

Selenium

Created by Jason Huggins in 2004.

classmate

Date 11th June, 2016
Page

Why selenium?

1. → opensource (Free) (BTP)
2. → supports multiple OS. (3) Paid Windows, Linux, Macintosh
3. → supports multiple Browsers only IE
4. → supports multiple Languages (6) only VBScript

object oriented lang. language
Framework

Java
C#
Python
Perl
Ruby

Csharp

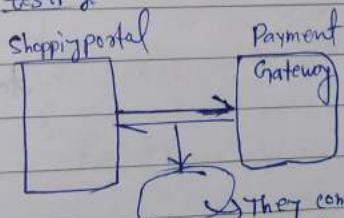
Used only for Web Application

(not used for window based app
or client server app)

opensource
Eclipse
Appium
Java
Selenium

No need to pay for these tools.

SOAP UI testing → To test the interfaces (BTP)



They communicate as per some protocols, called web services.

driver.findElement(By.tagName("iframe"));

→ used instead of iframe generally.

of Locators

driver.findElement(By.id("..."))

driver.findElement(By.name("..."))

driver.findElement(By.className("..."))

driver.findElement(By.partialLinkText("..."))

driver.findElement(By.tagName("input"))

driver.findElement(By.xpath("//input[@type='text']"))

driver.findElement(By.cssSelector("input[type='text']"))

Absolute xpath
Relative xpath
and, or, contains, startsWith
text() = 1 Axes

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
<p><u>W3C methods</u></p> <p>d). get() getTitle() getPageSource() getCompleteUrl() navigate().to() • back() • forward() • refresh</p> <p>close() quit()</p> <p>findElement(), findElements() → List<WE></p> <p>ele.sendKeys() clear click() isEnabled() isDisplayed() isSelected()</p> <p>Alert al = dr.switchTo().alert();</p> <p>d). manage(). window() / misc() timeouts(). implicitlyWait() dropdown → dropdown</p> <p>Select s = new Select(ele); → for <select> tag only.</p> <p>s.selectByVisibleText(" "). { By Index { 1 }; By value ("xyz"). s.getOptions() }</p>				

Selenium

classmate

Date _____

Page _____

Wait Commands:-

✓ Applicable for all ele. (to get scanned by web browser instance)

it makes code to wait for an Element to appear in DOM

{ 1) implicit →

dynamic

wait

for not

halted

"cond" met

static

wait

(script ex)

is halted for

specified time t

even if cond is met

PSVM (S...a)

{

implicit wait

dmt

dr.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);

doGet("url");

Explicit wait for part of ele.

seconds

WebDriverWait wait = new WebDriverWait(dr, 10);

wait.until(ExpectedConditions.visibilityOfElement
Located(By.id("email")));

dr.findElement(By.id("email")).sendKeys("vdt@");

do -> dr.findElement(By.id("signIn")).click();

}

Interface

Class

WebDriver driver = new FirefoxDriver();

Explicit wait used when →
if some ele takes more time as compare to other ele to load

✓ ExpectedConditions.elementToBeClickable(By.name("text"))

Convert Keys of Map to List
↓ Values

Map.keySet().
↓ interface Comparable vs Comparator

① java.lang

② used for
natural sorting

↓ interface Comparable

↓ java.util

↳ used for
customised
sorting

↳ can sort employee 6) starts with => ^

Obj by name

Age
Salary etc.

7) ends with => \$

③ ↳ sorting logic

must be in same class ↳ Separate
class

Xpath (lines) / or //
↳ parent child
↳ descendent

CSS > or <space>

④

- int compareTo (Object ob) int compareTo (Object obj1, Object obj2) ⇒ To find

⑤ method call

Collections.sort(list);

Collections.sort (list, new otherSortClass());

HashMap Methods:

{ put(K,V) → add (Replace value of a key also)

{ get(K) → value of given Key or null if

remove(K) → remove mapping for given Key, no mapping

entrySet() → return a Set of K-V pairs

{ keySet() → returns a "Set" of keys in the hm.

{ values() → returns a "List" of values in hm.

{ containsKey(K) → T/F

{ containsValue(V) → T/F

wear(), isEmpty(), size(), equals()

putAll(hm) → copy all mappings from hm

Print map elements:- do this map.

Iterator<String> itr = hm.keySet().iterator();

while (itr.hasNext())

{ sop(hm.get(itr.next()));

 ↑ Key

otherwise you will get

"ConcurrentModificationException"
thrown by iterator.

CSS SY:

1) Using id => #id

2) Using className => .classname
Date
age

3) Using Attribute

tagName[attribute='value']

4) Using Multiple attribute

tagName[Attribute='value1'][attribute2='value2']

5) contains => *

6) starts with => ^

7) ends with => \$

Xpath (lines) / or //

CSS > or <space>

Xpath = //table//tr[3]/td[1]

In given table, select checkbox
& click on submit for India

for (int j=0; j< chist.size(); j++)

{ if (cclist.get(j).getText().equals("India"))

{ cclist.get(j-1).click();

 cclist.get(j+1).click();

 }

 }

d.findElement(By.xpath("//*[contains(text(),'India'))]

 .click();

 }

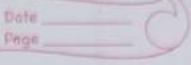
 if you are looking for modification
 List/ Set / map while iterating (like

 you should use class from java.util
 ArrayList → CopyOnWriteArrayList

 ArrayList → CopyOnWriteArrayList

 HashMap → ConcurrentHashMap

→ iterator's next() method throws ConcurrentModificationException if list size is changed during iteration?



CSS Selectors for Selenium:

→ Cascading style sheets → it is defined to display HTML in structured & colourful styles

→ Best alternative when there is no ID or name.

Benefits:

- more advantage than Xpath
- CSS is much more faster & simpler than the Xpath
- In IE Xpath works very slow, whereas CSS works faster when compared to Xpath.

Syntax:

css = [tagName [attributeName = 'attributeValue']]

e.g. input [id = 'email'] → input #email or #email
 input [name = 'email'] [type = 'text']

Special characters in CSS:

⇒ Id

• ⇒ Class

Caret → ^ ⇒ represents starting text in a string

· \$ ⇒ ending text "

* ⇒ contain text "

("India")

Note - class attribute is not unique like id.

So there can be multiple classes used for a single data element (like button)

e.g. td /

 spreading sibling; input [type = "button"]. click();
 button [class = "submit btn primary-btn flex-table-btn js-submit"]

 type = "submit" > login </button >

→ Java.util package
 classes gives

- fail-fast Iterator

→ Java.util.concurrent

@n ↓
 fail-safe iterator

e.g1 CSS = .primary-btn

e.g2 CSS = .btn.primary-btn

e.g3 CSS = .submit.primary-btn

WebElement b = d.fE(By.cssSelector(".primary-btn"));



classmate

Date _____
Page _____

`<input id="signIn" class="rc-button rc-button-submit">`

`css = input.rc-button.rc-button-submit`

or

`css = .rc-button.rc-button-submit`

`<input id="first-name" class="myForm">`

`css = input.myForm or css = .myForm`

Locating ele by combining more than one attribute

`<div class="ajax_enabled" style="display: block">`

`css = div[class=ajax_enabled][style=display:block]`

Locating child ~~ele~~ Element

`<div id="child"></div>`

`css = div#child img`

Choosing a specific Match:

used after: After: I target multiple children (e.g. list elements)

(n^{th} -of-type(n)

n^{th} -child(n)

Last-child

` # id = "fruit" >`

To locate part

`① Apple `

n^{th} -of-type

`② Orange `

To locate orange

`③ Banana `

`css = ul#fruit li : n^{th} -of-type`

To select last child ele -> Banana `css = ul#fruit li :last-child`

`css = ul#fruit li :last-child`

classmate

Date

Page

Strings & Randomly generated Ids

> Dynamically

`<div id = "123_randomId"`

`<div id = "randomId_456"`

`<div id = "123_Pattern_randomId"`

`css = div[id^='123']` → starts with

`css = div[id$='456']` → ends with

`css = div[id*='pattern']` → contains
or sub-string

~~`css = div[contains(_pattern)]` → obsolete~~

`div#A123`
will work?

Inner text:

e.g. "Need Help?" hyperlink given on webpage.

`css = <HTML tag><i><contains><(text)>`

⇒ symbolizes 'contains' method.

CSS Element loc using Adjacent sel?

↓
text that is
displayed anywhere
on the webpage
irrespective of its location

Next sibling

<input> </input>

<input> </input>

✓ <input> </input>

To locate 3rd input

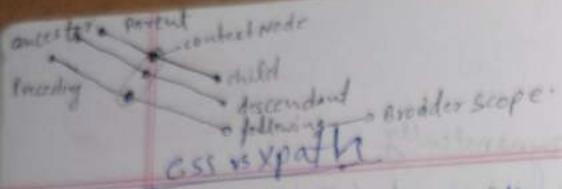
`css = input + input + input`

Next sibling ⇒ next adjacent element inside the same parent.

`css = form .input.username + input`

→ For immediate child we use '>' for descendants we use 'spans'
xpath '/>' xpath '//'

`css = form > label > input [id = PersistentCookie]`



CLASSMATE

Date _____
Page _____

1) Direct child:

Xpath uses '/'
CSS uses '>'

//div/a

CSS = div>a

2) Descendants (child or subchild)

if ele is inside another or one of its child.

Xpath uses '//'

CSS uses a whitespace

//div//a

CSS = div a

3) Id:

//div[@id='example']//a

CSS = div#example a

$\left\{ \begin{array}{l} @id \Rightarrow # \\ \end{array} \right.$

4) Class:

//input[@id='email'] \Rightarrow CSS = input#email

//*[@id='email'] \Rightarrow CSS = #email

using index to define path

//*[@name='email'][1]

dir[1].text() = 'sign up'

> sign up <

5) //div[contains(text(), 'sign up')]

or //div[contains(text(), 'Sign')]

we can replace 'text()' with '.*' dot

↓

//a[contains(., 'sign')]

Not good syntax
 \Rightarrow CSS = div:contains('sign')

CSS = a:contains('sign');

\Rightarrow contains work with normal text as well as link text

sometimes we may need to work with href

$\left\{ \begin{array}{l} //a[@href='url'] \\ CSS = a[href='url'] \end{array} \right.$

↪ Child, ↪ descendant
 ↪ parent \Rightarrow (td/..) parent of td
 (Two periods)

In programming * means everything
 • xpath wildcard *
 • xpath predicate E
 • xpath vertical bar or (pipe char) !
 ↪ And

Date _____
Page _____

→ By.cssSelector("a[onclick='clickClose();']") ✓
 By.xpath("//a[@onclick='clickClose();']") ✓
 refers current node
 amount > 10 < amount

(1) /
(2) //

inventory/drink/pop/amount [· < 20]
 inventory/*/*/amount [· < 20]

inventory/drink/pop/amount < 15

\nearrow parent \searrow child

XPath Axes

1) ancestor: parent & parent's parent etc.

e.g. ancestor for <div> \Rightarrow <td> <tr> <table>

2) ancestor-or-self

context node itself + ancestor

3) child: children of context node

for <tr>, <td>① & <td>②

"a"

4) descendant: children & their children etc.

for <tr>, <td>, <div>, <a>

5) descendant-or-self: context node + descendant

6) following: following siblings & their children & their children etc.

e.g. for <tr>I, <tr>II, <td> <div> <a> of <tr>II & <tr>III

7) following-sibling: following siblings

for <tr>I, <tr>II of <tr>III only not their children

8) parent: The parent of context node

for <div> its <td>

"single e"

9) preceding: reverse of following

10) preceding-sibling: //

11) self: context node

- classmate
Date _____
Page _____
- ↳ Ability to re-execute failed TCs.
TestNG → opensource
 ↳ Supports Data Driven Testing (with **DataProvider**) ↳ Add on to eclipse
 ↳ Support for parameters ↳ Report Generation
 ↳ Powerful Execution Model (One more TestSuite) ↳ Parallel Execution
 ↳ Assertion → To avoid if-else
 ↳ Test that your code is Multithreaded ↳ No need of main() method
 ↳ Support for variety of tools & plug-ins ↳ Logging exec status: helps in debugging in case of TGE errors
 ↳ Eclipse maven public class TestNGDemo

```
private WebDriver driver;
```

```
@BeforeTest  
public void setUp()
```

```
{  
  driver = new FirefoxDriver();  
  driver.manage().window().maximize();  
  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
```

(@Test)

```
public void MTC()
```

After exec
Refresh the project {
check the "test-output" folder
created

```
driver.get("http://www.google.com");  
WebElement wait = new WebDriverWait(driver, 10);  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.Id("SearchInput")));  
driver.findElement(By.Id("SearchInput")).clear();  
driver.findElement(By.Id("SearchButton")).submit();
```

@AfterTest

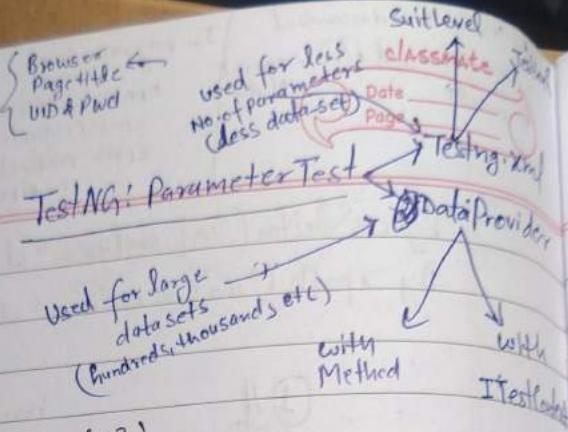
emailable-report.html
↓ open with Browser
↓ to see the report

```
public void tearDown()
```

```
{  
  driver.quit();
```

A simple report summarising ur test report

index.html: A info based report which provides info about the testcases & methods, also the pass/fail status of each method



@Test
 @Parameters({ "browser", "url" })
 p v mx(string b, string v)
 {
 3

→ Add TestNG Library in ur proj.
 from Project → Build path → Add Library.

~~Definition from official website~~

→ TestNG is a testing framework inspired by JUnit & NUnit but more powerful because of some new features & easy to use.

↳ It is designed to cover all categories of tests:

Unit, functional, end-to-end, integration.

↳ No need to extend a class or implement an interface.

↳ No need of main() method.

↳ **② Test(groups = {"fast"})** → A test method can belong to one or several groups.

→ To run testcases/testclasses in parallel

use parallel-classes attribute in testing.xml.

Setting Priority: To set exec order of T @Classmate
④ ~~dependsOnMethods~~: "0" represents the highest priority
@Test (priority = 1) (case-sensitive)
(PRIORITY = 1) ⇒ compile error.

Independent method → P v Login() { sop("Login Test Code"); }
Assert.assertEquals(5 > 6, "and False").
3 ↴ msg with printer

Dependent test case method ↗
② Test (dependsOnMethods = {"Login"}) ↗ if login file exists with be skipped
P. v. checkMail () { sop("checkMail code"); } ↴ same as?

② Test (dependsOnMethods = {"Login", "checkMail"}) ↗ just name not the symbol
P v logout() { sop("Logout Code"); }

② Test (priority = 2)

Independent method → P v checkLoginValidation() { sop("valid Code"); }

Multiple Parameters in Annotations ↗
② Test (priority = n, dependsOnMethods = {"meth1", "meth2"}) ↗ Test case will have dependency on multiple methods

① At runtime, by default, TestNG assigns priority 0 to all tests if priority not defined
② If we don't mention any priority, then test No1 meth will be executed based on alphabetical order of their names (respective ASCII value increasing order)

→ TestWIn dependency feature will work only if dependency test method is part of same class or part of inherited base class.
exceptionHandling

Q Test (expectedExceptions = ArithmeticException.class)
P V dividedByZeroDemo()

@Test(P=6)

P.V. a_method() {

{ sop("methodA") ; }

{

int e = 1/0;

⇒ T.C. will be

@Test(P=0)

P.V. e_method() {

{ sop("methodE") ; }

}

@Test(P=6)

P.V. d_method() {

{ sop("methodD") ; }

}

@Test(P=4)

P.V. c_method() {

{ sop("methodC") ; }

}

@Test(P=2)

P.V. b_method() {

{ sop("methodB") ; }

}

Working with IE & chrome

meth B
meth C
meth E
meth A
meth D

meth didn't throw
the given exception
therefore some other
TC. will marked

as failed.

download drivers from

SeleniumHQ website

System.setProperty("webdriver.ie.driver", "D:\\IEDriverServer.exe");
I
Chrome

WIP driver = new InternetExplorerDriver();
ChromeDriver

Multiselect

toolsqa.com/automation-practice-form

③ if 2 meth have same priority (say ±) then alphabetical order of meth names are considered. 3. deselectAll(); → can only be used with multiselect

④ we can have negative priority as well. 3. deselectByIndex(1); → for both By value By VisibleText

⑤ @Test(priority = -2) will execute before P=-1 or P=0 or P=+2

⑥ @Test(priority = 1 + 2) = will work fine on calculated result. ⑦ Priority of meth is respected across classes

Their place of implementation in the code:

By.xpath("//tagName[@Attribute = 'value'])
 Any tag of classmate
 Date Page

~~Alert Handling~~ click "OK" click "Cancel"
 + oper: alert.accept(), dismiss(), Type text in textbox prompt
 alert.getText(); alert.sendKeys("Deepak")
 ↳ To read text from alert pop-up

Q Before Test
 pr setup() throws Exception
 { driver.get("http://only-testing-blog.blogspot.in/2014/01/test.html")

Q Test 3 // Alert Pop-up
 pr text() throws InterruptedException
 { d.findElement(By.xpath("//input[@value='Show Me Alert?']")).click();

Hi, This is alert message
 [OK]

Alert a1 = driver.switchTo().alert(); // To locate alert
 System.out.println(a1.getText()); css = input::contains('SMALL')
 a1.accept();

|| Confirmation Pop-up <button onclick="myFunction()">SM Conf</button>
 d.findElement(By.xpath("//button[@onclick='myFunction()']")).click();

Alert a2 = d.switchTo().alert();
 System.out.println(a2.getText()); css = button::contains('SMC')

[OK] [Cancel]

a2.dismiss();

|| Prompt pop-up

d.findElement(By.xpath("//button[contains(., 'Show Me Prompt')]")).click();

Alert a3 = d.switchTo().alert();
 System.out.println(a3.getText()); css = button::contains('SMP')

a3.sendKeys("This is John");
 Thread.sleep("3000");

a3.accept();

How to check whether alert is present on the page or not

public boolean isAlertPresent()

{ try {

driver.switchTo().alert();

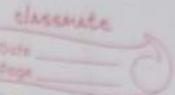
} return true;

catch (Exception e) if alert is not present

{ System.out.println("NoAlertPresentException will be thrown if false will be returned");

} return false;

AVI: Advanced User Interactions:-



~~Drag & Drop~~ Action class

→ in mobile app it's called

"Action"

3 ways to drag & drop

by dragAndDrop() method

by moveToElement()

by offset

```
driver.get("http://jqueryui.com/droppable/");
```

```
driver.switchTo().frame(0);
```

```
WEt element = driver.findElement(By.id("draggable"));
```

```
WEt target = driver.findElement(By.id("droppable"));
```

(new Actions(driver)).dragAndDrop(element, target).
perform();

for iOS & Android
→ this is used
WE^t = findby (id)

for any Q Test method default priority will be "0" → so if there is no priority set they will run sequentially.

Q Test → By default priority = 0

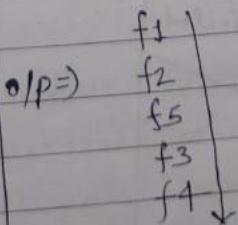
pr f1()

{ sop("f1");

3

Q Test(priority = 0)

pr f2()



Q Test(priority = 1)

pr f3()

Q Test(priority = 2)

pr f4()

Q Test

pr f5()

{ sop("f5");

3

* Reading values from dropdown :-

```
Select dd = new Select (de);
List <WebElement> list = dd.getOptions();
for (int i=0; i < list.size(); i++)
    System.out.println(list.get(i).getText());
```

```
for (WebElement e : list)
    System.out.println(e.getText());
```

~~assertEqual(AE)~~

If both values match then its fine if execute will continue. But if don't match that specific test will be marked as fail & it will exit from the test method.

Execution of that specific test method will be stopped.

Assert.assertEquals (ele.isSelected())

Hard Assertions → remaining code of — .isDisplayed()
method is not executed. — .isEnabled()

Soft Assertions

To continue remaining test execution in test method even if assertion fails & also you want to report assertion failure in testing report.

SoftAssert soft = new SoftAssert();

① Test if M1() → will be overall reported as failed.

System.out.println("M1");

soft.assertTrue(false);

System.out.println("M2");

soft.assertTrue(true);

System.out.println("M3");

soft.assertAll();

System.out.println("M4");

O/P: M1
will not execute M2
M3

TestNG Assertions

classmate

Date _____
Page _____

→ Assertion helps us to verify the condition of the test & decide whether test has failed or passed

"Assert" ← class

↓
only if test is completed
without throwing any exceptions.

→ TestNG supports assert" using "Assert" class.

e.g. → This can be used to check presence of web elements on webpages.

Commonly used Assertions:-

Not

from app

As of Testcase
recently expected

Works same as
assertTrue/assertFalse

1) assertEquals(String actual, String expected) (A,E) → opposite
assertEquals(String actual, String expected, String message)

2) assertNotEquals()

if assertion fails "java.lang.AssertionError" will be thrown & "message", if given, will be printed on console & Execution will be stopped.

3) assertTrue(boolean cond)
generally used for boolean cond
assert True (boolean cond, String msg) → if cond is true, test is passed & msg will NOT be shown.

4) assertFalse(boolean cond)
assertFalse (boolean cond, String msg) } → work opposite of assertTrue.
assertFalse (boolean cond)
assertEquals(A,E)

→ Parameters in above can be of any data type:
String, int, double, Array, object, collection etc.

→ Assert reports only failed scenario

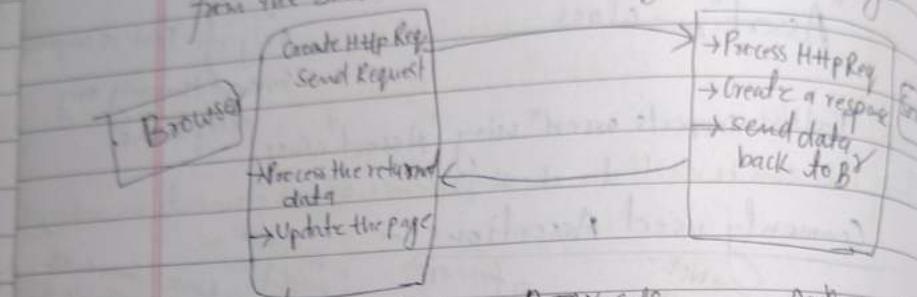
Assert.assertTrue(signbutton.isDisplayed());

Assert.assertTrue(login());

→ Assert.assertEquals("Google", driver.getTitle());
↓ error

java.lang.AssertionError: expected[Google] but found [Google]

AJAX: Asynchronous Javascript & XML used to store data in Ajax
 It allows the webpage to retrieve small amounts of data from the server without reloading the entire page.



AJAX Call

Ajax update a part of webpage without reloading the whole page.

App using Ajax technology:

{ Gmail
YouTube
Facebook
Google Map }

So when ajax call is made, we are not pretty sure about the time taken by server to send us response.

Ways to handle it:-

- 1) Thread.sleep(<time in ms>) → not recommended in selenium → only use in exceptional case if you do not find any alternative
- 2) Implicit Wait → will be applicable to all ele
- 3) Explicit Wait (WebDriverWait) → wait for certain condⁿ to occur before proceeding
- 4) Fluent Wait → To handle uncertain wait, Define max. amount of time after No need to mention time

W/Deg:

1) isElementPresent → (Timeout + frequency to check condⁿ)
 wait.until(EC.presenceOfElementLocated(By.tagName("xyz")))

↓
 Here we don't care whether till the ele is present in DOM or not appears
 Below Syntax can be used to check if ele is present or not

List<WebElement> l = driver.findElements(By.xpath("//"))
 l.size() > 0

Present in DOM

& Visible → EC.visibilityOfElementLocated(By -); or EC.visibilityOf(ele)

EC.invisibilityOfElementLocated(By -);

EC.elementToBeClickable(By -);

{ EC.titleContains("google")
EC.titleIs("Google"); }

{ EC.alertIsPresent();
EC.frameToBeAvailableAndSwitchToIt(); }

EC.visibilityOfAllElementsLocatedBy()

Locator

{ ele.isEnabled()
ele.isDisplayed() }

ele or index or str

Mouse Hover

classmate

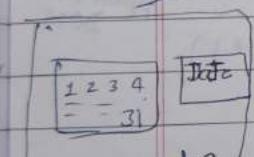
Date

Page

- Actions action = new Action(driver);
- SearchEngine Google
 - WebElement first = driver.findElement(By.xpath("//a[@href='http://www.google.com']"));
 - action.moveToElement(first).moveToElement(driver.findElement(By.xpath("//a[@href='http://www.google.com']"))).click();
 - action.moveToElement(first).build().perform();
- WebDriverWait wait = new WebDriverWait(driver, 15);
 - wait.until(ExpectedConditions.titleContains("Google"));
 - or hold at 1st ele
- action.moveToElement(first).build().perform();
- driver.findElement(By.xpath("//a[@href='http://www.google.com']")).click();

Handling Ajax calls

handled by implicit & explicit wait

- 
- Wait for Grid to appear. wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(".demo-container")));
 - WE dateText = driver.findElement(By.xpath("//input[@id='dp']")).getAttribute("value");
 - String txtBeforeAjaxCall = dateText.getText().trim();
 - driver.findElement(By.linkText("2")).click();
 - By loader = By.className("zadiv");
 - wait.until(ExpectedConditions.invisibilityOfElementLocated(loader));
 - WE dateText = driver.findElement(By.id("dp")).getAttribute("value");
 - String txtAfterAjaxCall = dateText.getText().trim();
 - Assert.assertEquals(txtAfterAjaxCall, txtBeforeAjaxCall);
 - Assert.assertEquals(txtAfterAjaxCall, "Tuesday, September 1, 2015");

Autocomplete / Live Search

③ Enter classmate

④ press enter

Tags: Ta

ActionScript
AppleScript
Asp
Java

iframe
classmate
Date
Page

```
① Test p.v.metacode {
    d.navigate().to("http://jqueryui.com/autocomplete/");
    WEt frame = d.fE(By.className("demo-frame"));
    d.st.frame(frame);
    waitUntil(ExpectedConditions.presenceOfElementLocated(By.id("tags")));
}
```

```
WEt textbox = d.fE(By.id("tags"));
    textbox.sendKeys("a");
    selectOptionWithText("Java");
    } for
```

p.v.selectOptionWithText(String start)

{

```
try {
    WEt ul = d.fE(By.id("ui-1d-1"));
    waitUntil(EC.visibilityOf(ul));
}
```

```
List<WE> options = ul.fEs(By.tagName("li"));
for (WebElement o : options)
```

```
{ if (o.getText().equals(start))
```

```
{ sop("Trying to select:" + start);
```

```
o.click();
break;
}
```

3 3 3

catch (Exception e)

```
{ sop(e.getStackTrace());
}
```

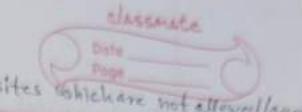
→ can not be
handled by Select class
X selectByVisibleText()

can be used only when
<select> tag is there in HTML.

```
<ul id="ui-1d-1">
<li id="> ActionScript </li>
<li>
<li id="> Java </li>
</ul>
```

Setting Proxy Settings

Used to access websites which are not allowed/given always
1) FirefoxProfile 2) DesiredCapability class



Once instance is closed
setting of B is reset to
default or old one.

@Before

p.v.setup()

{

String serverIP = "199.201.155.127"; // localhost
int port = 199; // 8080

FirefoxProfile profile = new FP();

profile.setPreference("network.proxy.type", 1);

profile.setPreference("network.proxy.ftp", serverIP);

http

socks

ssl

on the basis of
option selected
on the
Date
Page
Radio
button

profile.setPreference("network.proxy.httpport", port);

http
sock

ssl

[WebDriver driver = new FirefoxDriver(profile);]

@Test

p.v.m1()

{
driver.get("");
}

}

Firefox → options → Advance → Network → settings

updated settings
can be seen when browser
is opened

Synchronization in Selenium Web

↳ mechanism which involves more than one component, work parallel with each other, with same speed.

We synchronize App Under Test & Test Automation Tool to work with same speed to avoid "Element Not Found".

Synchronization can be of two types:

- 1) Unconditional Sync → Thread.sleep & wait()
- 2) Conditional Sync → implicit / explicit / Fluent wait

Firefox profile example

setting startup home page

```
FirefoxProfile profile = new FirefoxProfile();
profile.setPreference("browser.startup.homepage", "http://www.google.com");
driver = new FirefoxDriver(profile);
```

Taking Screenshot

classmate

Date
Page

TakesScreenshot ← interface
getScreenshotAs() ← method

```
ps void () {  
    import java.util.Date;  
    java.sql.Timestamp;  
  
    Date date = new Date();  
    Timestamp t = new Timestamp(date.getTime());  
    sop(t);  
  
    String dateNow = t.toString().replace(":", "-").  
        replace(":", "-");  
    dr.get("url");  
    /  
    * To avoid file name syntax  
    * incorrect error.  
    File scrFile = ((TakesScreenshot) driver).getScreenshotAs(  
        OutputType.FILE);  
}
```

① // FileUtil.copyFile(scrFile, new File("./screenshots/" + dateNow + ".png"));

FileUtil.copyFile(scrFile, new File(dateNow + ".png"));

↓ source dest "abc.png"

sop("Screenshot taken");

Refresh
the project
to see screenshot.

3
JavascriptExecutor jse = (JavascriptExecutor) driver;

webTable (Calendar)

classmate

Date
Page

Handling Dynamic Table:-

```

dr.get("http://only-testing-blog.blogspot.in/2014/05/form.html")
WebElement table = dr.findElement(By.xpath("//table[@id='post-body-82']"))
List<WebElement> rows = table.findElements(By.tagName("tr"));
for (int i=0; i<rows.size(); i++) {
    List<WebElement> columns = rows.get(i).findElements(By.tagName("td"));
    System.out.println("No. of cells in row " + i + " are " + columns.size());
}
  
```

3	4	5
4	5	6
5	6	
6	7	8
7		

```
for (int j=0; j<columns.size(); j++) {
```

```
{
```

```
String cellText = columns.get(j).getText();
```

```
sop("values in row " + i + " and column " + j + " are " + cellText);
sop("-----");
```

Handling windows pop-up (Tab)

```
dr.get("url");
```

```
String pwin = dr.getWindowHandle();
```

```
sop(pwin);
```

```
dr.findElement(By.linkText("Blogger")).click();
```

```
Set<String> wins = dr.getWindowHandles();
```

```
wins.remove(pwin);
```

```
dr.switchTo().window(wins.iterator().next());
```

```
dr.findElement(By.id("Email")).sendKeys("deepakbbd2011");
```

```
dr.findElement(By.linkText("Create account")).click();
```

```
driver.close();
```

```
dr.switchTo.window(pwin);
```

```
dr.findElement(By.name("name")).sendKeys("Deepak");
```

Returns parent window

new window

switch to child window

switch to new window

close the current window

return to parent window

~~Hand~~ Logging in Selenium; By logger class from ~~selenium~~ package.

→ Helps in debugging the script
 → Alternative to System.out.println
 → Lighter than System.out.println
 ↓
 puts logs on console

```
Class xyz {
    @BeforeTest
    p.v setup()
    {
        for(WebElement c : cd)
        {
            if(c.getText().equals("India"))
            {
                c.click();
            }
        }
    }

    @Test
    p.v. method()
    {
        driver.get("https://www.google.com");
        log.info("url opened"); // Reporter.log("App launched successfully");
        d.findElement(By.id("lst-ib")).sendKeys("facebook");
        Reporter.log("Text entered in search field");
        d.findElement(By.xpath("//a[@name='btnG']")).click();
        Reporter.log("search is successful");
    }
}
```

↓ works in TestNG only, not in normal Java class with main method.

```
import org.openqa.selenium.logging.Logs;
import java.util.logging.Logger;
import org.apache.logging.log4j.LogManager;
```

→ System.out.println(); ⇒ puts load on Server.
 → Log4j; Audit logging Framework. → uses Logger & LogManager class from apache-log4j package.

↳ log file is generated which can be analyzed later.
 ↳ which stores all logs, similar to above

↳ to be added or as maven dependency

Enum → used to store constants
↳ Keys → Enum direction = { East, west, North, South }

direction.East → can be used to

similarly we have "Keys" enum e.g. "Keys.F5"

ENTER
TAB

Enum Constants

ele.sendKeys(Keys.TAB) → can be used to remove focus from current ele.

dr.get('url'); → Google home page

dr.findElement(By.id("lst-ib")).sendKeys("facebook");

dr.findElement(By.id("lst-ib")).sendKeys(Keys.F5);

ele.sendKeys(Keys.DOWN);

ele.sendKeys("wikipedia");

ele.sendKeys(Keys.ENTER); → without clicking search button

can be used when

Search box is given but

Search button is not

given

action.sendKeys(Keys.DOWN).sendKeys(Keys.ENTER);

↓
Same as

ARROW_DOWN

(similarly Keys.UP)

Keys.ARROW_DOWN?

• ARROW_UP ↑ To go up/down
• TAB ← LEFT → RIGHT in a list or dropdown

• ALT

• SHIFT

• CONTROL

• CLEAR

• BACK_SPACE

• DELETE

• DECIMAL

• ESCAPE

• F1 → F12

• HOME

• END

• NUMPAD0 → NUMPAD9

• PAGE_DOWN

• PAGE_UP

• SEMICOLON

• ADD +

• SUBTRACT -

• DIVIDE /

• MULTIPLY *

EQUALS =

To select 6th option in dropdown/list

for (int i = 0; i < 6; i++)

{ dr.findElement(By.id("sb")).sendKeys(Keys.ARROW_DOWN);

if (i == 5)

} dr.findElement(By.id("sb")).sendKeys(Keys.ENTER);

Debugging Script(s) in Eclipse

- put a breakpoint (using toggle breakpoint) in ur script
 - ↓ At the line where you want to stop script exec
 - & start debugging line by line.

R.C.

Debug As

Java App

Use A buttons for debugging

F5 → Step Into

or constructor

To debug script with
1000 lines of code

F6 → Step Over

first remove all breakpoint
from ur project from debug

F7 → Step Return

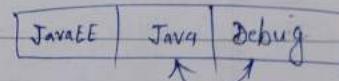
then set ur breakpoints. Eclipse

F8 → Resume

Continue running program until next breakpoint
or till the end of code if no more breakpoints

Terminate

To terminate program exec.



There can be conditional
breakpoints as well

doubleclick To switch
b/w the two
perspective

R.C.

Togglebreakpt

Breakpt properties

variables Expressions

$a \times b$
Select exprn

To see how
value of a var/expr
changes

→ R.C.

watch:

$a+b=20$

custom exprn
 $a+b=10/20$

testng.xml → can be integrated with Jenkins
+ contains the suite & the way exec will happen
→ generally only one testng.xml file is used.

<suite name="suite" parallel="tests" preserve-order="true" >

↓
3 threads would be created
in parallel in order of

t1 → t2 → t3.

If thread-count = '3'

If there are 5 tests then at a time 3 threads (for 1st step)
will be created when one of 3 threads closes then for 2nd
a thread will be created.

Parallel Tests

testng.xml → JRC
Learnt As TestNG Suite

classestests.html + space to get options A+B

```
<suite name="Suite" parallel="tests">
  <test name="test1">
    <parameter name="browser" value="firefox" />
    <classes>
      <class name="myTests.ParallelTests" />
    </classes>
  </test>
  <test name="test2">
    <classes>
      <class name="myTests.ParallelTests" />
    </classes>
  </test>
  <test name="test3">
    <classes>
      <class name="myTests.ParallelTests" />
    </classes>
  </test>
</suite>
```

Parallel Tests.java

All the 3 tests will run in parallel
→ 3 threads will be created simultaneously
→ if parallel attribute is not there
then they will run sequentially.

② RefersTest

① Parameters("browser") // Multibrowser Testing.

p v setProp(String browser)

```
{ if(browser.equalsIgnoreCase("firefox"))
```

```
{ driver = new FirefoxDriver();
```

```
{ System.setProperty("-Dwebdriver.chrome.driver");
```

```
driver = new ChromeDriver();
```

```
elseif(b.equalsI("ie"))
```

```
{ System.setProperty("-Dwebdriver.ie.driver");
```

```
driver = new InternetExplorerDriver();
```

```
else{ SOP("Wrong browser");}
```

② driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
driver.manage().window().maximize();

@Test

P. V. f() throws InterruptedException

```
{
    driver.get("http://www.google.com");
    dr.findElement(By.id("lst-ib")).sendKeys("Mumbai");
    Thread.sleep(3000);
}
```

@AfterTest

P. V. tearDown()

```
{
    driver.close();
}
```

Retry to execute

How to execute only failed testcases in S^m ?

using "IRetryAnalyzer" Interface

public class Retry implements IRetryAnalyzer

```
{
    int minRetry = 1;
    int maxRetry = 3; // will execute test 3 times
}
```

public boolean retry(ITestResult result)

```
{
    if (minRetry <= maxRetry)
    {
}
```

sop("Following test is failing :" + result.getName());

sop("Retrying count is :" + minRetry);

minRetry++;

return true;

for logging purpose

}

return false;

@Test(retryAnalyzer = Retry.class)

P. V. testM()

⇒ if this T.C. fails then it will

this will run until 3 times

call Retry class.

if test passes within this frame, it will come out of loop

Page Object Model: using properties file

→ For each webpage in app there should be a properties file to store the element details.

CalcPage.properties or Java class file

IDs:
One=1

Two=2

Three=3

Four=4

Five=5

Six=6

Seven=7

Eight=8

Nine=9

Zero=0

plus=plus

result=Resultbox

facebook=Facebook

Xpaths:

Login = //input[@id='u_0_0']

~~PageObjectModel.java~~

driver(url);
→ java.util prop

5+2>7
1 2 3 4 /
S 6 7 8 *
9 0 + - =

Properties prop = new Properties();

FileInputStream fis = new FileInputStream(System.getProperty("user.dir") + "/SeleniumProject/CalcPage.properties");

gives current directory
/SeleniumProject

prop.load(fis);

d.findElement(By.id(prop.getProperty("five"))).click();

"plus"

"two"

"equals"

.click();

String sum = d.findElement(By.id(prop.getProperty("result"))).getAttribute("value");

sop("sum is:" + sum);

d.findElement(By.linkText(prop.getProperty("facebook"))).click();

d.findElement(By.xpath(prop.getProperty("Login"))).click();

ParameterizedTest: Using DataProvider

classmate

Date _____
Page _____

```
@BeforeTest  
p.v.setup() { driver.get("http://www.google.com"); }
```

```
@Test (dataProvider = "dataPro")
```

```
p.v.f1(String author, String key)
```

```
{
```

```
d.findElement(By.name("q")).sendKeys(key);
```

↓ search box

```
sop("welcome " + author + " your search key is: " + key);
```

```
String textval = d.findElement(By.name("q")).getAttribute("value");
```

```
Assert.assertEquals(textval.equalsIgnoreCase(key));
```

```
}
```

```
@Test (dataProvider = "dataPro")
```

```
p.v.f2(String key)
```

```
{
```

```
d.findElement(By.name("q")).sendKeys(key);
```

```
sop("your search key is " + key);
```

String textval

Au

```
Assert.assertEquals(textval.equals(key));
```

```
}
```

```
@DataProvider (name = "dataPro")
```

```
public Object[][] getData(Method m)
```

```
{ if (m.getName().equalsIgnoreCase("f1"))
```

```
{ return new Object[][] {
```

↳ multidimensional

```
  { "Ria", "India" } }
```

```
  { "Deepak", "USA" } }
```

```
  { "Amit", "UK" } }
```

}

```
else
```

```
{ return new Object[][] { { "Canada", { "Russia", "USA" } }, { "Japan", "UK" } }
```

}

}

File Upload

Uploading.au3 → Compile → uploading.exe

classmate

Date
Page

- 1) Download & install AutoIT
- 2) Write the script on AutoIT editor
- 3) Save the script with .au3
- 4) compile & running F5
- + seconds
- 5) Give the path of exe file in your script

WinWait("[CLASS: #32770]", "", 10)

; set the focus

comment

ControlFocus("FileUpload", "", "Edit1")

Sleep(2000)

ControlSetText("FileUpload", "", "Edit1", "C:\Picard\abc.jpg")

Sleep(2000)

ControlClick("FileUpload", "", "Button1");

To run on autoIT editor

press F5

when upload window is opened.

FileUpload.java

Runtime.getRuntime().exec("D:\Downloads\uploading.exe")

AutoIT uses:

→ upload

→ Download

→ Window Authentication

→ open source.

GIT Hub: version control: common repository.
Tool

BitBucket:

X64 → will work for both 64 bit & 32 bit os.

X86 → will work for only 32 bit system.

Custom Logging:-

Listeners :-

- Used for logging on console
- Available in testing framework only.
- Used when we need to see the continuous status of the test execution when tests are getting executed.
- To keep an eye on execution flow for debugging in case of failure.
- Class has to implement "org.junit.TestListener" interface.
- These classes are notified by TestNG when the test starts.
- There can be multiple listeners added to the suite.

classmate

Date _____

Page _____

testng.xml

```
<suite name="suite">          preserve-order Attribute is also for suite tag  
    <listeners>                ↓  
        <listener class-name="pkg.ListenerClass" />  
    </listeners>              for classes execute in the order in which they're written  
    <test name="Test" preserve-order="true">  
        <classes>  
            <class name="testingPkg.LoggingClass" />  
        <methods>  
            <include name="addingNumbers" />  
            <include name="divideByZero" />  
            <include name="methodSkip" />  
        </methods>  
        <!Class>  
    </classes>  
    <test>  
</suite>
```

ListenerClass.java

package PK3;

import org.junit; IClass;

ITestResult;

TestListenerAdapter;

public class ListenerClass extends TestListenerAdapter

{

@Override

public void onTestStart(ITestResult tr) //invoked each time a test

{

Log("Test Started...");

}

@Override

public void onTestSuccess(ITestResult tr) //invoked each time a test

{

Log("Test " + tr.getName() + " Passed");

}

Log(tr.getTestClass());

Log("Priority of this method is:" + tr.getMethod().getPriority());

sop("----");

@Override

public void onTestFailure(ITestResult tr) //each time a test

{

Log("Test " + tr.getName() + " Failed");

Log(tr.getTestClass());

Log("Priority ----");

sop("----");

3

//takeScreenshot();

call a method to take screenshot

Alternative: taking screenshot in catch block

3

p. v. onTestSkipped(ITestResult tr)

//invoked each time a test is skipped.

Log("Test " + tr.getName() + " Skipped");

sop("----");

classmate

```
private void log (IClass testClass)
{
    sop (testClass);
}
```

```
private void log (String methodClass)
{
    sop (methodClass);
}
```

}

LoggingClass.java

```
public class LoggingClass
{
```

① Test (priority = 0)

p. v. addingNumbers()

```
{ sop ("In adding Numbers"); }
```

② Test (priority = 1)

p. v. divideByZero()

```
{ sop ("In divide by zero"); }
```

```
int a = 1/0; }
```

③ Test (dependsOnMethods = {"divideByZero"})

p. v. methodSkip()

```
{ sop ("In methodSkip"); }
```

}

O/P:

Test started...

In addiy Numbers

Test: addingNumbers Passed

[TestClass name = class LoggingClass.LoggingClass]

Priority of this method is: 0

Test Started...

In divideByZero

Test: divideByZero Failed

TestClass name → : 1

Priority → : 1

Test: methodSkip Skipped.

Page Object Model using PageFactory Model.

- ① By Page Classes, Extend \Rightarrow Then Page class
- ② By Page Interface, implement we create Page classes which stores page objects.
- ③ By Properties class, static M: \Rightarrow Date extended by T.G. as per requirement

Public class LoginPageFactory \rightarrow org.openqa.selenium Support "Find By"

```
① { ② FindBy(name = "username")  
    public static WebElement uname;
```

```
③ FindBy(how = How.NAME, using = "password")  
    P S WE PWD; ID XPATH CSS etc. value of locator
```

```
④ FindBy(name = "login")  
    P S WE login_button;
```

```
public LoginPageFactory(WebDriver driver)  
{  
    this.driver = driver;  
}
```

```
}
```

public class TestCase1
{

```
    @Test  
    {
```

```
        get(url);
```

```
        PageFactory.initElements(driver, LoginPageFactory.class)
```

```
        LoginPageFactory login = new LPF();  
        login.uname.sendKeys("abc");  
        login.pwd.sendKeys("abc@123");  
        login.loginbutton.click();  
    }
```

```
    driver.quit();
```

```
{ ② FindBy(-)
```

```
    ③ CacheLookup  $\Rightarrow$  If some ele is being used frequently / multiple times.  
    P. V. WE xyz; we can look in cache for quick access.
```

```
Better not to use for each & every ele
```

14

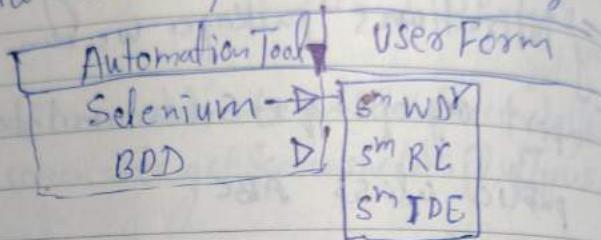
THURSDAY
JANUARY

014-352 • WK 03

Mouse Hover

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

We can hover the mouse to an webelement
using Action class of interactions pkg.



{
1 WebElement menu = d.findElement(By.id("Automation Tool"));
2 WE selenium = d.findElement(By.id("Selenium"));
3 WE selenWebdrive = d.findElement(By.id("selenWDn"));
4 Hover(driver, menu);
5 Hover(driver, selenium);
6 Hover(driver, selenium, seleniumWebdrive);

{
P.S. V. Hover(WD driver, WE element)

{
new Actions(driver).moveToElement
(element).perform();

{
P.S.V. HoverAndClick(WD driver, WE eleToHover,
WE eleToClick)
new Actions(driver).moveToElement(eleToHover).click(eleToClick)
• build().perform();

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F	15	16	17	18	19	20	21	22	23	24	25	26	27	28
E	29													
B		M	T	W	T	F	S	S	M	T	W	F	S	S

HTML pop-up
iframe pop-up
javascript pop-up
OS/windows pop-up

FRIDAY

JANUARY

15

Handling pop-up in Selenium

1) HTML Pop-up: elements are given in HTML

→ signIn part of BookMyshow.com code, find it & act on it

Hungama.com

(click)

2) HTML pop-up: iFrame : switch to it



12 driver.switchTo().frame("authIframe").
13 d.fE(By.Id("email"), sendKeys("abc@gmail.com"))
14 Goibibo → signIn
15 signUp



3) Non-HTML: Javascript pop-up

e.g. alerts

<iFrame id=">

3 driver.switchTo().alert().accept();

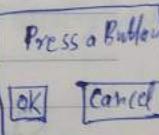
4 d.fE(By.xpath("//button[contains

(text(), 'Try It')]")).click();

dismiss();

getText();

sendKeys(String abc);



5 f) Non-HTML → OS pop-up.

generated by OS. ~~OS pop-up~~ e.g. upload window.

handled by AutoIT

switchTo() → alert(), frame("ID"), window()

2016

15 TUESDAY
MARCH

075-291 • WK 12

1 2 3 4 5 6 7 8 9 10
11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

M T W T F S U M T F S U

Testing Dependency Test
dependsOnMethods
dependsOnGroups

used in $\textcircled{1}$ Test

$\textcircled{1}$ Test (dependsOnMethods = { "ABC" })

Annot^h

$\textcircled{1}$ Test (dependsOnGroups = { " init.*" }) m0()

$\textcircled{1}$ Test (groups = { " init" }) m1() \downarrow m₁ & m₂ will

$\textcircled{1}$ Test (groups = { " init" }) m2() alwz be invoked
before m0

TestNG: Parameterized Test

<suite>

<test>

<parameter name="myName" value="Deepak"/>

<classes>

</classes>

</test>

</suite>

Slope: can be suite level (if under <suite>)

test level (if in <test>)
if both are there,

2016 test level will override

<suite> level

WEDNESDAY

* MARCH

16

→ value specified in testing and will automatically be converted to the type of parameter by TestNG.

e.g. int, String, float

Integer, float etc.

→ DataProvider may pass Integer, Boolean etc as param to
or Object as param

Executing existing JUnit testcases using TestNG

↳ we have to put Junit lib on TestNG classpath

testng.xml

<test name="test1" junit="true">

<classes>

= Junit classes

</classes>

</test>

TestNG-Results 2 ways to generate report:-

1) Listeners

2) Reporters

<methods> → using Reg exp "to include/exclude test methods from execution"

<include name="*, *two, *"/> → All test methods

<exclude name="*, *three, *"/> (containing word "two")

will be included in execution 2016

</methods>

16
Saturday

Xpath

January 2016
Week 03
Day 078 x 200
Date 16 + 01 + 2016

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

1. Id
`//div[@id='panel']`

2. Class
`//div[@class='panelClass']`

→ for element with multiple class we need to separate with space (" ") .
`//div[@class='panelClass1 panelClass2']`

3. Any Attribute
`//div[@type='button']`

 @name

4. Direct Child
`//ul/li`

5. Child/Subchild
`//ul//li`

6. nth Child
for 3rd child → Tea
`//ul[@id='drinks']//li[3]`
gt may be // as well

Note: 3rd child → Tea

for 3rd child → Tea

gt may be // as well

`<ul id="drinks">`
` coffee `
` Tea `
` Milk `
``

CSS

January 2016
Week 03
Day 079 x 200
Date 17 + 01 + 2016

17
Sunday

1. `div #panel` or `div[id='panel']`

2. `div.panelClass` or `div[class='panelClass']`
→ multiple class: . (dot)

`div.panelClass1.panelClass2`

3. `div[type='button']`

4. `ul>li`

5. `ul li`

6. `ul#drinks li:nth-of-type(3)`

or
→ will also work here.

Note: ul#drinks li:nth-child(3)

`p! nth-child(2) → India`
`p! nth-of-type(2) → Pak`

`<sections>`
`<h2> words <h2>`
`<p> India </p>`
`<p> Pak </p>`
`<sections>`

18

Xpath

Monday

January 2016
Week 04
Day 018 • 348
Date 18 • 01 • 2016

January 2016

Week 04

Day 018 • 348

Date 18 • 01 • 2016

7. Parent of an Element

/..

<li class="blue">
 second
 third
/li[@class = 'blue'] /.. → will represent

8. Next Sibling

↳ Next adjacent element inside same parent

/li[@class = 'blue'] /.. /li[2] →
li[3] ⇒ Second
li[3] ⇒ Third

9. Match By Inner Text.

//a[contains(text(), 'Login')]
or
//a[contains(string(), 'Login')]
or
//a[(., 'Login')]

10. match a substring

Note	Xpath	contains()	starts-with()	ends-with()
CSS	*	^=	=	\$=

19

CSS

Tuesday

January 2016
Week 04
Day 019 • 347
Date 19 • 01 • 2016

January 2016

Week 04

Day 019 • 347

Date 19 • 01 • 2016

7. li.blue.parent

(1) limitation as c. to xpath. This is Not supported so there is no way to take parent element with css selectors (xpath has ##footer-list a[href*='about'])

8. li.blue + li → second
li.blue + li + li → third

9. a:contains('Login') → not supported now

(2) limit w. - we can't use contains() for inner text of an element.

10. input[type^='sub'][name\$='thk']

20

XPath

Wednesday

January 2016
Week 04
Day 020 • 346
Date 20 • 01 • 2016

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

10. Match By Sub String (prefix, suffix, pattern)

Any
subString //div [contains (@id, 'pattern')]

//div [starts-with (@id, 'prefix')]

//div [ends-with (@id, 'suffix')]

AND, OR, Not operators

11. AND
//input[@type='submit'] and contains(@name, 'blk')

, //input[@type='submit'][contains(@name, 'blk')]

, //input[contains(@type, '')][contains(@name, '')]

12. OR

//input[contains(@value, 'Google')] or contains(@value, 'luck')]

, //input[contains(= '')] | //input[contains(= '')]

↓
Pipe
Symbol
(|)

Notes

CSS

21
Thursday

January 2016
Week 04
Day 021 • 345
Date 21 • 01 • 2016

10. div[id*='pattern']

div[id^='prefix']

div[id\$='suffix']

11.

11. ① input[type='submit'][name='blk']

12. input[type='submit'], input[name='blk']

↓
Comma
Separated

Notes

22

Friday

Xpath

B. Not

January 2016
Week 04
Day 02 - 348
Date 22-01-2016

not() → is a function

Xpath

|| li [not(contains(., 'Tea'))]
 bx

|| li [text() != 'Tea']

→ || input[@value != 'Google']

|| input[not(@value = 'Google')]

+ To select input which do not have
@class attribute

|| input[not(@class)]

B. Dynamic

ele that is disabled || E[@disabled]

enable =

checkbox/radio bts || * [not(@disabled)]

that is checked

|| * [@checked]

B. Parent/child

element with no children || A[count(*)=0]

|| with 1 child

|| E[count(*)=1]

January 2016
Week 04
Day 03 - 347
Date 21-01-2016

January 2016
Week 04
Day 03 - 347
Date 21-01-2016

23

Saturday

CSS

B. input : not([name='black']) [type='submit']

+
input [type='submit']: not([name='black'])

→

input: not([type='submit']): not([name='black'])

24

Sunday

X Path Axes /ækſɪz/ Jaxs

January 2016

Week 04
Day 024 • 342
Date: 24 • 01 • 2016

Axis

Method:::td(1)
position(1>1)

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

→ There are around 13 xpath axes.

An xpath axis tells the xpath processor which dirⁿ to head in the the hierarchical tree of nodes or by

Commonly Used Axes { //axis::*

child, parent, ancestor //axis::tagname

descendent, preceding, following, self

✓ 1. //child::* or /
defines children of the current node.
→ not the grandchildren

✓ 2. //descendant::* or //
children+Grandchildren
+ Grand Grand etc
of current node etc

Method:::td(1)
position(1>1)

January 2016

Week 05
Day 025 • 343
Date: 25 • 01 • 2016

25

Monday

/parent::* or //

defines parent node of current node
contains max one node.

min 0 node, if current is root ele (<html>)

//input[@id='email']/parent::*

or
//input[@id='email']//..

4. //ancestor::* → selects all ancestors

Parent + Grandparent + Greatparent etc

5. //preceding::* selects all nodes that come before the current node except ancestors

sibling & their children

6. //following::* selects everything after closing tag of current node

Notes

26

Tuesday

January 2016

Week 05

Day 05 • 340

Date 26 • 01 • 2016

S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31					

✓ 7. /preceding-sibling::*

selects all siblings before current node, but not the siblings' children.

✓ 8. /following-sibling::*

all siblings after current node
no children of sibling.

9. /self::* → selects current node itself
→ max 1 node.

10. /ancestor-or-self::* current node & ancestors

11. /descendant-or-self::* current node + descendants

Using Xpath Functions in Selenium

January 2016

Week 05

Day 06 • 341

Date 27 • 01 • 2016

27

Wednesday

When we get more than one element which are same in attribute & name like multiple checkboxes with the same name & same id, more than one button having same name or same id.

→ input type = 'checkbox' name = 'CHK' → first
 " " Second
 " " Third
 " " Fourth

Xpath expr^n can return
~~String~~ String, Number, Boolean, Node-Set
Functions

Node-Set: last(), position() etc.

String: starts-with(), contains(), string()

Boolean: not(), true(), false(), boolean()

Number: number(), sum(), floor(), ceiling()

If we have to select an ele based on pos^n
Notes then we can use

last() or position()

28

Thursday

January 2016
Week 01
Day 08 • 33h
Date 28-01-2016

for last checkbox

```
//input[@type='checkbox'][last()]
```

can be index as well
1, 2, 3, etc

can be
[last()=1]
for 2nd last
[last()-2]

for 3rd last

[4] → will also give
last checkbox but if
new checkboxes
are added this
index will not select
last checkbox.

for 2nd or 4th checkbox:

```
//input[@type='checkbox'][position()=2]
```

or position=2

for first 3 checkbox

[position() < 4]

for all checkbox except first 3, [position() > 3]

```
//input[@type='checkbox'][following::td[last()-1]]
```

```
//td[contains(., 'Pending User')]/following-sibling::td/child::
```

if Not a Link

button[contains(., 'Edit')]

&

```
<td><button>Edit</td>
```

29

Friday

Xpath

Table Element Selections

Name	PIN	City	Action	Status
Pending User	367	BOM	Edit	Active
ABC XYZ	678	DEL	Edit	Active

<tr>

<td> <a> Pending User </td>

<td> 367 </td>

<td> BOM </td>

<td> <button> Edit </button> </td>

<td> Active </td>

</tr>

or text()

//button[contains(., 'Edit')] and ./ancestor::*[./preceding-sibling::*[./child::*[contains(., 'Pending User')]]]]

or

//button[contains(., 'Edit')][./ancestor::*[./preceding-sibling::*[./child::*[contains(., 'Pending User')]]]]

or

//a[contains(., 'Pending User')]/parent::*[./following-sibling::td/child::span/child::button[contains(., 'Edit')]]

descendent

`singEdit()` introduces a relative locn path, starting
the context node
(current node)

30 Saturday

January 2016
Week 05
Day 030 - 336
Date 30-01-2016

`//a[contains(., 'pending user')]//parent::td/
parent::tr//button[contains(., 'Edit')]`

or

`//a[=]//..//button[=]`

`b "+name+"`

`//tr//a[contains(., 'Pending Us')]//button
[contains(., 'Edit')]`

It says if u find a link with given text
under any tr then check for button
under that tr with given button text

`//a[contains(., 'firstname')] and contains(., 'Edit')`

p.v. `clickEdit(string name)`

{
WE ele=d.findElement(By.xpath("//tr//a[contains(., "+name+")]"));
ele.click();

}

`%s` specifier is used for String or any type
%c char
%d integer
%f decimal no.

January 2016
Week 05
Day 031 - 336
Date 31-01-2016

31 Sunday

`String editBtn = "//tr//a[contains(., '%s')]//button";`

p.v. `clickEdit(string name)`

{
try{
WE ele=d.findElement(By.xpath(String.format(editBtn, name)));

ele.click();

catch(Exception e){

}

}

lastname String editBtn = "//tr//a[contains(., '%s')]//button";

d.findElement(By.xpath(String.format(editBtn, fname, lname)));

can be any
no. of times

sop(String.format("name is %s", "Sonu");
" value is %f", 32.67)

Notes

o/p:
name is Sonu
Value is

MONDAY

JANUARY

18

WK 04 • 018-348

Iterator i = set.iterator();
String currentWindowId;

while (i.hasNext())

{

 currentWindowId = i.next().toString();

 System.out.println(currentWindowId);

 if (!currentWindowId.equals(homePage))

{

 driver.switchTo().window(currentWindowId);

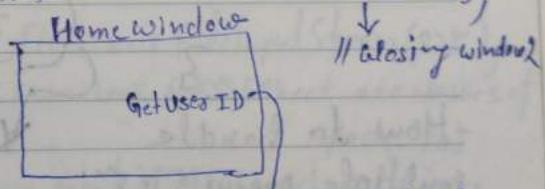
 Thread.sleep(5000);

 d.findElement(By.linkText("Know How")).click();

 d.switchTo().window(currentWindowId).close();

 }

 } } } } } } } } } } } } }



① getWindowsHandle()

 returns handle (or Id) of the current window

 return type: String

② getWindowsHandles()

 return a set of window handles (ids) which
 can be used to iterate over all opened windows

 return type: Set<String>

2016

19

TUESDAY
JANUARY

String pwin = d.getWindowHandle(); set.remove(pwin);
d.switchTo().window(set.iterator());

14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31

M T W T F S S M T W T F S

1 2
F 15 16
E 29
B M T

16

9

10

11

12

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

Handling Multiple windows!
019-347 WK 0

Set<String> set = d.getWindowHandles();

String window1 = (String) set.toArray()[0];

String window2 = (String) set.toArray()[1];

d.switchTo().window(window2);

for (String win : set)

{ dr.switchTo().window(win);

if ((dr.getTitle()).equals("STH"))

{

}

How to handle

multiple pop-ups in sh:

Convert the window Set<String>

Now using array index you can switch to

specific pop-up.

List<String> l = new AL(set);

String[] calc = toArray();

String[] win = (String) l.toArray();

set of type String.

if 1st row in excel is NOT

Aheadings:

String cellData = sheet.getRow(1).getCell(2).getStringCellValue();

then find 3rd cell ele of 2nd row

return type Object[]

row