Portnov Computer School presents:

Selenium
Web Test Tool Training

Discover the automating power of Selenium

Presented by:

Kangeyan Passoubady (Kangs)

3

Lesson

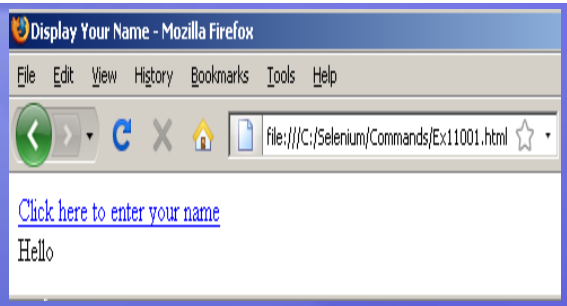
Selenese Commands

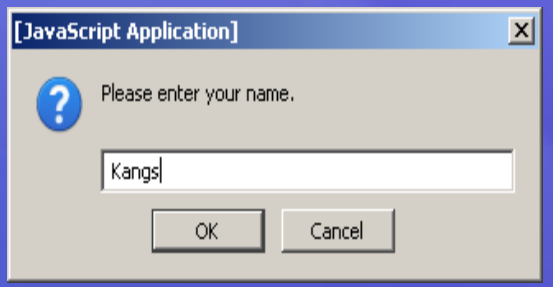
Variable Substitution #1

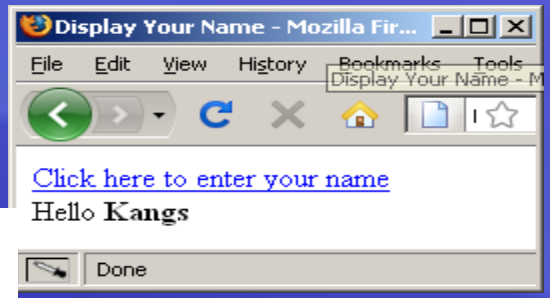
- You can store any value into a variable using selenium accessors.
- Variable substitution provides a simple way to include a previously stored variable in a command parameter
- If you give the variable name as intAmount. To use the variable anywhere else use ${intAmount}. This will substitute the actual value in the place of ${intAmount}.
- Use **echo** ${intAmount} to display the values in the Log.
- Any of the selenium accessors can be used to store a value.
- Some of the key accessors are
  - storeText, storeValue, storeCookie, storeAlert and storeEval

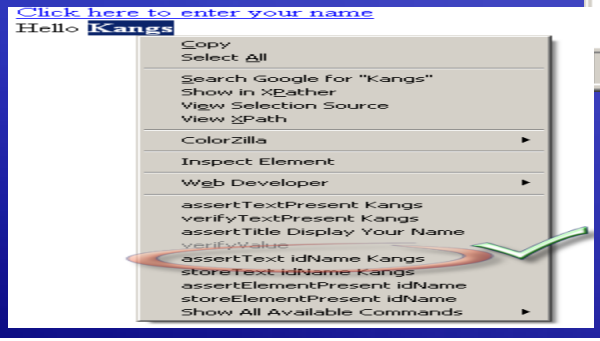
Variable Substitution

1. Open the HelloWorld.html in Firefox
2. Create a test case by opening Firefox→Tools→Selenium IDE (by default it is in Recording mode)
3. Update the base URL of Selenium IDE with the URL in the Firefox address
4. Click on the link “Click here to enter your name”, when prompter for your name, enter your name.
5. Your name will be displayed in next line.
6. Highlight your name Right click and select the assertText, then Stop the recording in Selenium IDE.
7. Rerun the test case in Selenium IDE



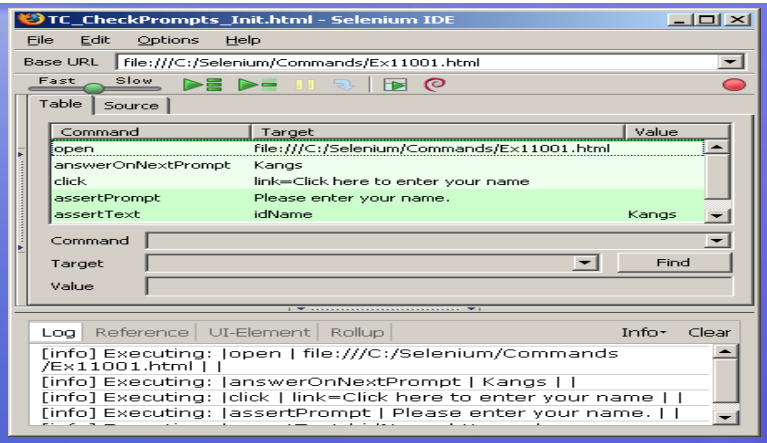






Variable Substitution

1. Your test case will look like in Selenium IDE as shown in the figure right side
2. Try to run multiple time and make sure your test case is passed without any failures.
3. Now save the test case as “Recorded_TestCase_HelloWorld.html”
4. Double click and open the “Recorded_TestCase_HelloWorld.html” either in IE or Firefox.
5. You can see all the commands in an HTML table.

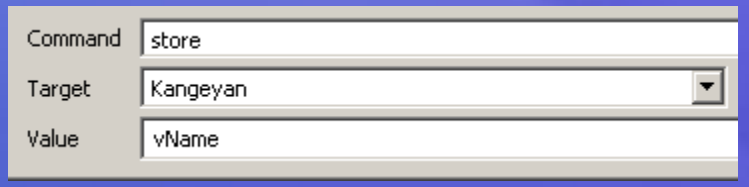


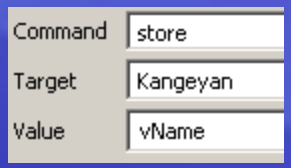
TC_CheckPrompts_Init		
open	file:///C:/Selenium/Commands/Ex11001.html	
answerOnNextPrompt	Kangs	
click	link=Click here to enter your name	
assertPrompt	Please enter your name.	
assertText	idName	Kangs

Your answerOnNextPrompt should be before than the click command.
Make sure it precedes otherwise your test case will fail.

Variable Substitution #5

1. Now we'll try to do the same exercise using variable substitution
2. In the Selenium IDE, select "answerOnNextPrompt", Right Click and Select a new command
3. Select the command "store"
4. Target "Kangeyan"
5. Value "vName"
6. Store | Kangeyan | vName
7. Underneath insert a new command and provide the below values
8. echo | ${vName}
9. Right click and copy the "echo" command, and paste before the assertText command line





Variable Substitution

#6

tc_EX11001_CheckPrompts		
open	file:///C:/Selenium/Commands/Ex11001.html	
answerOnNextPrompt	Kangs	
click	link=Click here to enter your name	
assertPrompt	Please enter your name.	
assertText	idName	Kangs

Your recorded test case will look something similar to this one

Change something similar to this one

tc_EX11001_CheckPrompts_Mod		
open	file:///C:/Selenium/Commands/Ex11001.html	
store	Kangeyan	vName
echo	${vName}	
answerOnNextPrompt	${vName}	
click	link=Click here to enter your name	
assertPrompt	Please enter your name.	
echo	${vName}	
assertText	idName	${vName}

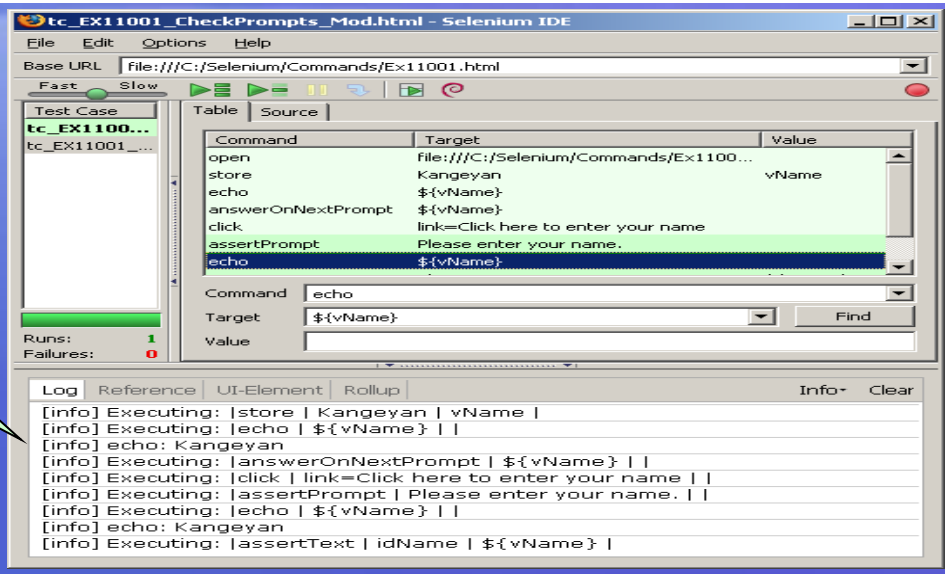
See how the ${vName} is used in multiple places

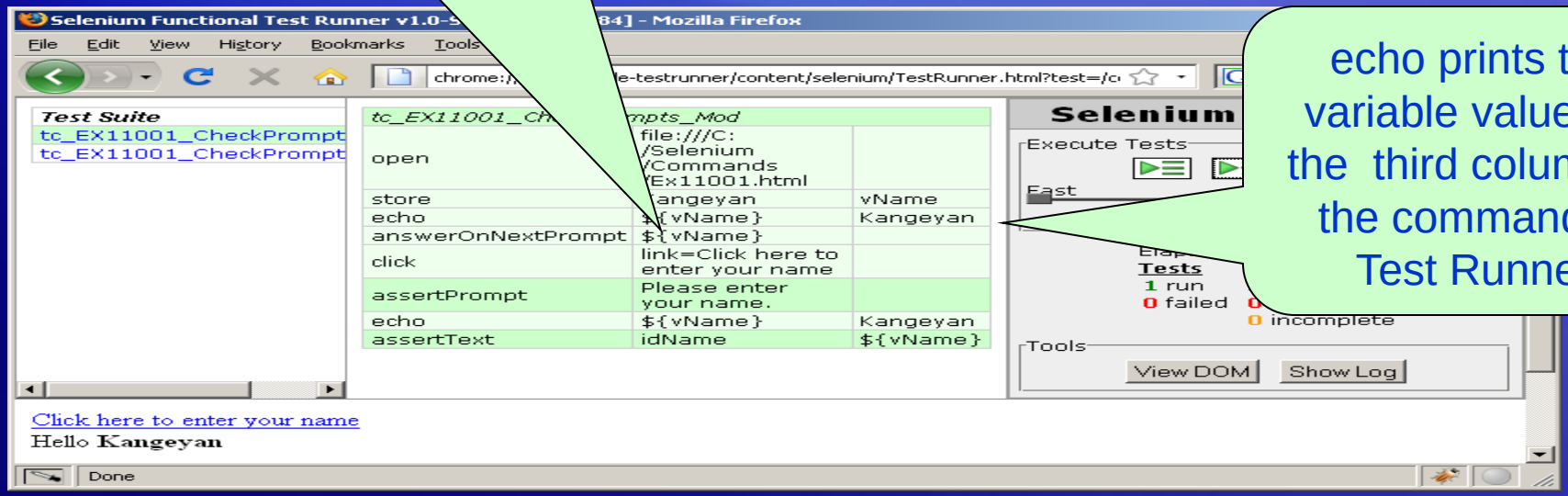
Variable Substitution

echo shows the variable values in the Log

See the usage of ${vName} in answerOnNextPrompt and assertText

echo prints the variable values in the third column of the command in Test Runner





storedVars #1

- All the variables are internally stored in a map named “storedVars”.
- storedVars allows you to access the ${varName} using the “varName” key within the map.
- Using storedVars you can reference the values within Javascript Evaluation code.

What is a Map?

Maps provide a more general way of storing elements. An object that maps keys to values. A map cannot contain duplicate keys (duplicate variable names); The Map collection type allows to store pairs of elements, termed "keys" and "values", where each key maps to one value. Here Keys refers Selenium variable Names and values refers to their values.

storedVars #2

1. Open URL: http://mail.yahoo.com
2. Type username: <username>
3. Type password:<passwd>
4. Press Sign In button
5. AssertTextPresent <username>
6. Press SignOut
7. waitForTextNotPresent <username>

storedVars #3

tc_yahoo_sVars		
open	https://login.yahoo.com/config/login_verify2?&.src=ym	
type	username	kpassoubady
type	passwd	*******
clickAndWait	.save	
assertTextPresent	kpassoubady	
click	_test_sign_out	
waitForTextNotPresent	kpassoubady	

Your recorded test case will look something similar to this one

Change something similar to this one

tc_yahoo_sVars_Mod		
open	http://mail.yahoo.com	
store	kpassoubady	vUName
type	username	javascript{storedVars['vUName']}
type	passwd	********
clickAndWait	.save	
assertTextPresent	javascript{storedVars['vUName']}	
click	_test_sign_out	
waitForTextNotPresent	javascript{storedVars['vUName']}	

See how the storedVars is used in multiple places

JavaScript Evaluation #1

- JavaScript evaluation allows full power of JavaScript code in constructing the Command Parameter
- JavaScript snippet can be given using the following syntax.
  - Javascript { <code snippet goes here> }
  - Javascript keyword is optional
  - The code given is treated as a JavaScript code and executed.
  - storedVars Map can be used to access the previously stored variables
  - Variable substitution should be handled carefully within the JavaScript code.

JavaScript Evaluation #2

1. You can use any of the following **Eval** commands
   - assertEval, assertNotEval, VerifyEval, verifyNotEval, waitForEval, waitForNotEval, storeEval
2. You can use any of the following **Expression** commands
   - assertExpression, assertNotExpression,verifyExpression, verifyNotExpression, waitForExpression, waitForNotExpression, storeExpression, store and WaitForCondition

Barnes and Nobles Problem #3

- Open a specific URL (http://www.barnesandnoble.com/)
- Search for a specific text (“Javascript”) in #1 page
- Sort by “Prize^{v}”
- How do you check “Online Price: $$$” is in sorted order?

Answer #4

- In this case I have decided to check the first two Amounts displayed on that page are in the ascending order.
- The first value is A, the second value is B
- If A <= B then we assume the first two listed prices are in ascending order.
- Now get the third value C
- If B <= C then we assume that A, B and C are in ascending order. (i.e., A <= B <=C)

Answer

#5

Barnes & Noble Price List		
open	http://www.barnesandnoble.com/index.asp	
type	search-input	javascript
clickAndWait	quick-search-button	
clickAndWait	link=Price	
storeText	//div[@id='bs-center-col']/div[3]/div[1]/div[2]/div/div/div/ul[1]/li[2]/strong	T1
echo	${T1}	
storeText	xpath=id('bs-center-col')/div[3]/div[3]/div[2]/div/div/div/ul[1]/li[1]/strong	T2
echo	${T2}	
storeEval	var A= Number("${T1}".substr(1));var B=Number("${T2}".substr(1)); var C=false; if (A<B) C=true;C	T3
echo	${T3}	
storeText	//div[@id='bs-center-col']/div[3]/div[5]/div[2]/div/div/div/ul[1]/li[1]/strong	T4
storeEval	var A= Number("${T2}".substr(1));var B=Number("${T4}".substr(1)); var C=false; if (A<B) C=true;C	T5
echo	${T5}	

Answer

#6

Xpath for first amount

storeText in a variable T1

Echo Prints the values in the Log

Barnes & Noble Price List		
open	http://www.barnesandnoble.com/index.asp	
type	search-input	jav
clickAndWait	quick-search-button	
clickAndWait	link=Price	
storeText	//div[@id='bs-center-col']/div[3]/div[1]/div[2]/div/div/div/ul[1]/li[2]/strong	T1
echo	${T1}	
storeText	xpath=id('bs-center-col')/div[3]/div[3]/div[2]/div/div/div/ul[1]/li[1]/strong	T2
echo	${T2}	
storeEval	var A= Number("${T1}".substr(1));var B=Number("${T2}".substr(1)); var C=false; if (A<B) C=true;C	T3
echo	${T3}	
storeText	//div[@id='bs-center-col']/div[3]/div[5]/div[2 div/div/ul[1]/li[1]/strong	
storeEval	var A= Number("${T2}".substr(1));var B=Nu "${T4}".substr(1)); var C=false; if (A<B) C=true;C	
echo	${T5}	

Substr(1) Removes '$' and only takes the numeric value.

Number converts the string type into a Number type

Answer

#7

1. Now we can store “true” in T6 and compare with value with T5
2. If that comparison yields true then it is Ascending Order

Store T6 with true

Check whether T5 and T6 both contains true

echo	${T4}	
storeEval	var B= Number("${T2}".substr(1));var [illegible]Number("${T3}".substr(1)); var E=fals[illegible]<D) E=true;E	T5
echo	${T5}	
store	true	T6
echo	${T6}	
assertExpression	${T5}	${T6}
echo	${T6}	
storeEval	var isSorted = new Boolean("${T6}"); var strResult ='Not in Sorted Order'; if (isSorted) strResult='Ascending Order'; strResult	vSorted
echo	${vSorted}	

Now echo Sorted or Not Sorted

Alert #1

- storeAlert(seleniumVariableName)
  - Checks for JavaScript Alert, stores the alert message. If no alert generated then throws an exception.
  - Getting an alert has the same effect as manually clicking OK.
  - If an alert is generated but you do not get/verify it, then the next Selenium action will fail.
  - JavaScript alerts will NOT pop up a visible alert dialog.
  - Selenium does NOT support JavaScript alerts that are generated in a page's onload() event handler. In this case a visible dialog WILL be generated and Selenium will hang until someone manually clicks OK.
- Returns:
  - The message of the most recent JavaScript alert.
- Other Alert commands are:
  - assertAlert (pattern), assertNotAlert (pattern), verifyAlert (pattern)
  - verifyNotAlert (pattern),waitForAlert (pattern),waitForNotAlert (pattern)

AlertPresent #2

- verifyAlertPresent()
  - The best way to check the alerts are using this command
  - This command never throws an exception
- Returns:
  - True or False.
- Other AlertPresent Commands are:
  - storeAlertPresent (seleniumVariableName)
  - assertAlertPresent ( )
  - assertAlertNotPresent ( )
  - verifyAlertNotPresent ( )
  - waitForAlertPresent ( )
  - waitForAlertNotPresent ( )

Alert

- Download the ClickAlert.html under the Alert section
  - Record a test
  - Are you able to test it?
    - (This is do-able)
- Download the OnLoadAlert.html
  - Are you able to test it?
    - (This should be challenging)

Alert

Test Case Alert Onload Event	
open	file:///C:/Selenium/Commands/Ex1.html
assertTextPresent	Displaying the alert Message

On load Alert will not be sensed by Selenium. You need to manually press “Ok” to continue the test.

Selenium able to intercept the button Click Alert.

Test Case Alert Button Click	
open	file:///C:/Selenium/JavaScriptEx/Ex1.html
click	//input[@value='Click and get the Welcome Message']
assertAlert	Welcome to Portnov!

goBack #1

- **goBack** and **goBackAndWait** are the two commands simulates a user clicking on the “back” button of the browser.
- Download the SelectAWebSite.html under Exercises.
- Record the test as listed below:
  - Select Google, after going to Google assertTitle then go back
  - Select Portnov, after going to PortNov assertTitle then go back
  - Select Microsoft, after going to MicroSoft assertTitle then go back
  - Select Yahoo, after going to MicroSoft assertTitle then go back
- Run the test
  - Why it fails?
  - How do you fix it?

goBackAndWait #2

Test Case Go Back And Wait_Mod		
open	file:///C:/Selenium/JavaScriptEx/Ex8.html	
select	OptWeb	label=Google
clickAndWait	btnGo	
assertTitle	Google	
select	OptWeb	label=Portnov
clickAndWait	btnGo	
assertTitle	Career Training & Career Change: Software Testing and Software QA (Quality Assurance) @ Portnov Computer School	
select	OptWeb	label=Microsoft
clickAndWait	btnGo	
assertTitle	Microsoft Corporation	
select	OptWeb	label=Yahoo
clickAndWait	btnGo	
assertTitle	Yahoo!	

Try goBack, and goBackAndWait, which one works well?

Test Case Go Back And Wait		
open	file:///C:/Selenium/JavaScriptEx/Ex8.html	
select	OptWeb	label=Google
clickAndWait	btnGo	
assertTitle	Google	
goBackAndWait		
select	OptWeb	label=Portnov
clickAndWait	btnGo	
assertTitle	Career Training & Career Change: Software Testing and Software QA (Quality Assurance) @ Portnov Computer School	
goBackAndWait		
select	OptWeb	label=Microsoft
clickAndWait	btnGo	
assertTitle	Microsoft Corporation	
goBackAndWait		
select	OptWeb	label=Yahoo
clickAndWait	btnGo	
assertTitle	Yahoo!	

Add goBackAndWait, see the results

waitForPopup #1

- **waitForPopUp (windowID,timeout)** and **selectWindow (windowID)** are the two commands allows you to test the Popup Windows.
- selectWindow selects a specific Popup, use null to select Parent window.
- Download Ex1.html to Ex3.html under wait for Popup, Open CreatePopUps.html in Firefox browser.
- Record the test as listed below:
  - Click Create Windows button
  - Select win1, click the button “Click and get the Welcome Message”, minimize win1
  - Select win3, select any option, press “Submit” button
  - Go back to the parent window, press “close button”
- Run the test
  - Is it fails?
  - How do you fix it?

waitForPopup #2

Reviewed Test Case Popup		
open	file:\\C:\2009 Selenium\Day 3\Ex\CreatePopUps.html	
click	winBut	
waitForPopUp	win1	30000
waitForPopUp	win2	30000
waitForPopUp	win3	30000
selectWindow	name=win1	
click	//input[@value='Click and get the Welcome Message']	
assertAlert	Welcome to Portnov!	
selectWindow	name=win3	
click	//input[4]	
click	Submit	
assertConfirmation	Are you sure you want to submit this answer?	
assertAlert	submitted	
selectWindow	null	
click	//input[@name='winBut' and @value='Close Windows']	

Fix the code something similar to shown here

To select Parent Window use “null”

Navigator Properties #1

- To get Browser information, you can use navigator object
  - The common properties of **navigator** object is like
    - appName
    - appCodeName
    - appEnabled
    - JavaEnabled
    - language
    - cookieEnabled
    - navigator.userAgent
    - navigator.plugins
    - navigator.platform
    - navigator.mimeTypes
  - The common properties of **browserVersion** object is like
    - browserVersion.name
    - browserVersion.browser
    - browserVersion.isFirefox

ArrayTypes: has multiple values

Not all the browsers support these properties

Navigator Properties #2

- Download "Test Case Navigator Properties.html" from the Exercises Section
- Run the Test Case
- Look at the Selenium IDE Log
- You can See all the Navigator Properties

Navigator Properties #3

Test Case Navigator Properties		
echo	javascript{navigator.appName}	
echo	javascript{navigator.appCodeName}	
echo	javascript{navigator.appVersion}	
echo	javascript{navigator.javaEnabled()}	
echo	javascript{navigator.language}	
echo	javascript{navigator.cookieEnabled}	
echo	javascript{navigator.userAgent}	
echo	javascript{navigator.platform.length}	
echo	javascript{navigator.mimeTypes.length}	
echo	javascript{browserVersion.name}	
echo	javascript{browserVersion.browser}	
echo	javascript{browserVersion.isFirefox}	
storeEval	var jLen = navigator.plugins.length;jStr=jLen+ " Plugin (s)"+" "+"Name | Filename | description"+" ";for(var i=0; i<jLen; i++) {jStr+=navigator.plugins[i].name+" | "+navigator.plugins[i].filename+" | "+navigator.plugins[i].description+" ";}	selPlugIn
echo	${selPlugIn}	

```
[info] Executing: |echo | javascript{navigator.appName} | |
[info] echo: Netscape
[info] Executing: |echo | javascript{navigator.appCodeName} | |
[info] echo: Mozilla
[info] Executing: |echo | javascript{navigator.appVersion} | |
[info] echo: 5.0 (Windows; en-US)
[info] Executing: |echo | javascript{navigator.javaEnabled()} | |
[info] echo: true
[info] Executing: |echo | javascript{navigator.language} | |
[info] echo: en-US
[info] Executing: |echo | javascript{navigator.cookieEnabled} | |
[info] echo: true
[info] Executing: |echo | javascript{navigator.userAgent} | |
[info] echo: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.3)
Gecko/2008092417 Firefox/3.0.3 (.NET CLR 3.5.30729)
[info] Executing: |echo | javascript{navigator.platform.length} | |
[info] echo: 5
[info] Executing: |echo | javascript{navigator.mimeTypes.length} | |
[info] echo: 62
[info] Executing: |echo | javascript{browserVersion.name} | |
[info] echo: Netscape
[info] Executing: |echo | javascript{browserVersion.browser} | |
[info] echo: Firefox
[info] Executing: |echo | javascript{browserVersion.isFirefox} | |
[info] echo: true
```

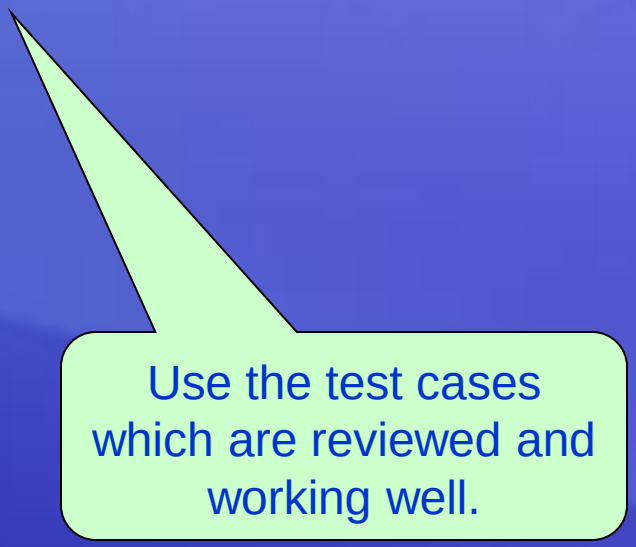
Create a string to display the object properties

Look at the log for results

Test Suite on Commands #1

Create a Test Suite based on the previous exercises:

- We'll use the following three test cases to create one Test Suite (Test Script Package)
  - Test Case Alert Button Click.html
  - Test Case Navigator Properties.html
  - Reviewed Test Case Popup.html

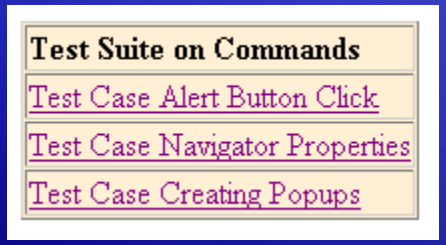


Test Suite on Commands #2

You can download the test suite from Exercises Section:

Test Suite on Commands.html

```
<tr><td><a href="Test Case Alert Button Click.html">Test Case Alert Button Click</a></td></tr>
<tr><td><a href="Test Case Navigator Properties.html">Test Case Navigator Properties</a></td></tr>
<tr><td><a href="Reviewed Test Case Popup.html">Test Case Creating Popups</a></td></tr>
```



Use the test cases which are reviewed and working well.

Test Suite on Commands #3

To CREATE a test suite from the scratch do the following activities:

1. Open “Test Case Alert Button Click.html” in the Selenium IDE
2. Select File → Save Test Suite As “Test Suite on Commands.html”
3. Open the “Test Suite on Commands.html” in notepad
4. Add the below HTML codes before the </tbody>
5. Select File → Save to save the test case.

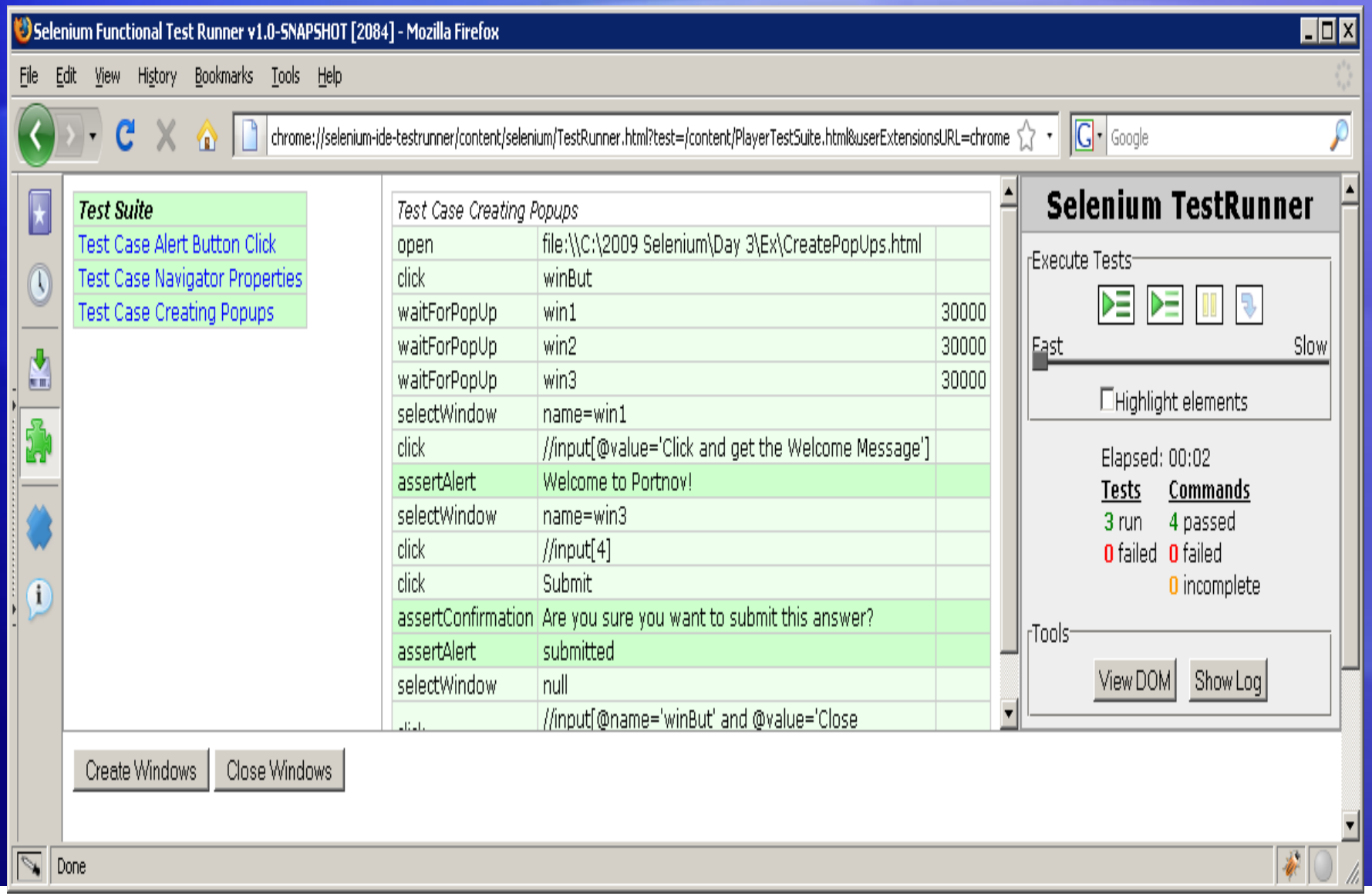
```
<tr><td><a href="Test Case Navigator Properties.html">Test Case Navigator Properties</a></td></tr>
<tr><td><a href="Reviewed Test Case Popup.html">Test Case Creating Popups</a></td></tr>
```

All the test cases in the same folder

Creating Test Suite From Scratch #4

- Open the Firefox browser
- Open the Selenium IDE
- Select File → Open Test Suite
- Choose the “Test Suite on Commands.html” from your local folder
- Press the TestRunner icon to run the test suite
- It will show the Test Suite in the firefox browser
- Click the “Run All Tests” button to run all the test cases
- Click the “Run the Selected Tests” button to run a specific test case

Run Test Suite on Commands #5



Exercise 1 – DevRy Popup Window

- Create a test case with the following steps:

1. Open http://www.devry-degrees.com
2. At the bottom of the page, you will see 5 footer links
3. Click a link at a time, this will create a popup
4. AssertTitle on the Popped Up window
5. Close the Popup
6. Click on the next link in the parent window, this will create a popup
7. AssertTitle on the Popped Up window
8. Close the Popup, Continue the same for all the 5 links

Exercise 1 – Questions

- Answer the below Questions once you complete your Devry Popup Window Test Case:
  - What are the Popup's link works without any problem?
  - Why one of the Popup's assertTitle produces an Error?