

# Selenium Tutorial



# What is Selenium?

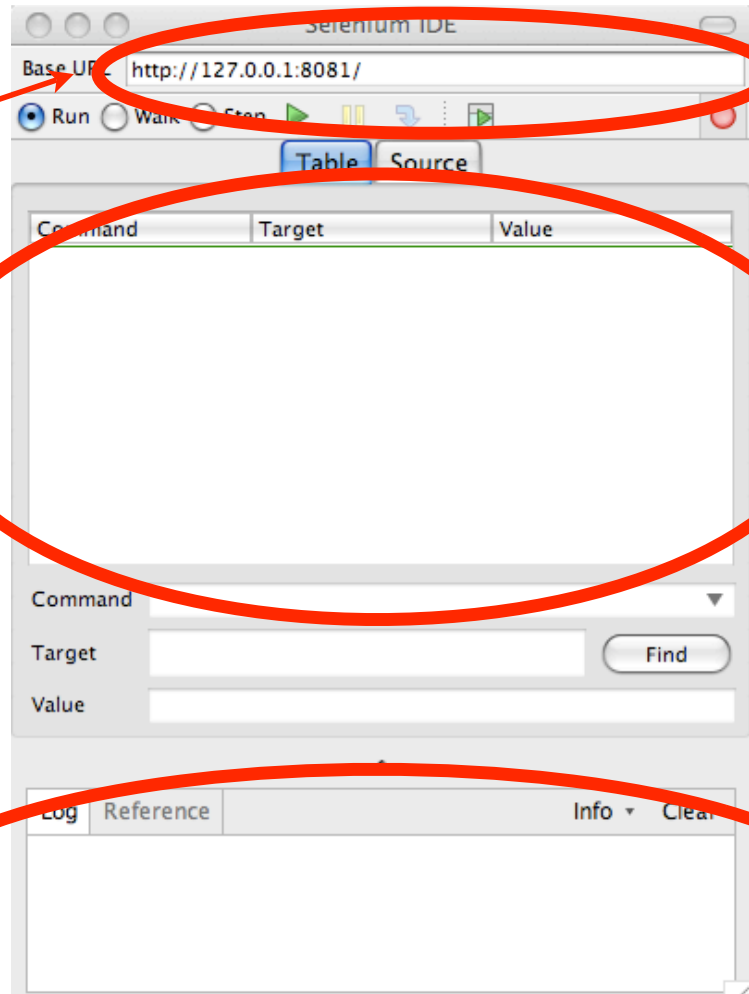
- Javascript framework that runs in your web-browser
- Works anywhere Javascript is supported
- Hooks for many other languages
  - Java, Ruby, Python
- Can simulate a user navigating through pages and then assert for specific marks on the pages
- All you need to really know is HTML to start using it right away

# Where to get it?

- You can use Selenium-Core and customize everything
- But it is easier to just get a Firefox plug-in “Selenium-IDE” that helps you “record” test cases
- You can record how an app is being used and then play back those recordings followed by asserts
- Get everything at: [www.openqa.org/selenium/](http://www.openqa.org/selenium/)

# Selenium IDE

The root of web application you want to test



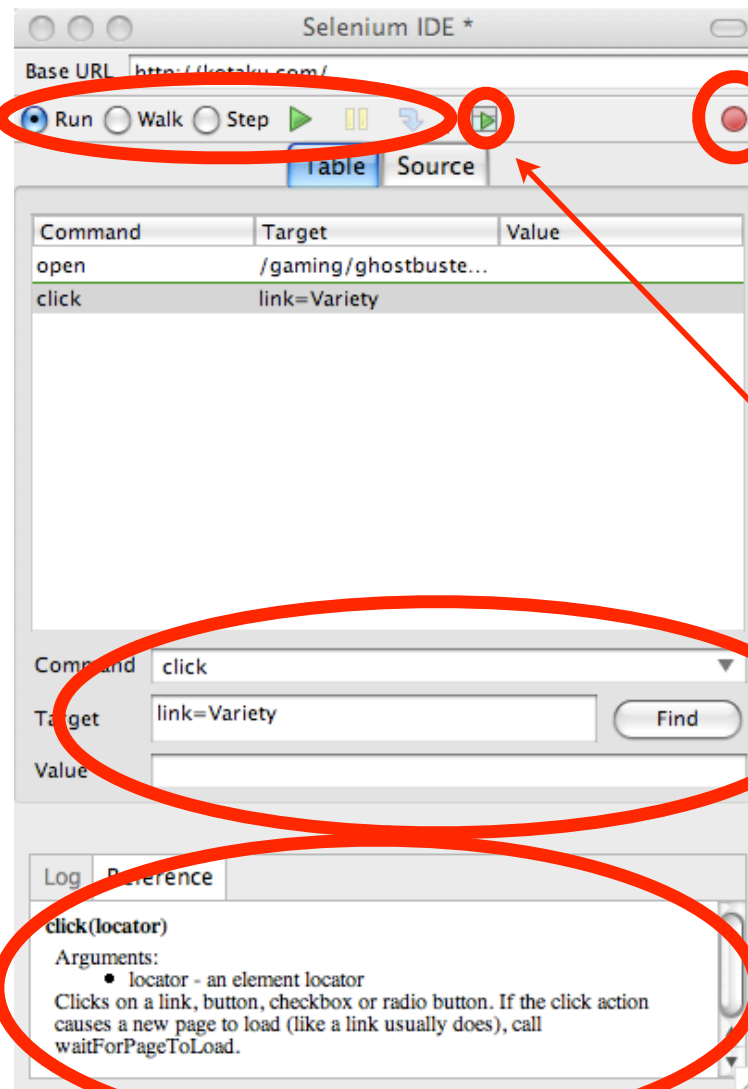
The list of actions in the actual test case to execute

The log of the events that were executed, including any errors or warning that may have occurred

# Selenium IDE

Execution  
Commands

Record test  
actions



Try the test in  
the Web based  
TestRunner

Reference of the  
currently selected  
command

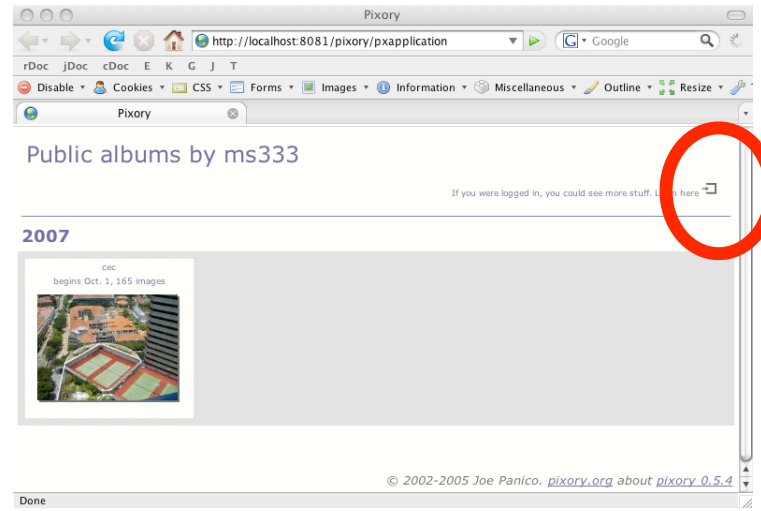
Specify commands,  
including asserts

# Test Creation Demo

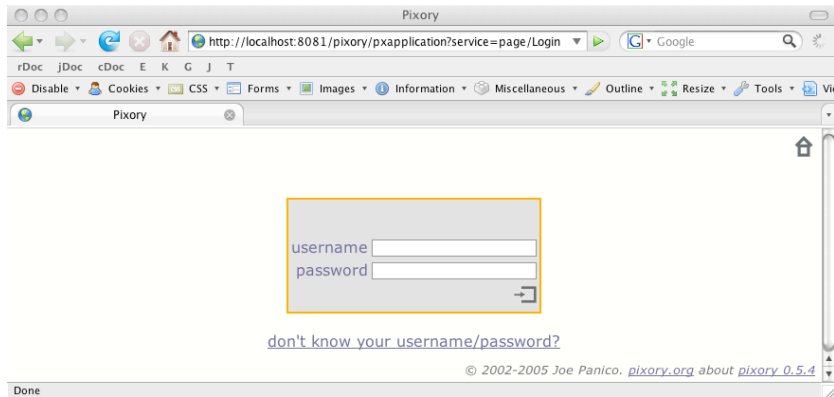
- Create test case to log into the gallery
- Create test case to log out of the gallery

```
java -- zsh -- Homebrew -- 3
[ms333 @ n1-12-125 : 12:48-07-11-15]
~/projects/classes/running/v_and_v/hw3/pix
$java -Djava.awt.headless=true -jar pixory.jar
PXLogConfiguration.INFO - Configured from file log/log.configuration.xml
█
```

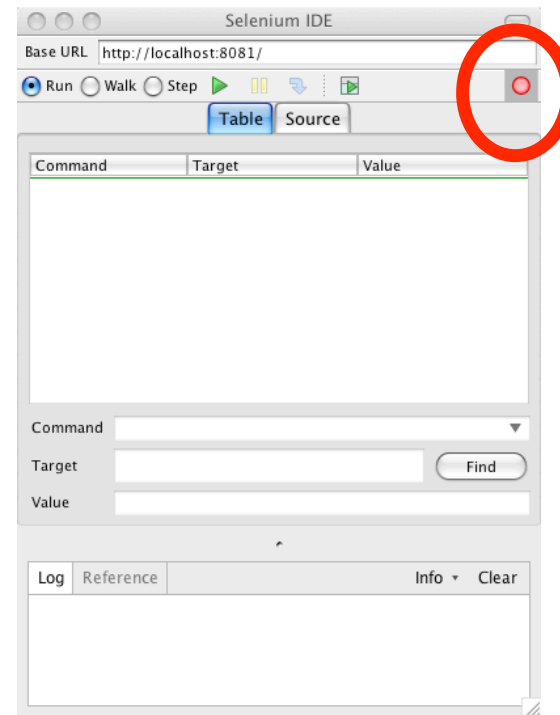
Start Pixory



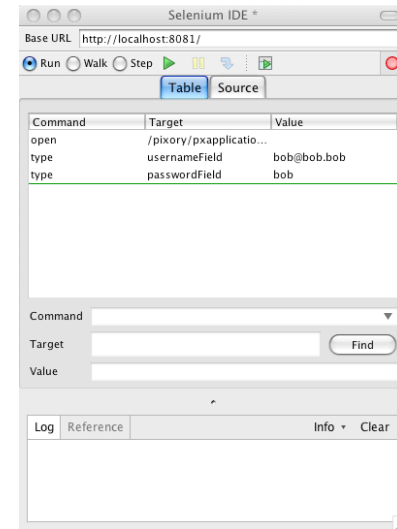
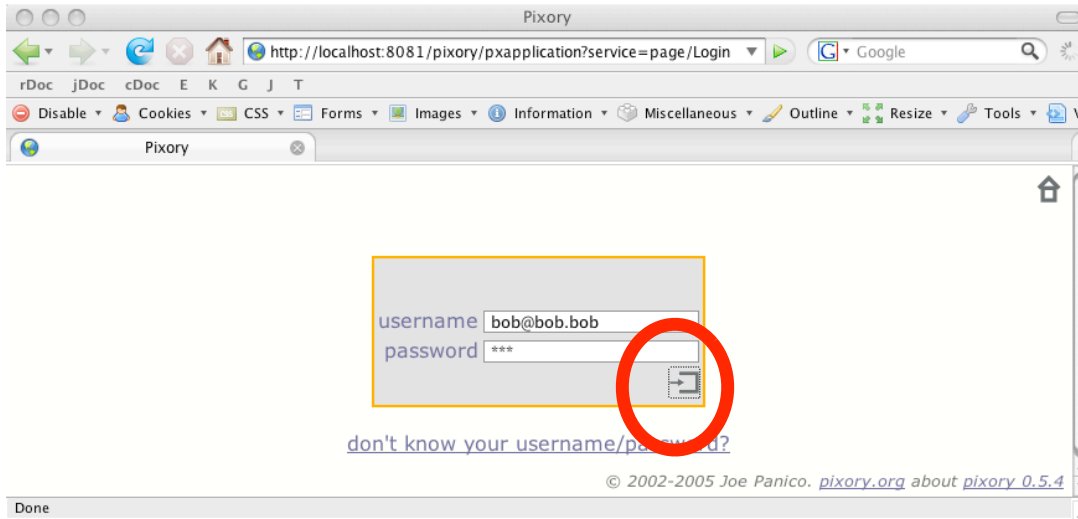
Connect to the Server



Go to the Login Screen

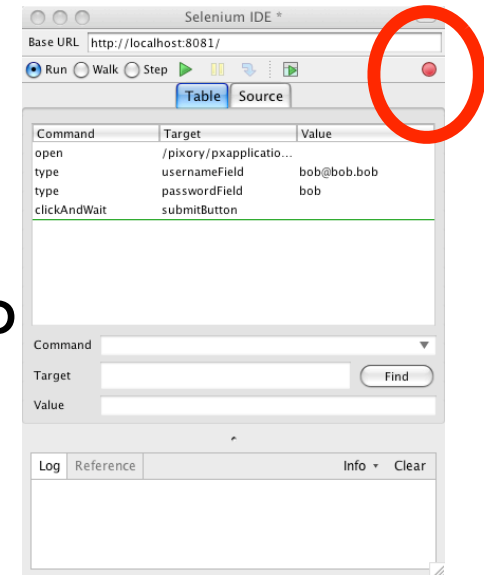
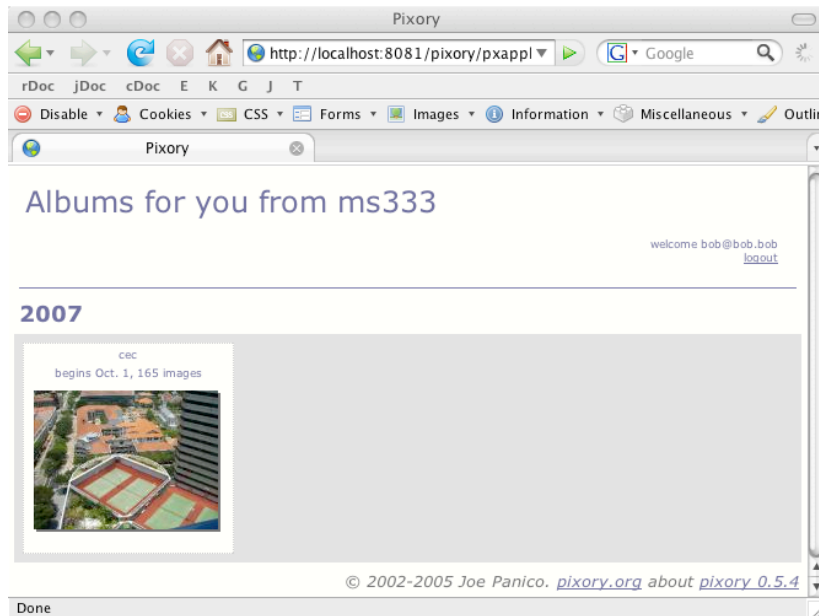


Hit the Record Button



Type in Username and Password

IDE should update



Hit record again to stop

Hit submit



Selenium IDE \*  
Base URL http://localhost:8081/

Run Walk Step

Table Source

Command	Target	Value
open	/pixory/pxapplicatio...	
type	usernameField	bob@bob.bob
type	passwordField	bob
clickAndWait	submitButton	
assertTextPresent	welcome bob@bob.bob	

Command: assertTextPresent  
Target: welcome bob@bob.bob  
Value:

Log Reference

**assertTextPresent(pattern)**  
Generated from **isTextPresent(pattern)**  
Arguments:  
• pattern - a pattern to match with the text of the page  
Returns:  
true if the pattern matches the text, false otherwise

Add `assertTextPresent` and type the text to search for

Selenium IDE \*  
Base URL http://localhost:8081/

Run Walk Step

Table Source

Command	Target	Value
open	/pixory/pxapplicatio...	
type	usernameField	bob@bob.bob
type	passwordField	bob
clickAndWait	submitButton	
assertTextPresent	welcome bob@bob.bob	

Command:  
Target:  
Value:

Log Reference

**clickAndWait(locator)**  
Generated from **click(locator)**  
Arguments:  
• locator - an element locator  
Clicks on a link, button, checkbox or radio button. If the click action causes a new page to load (like a link usually does), call

Hit play to make sure your test is successful

# Creating a Test Suite

A Test Suite in Selenium is just an HTML file that contains a table of links to tests

```
<html>
<head>
<meta content="text/html; charset=ISO-8859-1"
http-equiv="content-type">
<title>Demo Test Suite</title>
</head>

<body>
  <table id="suiteTable">
    <tbody>
      <tr><td>
        <b>Demo Test Suite</b>
      </td></tr>
      <tr><td>
        <a href="./testLogin.html">TestLogin</a>
      </td></tr>
      <tr><td>
        <a href="./testLogout.html">TestLogout</a>
      </td></tr>
    </tbody>
  </table>
</body>
</html>
```

# Executing the Test Suite

- Selenium Core is a collection of Javascript and HTML with iFrames
- Due to security concerns Core must be deployed within the same server as the application being hosted
- The simplest way to run Pixory is to just run the Java application and let it use its own server
- Problems using Core with Pixory
- Selenium IDE is a plug-in for Firefox and thus can go around these restrictions

# Running the Test Suite

- We basically want to execute the test suite using the Selenium IDE plug-in TestRunner.html

```
chrome://selenium-ide/content/selenium/TestRunner.html?  
baseURL=<BASE>&test=file:///<TEST SUITE FILE>&auto=true
```

```
chrome://selenium-ide/content/selenium/TestRunner.html?  
baseURL=http://localhost:8081&test=file:///Users/ms333/  
projects/classes/running/v_and_v/hw3/selenium/test/  
TestSuite.html&auto=true
```

# Test Suite

The screenshot displays the Selenium TestRunner v0.8.2 interface. The browser window shows the URL `chrome://selenium-ide/content/selenium/TestRunner.html?baseUrl=http://localhost:8080`. The main content area is divided into three sections:

- Demo Test Suite:** A list of test cases: `TestLogin` (highlighted in green) and `TestLogout`.
- testLogin:** A table of test steps:

open	/pixory/pxapplication?service=page/Login	
type	usernameField	bob@bob.bob
type	passwordField	bob
clickAndWait	submitButton	
assertElementPresent	viewer_home_title_cell	
- Selenium TestRunner:** A control panel with "Execute Tests" buttons (Fast, Slow), a "Highlight elements" checkbox, and a status summary:
  - Elapsed: 00:01
  - Tests: 0 run, 0 passed, 0 failed, 0 incomplete
  - Commands: 0 run, 0 passed, 0 failed, 0 incomplete

Below the interface, a login form is shown with fields for "username" and "password", and a submit button. A link below the form reads: [don't know your username/password?](#)

At the bottom right, a copyright notice states: © 2002-2005 Joe Panico. [pixory.org](http://pixory.org) about [pixory 0.5.4](http://pixory.org)

The status bar at the bottom left shows "Done".

# Test Suite

# Execution Control

Test Cases

Steps of the test case

Application being tested

The screenshot displays the Selenium TestRunner interface. On the left, a sidebar shows a 'Demo Test Suite' containing 'TestLogin' and 'TestLogout'. The main area shows the steps of the 'TestLogin' test case in a table:





Step	Action	Target	Value
open		/pixory/pxapplication?service=page/Login	
type	usernameField	bob@bob.bo	
type	passwordField	bob	
clickAndWait	submitButton		
assertElementPresent	viewer_home_title_cell		

On the right, the 'Selenium TestRunner' control panel includes 'Execute Tests' buttons, a speed slider (Fast to Slow), a 'Highlight elements' checkbox, and a status section showing 'Elapsed: 00:01' and a summary of 'Tests' and 'Commands' (run, passed, failed, incomplete). Below the table is a screenshot of the application being tested, which is a login form with 'username' and 'password' fields and a submit button. A link below the form reads 'don't know your username/password?'. The footer of the application shows '© 2002-2005 Joe Panico. pixory... about pixory 0.5.4'.

# Test Runner Control

**Selenium TestRunner**

Execute Tests

Fast Slow

Highlight elements

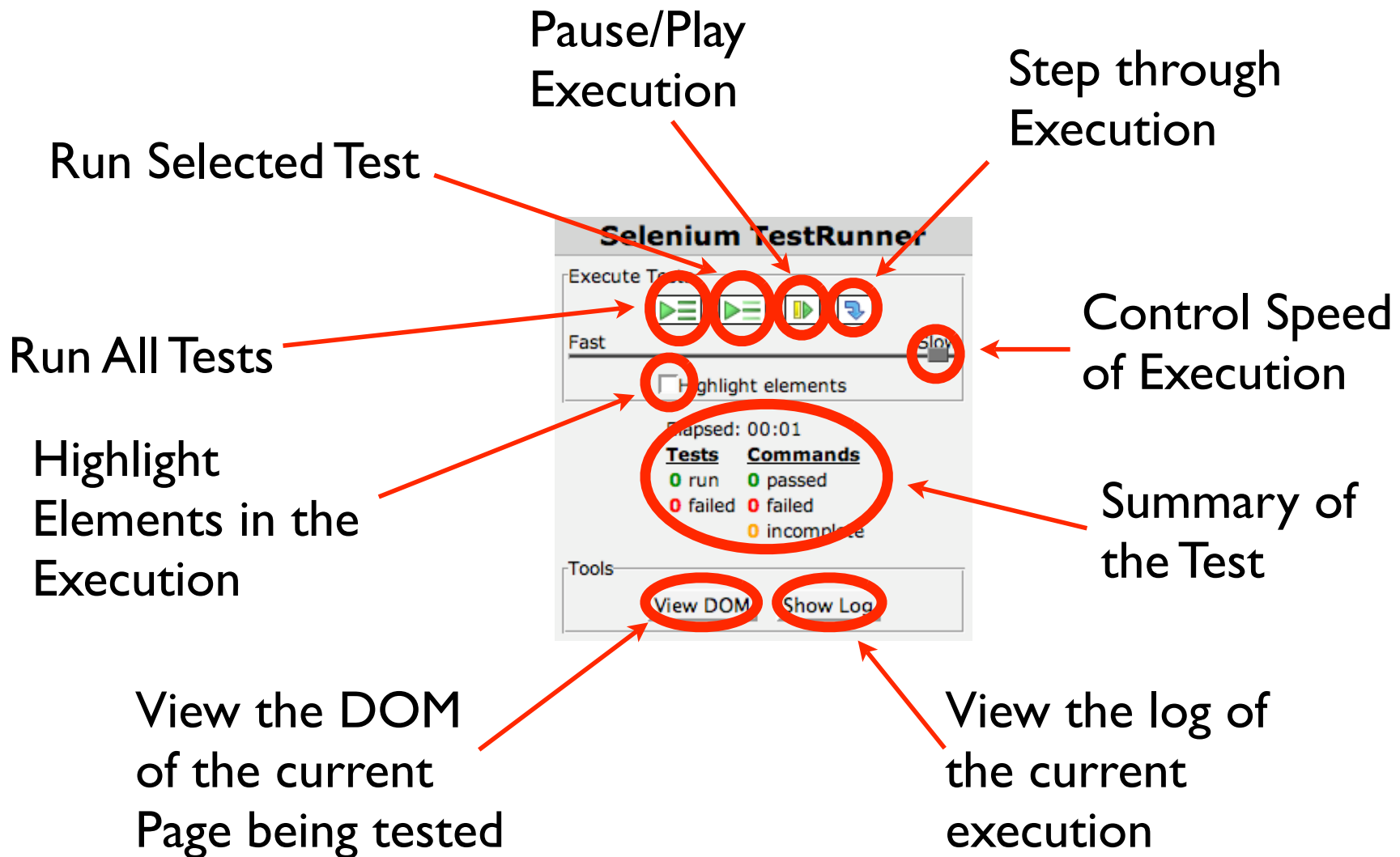
Elapsed: 00:01

<u>Tests</u>	<u>Commands</u>
<span style="color: green;">●</span> run	<span style="color: green;">●</span> passed
<span style="color: red;">●</span> failed	<span style="color: red;">●</span> failed
	<span style="color: orange;">●</span> incomplete

Tools

[View DOM](#) [Show Log](#)

# Test Runner Control





# TestRunner Demo

- Execute Tests created inside the Firefox TestRunner