JAVA, RUBY,PYTHON,PERL,PHP,JS

Prereq: (Pre recorded Sessions on JAVA)

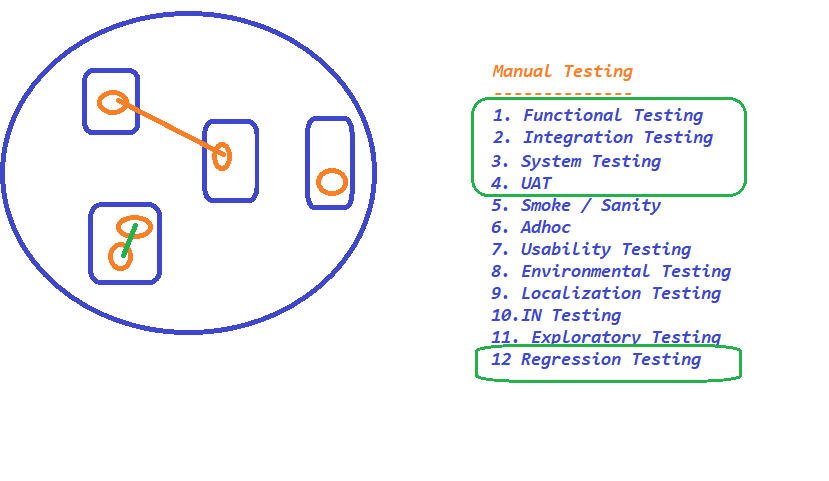
JAVA –

1. Basics of JAVA
   1. How to write a Class
   2. How to write a Method
   3. How to Call the methods

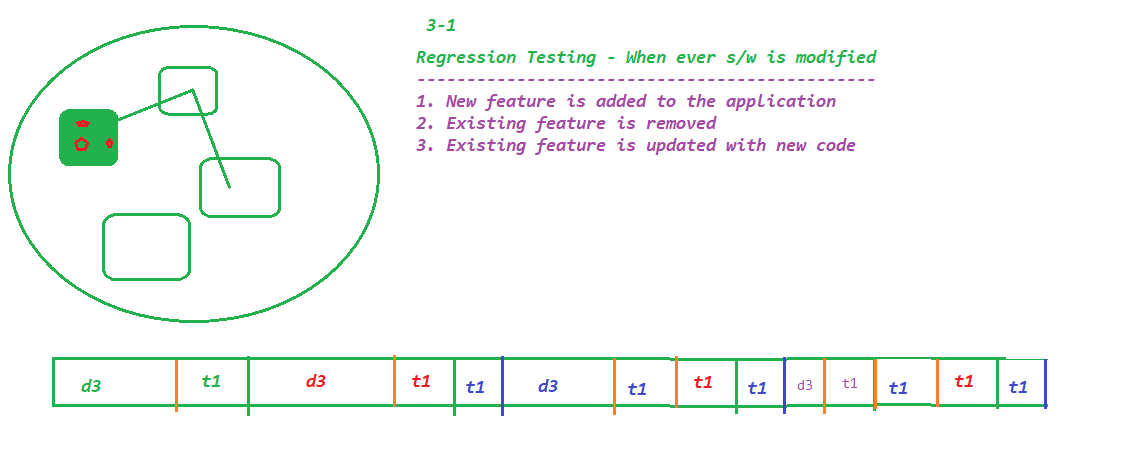
* OOPS Concept ( abstraction, Encapsulation, polymorphism, inheritance )
* Class
* Objects
* Interfaces
* package

1. Variables
   1. local variables
   2. global variables
      1. instance variables
      2. static variables
      3. constants
2. Methods / functions
   1. methods without arguments
   2. methods with arguments
   3. methods without return type
   4. methods with return type
   5. recursive ( method inside a method )
3. Constructors
   1. default
   2. parameterized
4. Access specifiers ( data hiding )
   1. private
   2. default
   3. protected
   4. public
5. Access modifiers ( change the behaviour of a members of a class)
   1. static
   2. final
   3. abstract
   4. synchronized
6. String in JAVA(java.lang)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*(reverse, reverse word, change all characters from ‘a’ to ‘A’)
   1. String functions
   2. StringBuffer
   3. StringBuilder
7. Threads
   1. How to achieve multithreading
8. Exception Handling
   1. try
   2. catch
   3. throw
   4. throws
   5. finally
9. File Handling in Java – properties or text
   1. Read a file (InputStream)
   2. Write into a file (OutputStream)
10. Collections
    1. List
    2. Set
    3. Map
11. Generics – Parameterized Collections

Manual Testing :



Regression Testing :



Selenium:

- What is Automation Testing ?

- Why Automation is required ?

- Different Tools ?

1. Test Complete
2. UFT (QTP)- HP
3. Sahi
4. Selenium
5. Katalon Studio
6. SoupUi
7. Cucumber
8. Waitr
9. RFT (Rational Fu Tester) - IBM
10. Test Studio
11. Rspec
12. SilkTest
13. EggPlant
14. Sikuli
15. AutoIT
16. .....

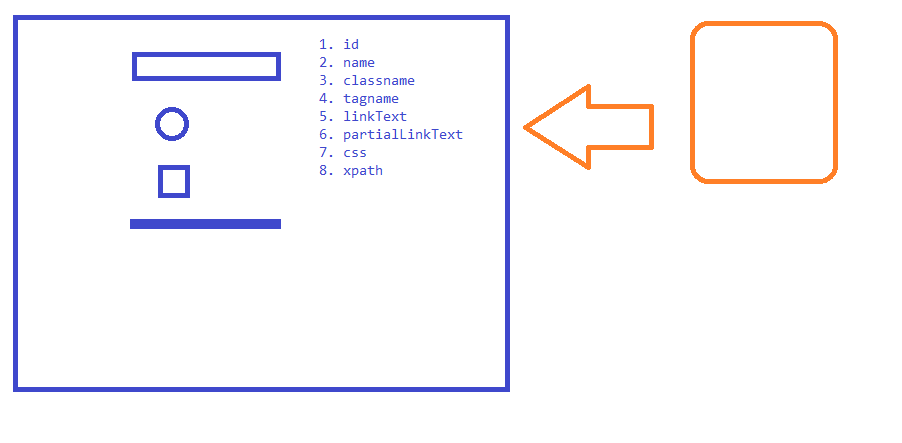
- Why Selenium is popular ?

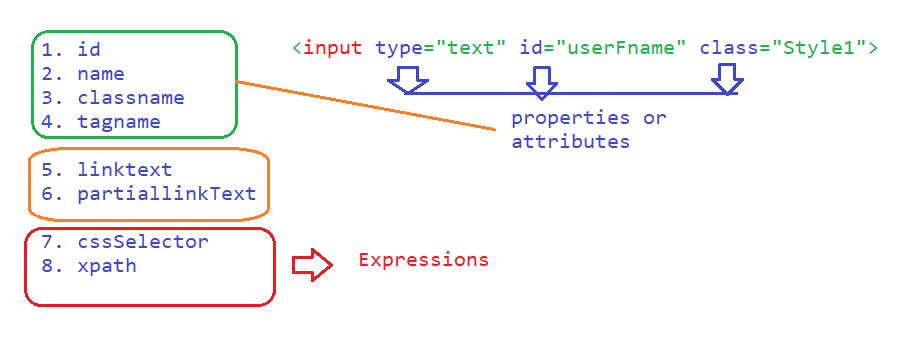
1. Free
2. Support Multiple Programming Languages
   1. c#
   2. ruby
   3. perl
   4. python
   5. java
   6. javascript
3. It support almost all the popular browsers available in market
4. No Dedicated machine is required to execute the script
5. Parallel Execution
6. Distributed Execution
7. Platform Independent

- Selenium IDE

* Beginners
* POC
* Add-on with FF and Chrome

Identifiers :





Priority –

First preference – *id*

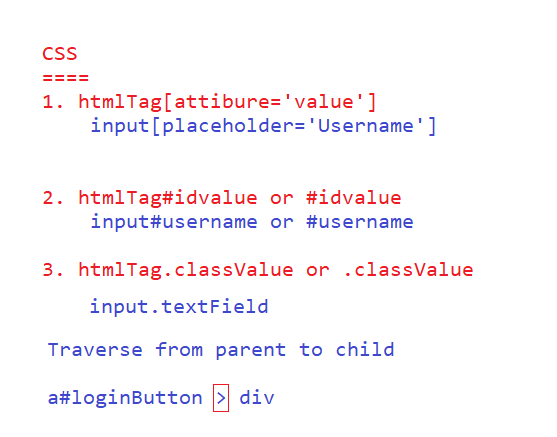
Second preference – *name*

Third preference – *xpath* / linktext /

***NOTE***

- don’t use id if it is having alphanumeric value (auto-gen-\* , ext-gen\*)

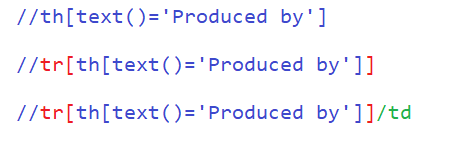
-CSS



a[class='initial'] > div

-xpath

1. Absolute xpath -> practically difficult to write or mantain
2. Relative xpath

* basic 🡪
  + //htmltag
* with filters
  + //htmltag[@attribute=’value’]
  + //input[@name='pwd']
* with logical operators
  + and
    - //input[@id='username' and @name='pwd']
    - //input[@id='username' and @name='username']
    - //td[text()='16' and @class!='past day']
    - //td[text()='16' and (@class='current day' or @class='wd day' or @class='we day')]
  + or
    - //input[@id='username' or @name='pwd']
* with functions
  + text()
    - //div[text()='Login ']
  + contains(*arg1*, *arg2*)
    - arg1 🡪 attribute or function
    - arg2 🡪 partial value respectively
    - Google Search - //h3[contains(text(), 'Intellipaat:')]
    - acti-time logo - //img[contains(@src,'timer.png')]
  + starts-with(arg1,arg2)
    - arg1 🡪 attribute or function
    - arg2 🡪 partial starting value of a Element
    - Google Search - //h3[starts-with(text(), 'Intellipaat')]
    - actittime task page - //button[starts-with(@id,'ext-gen')]
  + Travers from **parent to child**
    - //a[@id='loginButton']/div
    - //table[@id='loginFormContainer']//a/div
  + Travers from **child to parent ( whenever we are playing with dependent and independent elements )**
  + 
    - //tr[th[text()='Produced by']]/td
    - *gsmarena* - //tbody[tr[th[text()='Platform']]]//td[@class='nfo']
    - //tbody[tr[th[text()='Display']]]//td[@class='nfo']
* with axes functions
  + traverse between siblings Only search for sibling tags
  + 1. Following-siblings
    - Syntax -//xpath\_of\_Ind\_Ele/**following-sibling::**SiblingTAG
    - Ex - //th[text()='Produced by']/following-sibling::td
    - //li[a[span[text()='Soundtrack']]]/following-sibling::li
  + 2. Preceding-Sibling
    - //li[a[span[text()='Soundtrack']]]/preceding-sibling::li
  + 3. Following – it will search till the end of the page for the tag specified
  + 4. preceding - – it will search till the begining of the page for the tag specified
  + traverse to parent
    - //th[text()='Produced by']/parent::tr/td
  + traverse to child
    - //a[@id='loginButton']/child::div
  + traverse to grand parent
    - ancestor
    - //th[text()='Display']/ancestor::tbody//td[@class='nfo']
  + traverse between grand child

Syntax:

1. Relative xpath --------------> //htmltag[@attribute=’value’]
2. Logical operators---->//htmltag[@attribute1=’value1’ and @attribute2='value2']
   * //htmltag[@attribute1=’value1’ or @attribute2='value2']
3. Functions  
    //htmltag[text()=’value1’]  
    //htmltag[contains(text(),'partial-value')]  
    //htmltag[contains(@attribute,'partial-value')]  
    //htmltag[starts-with(text(),'value')]  
    //htmltag[starts-with(@attribute,'partial-value')]
4. Traversing from parent to child  
    //htmltag[contains(@attribute,'partial-value')]/childElementxpath  
    //htmltag[contains(@attribute,'partial-value')]//childElementxpath  
   //htmltag[@attribute=’value’]/child::ChildTAG
5. Traversing from child to parent

//childTAG[patentXpathExpression]

//child2[childTAG1[patentXpathExpression]]

//htmltag[@attribute=’value’]/child::ChildTAG1/child::ChildTAG2

6. Traversing between Siglings

//xpathForAnyTag/following-sibling::siblingTag

//xpathForAnyTag/preceding-sibling::siblingTag

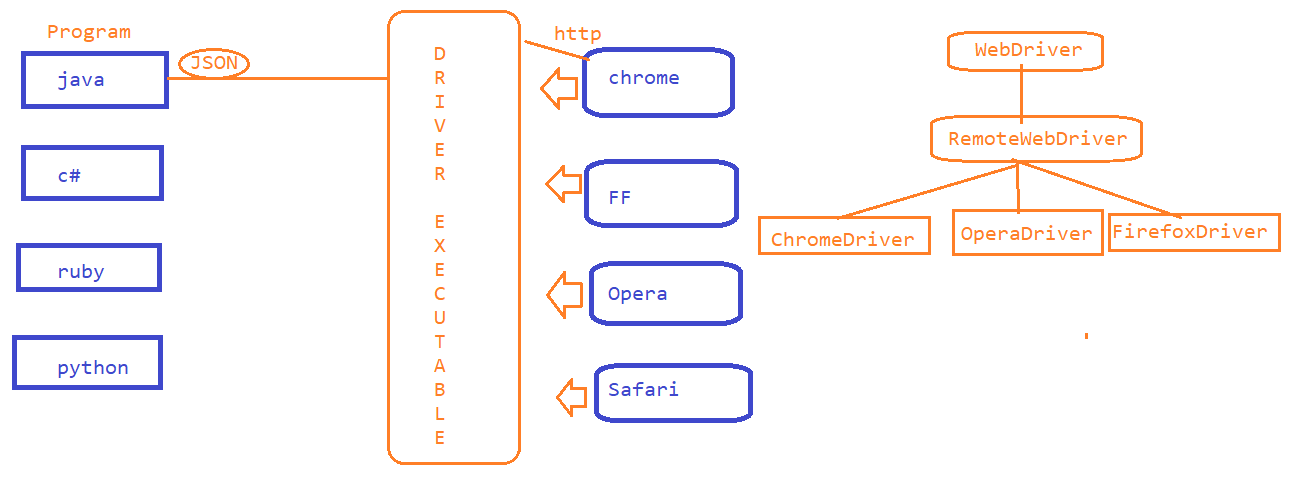
//xpathForAnyTag/ancestor::siblingTag

~~- Selenium RC~~

**- Selenium WebDriver**

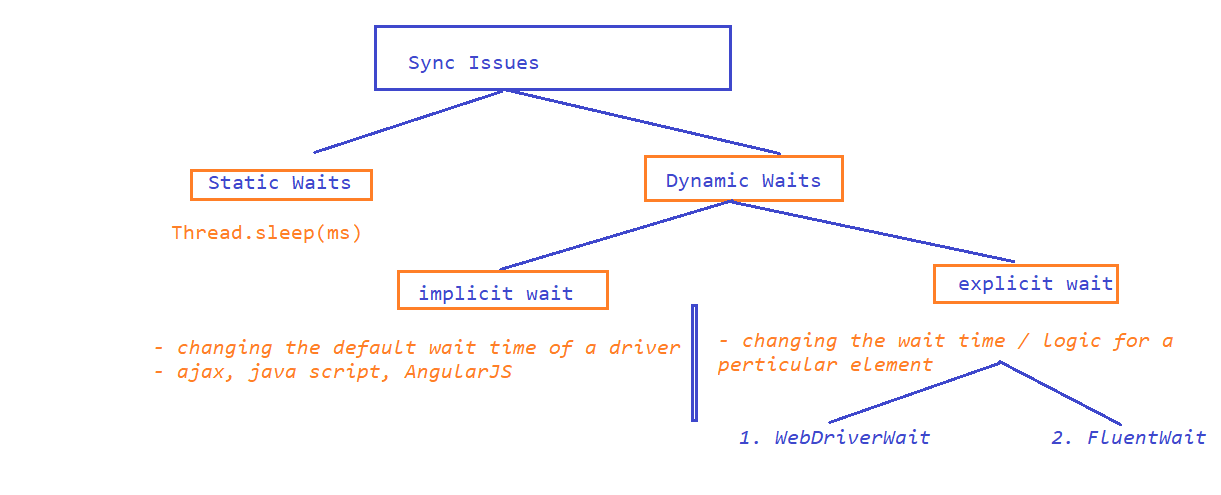
***Installation or Configuration***

1. *Create Java Project*
2. *Add Selenium libs to the java project created in Step1*
3. *Download Driver Executables*
4. *Start using the libraries and drivers to execute the test*

**

*Auto-Suggestions :*

*Sync Issue or Synchronization issue :*

**

*WebDriverWait –*

1. *Create a wait object by using WebDriverWait and specify the max time out*
2. *use until method present in wait Object*
3. *use ExpectedConditions to specify the condition*

*Drawbacks –*

*1. we have to rely on the existing methods present inside ExpectedConditions*

*2. we cannot change pooling time*

*3. If any exceptions you want to ignore*

*4. You can not write your own wait logic*

*Fluent Wait :*

- Selenium GRID

AutoIT : Third party tool

Frameworks :

- TestNG

- Page Object Model

Build Automation tools:

Maven

- Batch execution without opening a tool

Jenkins

- BDD -> Cucumber

Demo Application ->