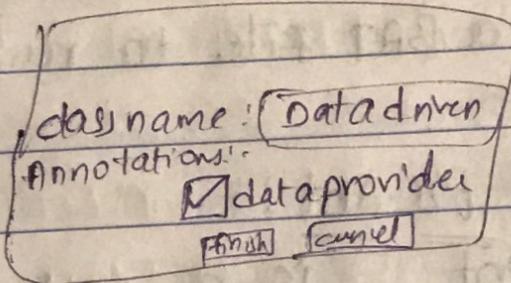
Now this test is skipped.

To create 3 test cases:-

what - Selenium London  
cucumber Glasgow  
testing manchester.

we can repeat the code 3 times but it is not recommended since it occupies <sup>more line or</sup> code

right we use data driven technique.  
right click on package name → Test → G → Create new TESTING class →



Now an auto generated code is generated  
change it according to our code.

```
package com.mysite.tests;  
import org.testng.annotations.Test;  
public class DataDriven {
```

Using Ant to run Selenium Tests.

### → Introduction to Ant

- Downloading and configuring Ant for Selenium.
- Creating & working with Ant Build.xml file.
- Running JUnit tests with Ant and report generation.
- Running TestNG tests with Ant and report generation.
- More focus on Results and Reports.
- Building a BAT file to run tests easily using ANT.

### Introduction to Ant :-

1. Apache Ant is a software tool for automating build processes. It's mainly used for java applications.
2. Ant is written in Java.
3. Ant configuration files are XML-based and the main configuration file is build.xml.
4. Ant is really useful if you want to run your project from command line or on other operating systems like Unix, Solaris where Eclipse is not available.
5. Ant provides portability so that we can build and run Ant on any platform.

## ④ Downloading ANT:-

1. Ant require Java to run it. So make sure that before running downloading Ant you have Java in your system.

search in google Ant.

apache Ant - welcome ← click.

→ Binary Distributions.

click on zip file.

Save it.

extract all

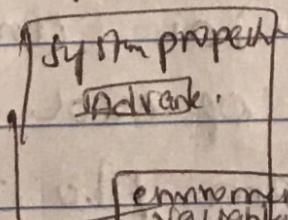
go inside →. → Inside bin we have bat file.

→ Configure Ant :- we have to go to system environment variables to do that

→ right click on PC → system environment variables

→ Click on properties.

→ Advance System Settings.



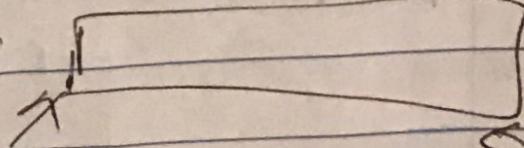
click.

click on new

variable name

ANT-HOME

variable value



Ant files location. copy past link

to access file  
Syntax      \$ (property name)

make sure that Java-HOME system variable is available.

Add Ant bin location to system pathvariable  
path click.

or location or  
ant files  
C:\softwares\apache-ant-1.10.0\bin  
After ; %ANT\_HOME%\bin; place ; at last  
C:\softwares\apache-ant-1.10.0\bin  
you can see ANT version also.  
by writing ant -version  
tools.jar unable to find → if we  
get this type of error.

C:\program files\java\jdk 1.8.0-112  
:lib  
you will find tools jar file copy  
it and ~~paste it in~~ go to jre 1.8.0-102 :lib  
paste it here Now goto command  
prompt and type ant enter

1. G1: to go into that folder in cmd ->  
2. cd - path (clone) of build.xml file.  
3. run ant & run ant & run  
4. call cmd & open  
5. This is the way to run a build at Junit.

There is no default build.xml file. we need to create one based on our requirements.  
Here we will create one for Junit and one for TestNG.

html reports are generated in Eclipse.

test suite property. enter  
run target specific to TestNG  
taskdef tag enter.  
run suite target Spec in "

Save the batch files with .bat extension.  
Now create batch file for Junit projects also