* Constructor :-

1. It is special type of method.
2. It is invoked at the time object creation.
3. It is used to initialise the value to the object of the class.
4. Constructor name and class name should be always same.
5. Constructor should create in its own class. It doesn’t have any explicit return type.

\*syntax=> class Demo

public class Demo

{

public Demo( )

{

}

}

* What is Manual Testing?

Tester Manually Performs Steps on Application then It is called as Manual Testing.

* What is Automation Testing?

Test Case:

1. Enter UserName

2. Enter Password

3. Click on Login

4. Click on Logout

* Tool Automatically Performs steps on Application then it is called as Automation Testing.
* Test Scripts: - When test case is converted in the language of tool then it is called as “Test Script”.
* Limitations of Manual Testing:

1. Takes more resources

2. Takes more time

3. Human errors are introduced

4. Inconsistent

5. Not Reliable

6. Performance Testing is not Possible Manually

* Advantage of Automation Testing

1. Less Resources

2. Less Time

3. Human Errors are minimized

4. Consistent in Output

5. More Reliable

6. Performance Testing is Possible

* Which test cases we will put for automation testing??

1. Repetitive test cases/re-testing
2. Regression testing
3. Performance testing
4. Data driven testing---multiple set of data/parameterization
5. Cross-browser testing
6. Cross-platform testing
7. Multilingual application
8. High risk business test cases

* Which test cases we will NOT put for automation testing??

1. Test cases which are not tested manually at least once…
2. Test cases which are tested on Ad-hoc basis/ random testing.
3. Test cases for which requirements are changing frequently.

* According to the purpose of testing there are diff. tools available in market :-

1. Test Management Tool: QC(ALM), JIRA, TestLink, Tarantula (STLC phases)

2. Defect Reporting Tool: Mantis, Bugzilla, QC-ALM(HP), Redmine, JIRA etc. (Defect Life Cycle)

3. Functional Testing Tool : Selenium, QTP(UFT), RFT

4. Performance Testing Tool : Jmeter, LoadRunner, RPT

Selenium:-

1. Functional testing tool
2. Designed to test web applications
3. Free and open source
4. Best suited for regression testing.
5. Supports multiple languages …. Java, python, c#, ruby etc.
6. Supports multiple browsers ….. IE, chrome, firefox, edge etc..
7. Supports multiple platforms --- linux, windows, mac etc.

7) Selenium suite ----

A] Selenium IDE

B] Selenium RC

C] Selenium Grid

D] WebDriver

\* **Birth of selenium** :🡪

-- Jasson huggins in 2006

-- JavaScriptTestRunner

-- Selenium core

* **Same Origin Policy** :🡪

|  |
| --- |
| Selenium  Core/ java script |

|  |
| --- |
| AUT --- google |

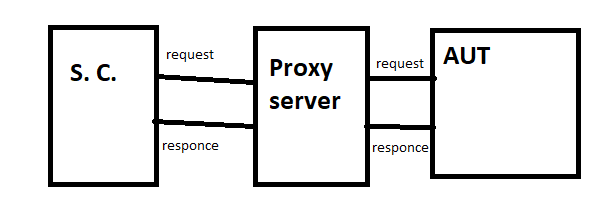
|  |
| --- |
| AUT ---yahoo |

www.google.com :8020

www.yahoo.com

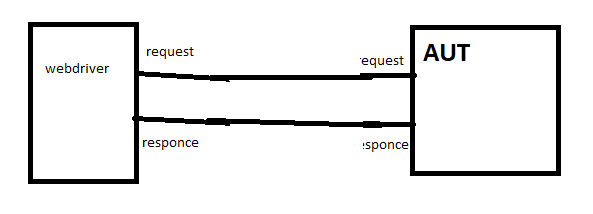
1. Selenium RC—Remote control/Selenium 1

Paul Hammant



Single machine + multiple browsers

1. **Webdriver**
2. Functional testing tool
3. Designed to test web applications
4. Free and open source
5. Best suited for regression testing.
6. Supports multiple languages …. Java, python, c#, ruby etc.
7. Supports multiple browsers …... IE, chrome, firefox, edge etc...
8. Supports multiple platforms --- linux, windows, mac etc.
9. Webdriver directly communicates with web browser and uses its native compatibility properties to automate the application.
10. Test design tool
11. It is created by Simmon Stewert.

Cross platform testing tool

AUT-application under test

Single machine + multiple browsers

1. Selenium grid

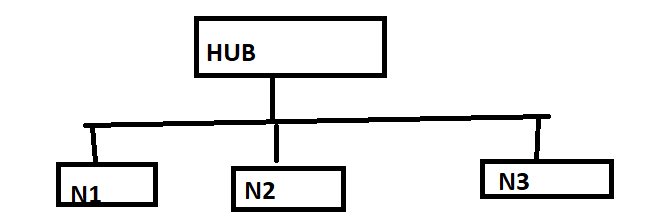
Patrick Lightbody

Test execution tool

HostedQA-----Selenium Grid

Parallel execution --- can execute multiple test cases at the same time.

Multiple machines + multiple browsers



1. Selnium IDE

Shinaya Kasatani ….. japan

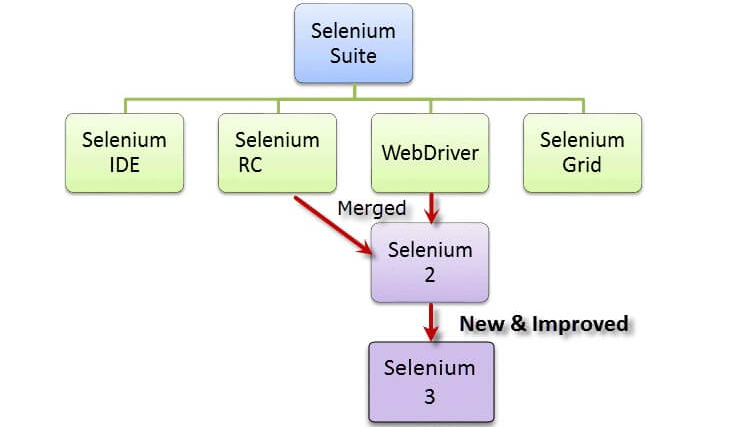
------ record & playback tool

------ available as firefox plugin + chrome plugin

\*Limitations of Selenium IDE

1. It doesn’t support oop concept.
2. It doesn’t support database like excel, oracle etc.
3. It generates only logs not detailed report.

2008



Selenium 2-------🡪 selenium 3---3.141.59

Selenium 4.0 – alpha version--- not stable

* Why Selenium ?

QTP – HP- Quality Test Pro

Mercury interactives ---

\* Environment setup for selenium -----

1) jdk

2) ide--- eclipse, myeclipse, altova

3) selenium jar files

(seleniumhq.org) selenium.dev

4) browser drivers ---exe files

i) chrome – chromedriver- webdriver.chrome.driver

ii) firefox – geckodriver - webdriver.gecko.driver

* get(String arg) :🡪 It is used to open application url in current opened instance of web browser.
* close() 🡪 to close current single opened instance of web browser
* quit() 🡪 to close current multiple opened instances of web browser
* **Locator** :--- It is a unique value by using which we can identify web-element over webpage very accurately.

Types of locators ----

1. Id
2. Name
3. className

If space is present in between value of class attribute then instead of space put “.” over there.

1. linkText

when anchor tag i.e. <a>….</a> is present in page source ….then we can use this locator. It will take actual visible text as a parameter which has hyperlink..

e.g. -- <a href="/index.php/auth/logout">Logout</a>

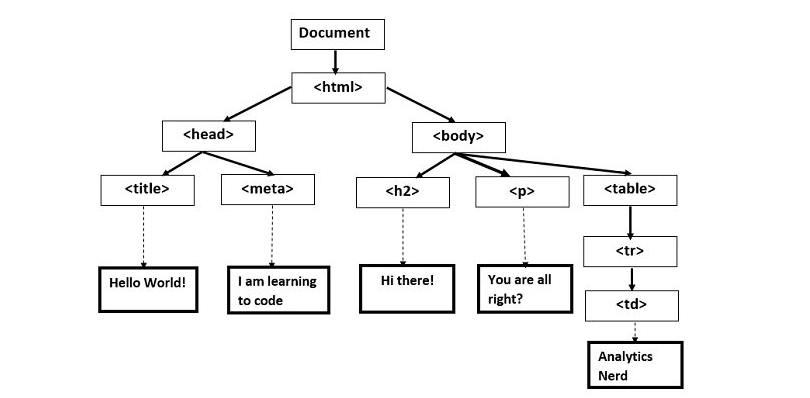
wd.findElement(By.linkText(“Logout”)).click();

1. cssSelector

css -- cascading style sheet

\* ways of cssSelector

1. tagname#id
2. tagname[attribute name=value of attribute]
3. tagname.className
4. xpath :--
5. In xpath, x stands for xml path. It is basically used to find xml elements but we can use xpath to locate html elements also, as both are markup languages.
6. It manipulates HTML DOM structure to find webelement over webpage
7. HTML DOM (Document Object Model) structure



A] **Absolute xpath**

1) It is direct way to find webelement as it starts with single forward slash (/). It means we can select the webelement from root node.

2) It gives xpath from root node i.e. <html>

3) Disadv.— If there are any changes made in absolute xpath then that xpath will get fail.

e.g.--- /html/body/table/tr/td[2][@id=username]

**B]** **Relative xpath**

1) it starts with //, which means it can search webelement anywhere on the webpage.

2) it starts in between html DOM structure

Ex.--> //\*[@id=\"txtUsername\"]\td\.... ---- system generated relative xpath

Syntax for creating xpath: --

//tagname[@atrribute=”value of attribute”]

* **Controls---**

1. **Dropdown** :----
2. To select one element from multiple elements of dropdown we can use “Select” class. It is predefined class of selenium.support.ui package.
3. When <select> tag is present in html source code then it is 100% dropdown.
4. There are some methods of select class to select 1 element from dropdown….

**A] selectByIndex(int index)**:🡪

by using index no.-----index starts with 0

**B] selectByValue(String args)** :🡪

by using value attribute of option tag.

ref.--> <option value=”123”>jkjkkh</option>

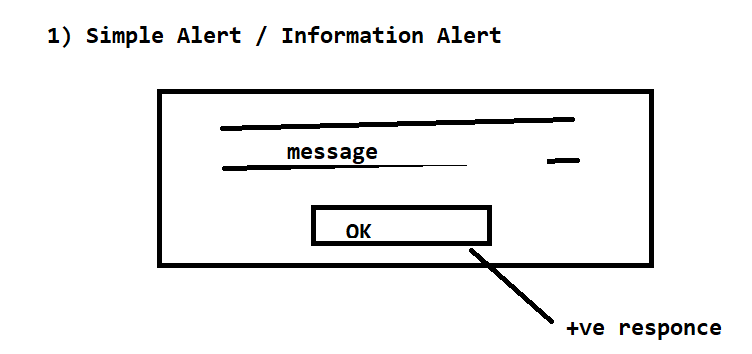
**C] selectByVisibleText(String args)** :🡪

by using actual visible text of option

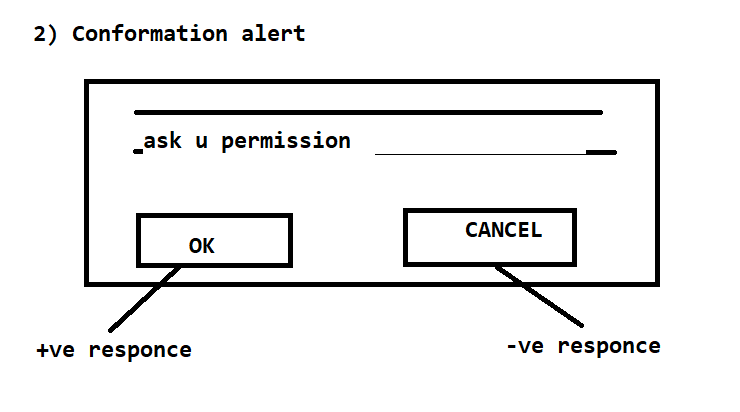
**\* Alert :🡪**

\* To handle onscreen notifications, we can use “Alert” interface of Selenium.

\* Types of Alert :--



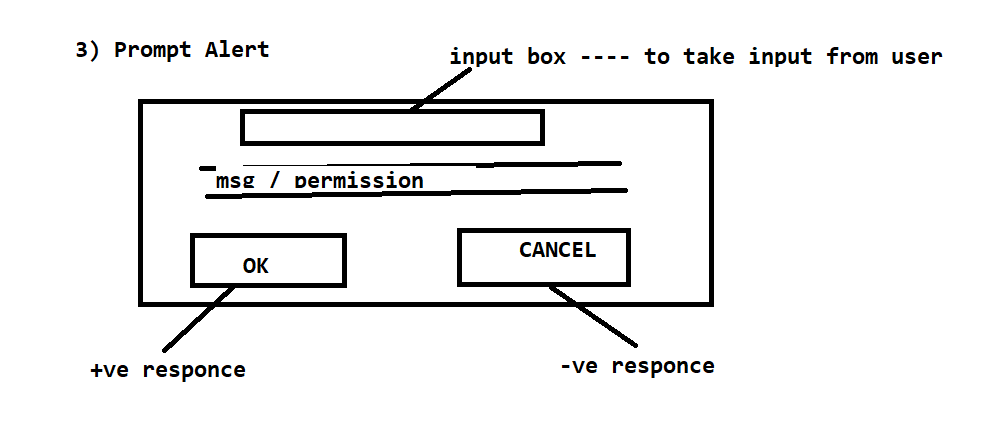
* This kind of alert can be used to display some information/ message/ warning on screen.



* This kind of alert can be used to ask permission to perform some kind of operation.
* It contains two buttons –

1) Ok -- +ve responce

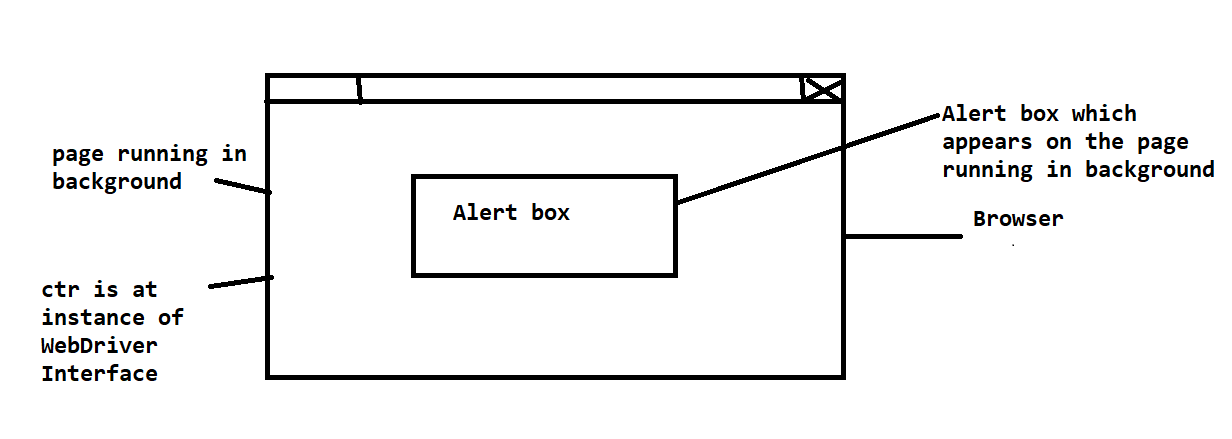
1. Cancel -- -ve responce



* It can be used when some kind of input is required from user.

According to the response, there are diff. methods of Alert Interface is present ----

|  |  |
| --- | --- |
| **Response** | **Method** |
| +ve response (ok button) | accept() |
| -ve response (cancel button) | dismiss() |
| Input data | sendKeys() |
| Capture text | getText() |



Now we have to switch the ctrl from page running in background to the Alert.

For this purpose, we can use swithTo() method of WebDriver Interface.

Alert alt=wd.switchTo().alert();

\***Actions** :-

i> To handle special events of mouse and keyboard, we can use AdvancedUserInteractions API. It contains two classes - Actions and Action classes of selenium.interactions package .

ii>

|  |  |
| --- | --- |
| **Actions** | **Action** |
| To capture single special event | To build/create series of special events |
| .perform( ) | .build( ) |

iii> Syntax :-

Actions act=new Actions(wd);

iv> To move the scope of the cursor to the middle of the element, we can use moveToElement( ) method of Actions class.

\***List** :🡪

i> To handle multiple elements which are having same property, we have to first group them together. To create group, we can use List Interface of java.util package

ii> Syntax :-

List<WebElement> instance=wd.findElements(target);

iii> findElements() :🡪 This method is used to find multiple elements which are having same property. It returns list of web elements.

\*CheckBox

\***isSelected()** :🡪 it is predefined method of WebElement Interface. It returns boolean value. It is used to check whether particular webelement is selected or not.

\***isDisplayed()** :🡪 it is predefined method of WebElement Interface. It returns boolean value.

It is used to check whether particular webelement is getting displayed on webpage or not.

\***isEnabled( )** :🡪 It is predefined method of WebElement Interface. It returns boolean value. It is used to check that webelement is enable or disable.

\* **Wait** :🡪

Here two applications are interacting with each other ---

A] Selenium webdriver- which is firing queries on AUT with constant speed

B] AUT – which is responding to the queries with same speed

Sometimes AUT is not able to respond to the queries in desired time.

This is called as synchronisation issue between two applications.

To handle synchronisation issue between two applications, we can use wait mechanism of selenium WebDriver.

**Types of wait** :-

**1) Implicit Wait / Global wait** =>

i> As it is applicable to all line of statements present in a script, hence it is called as Global wait…

**syntax** :-

wd.manage().timeouts().implicitlyWait(timeduration, TimeUnit.**SECONDS**);

complete time (to throw an exception)

1. It takes only required time for webelement to get visible on webpage not a complete time.
2. When complete time will exceed, then it will throw an exception.

**2) Explicit Wait** ----

i> Applicable to particular line of stmt in a script.

ii> This type of wait is used to tell the web driver that wait for certain conditions i.e. expected conditions before throwing an exception if complete time period will exceed.

syntax :-

WebDriverWait wt=**new** WebDriverWait(wd, timeOutsInSeconds);

Instance of WebDriver Complete time

Ex.-->

wt.until(ExpectedConditions.visibilityOfElementLocated(ByLocator));

**3) Fluent Wait** =>

**\*Screenshot :-**

TakesScreenshot ts=(TakesScreenshot)wd;

//File sf=ts.getScreenshotAs(OutputType.FILE);

//File df=new File("./OHM\_Home.png");

FileHandler.copy(ts.getScreenshotAs(OutputType.**FILE**), **new** File("./OHM\_Home1.png"));

**\*Framework** :-

A] It is structure, by using which we can achieve quality product.

B] Without framework, we place code as well as data in same script which is not readable/reusable.

C] Adv.-

i> code reusability

ii> code maintenance will be easy

iii> length of code will reduce

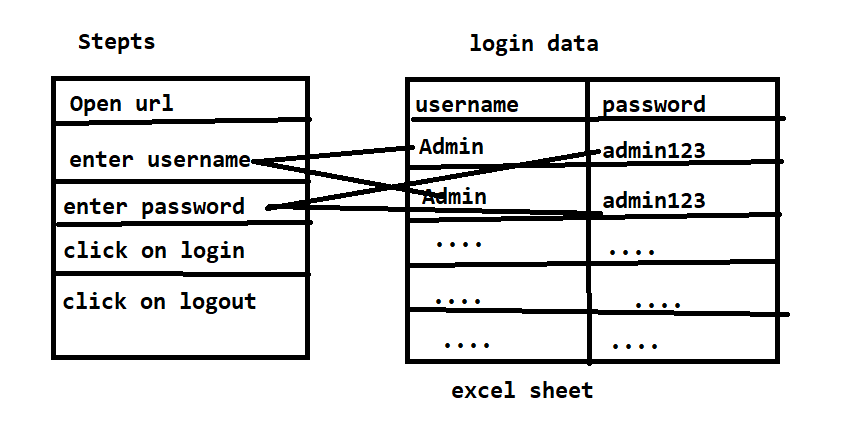
\* Types of Framework :🡪

1. **Data Driven Framework / Parameterization**

a] Framework which is driven by data is called as Data Driven Framework

b] To test application across multiple set of data we have to use this framework.

C] In DDF, we are going to separate data from the script. Data will be at other location and script will be at another location



d] In script, instead of actual credential we will put variable at sendKeys(un).

e] FileInputStream class --- to read the file we can use FileInputStream class of java.io package

Syntax:🡪

FileInputStream object=new FileInputStream(path of the file);

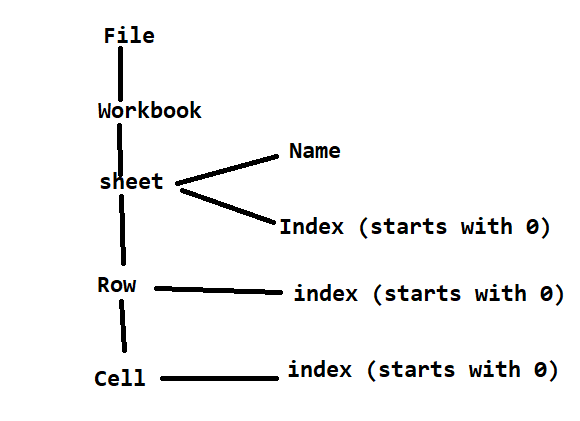
f] Structure of excel sheet:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Workbook | | | | | |
| Un | pwd |  |  |  |  |
| F1 | F1 |  |  |  |  |
| F2 | F2 |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |

cells

EXCELL FILE

1. xls ---- Excel 97-2003 Workbook
2. xlsx ---- excel workbook



G] To actually read data from the excel file, we have to use POI.

\***POI** 🡪

i> product of apache

ii> available in jar format

iii> it is used to perform read and write operation on MS-Office products.

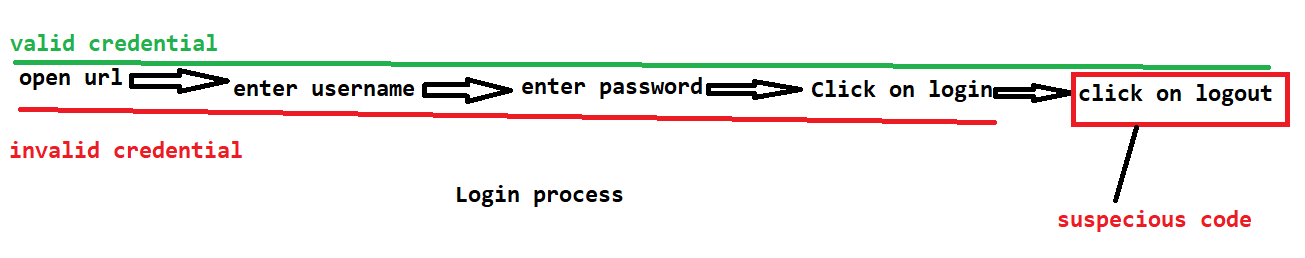
e.g. 🡪 Excel

How to download :--- [www.apache.org](http://www.apache.org)

We can save excel file in 2 formats – xls & xlsx

|  |  |
| --- | --- |
| **xls** | **xlsx** |
| HSSFWorkbook | XSSFWorkbook |
| HSSFSheet | XSSFSheet |
| HSSFRow | XSSFRow |
| HSSFCell | XSSFCell |

Predefined classes in jar files of POI



**ii> database testing (phpMyAdmin)-jdbc**

* to crate connection with database – download jar file--- mysqlconnector.jar

It contains some predefined classes, interfaces to create connection with database 🡪

1. Connection Interface -to create connection with database
2. getConnection()method of DriverManager Class – for actually creating connection with database

getConnection(“url”,”username”,”password”)

1. url(database url)

jdbc:mysql://db4free.net:3306/information\_schema

jdbc:mysql://dbpanel.squadinfotech.in/localhost:3306/databasename

1. database username
2. database password
3. To fire query on database , we have to use Statement Interface. According to the type of the stmt, diff. methods of Statement Interface are available ----
4. execute()-- returns boolean value

**DDL – data definition lang**

**Create, alter, rename, drop, truncate**

1. executeUpdate()--returns integer value

**DML—data manipulation lang**

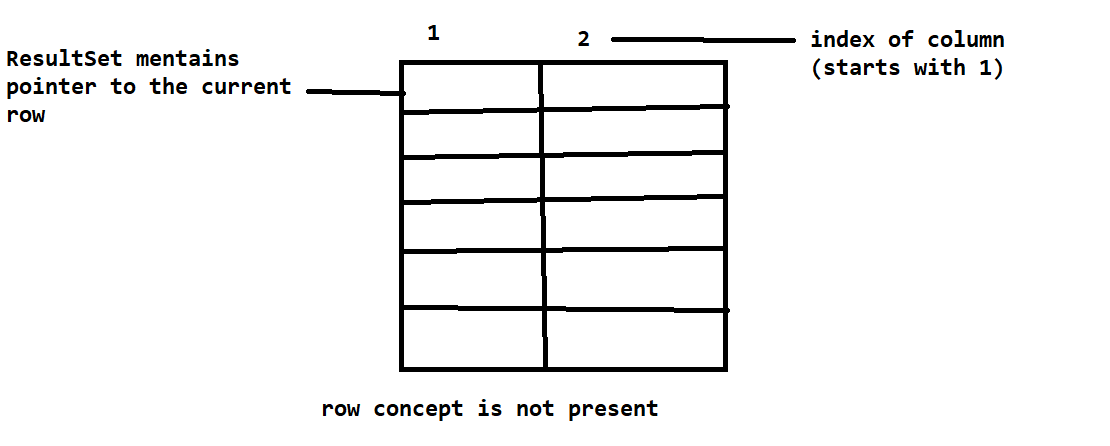
**Insert, update, delete**

iii> executeQuery() – it returns ResultSet

**DQL—data query lang**

**Select**

\*it returns table kind of structure. To handle this table kind of structure, we have to use ResultSet Interface. It maintains pointer to current row. As row concept is not present, to move the pointer to the next row, we have to use next() method of ResultSet interface

****

**update** [ENGINES](https://db4free.net/phpMyAdmin/sql.php?db=information_schema&table=ENGINES) **column1=’abc’ where column2=’value’**

**\*POM – (Page Object Model)**

A] e.g.--- Mantis application ---- 50 test cases ----

i> view issue

ii> report issue

b] Design pattern not a framework.

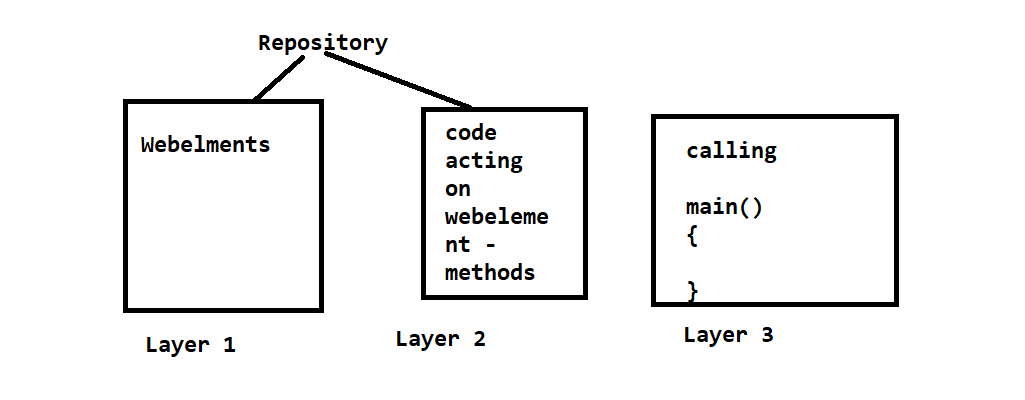
C] Can implement this design pattern in any framework.

D] We can design framework in more structured way by using POM.

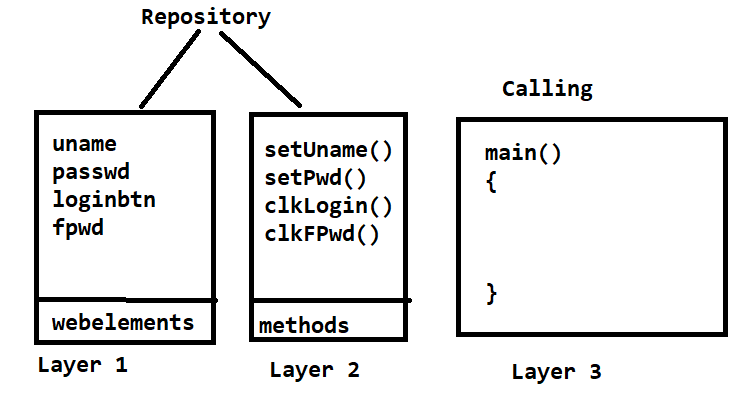
E] There is page wise distribution in POM.

F] In POM, we are creating repository of webelements and code acting on these webelements i.e. methods.

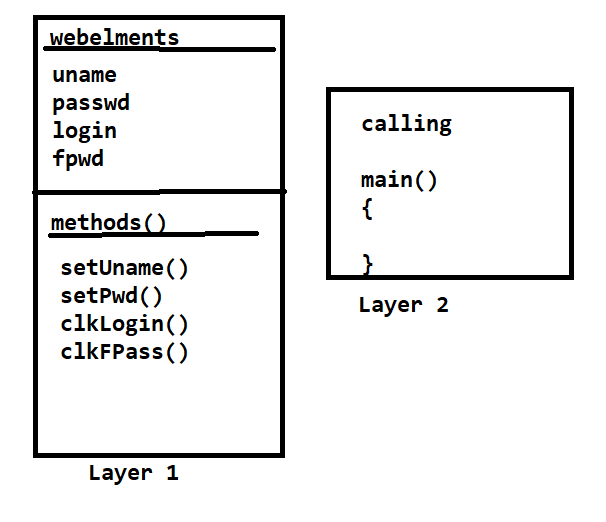
(Repository :--- It is a common location for storing objects.)



Ex.- Login Page Repository



g] Now a days in industry, layer 1 and layer 2 are merged together.



\* Adv. Of POM :-

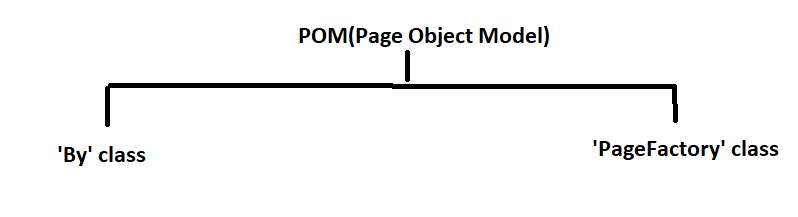
I] Code reusability

II] Code maintenance will be easy

III] Easily find out error location

IV] Length of code will reduce.

H] We can implement POM by two ways –



2**) Keyword Driven Framework**

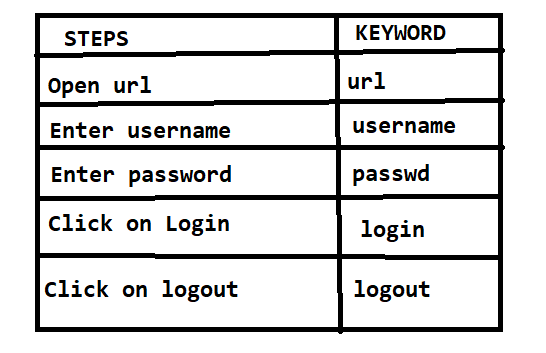
i> Which is driven by the keyword is called as KDF.

ii> Keyword has some predefined meaning.

e.g.- int a;---can store integer type data

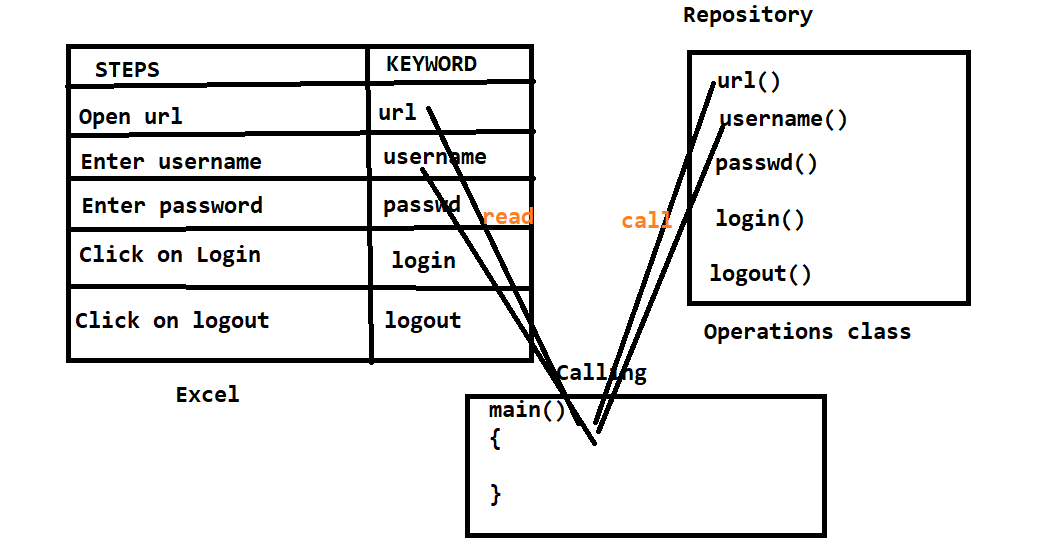
iii> Keywords are used to represent actions/steps which are involved in script writing.

Ex.- Login Test script



**Excel File**

iv> In industry generally this sheet is created by sr. or experienced tester and will be given to jr. tester for testing purpose only.

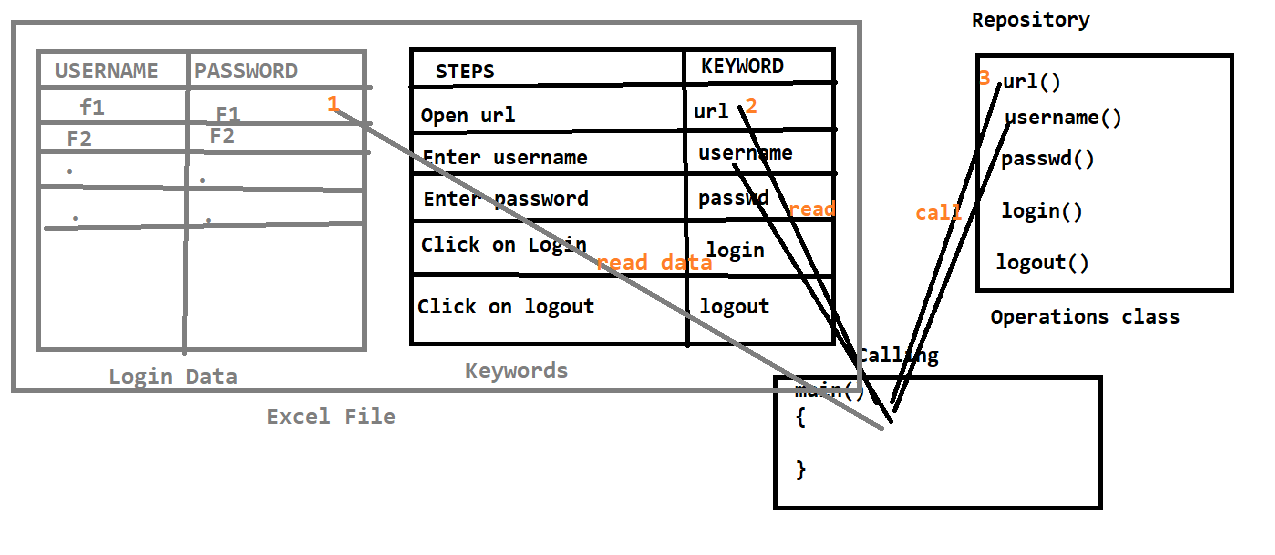


**Keyword Driven Framework**

v> In KDF, no. of keywords in excel file and no. of methods in repository should be same and name also should be same.

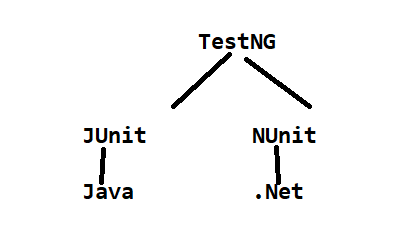
**3) Hybrid Framework**

DDF + KDF = Hybrid Framework



**4) TestNG-Next Generation**

JUnit & NUnit (for unit testing)



As it is inspired from these 2 frameworks, it also contains annotations (@).

It has various advantages over Junit.

**Features of TestNG :🡪**

1. **Annotations** :-

A] It gives information about the method written below it.(order of execution). It has 10 no. of annotations.

**\* Order of execution of TestNG annotations**

**@BeforeSuite**

**@BeforeTest These annotations will execute only once.**

**@BeforeClass**

**@BeforeMethod**

**@Test Execution of these two annotations is**

**@AfterMethod depend upon @Test annotation**

**@AfterClass**

**@AfterTest These annotations will execute only once.**

**@AfterSuite**

B] TestNG class can contain multiple @Test annotations

C] Priority :--

1) To set priority of @Test annotation we can use “priority” attribute.

2) The @Test which has low priority will execute first.

3) For execution of @Test annotation, TestNg considers alphabetic order

1. **Reports** :- It generates reports in html format.

A] index.html

B] emailable-report.html

C] Default test.html

1. **Dataprovider** :-

A] To achieve parameterization we can use @DataProvider annotation.

B] @DataProvider provides data to test annotation.

Ex. 🡪

@Test(dataProvider = "dp")

**public** **void** f(Integer n, String s) {}

@DataProvider

**public** Object[][] dp() {

**return** **new** Object[][] {

**new** Object[] { 1, "a", },

**new** Object[] { 2, "b", },

};

}

1. **TestSuite** :- Test Suite is collection of multiple test cases. With the help of TestNg Xml Suite, we can execute multiple test cases at the same time. This is called as Parallel execution. It saves test execution time.