

SELENIUM 2

Umadevi Balakrishnan

SELF INTRODUCTION



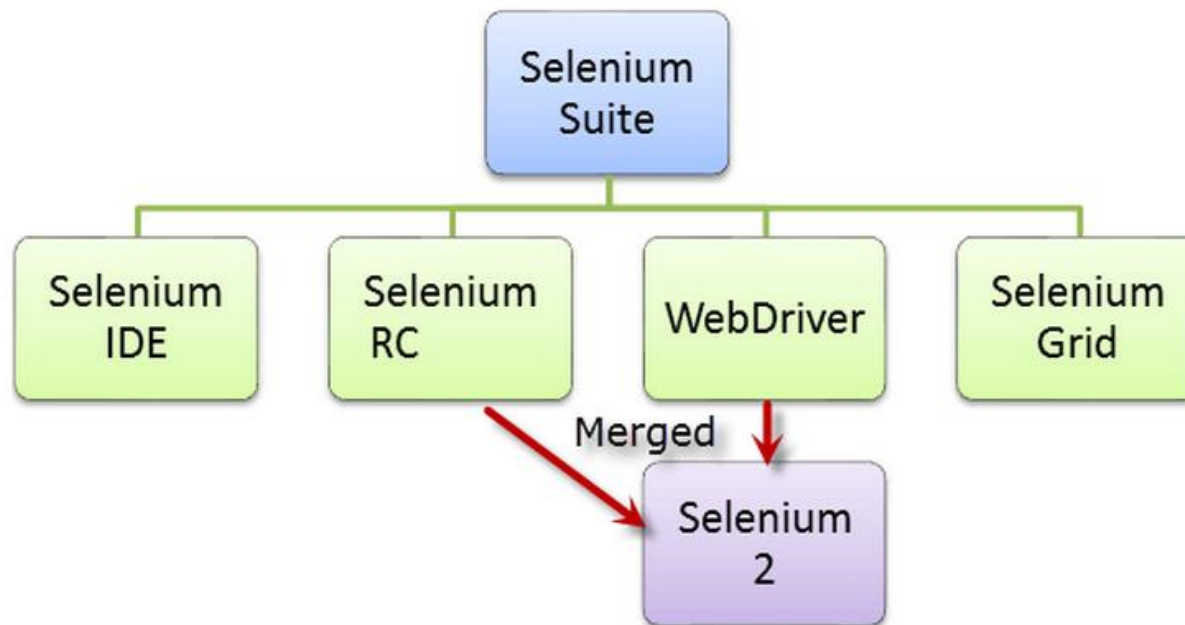
AGENDA

- Selenium IDE
- Selenium WebDriver



SELENIUM????

- Free (open source)
- Automated testing suite for web applications
- Works across different browsers and platforms



SELENIUM SUITE

○ Selenium IDE

- Firefox add-on that you can only use in creating relatively simple test cases and test suites.

○ Selenium Remote Control

- Also known as **Selenium 1**, is the first Selenium tool that allowed users to use programming languages in creating complex tests

○ Web Driver

- Allows the test scripts to communicate directly to the browser, thereby controlling it from the OS level

○ Selenium Grid

- Tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems.



WHY IS THE TOOL NAMED AS SELENIUM?



Mercury Interactive

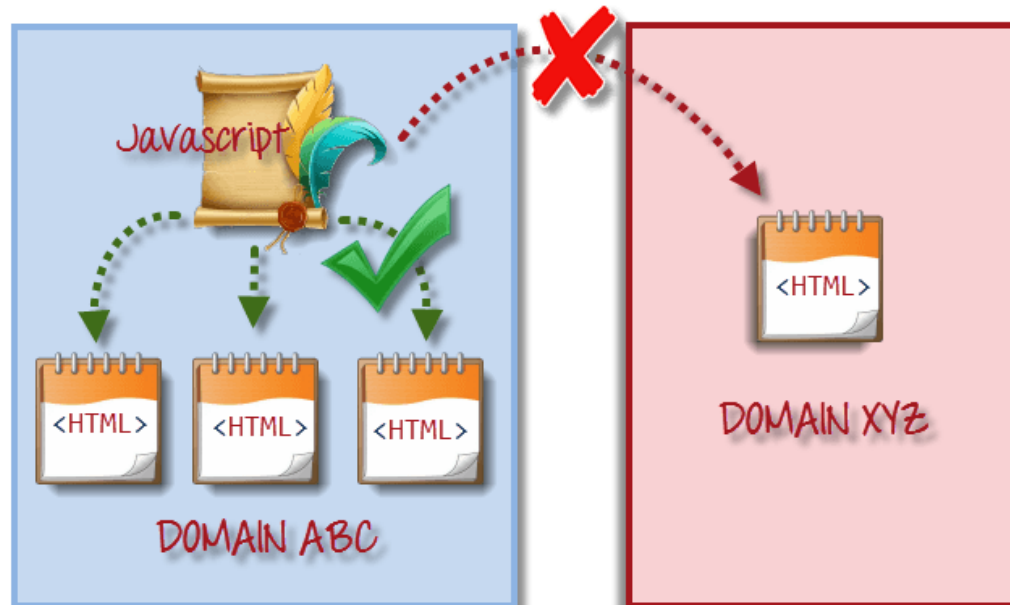


SELENIUM CORE

- Jason Huggins of ThoughtWorks in 2004 was working on the testing of an internal Time and Expenses application
- Created a JavaScript program that would automatically control the browser's actions
- He named this program as the "JavaScriptTestRunner" ,later named it Selenium Core



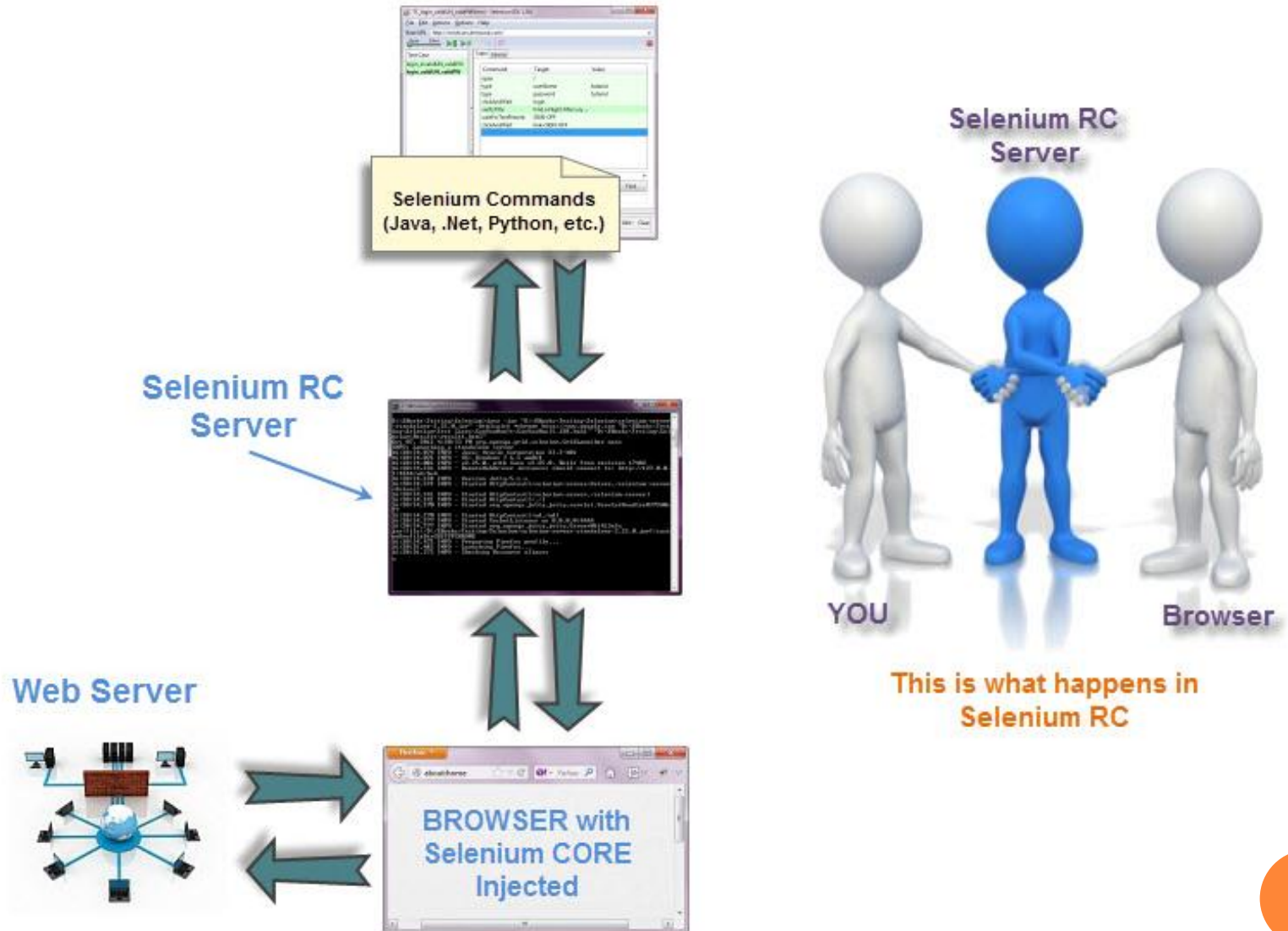
THE SAME ORIGIN POLICY ISSUE



- Testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the **same origin policy**



SELENIUM REMOTE CONTROL (SELENIUM RC)



SELENIUM GRID

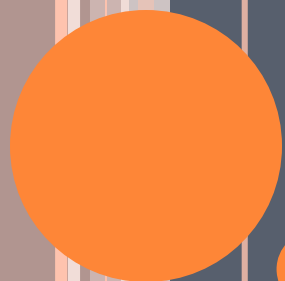
- To address the need of minimizing test execution times as much as possible



SELENIUM WEB DRIVER

- First cross-platform testing framework that could control the browser from the OS level





SELENIUM IDE

INSTALLATION OF SELENIUM IDE

- Navigate to <http://seleniumhq.org/download/>
- Choose Selenium IDE current version
- Selenium IDE would be installed as a firefox plug-in
- To launch firefox
 - Ctrl+Alt+S
 - Firefox menu button > Web Developer> Selenium IDE



INSTALLATION OF FIREBUG


- Firebug is a Firefox add-on that we will use to inspect the HTML elements of the web application under test
- It will provide the name of the element that the Selenese command would act upon
- Navigate to <https://getfirebug.com/downloads/> and click download link
- To launch fire-bug
 - Press F12
 - Click on the Firebug button on the upper right corner of the Firefox window



File Edit View History Bookmarks Tools Help

Mozilla Firefox Start Page

Firefox Search or enter address Search



mozilla

Search

Keep your brain sharp with our Puzzle Games Collection. Play one or play them all, now in the [Firefox Marketplace](#).

Console HTML CSS Script DOM Net Cookies FirePath

Top Window Highlight XPath: `//*[@id='searchText']` Parent: `/` Eval

```
<document>
  <html xmlns="http://www.w3.org/1999/xhtml" hasBrowserHandlers="true" snippetsURL="https://snippets.cdn.mozilla.net/4/Firefox/36.0.1/20150305021524/WINNT_x86-msvc/en-US/release/Windows_NT%206.1/default/default/" snippetsVersion="4" searchEngineName="Google">
    <html:head>
      <html:body dir="ltr">
        <html:div class="spacer"/>
        <html:div id="topSection">
          <html:div id="brandLogo"/>
          <html:div id="searchContainer">
            <html:form id="searchForm" name="searchForm" onsubmit="onSearchSubmit(event)">
              <html:div id="searchLogoContainer">
```

1 matching node

SELENIUM IDE???

- Simplest tool in the Selenium Suite
- A Firefox add-on that creates tests very quickly through its **record-and-playback** functionality
- It is effortless to install and easy to learn
- To be used only as a prototyping tool – not an overall solution for developing and maintaining complex test suites



Menu bar

Base URL
bar

Toolbar

Test case pane

Log/Reference/
UI-Element/
Rollup pane

Valid_login.html - Selenium IDE 1.9.1

File Edit Actions Options Help

Base URL http://newtours.demoaut.com/

Fast Slow

Editor

Test Case

Invalid_login

Valid_login

Runs: 2

Failures: 1

Command	Target	Value
open	/	
type	userName	tutorial
type	password	cause_of_failure
clickAndWait	login	
verifyTitle	Find a Flight: Mercu...	
clickAndWait	link=SIGN-OFF	
verifyTitle	Sign-on: Mercury T...	
clickAndWait	link=Home	
verifyTitle	Welcome: Mercury ...	

Command

Target

Value

Find

Log Reference UI-Element Rollup

Info Clear

[info] Changed test case

[info] Executing: |open | / | |

[info] Executing: |type | userName | tutorial |

[info] Executing: |type | password | cause_of_failure |

[info] Executing: |clickAndWait | login | |

[info] Executing: |verifyTitle | Find a Flight: Mercury Tours: | |

[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'

[info] Executing: |clickAndWait | link=SIGN-OFF | |

[error] Element link=SIGN-OFF not found

SELENIUM IDE SCRIPT

- First Script – Demo
 - <http://newtours.demoaut.com/>



EXPORT TESTCASE

- Test cases can be exported only to the following formats:
 - .cs (C# source code)
 - .java (Java source code)
 - .py (Python source code)
 - .rb (Ruby source code)



SELENIUM COMMANDS - SELENESE

Actions	<p>These are commands that directly interact with page elements.</p> <p>Example: the "click" command is an action because you directly interact with the element you are clicking at.</p> <p>The "type" command is also an action because you are putting values into a text box, and the text box shows them to you in return. There is a two-way interaction between you and the text box.</p>
Accessors	<p>They are commands that allow you to store values to a variable.</p> <p>Example: the "storeTitle" command is an accessor because it only "reads" the page title and saves it in a variable. It does not interact with any element on the page.</p>
Assertions	<p>They are commands that verify if a certain condition is met.</p> <p>3 Types of Assertions</p> <ul style="list-style-type: none">• Assert. When an "assert" command fails, the test is stopped immediately.• Verify. When a "verify" command fails, Selenium IDE logs this failure and continues with the test execution.• WaitFor. Before proceeding to the next command, "waitFor" commands will first wait for a certain condition to become true.<ul style="list-style-type: none">◦ If the condition becomes true within the waiting period, the step passes.◦ If the condition does not become true, the step fails. Failure is logged, and test execution proceeds to the next command.◦ By default, timeout value is set to 30 seconds. You can change this in the Selenium IDE Options dialog under the General tab.



COMMON COMMANDS

Command	Number of Parameters	Description
open	0 - 2	Opens a page using a URL.
click/clickAndWait	1	Clicks on a specified element.
type/typeKeys	2	Types a sequence of characters.
verifyTitle/assertTitle	1	Compares the actual page title with an expected value.
verifyTextPresent	1	Checks if a certain text is found within the page.
verifyElementPresent	1	Checks the presence of a certain element.
verifyTable	2	Compares the contents of a table with expected values.
waitForPageToLoad	1	Pauses execution until the page is loaded completely.
waitForElementPresent	1	Pauses execution until the specified element becomes present.



CREATE A SCRIPT MANUALLY WITH FIREBUG

- Demo



TYPES OF LOCATOR

- ID
- Name
- Link Text
- CSS Selector
 - Tag and ID
 - Tag and class
 - Tag and attribute
 - Tag, class, and attribute
 - Inner text
- DOM (Document Object Model)
 - getElementById
 - getElementsByName
 - dom.name
 - dom.index
- XPath



LOCATE ELEMENTS

- Locating by ID
- Locating by Name
- Locating by Name using Filters

Firebug

```
<b>  
  <font size="2" face="Arial, Helvetica, sans-serif">  
    <input type="radio" checked="" value="roundtrip" name="tripType">  
    Round Trip  
    <input type="radio" value="oneway" name="tripType">  
    One Way  
  </font>  
</b>
```

Flight Details

Type: ☒ Round Trip ☐ One Way

Passengers: 1 ▼

Departing From: Acapulco ▼

Mercury Tours Flight Finder page

Command	Target	Value
click	name=tripType value= oneway	

Command	click	
Target	name=tripType value= oneway	Find
Value		

CONTD...

○ Locating by Link Text



```
'mouseover')" onmouseout="changeStyle(th
is, 'mouseout')">
<a href="mercuryregister.php">REGISTER<
/a>
</td>
<td class="mouseout" width="73" height="
```

Command	Target	Value
	link=REGISTER	

Command	<input type="text" value="link=REGISTER"/>	<input type="button" value="Find"/>
Target		
Value		



○ Locating by CSS Selector

- Tag and ID
- Tag and class
- Tag and attribute
- Tag, class, and attribute
- Inner text



Method	Target Syntax	Example
By ID	<code>id= <i>id_of_the_element</i></code>	<code>id=email</code>
By Name	<code>name=<i>name_of_the_element</i></code>	<code>name=username</code>
By Name Using Filters	<code>name=<i>name_of_the_element</i> filter=<i>value_of_filter</i></code>	<code>name=tripType value=oneway</code>
By Link Text	<code>link=<i>link_text</i></code>	<code>link=REGISTER</code>
Tag and ID	<code>css=<i>tag</i>#<i>id</i></code>	<code>css=input#email</code>
Tag and Class	<code>css=<i>tag</i>.<i>class</i></code>	<code>css=input.inputtext</code>
Tag and Attribute	<code>css=<i>tag</i>[<i>attribute=value</i>]</code>	<code>css=input[name=lastName]</code>
Tag, Class, and Attribute	<code>css=<i>tag.class</i>[<i>attribute=value</i>]</code>	<code>css=input.inputtext[tabindex=1]</code>



ENHANCE A SCRIPT USING SELENIUM IDE

○ Verify Presence of an Element

- **verifyElementPresent** – returns TRUE if the specified element was FOUND in the page; FALSE if otherwise
- **verifyElementNotPresent** – returns TRUE if the specified element was NOT FOUND anywhere in the page; FALSE if it is present.

○ Verify Presence of a Certain Text

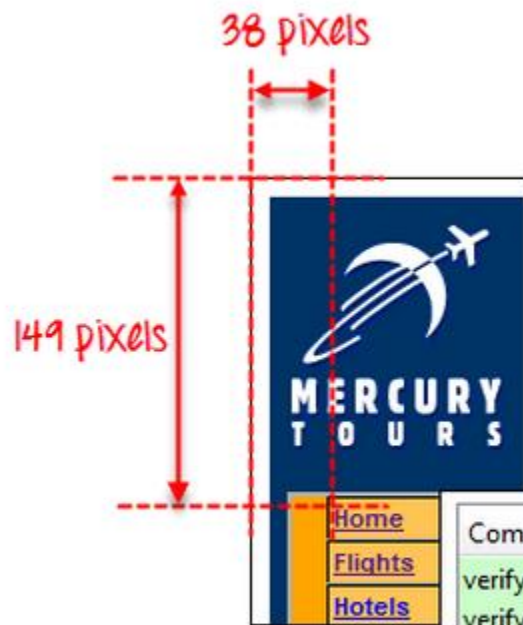
- **verifyTextPresent** – returns TRUE if the specified text string was FOUND somewhere in the page; FALSE if otherwise
- **verifyTextNotPresent** – returns TRUE if the specified text string was NOT FOUND anywhere in the page; FALSE if it was found



VERIFY SPECIFIC POSITION OF AN ELEMENT

- Selenium IDE indicates the position of an element by measuring (in pixels) how far it is from the left or top edge of the browser window.
 - **verifyElementPositionLeft** – verifies if the specified number of pixels match the distance of the element from the left edge of the page. This will return FALSE if the value specified does not match the distance from the left edge.
 - **verifyElementPositionTop** – verifies if the specified number of pixels match the distance of the element from the top edge of the page. This will return FALSE if the value specified does not match the distance from the top edge.





Command	Target	Value
verifyElementPositionLeft	link=Home	38
verifyElementPositionTop	link=Home	149



○ Wait commands

- clickAndWait
- typeAndWait
- selectAndWait

PASSED

Command	Target	Value
open	/	
type	name=userName	tutorial
type	name=password	tutorial
clickAndWait	name=login	
assertTitle	Find a Flight: Mercury Tours:	

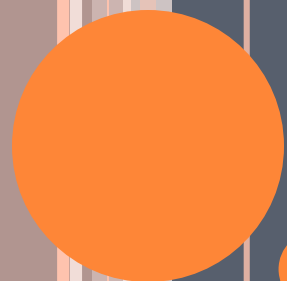
This PASSED because "clickAndWait" was used. Selenium IDE first waited for a new page to load before executing the "assertTitle" command.

FAILED

Command	Target	Value
open	/	
type	name=userName	tutorial
type	name=password	tutorial
click	name=login	
assertTitle	Find a Flight: Mercury Tours:	

This 2nd test case FAILED because "click" was used. Selenium IDE executed the "assertTitle" command without waiting for a new page to load.





WEBDRIVER

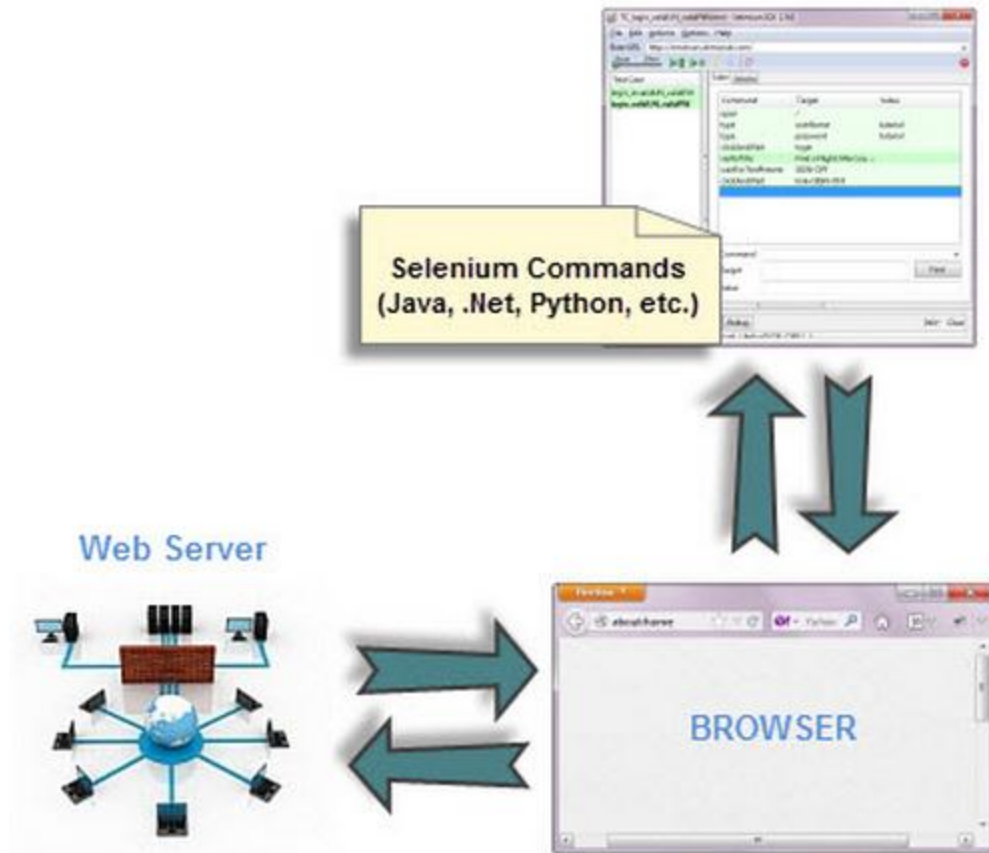
WHAT IS WEBDRIVER?

- It is a web automation framework that executes tests against different browsers, not just Firefox
- It also enables to use a programming language in creating the test scripts
- Conditional controls
 - if-then-else or switch-case
- Looping controls
 - While , do-while.
- Supports various programming languages
 - Java , .Net , PHP ,Python ,Perl ,Ruby





ARCHITECTURE



This is what happens in WebDriver



WEBDRIVER FEATURES

- WebDriver is a tool for testing web applications **across different browsers** using different programming languages
- WebDriver **allows you to use a programming language** of your choice in designing your tests
- WebDriver is **faster than Selenium RC** because of its simpler architecture
- WebDriver **directly talks to the browser** while Selenium RC needs the help of the RC Server in order to do so
- WebDriver's API is **more concise** than Selenium RC's
- WebDriver **can support HtmlUnit** while Selenium RC cannot



DRAWBACKS OF WEBDRIVER

- It cannot readily support new browsers, but Selenium RC can
- It does not have a built-in command for automatic generation of test results



SELENIUM WEBDRIVER - INSTALLATION

○ Install Java

- Java Development Kit (JDK). <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

○ Install Eclipse IDE for Java Developers

- Eclipse IDE – <http://www.eclipse.org/downloads/>

○ Download the Selenium Java Client Driver

- Java Client Driver – <http://seleniumhq.org/download/>

○ Configure Eclipse IDE with WebDriver



DRIVERS

- HTMLUnit and Firefox are two browsers that WebDriver can directly automate
- For other browsers, a separate program is needed called as the **Driver Server**
- A driver server is different for each browser.
 - For example, Internet Explorer has its own driver server which you cannot use on other browsers



DRIVER SERVERS

Browser	Name of Driver Server	Remarks
HTMLUnit	(none)	WebDriver can drive HTMLUnit without the need of a driver server
Firefox	(none)	WebDriver can drive Firefox without the need of a driver server
Internet Explorer	Internet Explorer Driver Server	Available in 32 and 64-bit versions. Use the version that corresponds to the architecture of your IE
Chrome	ChromeDriver	Though its name is just "ChromeDriver", it is in fact a Driver Server, not just a driver. The current version can support versions higher than Chrome v.21
Opera	OperaDriver	Though its name is just "OperaDriver", it is in fact a Driver Server, not just a driver.
PhantomJS	GhostDriver	PhantomJS is another headless browser just like HTMLUnit.
Safari	SafariDriver	Though its name is just "SafariDriver", it is in fact a Driver Server, not just a driver.



WEB DRIVER CODE

○ Importing Packages

- org.openqa.selenium.* – contains the WebDriver class needed to instantiate a new browser loaded with a specific driver
- org.openqa.selenium.firefox.FirefoxDriver – contains the FirefoxDriver class needed to instantiate a Firefox-specific driver onto the browser instantiated by the WebDriver class

○ Instantiating objects and variables

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

○ Launching a Browser Session

```
driver.get(baseUrl);
```



- Get the Actual Page Title

```
actualTitle = driver.getTitle();
```

- Compare the Expected and Actual Values

```
if (actualTitle.equals(expectedTitle)) {  
    System.out.println("Test Passed!");  
} else {  
    System.out.println("Test Failed!");  
}
```

- Terminating a Browser Session

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



LOCATING GUI ELEMENTS

- Id
- Xpath
- CSS Selector



HTML ATTRIBUTES

```
/**
Locating elements by ID
*/

<div id="toolbar">.....</div>

WebElement Ele = driver.findElement(By.id("toolbar"));

/**
Locating elements by name
*/

<input name="register" class="required" type="text"/>

WebElement register= driver.findElement(By.name("register"));

/**
Locating elements by LinkText
*/

<a href="seleniumhq.org">Downloads</a>
|
WebElement download = driver.findElement(By.linkText("Downloads"));
```



HTML ATTRIBUTES – CONTD...

```
/**
Locating an Element By TagName
**/
<select name="Language" size="1">
  <option value="11">Nov</option>
  <option value="12">Dec</option>
</select>

Select select = new Select(driver.findElement(By.tagName("select")));

select.selectByVisibleText("Nov");
select.selectByValue("11");

/**
Locating an Element By Class Name
**/
<div>
  <input type='text' class='firstName' value=''>
</div>

WebElement box = driver.findElement(By.className("firstName"));
```



CSS SELECTOR

```
/**
CSS Selector
**/
<div class="own">
    <input type="checkbox" name="cars" value="i10">i10<br/>
    <input type="checkbox" name="cars" value="Ford-class">Ford Classic<br/>
    <input type="checkbox" name="cars" value="eon">eon<br/>
</div>

<div class="wishes">
    <input type="checkbox" name="cars" value="Skoda-Superb">Superb<br/>
    <input type="checkbox" name="cars" value="Audi-A8">A8<br/>
</div>

WebElement wishes = driver.findElement(By.cssSelector("div.wishes"));
//Select
WebElement superb = wishes.findElement(By.xpath("(input[@name='cars'])[1]"));
```



XPATH

```
/**
```

```
Xpath using multiple attrib values
```

```
*/
```

```
<input type="radio" name="Skill" value="no" id="random-9749570"><br>
```

```
By.xpath("//input[@name='Skill' and @id='random-9749570']");
```

```
/**
```

```
CSS Selectors nth-child
```

```
*/
```

```
// select first column in second row of the first table
```

```
table:nth-child(1) tr:nth-child(2) td:nth-child(1)
```



Questions???

