09/01/2019

- Selenium:

1. Selenium is open source, web application automation tool.

We can see the source code of selenium, we can download and customize it &we can also use it for commercial purpose without paying any license cost.

2. We can download Selenium from following websites:
www.seleniumhq.org

Que: What is automation? What are the advantage &disadvantage of automation? (Assignment)

Que: What is automation tool? List any 4 automation tool. (Assignment)

Que: What are the flavors of selenium? (Assignment)

Ans: 1.Selenium Core
2. Selenium RC
3. Selendroid
4. Selenium Web Driver
5. Appium
6. Winium

Que: What are the languages supported by selenium? (Assignment)

Ans: 1.Java 2.C# 3.Ruby
4. Node JS (java-script) 5.Python
6. Perl 7.Haskell 8.Objective-C
9. JavaScript 10.R
11. Dart 12.Tcl 13.Elixir
14. PHP

Que: What are the operating systems supported by selenium? (Assignment)

Que: What are the Bowers supported by selenium? (Assignment)

Que: What is the latest version of selenium Web Driver? (Assignment)

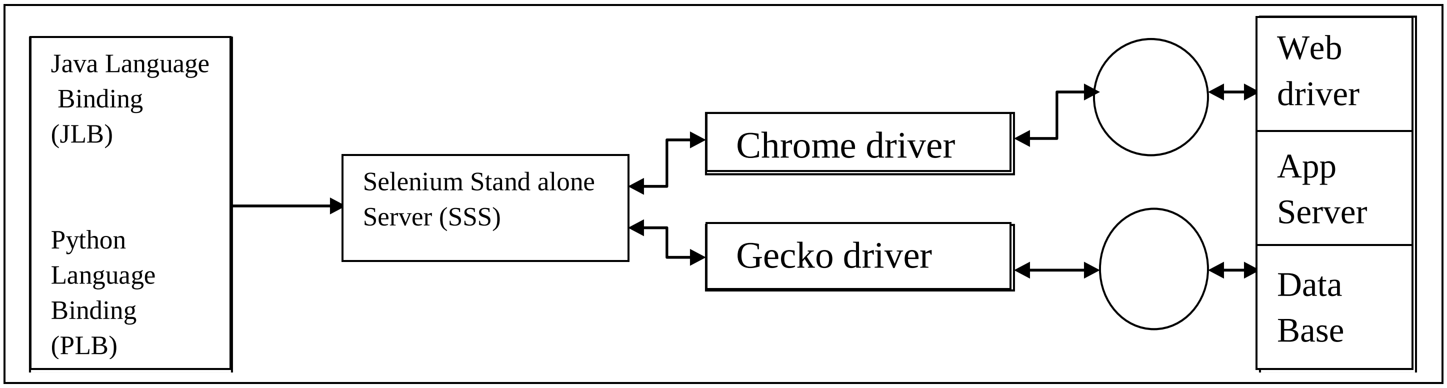
Ans: 3.141.59

- Architecture of Selenium:

1. Selenium supports many coding languages (14).each of them are called as "language Binding" or "Client Binding" which communicate with selenium stand alone server.

2. The server perform action on the browser with the help of "browser specific driver executable".

3. Selenium uses JSON Wire protocol, where JSON stands for "Java Script Object Notation"


Java Language Binding (JLB)
Python Language Binding (PLB)
Selenium Stand alone Server (SSS)
Chrome driver
Gecko driver
Web driver
App Server
Data Base

4.Selenium stand alone server has java language binding build into it, hence while writing selenium script using java we use only *selenium server* and *driver executable.*

Required Software and files:

1. JDK
2. Eclipse
3. Selenium Server
4. Driver Executable
5. Browser
6. Web Application

Step 1: Ensure that JDK is installed properly (java version)

Step 2: Ensure that Eclipse IDE is present.

Step 3: Ensure that browser and web application available.

Step 4: Go to download page of selenium & click on the download link present under selenium stand alone server, it will download jar file.

Step 5: Go to download page of selenium, scroll down till third party driver section and click on latest link which present next to Google chrome driver.Click on chrome driver, then click on the required link (such as chromedriver_win32.zip)

Step 6: Extract zip file.

Step 7: open the eclipse and create new java project (Qsp).

Step 8: In the java project, Create two new folder (driver & jar),copy *chromedriver.exe* file and paste it into driver folder.

Step 9: Copy selenium server jar file and paste it into jar folder.

Step 10: Expand jar folder *right click* on selenium jar file, go to build path and select *'add to build path'*.

Step 11: Except jar file never add any other file to build path, always use *package explorer window*. (Window show you package explorer)

10/01/2019:

Que: When do you get illegal exception? How to handle it?

Ans: 1. if the location of driver executable file is unknown then we get illegal state exception.

2. To handle this exception either we have to add the location to path *environment variable manually* or we have to set it programmatically using *System.setProperty*

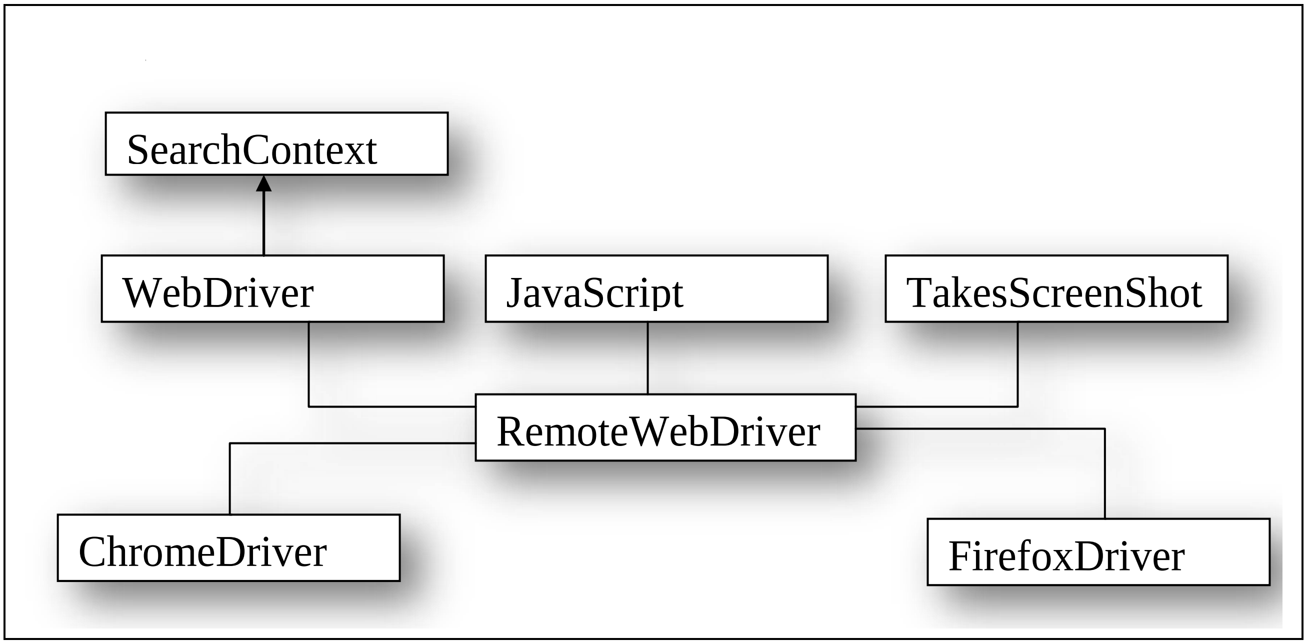
3. *System.setProperty ()* take a two arguments (key, value) where key is defined by a WebDriver, value is a path of a variable.

Note: 1. Generally we used static block to set the path of driver executable.

2.While setting the path we use relative path, Start with (.) which represent *current java project.*

11/01/2019:

- Selenium java client Architecture:



1. In Selenium java client binding, searchContext is the super most interface which extended by WebDriver.
2. Abstract ( ) of these interface is implemented by Remote WebDriver.
3. Remote WebDriver is a class which implements 13 interfaces.
4. All the browser specific classes such as: ChromeDriver, FirefoxDriver,
5. Internet Explorer Driver are extended by Remote WebDriver.

- WebDriver ( ):

Browser	*Class*	*Driver Executable file*	*Key*
Google chrome	Chrome Driver	Chrome Driver.exe	webdriver.chrome .drive
Mozilla Firefox	Firefox Driver	Firefox Driver.exe	webdriver.gecko. driver
Internet Explorer	Internet Explorer Driver	Internet Explorer Driver.exe	webdriver.driver

SearchContext ( ):

- get ( )
- getTitle ( )
- getCurrentUrl ( )
- getPageSource( )
- findElement ( )
- findElements ()
- getWindowHandle( )
- getWindowHandles( )
- switchTo( )
- manage ( )
- navigate( )
- close( )
- quit( )

JavaScriptExecutable ( ):

1. executeScript ( )
2. executeAsyncScript ( )

TakeScreenShot ( ):

1. getScreenShot ( )

Tools Required for Automation:

1. Jdk [10.]
2. Eclipse (Any version)
3. Selenium jar file
4. Driver Executable file
5. Browser(Any)
6. Application under Testing

Que: How do you hide object class method?

Ans: 1.Go to window then select Preferences expand the java language Expand appearance

2. Select the type filters. Click on ad button, (you will find type filter popup), add the package inside the text box, click OK.

3. Click Apply and close.

Que: How to open Source code?

Ans: 1.Go to seleniumhq.org

2.Click on download,(you will find source-code link),click on the source code link, Select java, select client, select src, then click on org/openqa/selenium, then select which browser that you want , click on chrome driver.java

Que: Explain the below step?

Ans: WebDriver driver=new ChromeDriver ();

1. The above example is a example of up casting where WebDriver is an interface
2. Driver is reference variable = assignment operator
3. New = keyword to create object
4. Chrome driver = constructor of object
5.; = Statement of delimit
6. in selenium we use up casting so that we can execute the same script on any browser.
7. We can up cast he browser object to both web driver are remote web driver but as per the selenium coding standard we up cast it to web driver.

16/01/2019:

Exception:

1. Illegal State Exception: java runtime unchecked.
2. Interpreted Exception: java checked.

```
Program 1:
  //open the browser
  ChromeDriver driver=new ChromeDriver();
  //Enter the URL
  driver.get("http://www.google.com");
  //Maximize the browser
  driver.manage().window().maximize();
  // find the active element(where cursor is blinking)
  WebElelment e= driver.switchTo().activeElement();
  Enter the input ‘java’.
  e.sendKeys ("java");
```

Que: How to enter the URL without using get ( )?

Ans: Using navigate ( ). to( ).

Que: What is the difference between get( ) & to( )?

Ans: get( )is used to enter the URL where as to( ) is also enter the URL, because it internally calls get( ).

Que: What is the difference between get( ) & navigate( )?

Ans: get( ) can only enter the URL, where as using navigate( ) we can enter URL & click back, forword & refresh.

```
  Program 2:
  //open the browser
  ChromeDriver driver=new ChromeDriver();
  //Enter the URL
  driver.get("http://www.gmail.com");
  //navigate to google.com
  driver.navigate().to("http://www.google.com");
  driver.navigate().back();
  driver.navigate().forward();
  driver.navigate().refresh();
```

Que: How do you get HTML code of the webpage?

Ans: Using get pageSource( )

```
Program 3:
ChromeDriver driver=new ChromeDriver();
driver.get("http://www.gmail.com");
String s=driver.getPageSource();
System.out.println(s);
```

Que: Write a code to print to window handle of the browser.

Ans: Using getWindowHandle( ) *(which handle only one window and give a address of only one window).*

```
Program 4:
WebDriver driver=new ChromeDriver();
String w = driver.getWindowHandle();
System.out.println(w);
```

Que: Write a script to count the number of browsers open on the desktop.

Ans: Using getWindowHandles( )*(which handle a multiple programs in a window and give a address of the windows which is been opend.)*

```
Program 5:
WebDriver driver=new ChromeDriver();
driver.get("http://www.naukari.com");
Set<String> w=driver.getWindowHandles();
System.out.println(w.size());
```

Que: What are the difference between getWindowHandle ( ) & GetWindowHandles ( )?

Ans:

getWindowHandle()	getWindowHandles()
1.Return Type: "String"	Return Type: Set<String>
2. Returns window handles of current browser.	Returns window handles of all browser.

Que: What is HTML?

Que: What is HTML element?

Que: Example of HTML tag?

Que: Example of Attributes of……

Que: How do you remove value present in the text box?

Ans: By using clear ( ) of WebElement interface.

Que: How do you remove values without using clear()

Ans: By pressing CONTROL "a" DELETE using SendKeys ( )

17/01/2019:

HTML: (Hyper Text Markup Language)

- Selenium is used only to automate UI testing. [Front End testing]
- UI is developed using HTML in web application.
- HTML stands for hyper text markup language.
- Using HTML we can create our own web page for this open the notepad or any other text editor, type the HTML code, save it in the required location with the file extension as. HTML.
- Anything present on the web page is called as element or WebElement
- Element contains tag, Attributes &text.
- In order to view Source code of specific element
  -Right click on the element
  -select the option inspect
- If right click is disable then
  -press *ctrl+shift+I* or press F12 it will display Developer toolbar.

- It Ensure that we are in the final tab – element click on the arrow button- Click on the required element.

Sample Web-page:

```
 <html>
<body>
<a id=”a1” name=”n1” class “c1” href=”http:// www.Qspiders.com”>Qspiders
</a>
</body>
</html>
```

Note: open the notepad, write the above html code. Save it as sample1.html on the desktop.
Double click on the html file present on a desktop it opens the page in the default browser.

- Selenium program to open the sample webpage:

17/01/2019:

```
package auto;
import java.util.Set;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo6
{
	static
	{
	System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
	}
	public static void main(String[] args) throws Exception
	{
		WebDriver driver=new ChromeDriver();
		driver.get("file:////c:/user/bhanu/desktop/sample1.html");
	}
}
```

Note: do not type URL; copy paste it from address bar.

Non-Static Method	Interface	Abstract class	Concrete class
Non-Static Abstract Method	True	True	False
Non-Static Concrete Method	False	True	True

Note: In abstract class we can have static method.

Ex: By.tagname(“a”);

Here By: Abstract class

Tag name: static method

Code : Abstract class by

```
{
static  By tag name (String s)
        {
        By b = new by tag name( );
        Return b;
        }
}
```

How to call:

By b =By.TagName(“a”);

- In selenium before performing any action on the element, we find element using locators.
- In selenium there are 8 types of locators, all of them are static method present in by class [abstract]
- All these static method takes 1 input argument of type string return reference variable or object of type “By”.

Following are the 8 locators present in Selenium:

1. tagName
2. id
3. name
4. className
5. linkText
6. partialLinkText
7. css (Cascade style sheet)Selector
8. xpath

Que: What is the return type of findElement method?

Ans: WebElement (interface).

Que: How many input element taken by find Element? What are those?

Ans: Find method takes only one input argument and the type is By.

Que: if the specified locator is not matching with any of the element present in the page then what FindElement throws?

Ans: FindElement throws NoSuchElement Exception.

Que: if the specified locator is not matching with many elements then what fin method does?

Ans: It returns address of first matching element.

Que: Selenium program to click on the link present on the sample page.

Ans:
```
WebDriver driver=new ChromeDriver();
driver.get("file:///c:/users/Qsp/Desktop/sample1.html");
WebElement e=driver.findElements(By.tagName(“a”));
e.click();
```

18/01/2019:

Out of 8 locators following are the frequently used :

1. id
2. name
3. linktext

- driver .findElement(by.id(“ ”))
- The above statements says in a browser find an element by id.

- driver .findElement (by.Name(“ ”))
- The above statements says in a browser find an element by Name.

- driver .findElement (by.className(“ ”))
- The above statements says in a browser find an element by class name.

- driver .findElement (by.linkText(“ ”))

The above statements says in a browser find an element by link Text.

- linkText & partialLinkText:

1. Both linkText & partialLinkText locator works only for linkTag, (i.e.-<a>)for other type of elements will throws an exception called as NoSuchElementException
2. partialLinkText is used to handle the link which is partially dynamic or when some part of text keep on changing at that moment we go with partialLinkText

- css Selector:

1. css stands for cascade Style sheet.
2. css selects an expression
3. css selector a support for all the attributes
4. Syntax: tag[Attribute name="Attribute value"]

21/01/2019:

Checking css selector in program:

-Inspect the required element it will display developers toolbar.

-press *control+f* it will display search field.

-Type the css selector expression.

Ex: Input [id='username']

-It will highlight the matching element & it's source code.

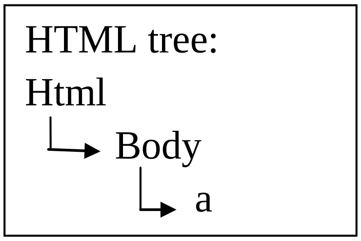
Xpath:

-Path of the element in html tree is called as "xpath".

-While writing xpath, we use (.) at the beginning which represent current webpage.

-using '.' Is not mandatory; while writing xpath we use single forward slash (/) which represent child element.

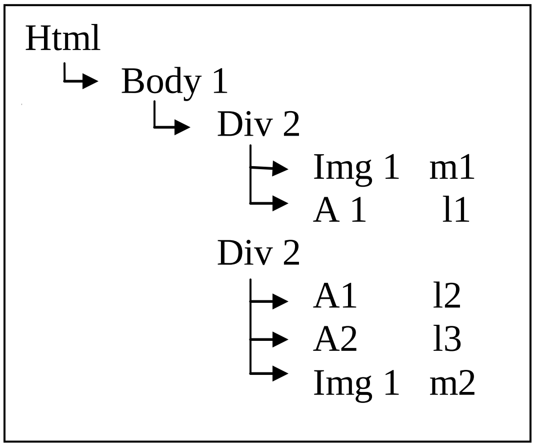
Ex:



-Xpath: ./html/body/a (or) /html/body/a

_-Note: We can used index in path which from 1.

Ex:



m1:/html/body/div [1] /img.

l1:/html/body/div [1] /a.

l2:/html/body/div [2] /a [1].

l3:/html/body/div [2] /a [2].

m2:/html/body/div [2] /img.

-xpath which matching with both m1& m2:

/html/body/div/img.

-xpath which matching with both l1&l2:

/html/body/dive/a [1].

-xpath which matching with both l2 & l3:

/html/body/div[2]/a.

Relative xpath:

-When we write complete path of expression in order to reduce the length of xpath expression, we go for relative path.

- In relative path we us *double forward slash* [//] which represent descendant.[child,grandchild,etc.]

-Ex: //img match with m1&m2.

//a matches with l1 & l2,l3.

-Relative path for m1:

//div[1]/img

-Relative path for l1:

//div[1]/a

-Relative path for l2:

//div[2]/a[1]

-Relative path for l3:

//div[2]/a[2]

-Relative path for m2:

//div[2]/img

Que: Write a xpath which matches with all the link which is present on the page

Ans: //img/a

We use pipe (|) to join the expression

//a: It matches with all the links.

//a [1]: it matches with all the links where index [1].

//table//a

22/01/2019:

Username: //input [@id=‘username’]
Password: //input [@name=‘pwd’]
Login Button: div[‘login’]
Checkbox: //input [@name=‘remember’]
Forgot password link: //a [@id=‘toPasswordRecoveryPageLink’]
    //a [text ( ) = ‘Forgot your password?’]
Acti-time Link: //a [text ( ) = ‘ACTITIME INC.’]
Version: nobr [contains (text( ),‘actitime’)]
Header: //td[@id=‘headercontainer’]
    //td[text( )=‘Please identify yourself’]
Actitime image: //img[contains(@src,‘logo’)]
Clock image: //img[contains(@src,‘timer’)]

Que: What is the difference between ‘/’ & ‘//’?
Ans: / represents child
    // represents descendant.

Using Attributes & text in xpath:

-While writing xpath expression we can include attribute & text of the element.

-The syntax for attribute is:

```
//tag[text( )=‘TextValue’]
```

-Ex: //input[@name=‘pwd’]

-Ex: //div[text( )=‘Login’]

-In the same xpath expression we can include multiple attributes & text.

```
//tag[@AttributeName1=‘AttributeValue1’
 and @AttributeName2=‘AttributeValue2’]

//tag[@AttributeName2=‘AttributeValue2’
 or @AttributeName2=‘AttributeValue2’]
```

Xpath	Elements
//input	A,B……C,
//input[@type=‘text’]	A1 B1
//input[@value=‘A’]	A1 A2
//input[@type=‘text’ and @value=‘A’]	A1
//input[@type=‘button’ and @value=‘A’]	A2
//input[@type=‘text’ and @value=‘B’]	B1
//input[@type=‘button’ and @value=‘B’]	B2
//input[5]	C1
//input[@type=‘checkbox’][1]	C1
//input[@checked]	C1
//input[@type=“checkbox” and @checked]	C1
//input[6]	C2
//input[@type=“checkbox”][2]	C2
//input[@type=‘checkbox’ and not (@checked)]	C2
//input[not(@type=‘checkbox’)]	A1B1A2B2
//input[@value=‘A’ or @value=‘B’]	A1B1A2B2
//input[@type=‘text’ or @value=‘A’]	A1B1A2B2

Que: What is the syntax for contains function?

Ans: Syntax 1: //tag [contains (@AN, ‘AV’)]

Ex: //img [contains (@src, ‘timer’)]

Syntax 2: //tag [contains (text ( ), ‘tv’)]

Ex: //nobr [contains (text ( ), ‘actiTIME’)]

Que: How to handle link without using partialLinkText?

Ans: using contains functions of xpath

Ex: //a [contains(text( ), ‘Inbox’)]

23/01/2019:

Que: How to handle partial dynamic element in xpath?
Ans: using contains function.
Que: What is the syntax for contains function?
Ans: //tag[contains@]

Xpath traversing:

1. Navigating from one element to another element is called as “traversing”, there are majorly two types:

A) Forward traversing
B) Backward traversing

Handling Completely Dynamic Element:

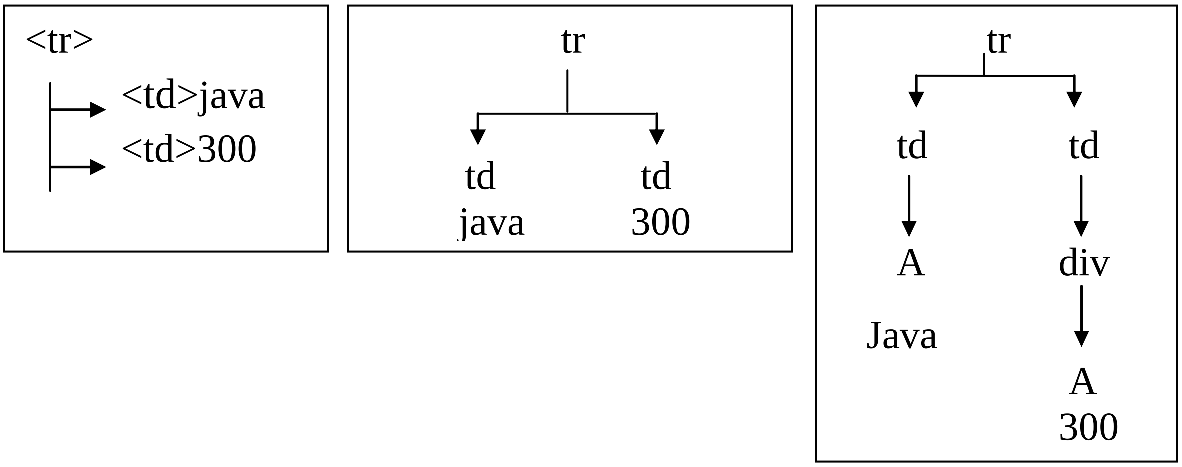
-In order to identify completely dynamic element we used a concept called “Independent & Dependant xpath”.

-Ex:

JAVA	300
UNIX	200

-In the above web table, the cost may change dynamically or it may be duplicate, & cost depends on subject, here cost is called as “*Dependant Element*”,subject is called as “*Independent Element*”.

-Ex:



Steps to construct HTML trick:

1. Inspect independent element and note down its source code.
2. Place the mouse pointer on the source code of independent element & move in upward direction till it highlights both independent & dependant element.

Note: The place at which both elements are highlighted will be the common parent, add the path to the common parent the above source code of independent element.

3. Use down arrow key to navigate from common parent to dependant element and add its path below the common parent.

Que: Write xpath to identify “client version” of “java” present in selenium download page?

Ans: *//td[text()='Java']/../td[2]*

Que: Write xpath to identify “Release date ” of “Ruby” present in selenium download page?

Ans: *//td[text()='Ruby']/../td[3]*

Que: Write xpath to identify “Download Link ” of “Python” present in selenium download page?

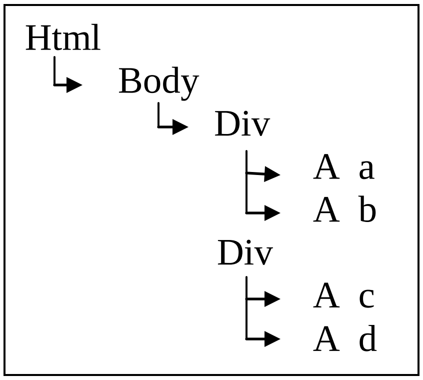
Ans: *//td[text()='Python']/../td[4]*

Que: Write a xpath to identify latest link of “Google Chrome Driver” present in Selenium download page

Ans: *//a[text()='Google Chrome Driver']/../..td[2]/a*

Xpath by group index:

1. While writing xpath expression when we used the index it may match with multiple elements in such cases we can used group index where we write the xpath *inside the braces* & we mention the index *outside the braces* as shown below.



A
B
C
D

2. When we used braces first it executes xpath present inside the braces & it will store the matching element in the xpath array then it will identify the element present in the xpath array based on the index which is specified outside the braces.

Ex: (//a[2])[1])

B
D

24/01/2019:

Que: In which scenario does all the following xpath matches with same image? //img, //img[1], (//img)[last ( )]

Ans: Only when the webpage contains only one image

Que: Write an xpath which matches with first and last image present on the webpage.

Ans: (//img)[1] | (//img)[last ( )]

Que: Write a xpath which matches with first and last link present on the webpage.

Ans: (//a)[1] | (//a)[last ( )]

Que: Write a xpath which matches with *first three links* present on the webpage.

Ans: (//a)[position ( ) <=3]

Que: Write a xpath which matches with *last three links* present on the webpage.

Ans: (//a)[position ( ) > last ( )-3]

Que: What is a procedure followed in your project to follow the elements?

Ans: 1.In our project out of 8 locators out of these we use 4 of them very frequently such as id, name, LinkText & xpath.

2. In xpath, first preference is given for attributes then text then contains ( ) then Independent-dependant xpath and then GroupIndex.

Handling the multiple elements:

In selenium to perform actions on multiple elements we use findElements ( ) Which returns list<webelements> & list must be important from java.util package.

```
Ex:   <html>
      <body>
      <a href="http://www.qspidrs.com">Qspiders</a>
      <a href="http://www.jspidrs.com">Jspiders</a>
      </body>
      </html>
```

```
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class sample4
{
	static
	{
	System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
	}
	public static void main(String[] args)
	{
	WebDriver driver=new ChromeDriver();
	driver.get("file:///C:/Users/MAHESH/Desktop/sample4.html");
	List<WebElement> allLinks =
	driver.findElements(By.tagName("a"));
	int count=allLinks.size();
	System.out.println(count);
	driver.close();
	}
}
```

Que: Write a script to count the number of the number of links, present in the given web page.

Ans: In most of the cases along with findElements we use xpath.

- IndexOutOfBounds Exception:

```
//24/01/2019
//write a script to count a number of link present on a page and
also print text of all.
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo8
{
    static
    {
    System.setProperty("webdriver.chrome.driver","./driver/chromedr
iver.exe");
    }
    public static void main(String[] args)
    {
    WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/MAHESH/Desktop/sample4.html");
    List<WebElement> allLinks = driver.findElements(By.xpath("a"));
    int count=allLinks.size();
    System.out.println(count);
    for(int i=0;i<count;i++)
    {
        WebElement link=allLinks.get(i);
        String text=link.getText();
        System.out.println(text);
        }
        driver.close();
    }
}
```

Que: What are the difference between findElement & findElements?

Ans:

findElement	findElements
1.Return type:WebElement	1.Return type:List<WebElement>
2.It Return a first matching element	2.It Returns all the matching elements
3.If the locators is not matching with any of the elements it will throw *NoSuchElementException*.	3.It will return empty list

Que: Write a script to count the number of images present on amazon.com

Ans:

25/01/2019:

Que: Automate following scenarios:

-Open the Browser.

-go to Google.com.

-Type SQL in search box, it will display auto suggestions.

-Find all the auto suggestions elements.

-Print the count.

-Print the text of all the suggestions.

-Select the first auto suggestion.

```
//25/01/2019
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo10{
     static{
     System.setProperty("webdriver.chrome.driver","./driver/chrome
     driver.exe");}
     public static void main(String[] args) throws
     InterruptedException{
          //open the Browser
          WebDriver driver=new ChromeDriver();
          //get the URl
          driver.get("http://google.com");
          //type "sql" in search box
         driver.switchTo().activeElement().sendKeys("sql");
        //wit fir 2 sec.
        Thread.sleep(2000);
        //it will give you multiple suggestions
        String xp="//span[contains(text(),'sql')]";
        //ASE=auto suggestion elements
     List<WebElement> allASE = driver.findElements(By.xpath(xp));
        //it will count the suggestion elements
          int count=allASE.size();
          //it will print the count
          System.out.println(count);
          for(int i=0;i<count;i++) {
               //to get a first suggestion
               WebElement ase = allASE.get(i);
               // assign the name text to first suggestion
               String text=ase.getText();
               //print the text
               System.out.println(text);}
          //get the first suggestion
          allASE.get(0).click();
          //close the browser
          driver.close();}}
```

Que: Automate the following:

- -Open the browser.
- -Go to cleartrip.com.
- -Type ban in ‘From’ field, it will display auto suggestion
- -Count the no. of auto suggestion displayed.
- -Print the text of all the auto suggestion.
- -Select the first auto suggestion.

```
//25/01/2019 //Automate Cleartrip.com
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class Cleartrip
{static{
     System.setProperty("webdriver.chrome.driver","./driver/chromed
river.exe");}
public static void main(String[] args) throws InterruptedException
{
         //open the Browser
         WebDriver driver=new ChromeDriver();
         //get the URl
         driver.get("http://cleartrip.com");
         driver.manage().window().maximize();
         driver.findElement(By.name("origin")).sendKeys("ban");
         Thread.sleep(2000);
     List<WebElement>text=driver.findElements(By.xpath("//a[contain
s(text(),'ban')]"));
         int count=text.size();
         System.out.println(count);
         for(WebElement web:text)
         {
              String s=web.getText();
              System.out.println(s);
         }
         text.get(2).click();
     }
}
```

Handling a list box:

-In Selenium in order to perform action on the list box, we use Select class of selenium.

-Select class should be imported from following package:

Org.openqa.selenium.support.ui.select;

-Select class has parameterized constructor, it takes single argument of type WebElement must be List box. (Tag of the element should be selected or else we get UnexpectedTagNameException)

-Select class has following methods:

A. selectByIndex( )
B. selectByValue( )
C. selectByVisibleText( )
D. deselectByIndex( )
E. deselectByValue( )
F. deselectByVisibleText( )
G. deselectAll( )
H. isMultiple( )
I. getAllSelectedOptions( )
J. getFirstSelectedOptions( )
K. getOptions( )
L. getWrappedElement( )

- Note: We cannot use deselect ( ) in single select list box, if we try to use it then we get UnsupportedOperationException.
- The value and text is *case sensitive.*
- If the specifies value or text is invalid, we get *NoSuchElementException*

```
//25/01/2019
package auto;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo11
{
    static
    {
    System.setProperty("webdriver.chrome.driver","./driver/ch
romedriver.exe");
    }
    public static void main(String[] args)
    {
        WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20no
tes/html%20Programs/hotel.html");
        WebElement listBox=driver.findElement(By.id("mtr"));
        WebElement mtrLB = driver.findElement(By.id("mtr"));
        Select select=new Select(mtrLB);
        select.selectByIndex(0);
        select.selectByValue("v");
        select.deselectByVisibleText("Dosa");
        select.deselectByIndex(0);
        select.deselectByValue("v");
        select.deselectByVisibleText("Dosa");
        WebElement slvLB=driver.findElement(By.id("slv"));
        Select slv=new Select(mtrLB);
        System.out.println(slv.isMultiple());
        slv.deselectAll();
        WebElement ccdLB=driver.findElement(By.id("ccd"));
        Select ccd=new Select(ccdLB);
        System.out.println(ccd.isMultiple());
        ccd.selectByValue("t");
        ccd.deselectByValue("t");
    }
}
```

28/01/2019:

Que: Write a script to count the number of options selected in the list box and print all the options which are selected in the list box:

```
Ans:
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo12
{
    static
    {
    System.setProperty("webdriver.chrome.driver","./driver/chromed
river.exe");
    }
    public static void main(String[] args) throws
InterruptedException
    {
    WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20notes/h
tml%20Programs/hotel.html");
        WebElement slvLB = driver.findElement(By.id("slv"));
        Select slv=new Select(slvLB);
        List<WebElement>allSOs=slv.getAllSelectedOptions();
        int count=allSOs.size();
        System.out.println(count);
        for(int i=0;i<count;i++)
        {
            WebElement option=allSOs.get(i);
            String text=option.getText();
            System.out.println(text);
        }
        driver.close();
    }
}
```

```java
//28/01/2019
package auto;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo13
{
    static
    {
System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException
    {
    WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20notes/html%20Programs/hotel.html");
        WebElement slvLB = driver.findElement(By.id("slv"));
        Select slv=new Select(slvLB);

        WebElement option=slv.getFirstSelectedOption();
        String text=option.getText();
        System.out.println(text);

            List<WebElement> allOptions=slv.getOptions();
            System.out.println(allOptions.size());

            for(WebElement v:allOptions)
            {
                String t=v.getText();
                System.out.println(t);
            }
        //print listbox options without using loop
            WebElement w=slv.getWrappedElement();
            System.out.println(w.getText());
            driver.close();
        }
    }
```

Que: Write a script to print the content of a list box in sorted order

```
Ans: //28/01/2019
package auto;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo14
{ static{
     System.setProperty("webdriver.chrome.driver","./driver/chrom
edriver.exe");
     }
     public static void main(String[] args) throws
InterruptedException
     {WebDriver driver=new ChromeDriver();
     driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20notes
/html%20Programs/hotel.html");
          WebElement mtrLB = driver.findElement(By.id("mtr"));
          Select select=new Select(mtrLB);
          List<WebElement> allOptions=select.getOptions();
          List<String> allText=new ArrayList<String>();
          for(WebElement option:allOptions){
                    String text=option.getText();
                    System.out.println(text); }
          Collections.sort(allText);
          for(String Text:allText)
               {
               System.out.println(Text);
               }
               driver.close();}}
```

Que: Write a script to print content of list box in sorted order without any duplicate options (without any repetations).

```
Ans://28/01/2019
package auto;
import java.util.List;
import java.util.Set;
import java.util.TreeSet;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo15
{
    static
    {
System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
    }
public static void main(String[] args) throws InterruptedException
    {
        WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20notes/html%20Programs/hotel.html");
        WebElement mtrLB = driver.findElement(By.id("mtr"));
        Select select=new Select(mtrLB);
        List<WebElement> allOptions=select.getOptions();
        Set<String> allText=new TreeSet<String>();
        for(WebElement option:allOptions){
                String text=option.getText();
                allText.add(text);}
            for(String text:allText)
            {
            System.out.println(text);
            }
            driver.close();
    }
}
```

29/01/2019:

Synchronization:

- Process of matching Selenium speed with application is known as "Synchronization".
- When we use driver.get, it enters the URL & it waits till the page is completely loaded.
- By default the waiting time for page load is unlimited.
- We can change the waiting time of the page load as shown below:
    1. DAYS
    2. HOURS
    3. MINUTES
    4. SECONDS
    5. MILLISECONDS
    6. MICROSECONDS
    7. NANOSECONDS
- Even after the specified duration if the page is not loaded we get TimeOutException.

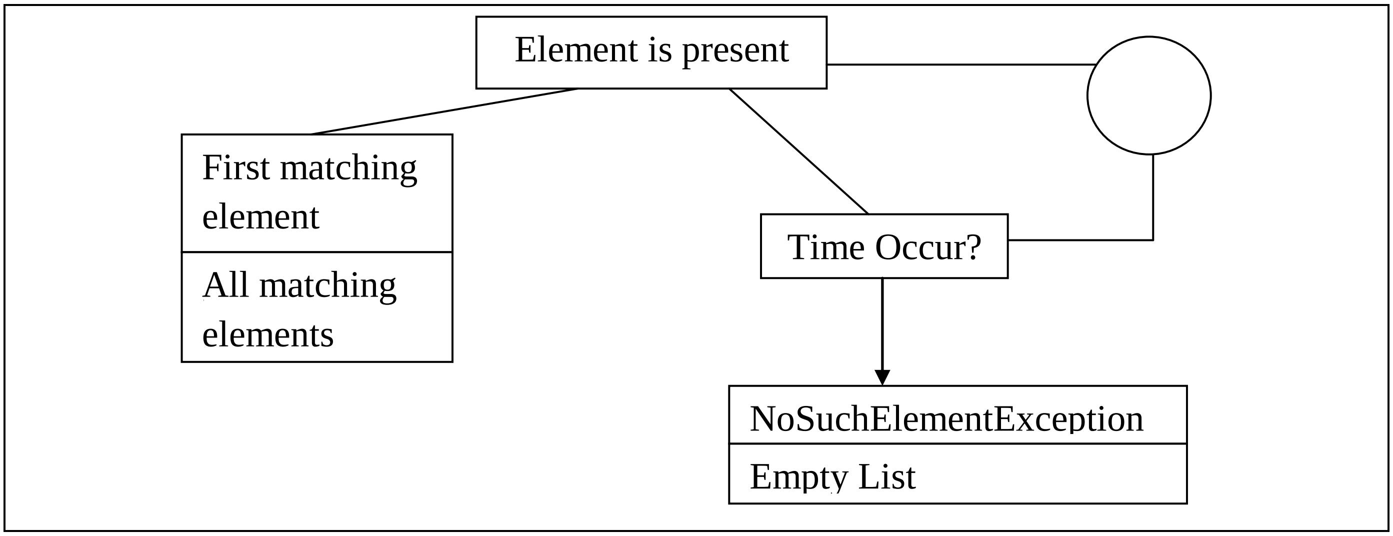
Que: Write a script to check whether the page is loaded or not within 3 sec.

```
Ans://29/01/2019
//Que:Write a script to check whether the page is loaded or not
within 3 sec.
package auto;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo17
{
     static
     {
     System.setProperty("webdriver.chrome.driver","./driver/chromedr
iver.exe");
     }
     public static void main(String[] args)
     {
          WebDriver driver=new ChromeDriver();
   driver.manage().timeouts().pageLoadTimeout(3, TimeUnit.SECONDS);
          try {
          driver.get("https://www.google.com");
          System.out.println("Page is loaded within 3 seconds");
          }
          catch(Exception s)
          {
          System.out.println("Page is not loaded within 3 seconds");
          }
     }
}
```

Implicit Wait:

In most of the cases selenium will faster than the application because of this we get NoSuchElementException even though specified locator is valid, in order to handle this we used implicit wait.

```
//29/01/2019
package auto;
import java.util.List;
import java.util.Set;
import java.util.TreeSet;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class demo15{
     static{
     System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
     }
public static void main(String[] args) throws InterruptedException{
          WebDriver driver=new ChromeDriver();
     driver.get("file:///C:/Users/MAHESH/Desktop/selenium%20notes/html%20Programs/hotel.html");
          WebElement mtrLB = driver.findElement(By.id("mtr"));
          Select select=new Select(mtrLB);
          List<WebElement> allOptions=select.getOptions();
          Set<String> allText=new TreeSet<String>();
          for(WebElement option:allOptions){
                    String text=option.getText();
                    allText.add(text);}
               for(String text:allText)
               {
               System.out.println(text);
               }
               driver.close();
     }
}
```



- When a control comes to findElement or findElements statement, it will check whether the specified element is present or not.
- If the present findElement ( ) returns first matching element where as findElements ( ) returns all the matching elements.
- If the element is not present then it will check for the time if time is over findElement ( ) will throw NoSuchElementException where as findElements ( ) returns empty list.
- If time is not over it will wait for 500ms & it will continue to search the element.

Explicit wait:

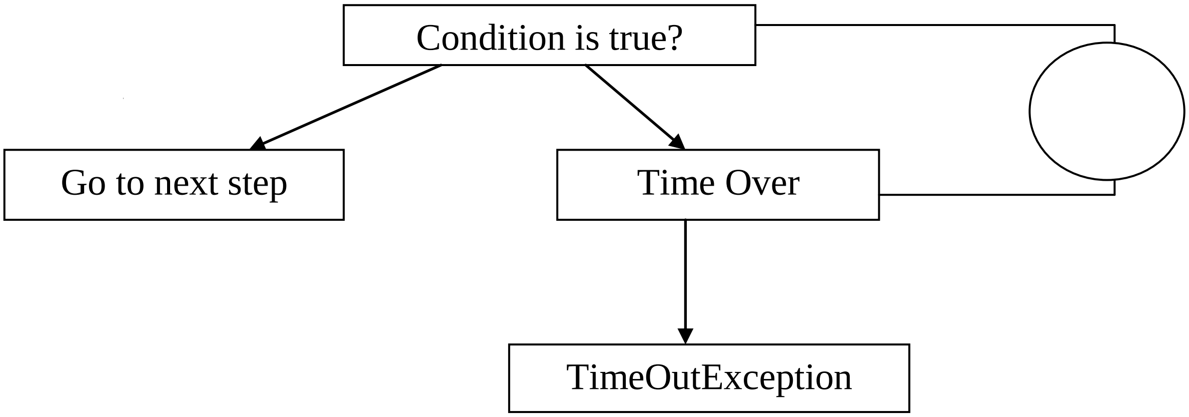
In order to handle synchronization of any method including findElement ( ) & findElements ( ) we use Explicit wait.

Generally use explicit wait to handle synchronization of any method except findElement & findElements.

In explicit wait first we create the object of WebDriver wait & the new specify the waiting condition explicitly using *wait.until.statement*, hence the name explicit wait.

```
package auto;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class demo16{
    static{
System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
    }
    public static void main(String[] args)
    {
        WebDriver driver=new ChromeDriver();
driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
        driver.get("https://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.name("pwd")).sendKeys("manager");
    driver.findElement(By.xpath("div[text()='Login']")).click();
        WebDriverWait wait=new WebDriverWait(driver,10);
        wait.until(ExpectedConditions.titleContains("Enter"));
        System.out.println(driver.getTitle());
        driver.findElement(By.id("logoutlink")).click();
        wait.until(ExpectedConditions.titleContains("login"));
        System.out.println(driver.getTitle());
    }
}
```

Predicate:



- When the control comes to wait until statement it will check the condition (predicate) if the condition is true, it will go to the next step is false it will check the time.
- If the time is over then it will throw TimeOutException or else it will wait for 500 ms [polling period] & it will continue to check the condition.

Que: What are the difference between implicit wait & explicit wait?

Ans:

implicit wait	Explicit wait
1. The waiting condition is builin.	1. We should specify waiting condition is separately.
2. We can handle synchronization of findElement & findElements.	2. We can handle synchronization of any method but only one at a time.
3. After the time we get NoSuchElementException or Empty list.	3. We get TimeOutException.
4. TimeUnit can be Days, Hours, Minutes, Seconds, Milliseconds	4. TimeUnit can only be Seconds.

30/01/2019:

Que: Write a script to login & logout for *actitime* where we get NoSuchElementException for logout link handle this without using implicit wait, Explicit wait or Thread.sleep( ).

Ans: We handle it using custom wait.

```java
//29/01/2019
package auto;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo18 {
     static{
System.setProperty("webdriver.chrome.driver","./driver/chromedriver
.exe");
     }
     public static void main(String[] args)
     {
          //Open the browser
          WebDriver driver=new ChromeDriver();
     driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
          //get the URL
          driver.get("https://demo.actitime.com");
          //Enter user name
          driver.findElement(By.id("username")).sendKeys("admin");
          //Enter password
          driver.findElement(By.name("pwd")).sendKeys("manager");
          //click on login
driver.findElement(By.xpath("//input[@type='submit']")).click();
     //driver.findElement(By.xpath("div[text()='Login']")).click();
          while(true){
               try{
          driver.findElement(By.id("logoutlink")).click();
          System.out.println("logout link is found & checked");
          break;}
               catch(Exception e){
System.out.println("logout link is not found will try again");}
          System.out.println("End");
          }
     } }
```

- Handling JavaScript popup:
  - Characteristics of the popup:
  1. We cannot inspect this popup.
  2. We cannot move this popup.
  3. Popup will be displayed just below the address bar & in the center section of the browser.
  4. Popup contains ok button (alert popup) or ok & cancel button (confirmation popup)

Solution:

To handle this JavaScript popup we used SwitchToAlert ( ) & after switching we used getText ( ) to get the text present in the popup we used accept to click ok & we used dismiss to click cancel.
If the popup is alert, then there is no difference between accept & dismiss.

Why performing any of the actions?

-If the popup is not present, then we get NoAlertPresentException

Que: automate the www.irctc.com

Click on login button.

Click ok button present on popup.

```
//30/01/2019
//Automate  irctc.com
package auto;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class irctc
{
	static
	{
	System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe");
	}
	public static void main(String[] args)
	{
		//open the Browser
		WebDriver driver=new ChromeDriver();
		//go to irctc
	driver.get("http://www.irctc.co.in/eticketing/pageUnderConstruction.jsf");
		//click login button
		driver.findElement(By.id("login button")).click();
		//wait it JS popup is displayed
		WebDriverWait wait=new WebDriverWait(driver,5);
		wait.until(ExpectedConditions.alertIsPresent());
		//click ok button present on the popup
		Alert alert= driver.switchTo().alert();
		//get msg & print
		System.out.println(alert.getText());
		alert.accept(); //click ok
		//alert.dismiss();//click cancel
	}
}
```

- Handling Hidden Division popup:

Characteristics:

1. We can inspect this popup.
2. We cannot move this popup.

Solution: We can handle this using findElement ( ) itself. Calendar popup are the best example of hidden division popup.

```
//automate flipkart.com
//print bhanu in the text field
//click on close once after printing bhanu
//30/01/2019
package auto;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class flipkart
{
	static{
	System.setProperty("webdriver.chrome.driver","./driver/chromedr
iver.exe");
	}
	public static void main(String[] args){
		//open the Browser
		WebDriver driver=new ChromeDriver();
		driver.manage().deleteAllCookies();
		//go to flipkart
		driver.get("http://www.flipkart.com");
		//click login button
		String xp="(//input[@type='text'])[2]";
		driver.findElement(By.xpath(xp)).sendKeys("bhanu");
		String xp2="(//button[not@type])";
		driver.findElement(By.xpath(xp2)).click();
		button[not (@type)]
	}
}
```

Que: Automate the following: Open the browser
Go to renew page of Religarehealthinsurance
Enter policy number
Select the date of birth
Specify the contact number
Click on lets renew

30/01/2019:

```
package auto;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class religare{
     static{
System.setProperty("webdriver.chrome.driver","./driver/chromedriver.exe
");}
public static void main(String[] args) throws InterruptedException
     {
          WebDriver driver=new ChromeDriver();
     driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
Stringurl="https://www.religarehealthinsurance.com/rhhicl/
proposalcp/renew/index-care";
          driver.get(url);
          driver.findElement(By.id("policynumber")).sendKeys("P123");
          driver.findElement(By.id("dob")).click();
          Thread.sleep(2000);
          String xp1="//select[@data-handler=''select Year']";
          WebElement yLB=driver.findElement(By.xpath(xp1));
          Select year=new Select(yLB);
          year.selectByVisibleText("1996");
          Thread.sleep(2000);
          String xp2="//select[@data-handler=''select Month']";
          WebElement mLB=driver.findElement(By.xpath(xp2));
          Select Month=new Select(mLB);
          Thread.sleep(2000);
          driver.findElement(By.linkText("1")).click();
driver.findElement(By.id("alternative_number")).sendKeys("9422556618");
          //click on "lets renew" button
          driver.findElement(By.id("renew_policy_submit")).click();
          }}
```

31/01/2019:

Que: what is the difference between text & text ().

text	text( )
1. text is the value of one of the attribute.	1. text( ) is xpath function
2.Ex: //input[type=text]	2. Ex: div[text( )= ‘login’]

Que: Write a script to select today’s date in the calendar.

Ans:
```
Date d= new Date ( );
        String s= d.toString ();
        System.out.println ();
```

OR

```
LocalDate v=LocalDate.now ( );
System.out.println (v.toString ( ));
System.out.println (v.plusdays (2));
System.out.println (v.plusdays (-32));
```

OR

```
Date d= new Date ( );
System.out.println (d);
SimpleDateFormat s= new SimpleDateFormat (“yyyy-mmm-dd”);
System.out.println (d.format (d));
```

Handling child browser popup:

-Characteristics:

1. We can inspect this popup.
2. We can move this popup.
3. The popup will have minimize & maximize button.

-Solution:

We handle this using *getWindowHandles ( )* & *switchTo ( )*.

Ex: Write a script to count the number of browsers.

Write a script to print title of all the browsers.

Write a script to close all the browsers without using quit.

```
//31/01/2019
package auto;
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo19
{
    static
    {
System.setProperty("webdriver.chrome.driver","./driver/chromedriver
.exe");
    }
public static void main(String[] args) throws InterruptedException
    {
        //Open the browser
        WebDriver driver=new ChromeDriver();
        //get the URL
        driver.get("https://www.naukari.com/");
        //handle all the browsers
        Set<String> allWHS = driver.getWindowHandles();
        // count size of all the browser
        int count = allWHS.size();
        //print the count
        System.out.println(count);

        for (String wh:allWHS)
        {
            driver.switchTo().window(wh);
            System.out.println(driver.getTitle());
            Thread.sleep(1000);
            driver.close();
        }
```

Que: Write a script to close only parent browser.
Ans: driver.close ( ).
Que: Write a script to close all child browsers.
Ans: driver.quit( ).
Que: Write a script to close specified browser.
Ans:
Que: Write a script to close all browsers in reverse order.
Ans:
Que: What is difference between close & quit.
Ans:

close( )	quit( )
close ( ) is used to close the current browser .	quit( ) is used to close all browsers.

Que: Automate following scenarios:

1. Open the browser.
2. Go to login page of actiTIME.
3. Click on view license, it will open license agreement page in new window
4. Find all the sections
5. Print the number of sections present
6. Print level or heading of all the sections.
7. Close the agreement window.
8. Close the login window.

```
//31/01/2019
package auto;
import java.util.List;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo20{
    static{
    System.setProperty("webdriver.chrome.driver","./driver/chromedr
iver.exe");
    }
public static void main(String[] args) throws InterruptedException{
        WebDriver driver=new ChromeDriver();
        String parent =driver.getWindowHandle();
    driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
        driver.get("http://localhost/login.do");
        driver.findElement(By.id("licenseLink")).click();
        Set<String> allWHS = driver.getWindowHandles();
        for (String wh:allWHS)
        {
            driver.switchTo().window(wh);
        }
        List<WebElement> allH =
driver.findElements(By.xpath("//h2"));
        int count = allH.size();
        System.out.println(count);

        for(WebElement h:allH)
        {
            System.out.println(h.getText());
        }
        driver.close();
        Thread.sleep(1000);
        driver.switchTo().window(parent);
        driver.close();
    }
}
```

Que: Automate following scenarios:

1. Open the browser
2. Go to actitime login page.
3. Click on the link ( ), it will open the new tab.
4. Count the number of tabs and print the title of tabs.
5. Click on the button “got it” which is present in second tab.
6. Close the second tab.
7. Close a first tab.

```
//31/01/2019
package auto;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
//import org.openqa.selenium.support.ui.ExpectedConditions;
//import org.openqa.selenium.support.ui.WebDriverWait;
public class demo21{
     static{
     System.setProperty("webdriver.chrome.driver","./driver/chromedr
iver.exe");
     }
public static void main(String[] args) throws InterruptedException {
          WebDriver driver=new ChromeDriver();
          String parent =driver.getWindowHandle();
     driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
          driver.get("http://localhost/login.do");
          driver.findElement(By.linkText("Actimind Inc.")).click();
          Thread.sleep(2000);
          Set<String> allWHS = driver.getWindowHandles();
          int count=allWHS.size();
          System.out.println(count);
          for (String wh:allWHS){
               driver.switchTo().window(wh);
               System.out.println(driver.getTitle());
          }
          //WebDriverWait wait=new WebDriverWait(driver,10);

     //wait.until(ExpectedConditions.visibilityOfElementLocated(By.i
d("cookie-button-got-it")));
          //driver.findElement(By.id("cookie-button--got-
it")).click();
          driver.close();
          Thread.sleep(2000);
          driver.switchTo().window(parent);
          driver.close();
     }
}
```