Selenium Java

Monday, April 5, 2021   1:59 PM

Domain :BA

Programming:
    Knowledge :
1) Data type
2) Variables
3) Arrays
4) Conditions
5) Loops
6) Data Structures
7) Class
8) Object
9) OOPS

Start: 2 PM
Break: 4 PM : 15 mins
End : 6 PM

Theoretical+ Practical + Practice

15 Days

JavaScript
Java
Selenium
POM/POF
Junit
GIT/Maven/Jenkins
Github/BitBucket

Apply on Project

1) Web Applications
    a. Server
    b. Client (Browser)
        i. JavaScript : Logic
        ii. Html : UI
        iii. CSS : Styling
2) Windows Applications
    a. Executables
3) Desktop Applications
    a. Require additional software
    b. Java, .Net…etc
    c. Power Builder
4) Mobile
    a. Web
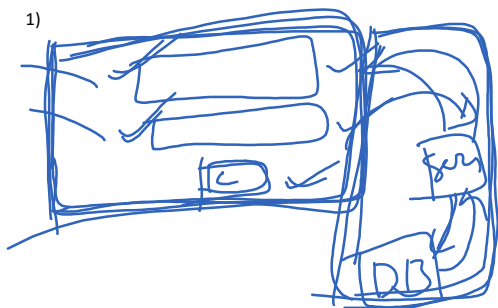   b. Native Applications (More Secured)
    c. Hybrid (Native + web)

V8
SpiderMonkey
WebKit
Chakra

JavaScript

1)

2009 : NodeJS

Install NodeJS to execute JS outside of browser

JavaScript can used in server side using NodeJS

Install NodeJS
Vscode : IDE

1) Register and login to https://github.com/
2) Click on New to Create Repository
3) Give Repository Name
4) Finish
5) Repository Created
6) Click on Creating a new file
7) Give file name as Demo.js
8) Click on Commit new file

https://github.com/qtpsudhakar/SynchronyJS

http://gitpod.io/#https://github.com/qtpsudhakar/SynchronyJS

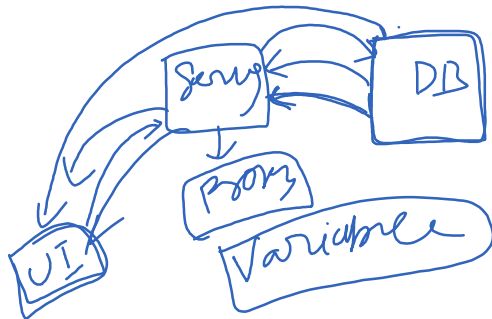http://gitpod.io/#https://github.com/UserName/RepositoryName

Datatypes:

Alpha Numeric Special

Numbers
Strings
Boolean

Variable: Which can stores data in run time

Datatype : Type of data

Variable : stores data :
We can store any type of data in a variable
We can use let to declare a variable
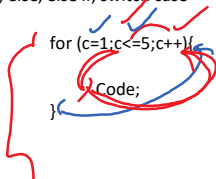Declaration is optional
"use strict" will forces declaring variable

Arrays:

Storing multiple values in a single variable using an index number

Condition: Used to execute block of code based on given condition
If, else, else if, switch case


```
for (c=1;c<=5;c++){
    Code;
}
```

Loop: Used to execute block of code again and again based on a condition or number of times
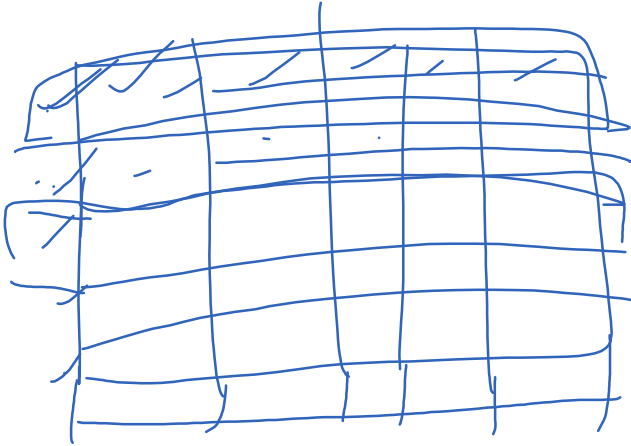
For Loop:
        i=1 : initialization
        i<=10 : condition
        i++ : incrimination

```
for(let i=1;i<=10;i++){
      Log(i)
}
```

Loops : Execute a block of code again and again base on condition or number of times

Functions: Reuse a block of code

Eclipse
IntelliJ

Java Development

Visual Studio : .net Development

Functions

1) Creating and reusing functions
2) Creating Parameters
3) Returning value
4) Returning multiple value
5) Passing multiple values

Functions: A reusable code block
Named Functions
Anonymous Functions : Assign as value to a variable, pass to parameter while calling function
        Call back
Parameters: normal, rest parameter (accepts any number of values)

Arrow Functions: short form for anonymous function (lambda expression)

We can return values using return keyword

How many return statements can be there in a function?

```
function isOdd(n){

      if(n%2==1){
             return true;
      }else{
             return false;
      }
}
```

Functional Programming

Write reusable code in functions
Call them where ever you want for creating test flow

Class
Object

Encapsulation
inheritance
Polymorphism

Developer develops
Developer tests

Improving quality

1995 : winrunner
Xrunner
Mercury interactive
QTP : 2002
6.5
2008 : res

2007 : iphone
2009 : android

Selenium 2006
WebDriver 2007

Selenium + WebDriver 2010
2011 : smart phones

Selenium for WebAutomation
World moves towards web

2014 : selenium crossed QTP

2018 : default for every tester

Java

Typed language
int x=10;

Datatype variable = value;

Data types and Variables

int x = 10;

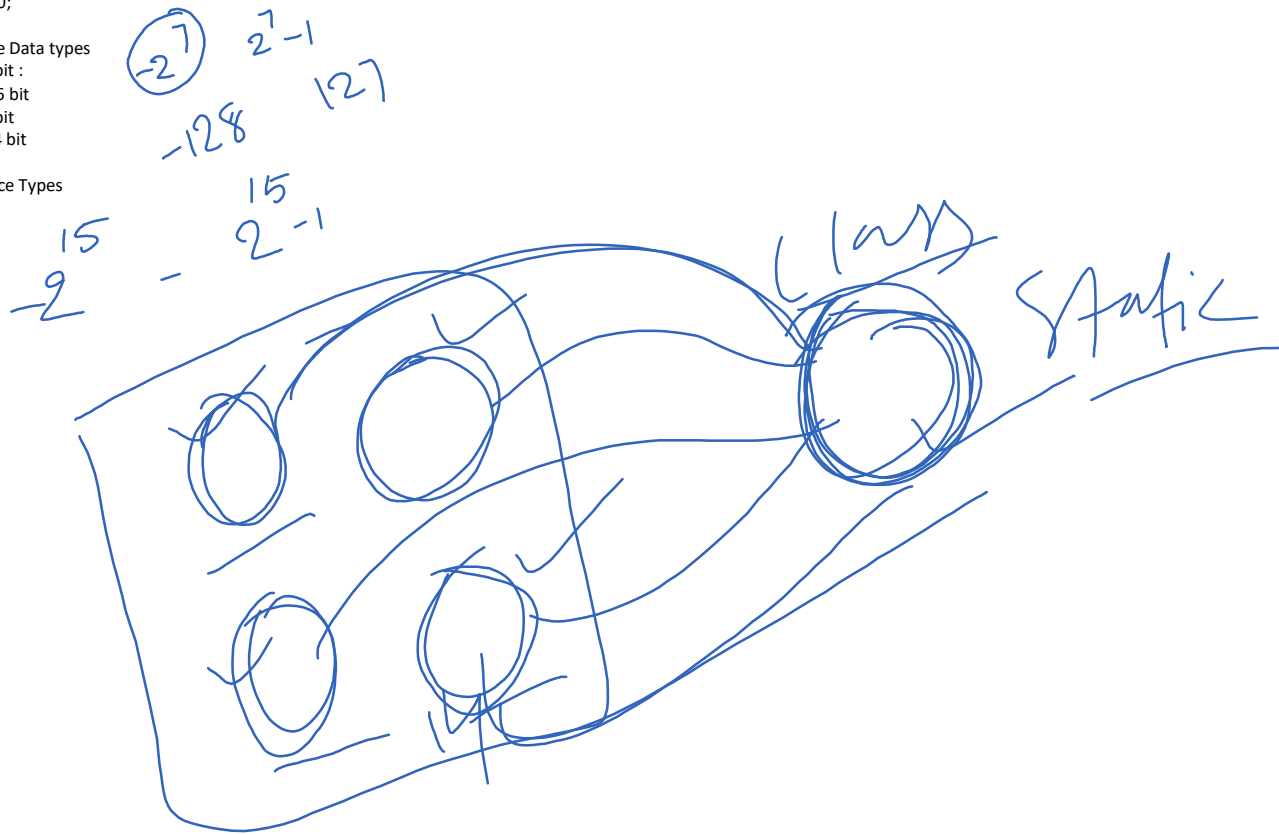Primitive Data types

$$2^7 \quad 2^7-1$$

int x = 10;

Primitive Data types
byte : 8bit :
short: 16 bit
int : 32 bit
long : 64 bit

Reference Types

-2^7

$2^7 - 1$

127

-128

15

$2^{15} - 1$

-2^{15}

(Class)

Static

Advantage of List over an array

Array is fixed length
Array can have primitive

List is dynamic
List can have Reference

Dynamic Data: Data generated in run time

Set : It can store only unique data
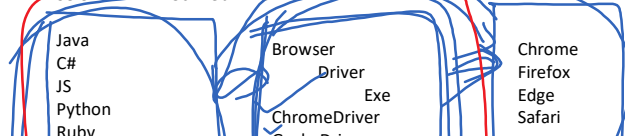Map : It can store data as key and value pair

Configure Selenium:

1) Goto https://github.com/qtpsudhakar/SeleniumJavaTraining
2) Download selenium-java-3.141.59.zip
3) Download chromedriver_win32 (2).zip
4) Create a folder in any drive
5) Copy these 2 files to that folder
6) Extract these files
7) One will have chromedriver.exe file
8) Selenium-Java-3.141.59 will have jar file and lib folder
9) In IntelliJ IDE -> File -> Project Structure-> Modules -> Dependencies
10) Click on + button and select jars
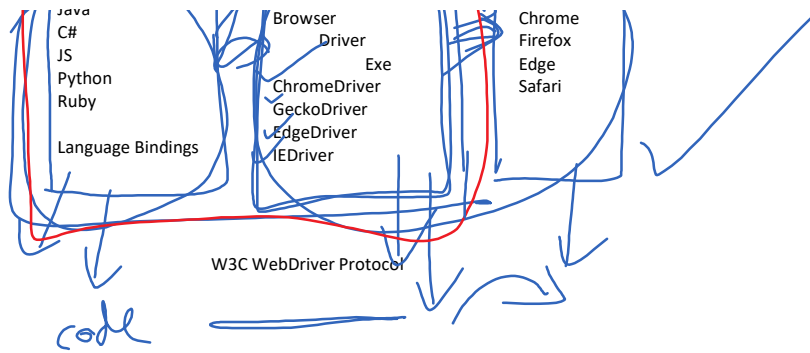11) Add all jar from that extracted selenium folder

Selenium 1 : Selenium RC : Remote Control
Selenium 2 : Selenium RC + WebDriver : Selenium WebDriver
Selenium 3 : Selenium 2 - Selenium 1
Selenium 4 : W3c WebDriver

Java
C#
JS
Python
Ruby

Browser
Driver
        Exe
ChromeDriver

Chrome
Firefox
Edge
Safari

Java
C#
JS
Python
Ruby

Language Bindings

Browser
Driver

Exe
ChromeDriver
GeckoDriver
EdgeDriver
IEDriver

Chrome
Firefox
Edge
Safari

W3C WebDriver Protocol

code

Name
ID

HTML + CSS + JS

Textbox	Input type="text"
Checkbox	Input type="checkbox"
Radio	Input type="radio" Radio buttons must groups with a name
Links	A href
Image	img
Lists/Dropdown	Select option
Para/header/display text	p, h1, span, li, label
File input	Input type="file"
Forms	We can submit
Button	Input type="button" Input type="submit" button
Calendar	Created using table or div
Table	Table, tr, td, th, tbody
Frame	iframe
Alerts	Created by javascript

Style attribute: to style a single element
Style tag:
1) Access all elements of specific type and assign a style
2) Store style with class names and assign that style to any element

Jquery
Angular
React
ExtJS
VueJS
……

Attributes:
1) Id
2) Name
3) Value
4) Type
5) Style
6) Class
7) Innertext

Enter Text	sendkeys
Click	click
Select	select
Mouse Actions	Actions class

Locators:

Id	<input type="text" id="fname" />	unique
Name	<input type="text" name="fname" />	Unique For radio buttons it's not unique
Class Name	<input type="text" class="demo" />	May not create uniqueness Use it with combination
Tagname	Html Tagname of that element Input, a, table, tr, td…etc	For finding elements of same type Find how many links
Link Text	 selenium Inner text of a link	For find links

Partial link text	Partial innertext of a link Ex: Sele, nium, leni	For find links
Css selector	The css path of an element	To find unique element It is faster than xpath
Xpath	The xpath of an element	To unique element It has more navigation features than css

CSS, Xpath : We can use any attribute of an element

CSS Selector:

Absolute Path:

html>body>table>tbody>tr:nth-of-type(1)>td:nth-of-type(2)>input
> : refers first level child of current element

Relative path:
input[id='fname']
Tagname[attribute='value']

Tagname#idvalue
Input#fname

Tagname.classname

Xpath

/ : refers first level child of current element
// : refers any level

html/body/table/tbody/tr[1]/td[2]/input

Relative Xpath: //tagname[@attribute='value']
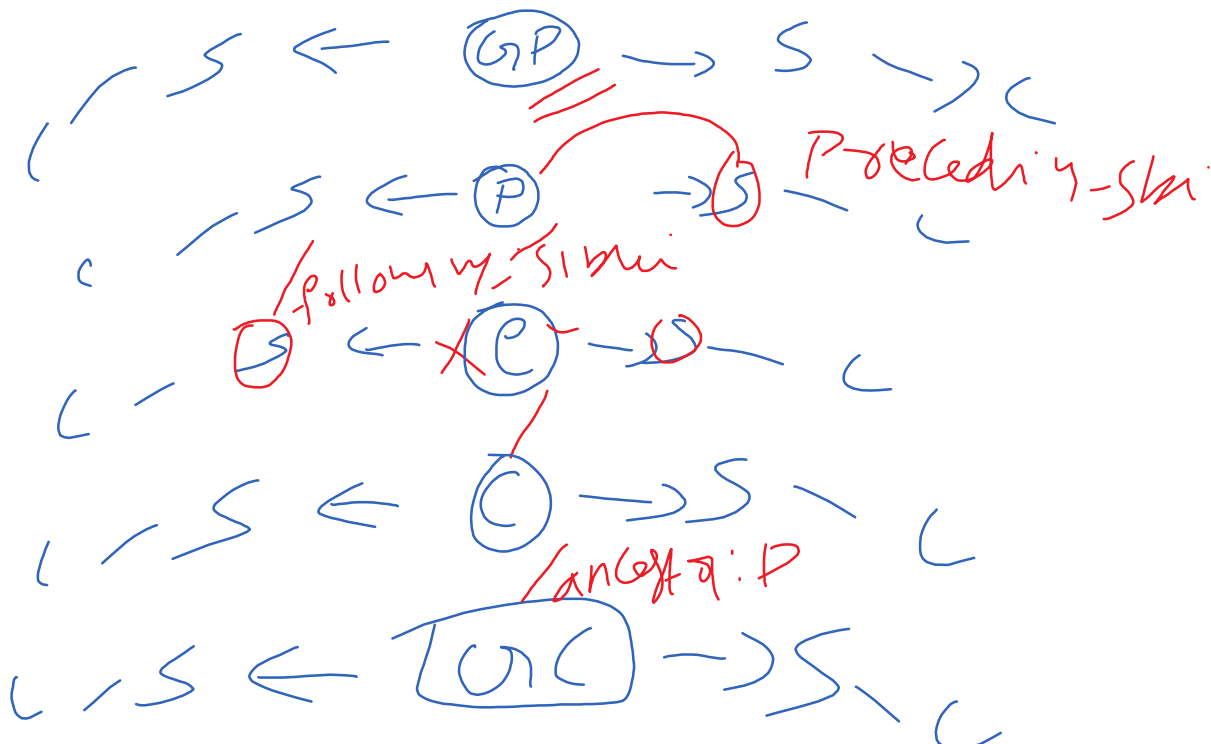
//tagname[function()='value']

//a[text()='value']
//a[text()='selenium']

//tagname[normalize-space()='text']

Try CSS
If not able to create uniqueness
Go for XPATH

1) Parent to Child
   a. / : first level child
   b. // : any level child
2) Child to parent
   a. /..
   b. /ancestor::tagname[attributes]
3) First to last sibling
   a. /following-sibling::tagname[attributes]
4) Last to first
      /preceding-sibling::tagname[attributes]

//td[text()='LastName:']/following-sibling::td/input
//td[text()='LastName:']/../input
//tr[contains(normalize-space(),'LastName:')]//input

1) Write xpath for related element
2) Write navigation from related element to common parent/common related element
3) Write navigation from common parent to actual element

//tagname[contains(@attribute,'value')]
//tagname[contains(function(),'value')]

1) Ways to write relative xpath
   a. Using self attributes
   b. Using parent/grand parent
   c. Using child /descendants
   d. Using siblings
   e. Using index
//a[text()='0016']/../preceding-sibling::td/input
//a[text()='0016']/../..//input
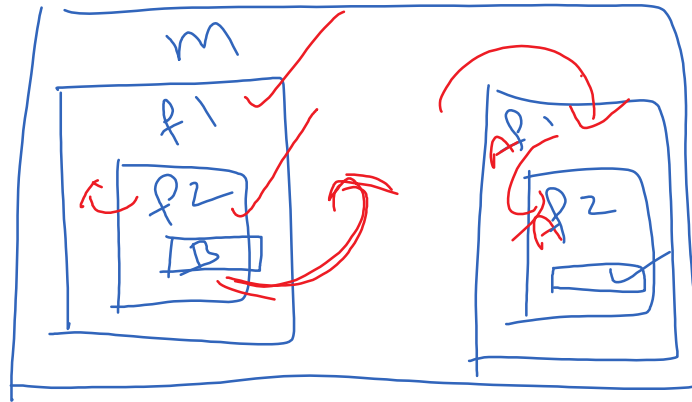//tr[contains(normalize-space(),'0016')]//input

Add Employee
Delete Employee

//img[@class='ui-datepicker-trigger']

//div[contains(@class,'srvceNO') and contains(normalize-space(),'1067')]/../../following-sibling::div[contains(@class,'col5')]/input
//div[@class='srvceNO' and normalize-space()='7143']/../..//input[@value='Select Seats']
//div[@class='srvceNO' and not(ancestor::div[@style='display: none;'])]

Mouse Actions : Actions class

Click	click
Hover	movetoelement
Scroll	movetoelement
Drag and Drop	draganddrop
Double click	doubleclick
Right Click	contextclick

Waiting:
Implicit waiting
     Pageloadtimeout : driver.manage().timeouts().pageLoadTimeout(20,TimeUnit.*SECONDS*);
     Implicitlywait : driver.manage().timeouts().implicitlyWait(10,TimeUnit.*SECONDS*);
Explicit wait
     Webdriverwait

1) Default wait (applicable for entire session)
2) Element Existence
3) Elements Existence
4) Element Attribute (enable, disable, visible…etc)
5) Time (Thread.sleep)

1) Multiple Machines with GRID Setup
2) Docker Setup
3) Cloud

Public : Can be accessed from anywhere
Private : Can be accessed only in where it is declared
Default : Can be accessed only with in package
Protected : Can be accessed with in package and also through extended classes of any package

Exception Handling

1) Nullpointerexception
2) ArrayOutofBoundException
3) Nosuchelementexception
4) Elementnotinteractable

Throwable https://docs.oracle.com/javase/7/docs/api/java/lang/Throwable.html
1) Error
2) Exception
    a. Runtime Exception
        i. WebDriverException
            1) Not found
                a) No such element exceptions
    b. All remaining Exceptions
        i. InterruptedException

Error : Not handle
Exception: Will implement handlers

Error or exception will be handled in Test Automation because we capture results and logs

Exception:
1) Checked Exception
2) Unchecked or runtime exception

If any statement throws a checked exception then
1) we need to create handler for it
2) use throws declaration with that exception name in method declaration

What is a handler?
A handler contains try, catch, finally blocks.
1) Try block contains the main code
2) catch block contains handler code
3) finally contains post try code
4) finally is an optional block, if used that will definitely execute
5) If there is an error in main code then, try> catch> finally blocks will get executed
6) If there is no error in main code then, try> finally blocks will get executed
7) From where we get a failure in try block, from that step catch block will get invoked depends on exception
8) We can create multiple catch blocks by giving variations in Exception class name

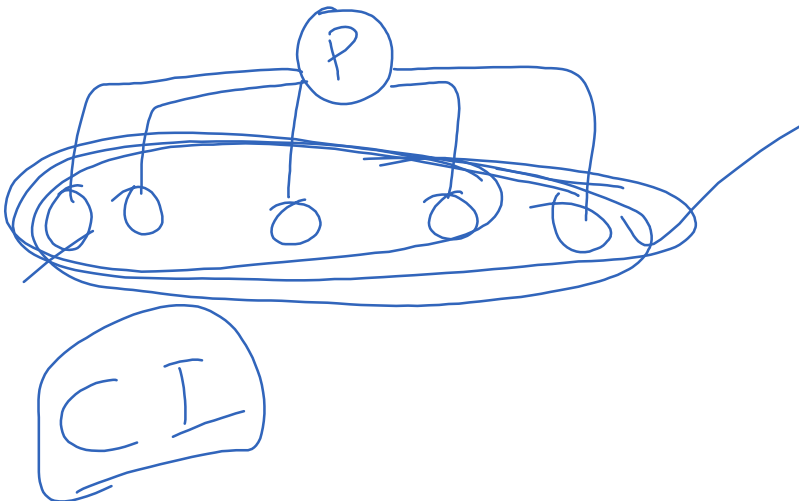How are testers using try catch blocks?
1) To skip failure
2) To provide alternate code
3) To create meaningful logs and results

For Error, Exception, Checked or unchecked we use try catch finally block. For only checked exceptions we require to create try catch while developing the code. For any other exception or error we can create try catch block whenever we want.
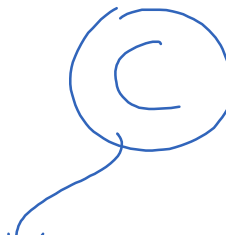
Configurations

1) WebDriver
2) Driver Executables

Build Tools
    ANT
    Maven
    Gradle

Apache Maven:

    Maven Project
        src
        test
        pom.xml : Project Object Model
            Dependency

```
src
test
pom.xml : Project Object Model
        Dependency
                Group
                Artifact
                version
```

Maven Central Repo -> Local Repository

Users/.m2 folder

Company level repository
Settings.xml -> .m2 folder

In POM file configure project to run with specific version of java

Keep below xml in project tag

```xml
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<maven.compiler.target>11</maven.compiler.target>
<maven.compiler.source>11</maven.compiler.source>
</properties>
```

Right click on Project -> Maven -> Reload Project (IntelliJ)
Right click on Project -> Maven -> Update Project (Eclipse)

Now configure project with Selenium:


```xml
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
   <groupId>org.seleniumhq.selenium</groupId>
   <artifactId>selenium-java</artifactId>
   <version>3.141.59</version>
</dependency>
```


File -> Settings-> Editor-> General - > Code Completion -> Uncheck match case -> click ok

Configure Driver Executables:

```xml
<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
   <groupId>io.github.bonigarcia</groupId>
   <artifactId>webdrivermanager</artifactId>
   <version>4.4.0</version>
</dependency>
```


POM: Dependencies

File types:

```
Txt
Csv
json
Excel
DB
```