

# Bài 8 – Kiểm thử tự động

Công cụ kiểm thử phần mềm  
Bộ môn Công nghệ phần mềm



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Nội dung

- ☐ **Kiểm thử tự động**
- ☐ Tổng quan Selenium
- ☐ Selenium IDE
- ☐ Selenium WebDriver

# Đặt vấn đề

- Bạn là một tester giỏi, bạn thiết kế các testcase tốt và thực hiện cẩn thận chúng nhưng còn thời gian và tiền bạc thì sao?

Again

Again

...and again

I have to do this test again.



How about my girlfriend?

# Định nghĩa

- Kiểm thử tự động (automated testing) là sử dụng công cụ để thực hiện tự động một phần hoặc toàn bộ quy trình kiểm thử và kiểm tra kết quả mà không cần sự can thiệp của con người

# Lợi ích

- ☐ Tiết kiệm thời gian và công sức
- ☐ Giảm sự nhầm chán và sai sót
- ☐ Nâng cao mức độ phủ của test suits
- ☐ Có thể sử dụng lại các bộ test case

➔ Tuy nhiên lợi ích của việc tự động hóa chỉ đạt được cao nhất khi nỗ lực cho việc tự động hóa không quá lớn

# Ứng dụng

- Kiểm thử tự động thích hợp
  - Kiểm thử hồi qui (Regression testing)
  - Kiểm thử hướng dữ liệu (Data driven testing)
  - Kiểm thử bản địa hóa (Localization testing)
    - Kiểm thử trên nhiều nền tảng
    - Kiểm thử trên nhiều ngôn ngữ
  - Kiểm thử khả năng chịu tải (Load testing)

# Khó khăn

- ☐ Test scripts là mã nguồn
- ☐ Xác nhận (validation) 1 testcase là Pass hay Fail
- ☐ Một số tình huống cần sự can thiệp của con người (ví dụ như captcha)

# Các thành phần

- ☐ Test script
  - ☐ Dùng công cụ record and replay
  - ☐ Tự viết script
- ☐ Test data
  - ☐ CSDL
  - ☐ Xml, Cfv
- ☐ Test validation



# Một số công cụ kiểm thử tự động

## ☐ Functional test

- ☐ QTP, Selenium, Windmill, ...

## ☐ Non-functional test

- ☐ Load runner, CloudTest, Load Storm, ...

# Nội dung

- ☐ Kiểm thử tự động
- ☐ **Tổng quan Selenium**
- ☐ Selenium IDE
- ☐ Selenium WebDriver

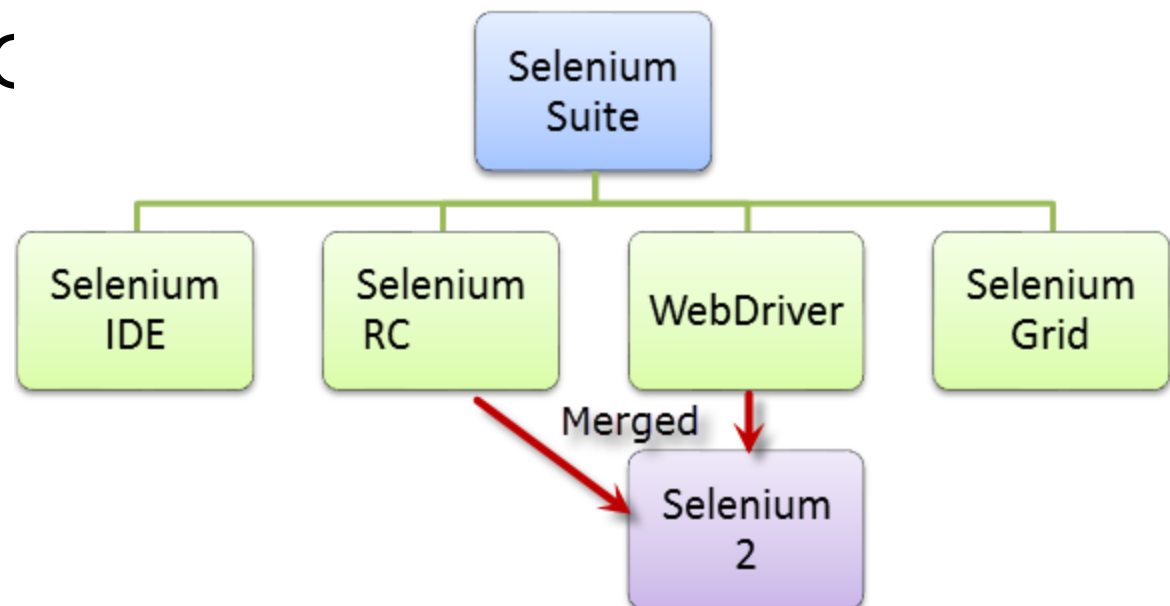
# Tổng quan Selenium

- Selenium là công cụ
  - Mã nguồn mở
  - Kiểm thử tự động
  - Ứng dụng website



# Tổng quan Selenium

- Selenium có 4 thành phần
  - ▣ Selenium IDE
  - ▣ Selenium RC
  - ▣ Selenium WebDriver
  - ▣ Selenium C



# Nội dung

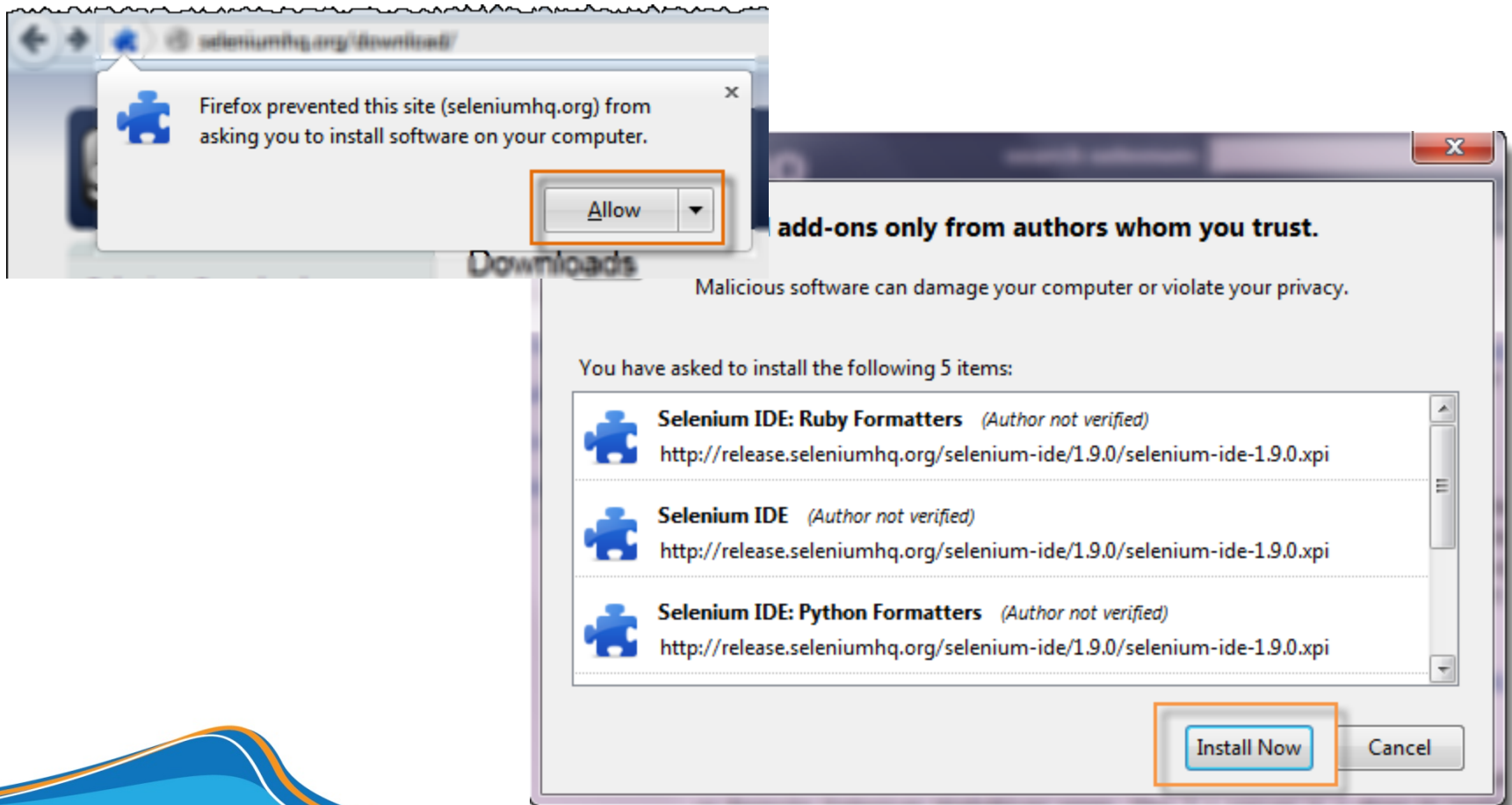
- ☐ Kiểm thử tự động
- ☐ Tổng quan Selenium
- ☐ **Selenium IDE**
- ☐ Selenium WebDriver

# Selenium IDE

- ☐ Plugin của Firefox
- ☐ Chức năng Record và Playback

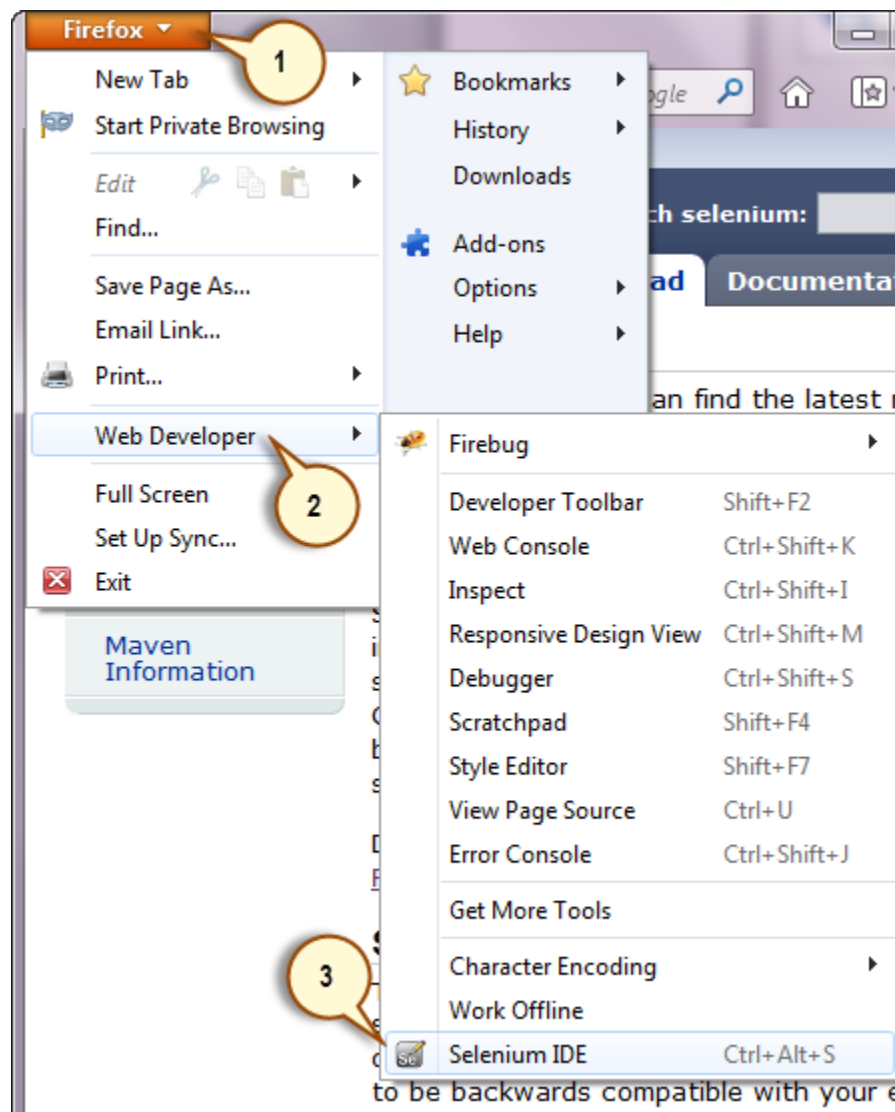
# Selenium IDE

□ Cài đặt <http://seleniumhq.org/download/>



# Selenium IDE

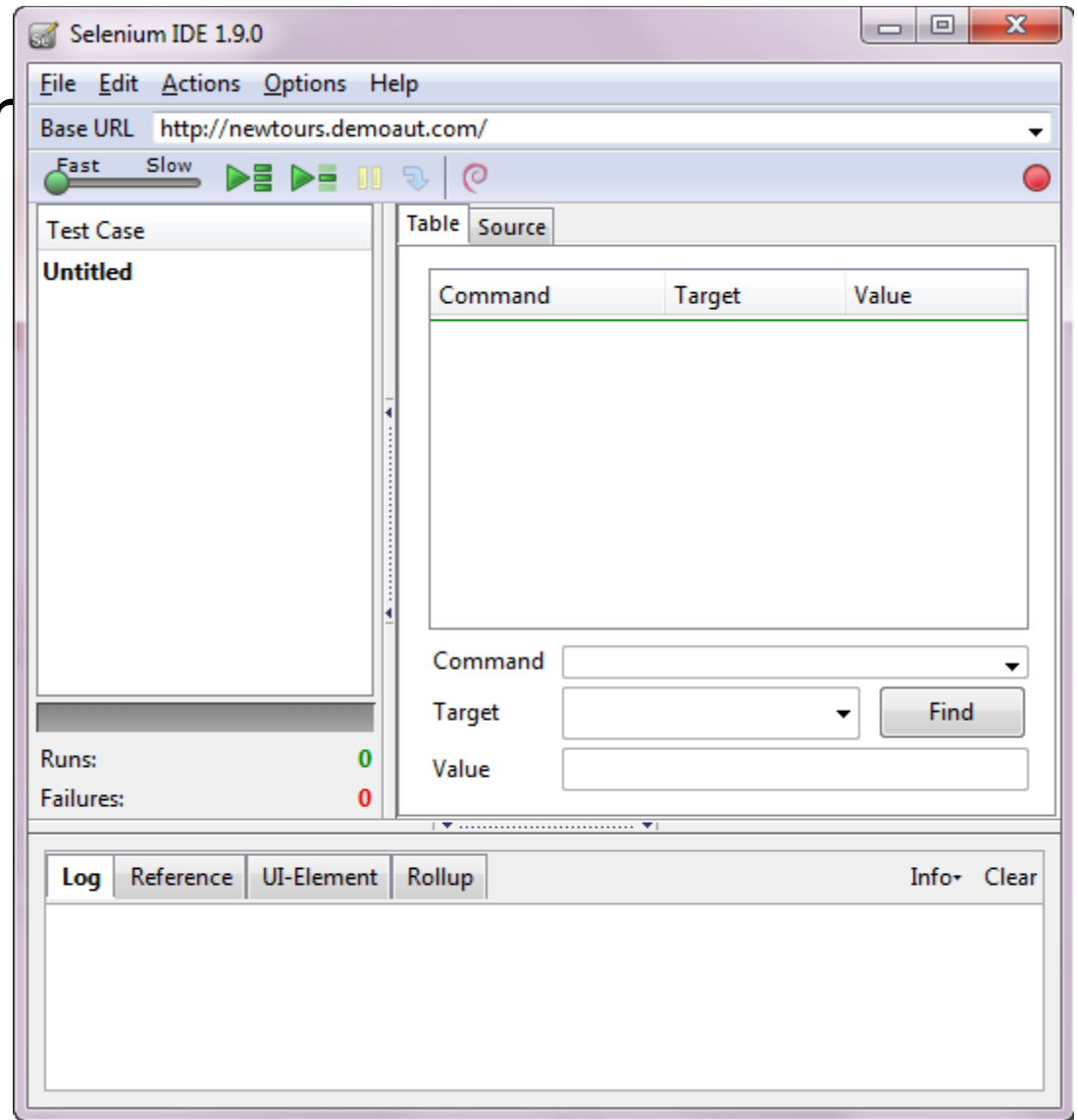
☐ Chạy plugin





# Selenium IDE

□ Giao diện chính



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

Fast Slow      Editor

Test Case	Table	Source																														
Invalid_login																																
Valid_login	<table border="1"> <thead> <tr> <th>Command</th> <th>Target</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>open</td> <td>/</td> <td></td> </tr> <tr> <td>type</td> <td>userName</td> <td>tutorial</td> </tr> <tr> <td>type</td> <td>password</td> <td>cause_of_failure</td> </tr> <tr> <td>clickAndWait</td> <td>login</td> <td></td> </tr> <tr> <td>verifyTitle</td> <td>Find a Flight: Mercu...</td> <td></td> </tr> <tr> <td>clickAndWait</td> <td>link=SIGN-OFF</td> <td></td> </tr> <tr> <td>verifyTitle</td> <td>Sign-on: Mercury T...</td> <td></td> </tr> <tr> <td>clickAndWait</td> <td>link=Home</td> <td></td> </tr> <tr> <td>verifyTitle</td> <td>Welcome: Mercury ...</td> <td></td> </tr> </tbody> </table>	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																														
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 ...																															

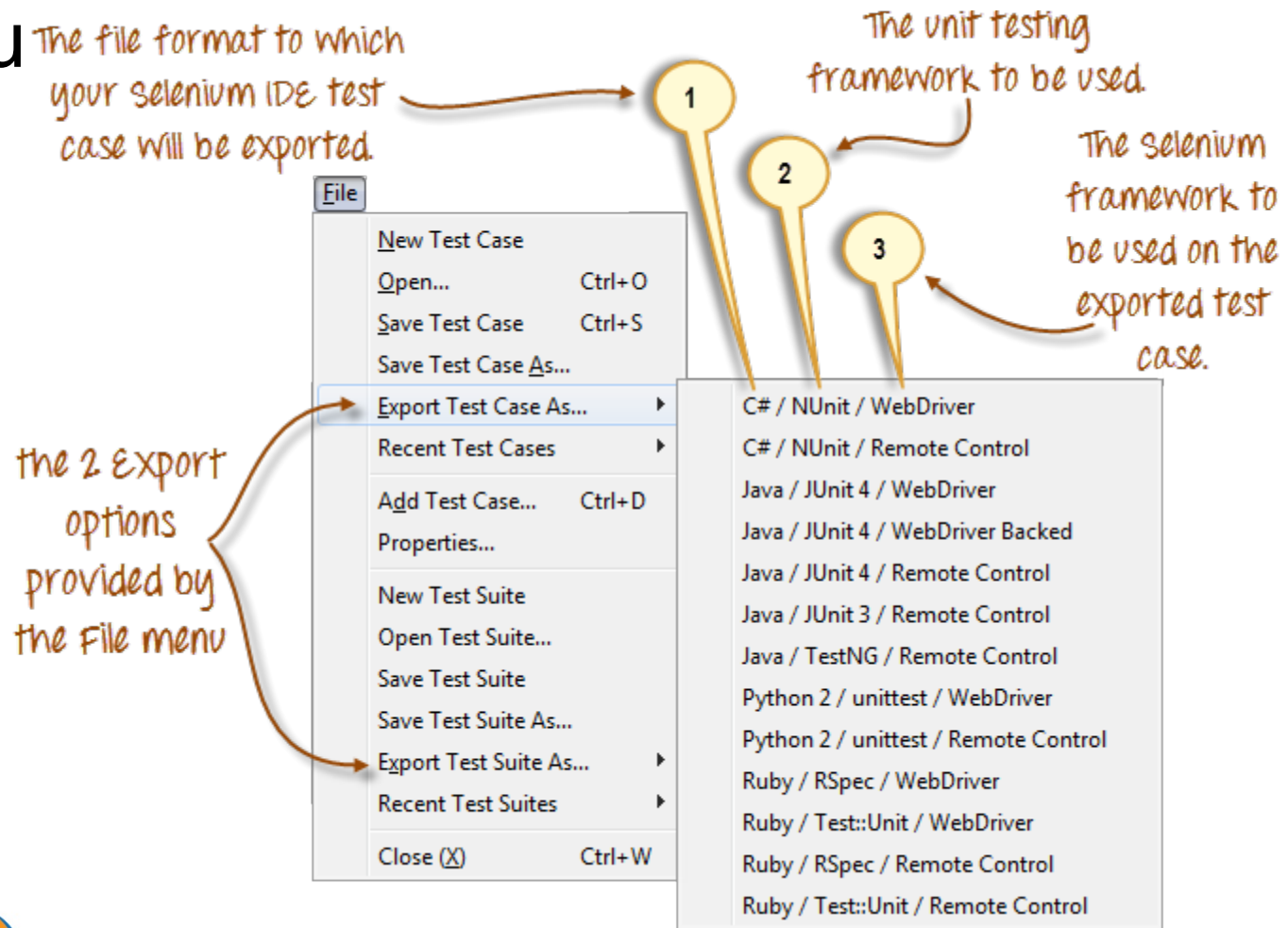
Runs: 2  
Failures: 1

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:					
<b>[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'</b>					
[info] Executing:  clickAndWait   link=SIGN-OFF					
<b>[error] Element link=SIGN-OFF not found</b>					

# Selenium IDE

## Menu

The file format to which your Selenium IDE test case will be exported.



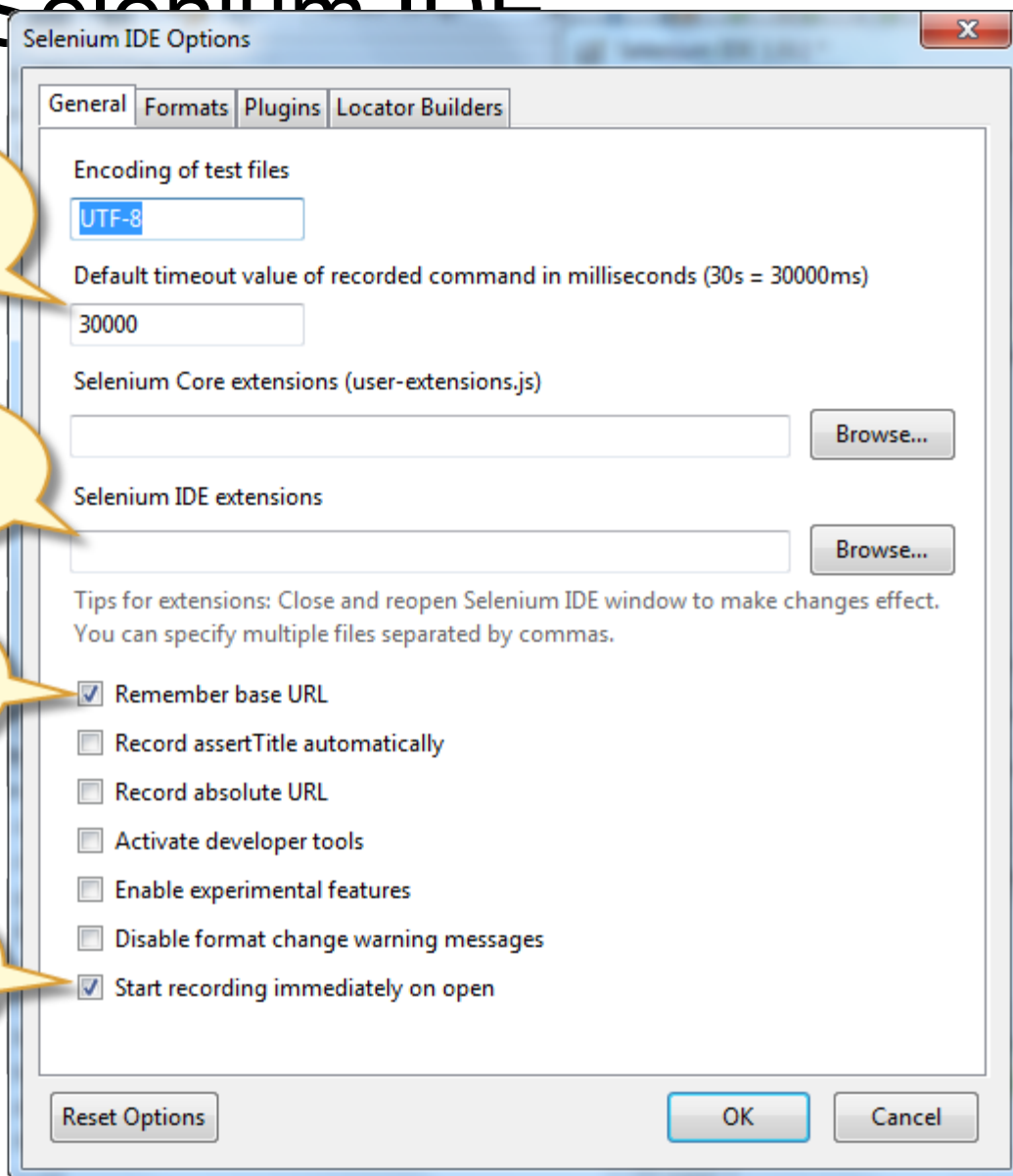
## □ Cấu hình

Default  
Timeout Value

Selenium IDE  
extensions

Remember  
base URL

Autostart  
record



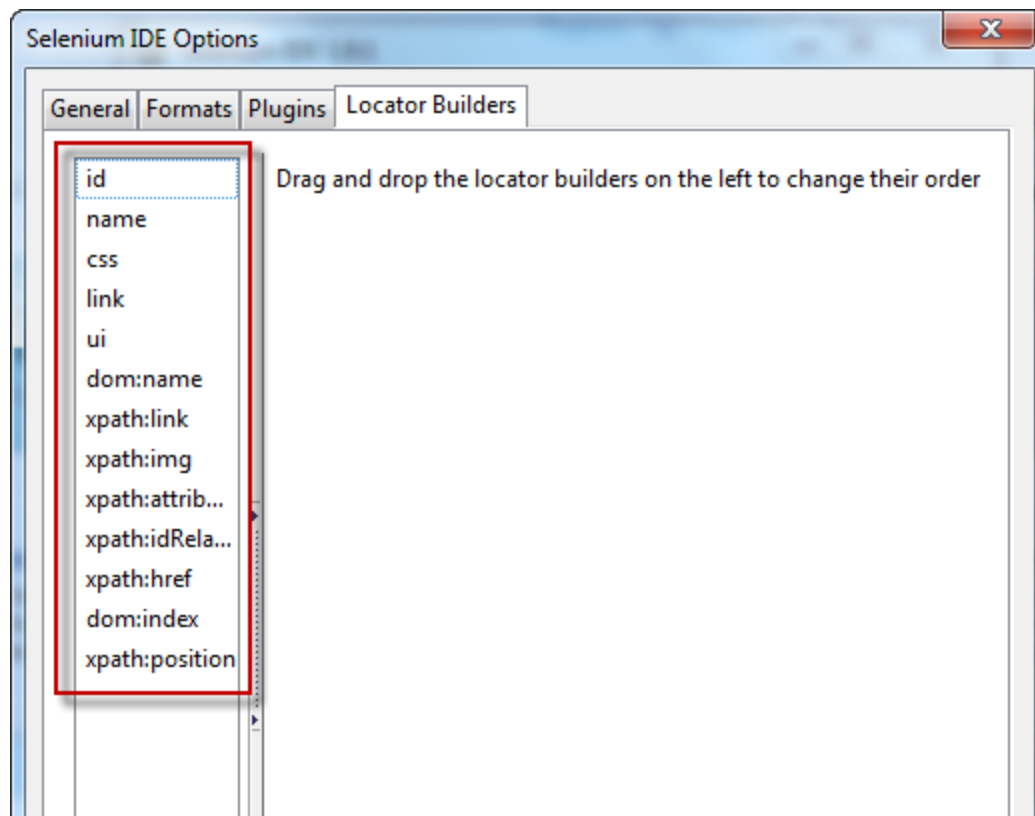
The screenshot shows the 'Selenium IDE Options' dialog box with the 'General' tab selected. The settings are as follows:

- Encoding of test files:** UTF-8
- Default timeout value of recorded command in milliseconds (30s = 30000ms):** 30000
- Selenium Core extensions (user-extensions.js):** (Empty text field with a 'Browse...' button)
- Selenium IDE extensions:** (Empty text field with a 'Browse...' button)
- Tips for extensions:** Close and reopen Selenium IDE window to make changes effect. You can specify multiple files separated by commas.
- Remember base URL:** ☒ (Checked)
- Record assertTitle automatically:** ☐ (Unchecked)
- Record absolute URL:** ☐ (Unchecked)
- Activate developer tools:** ☐ (Unchecked)
- Enable experimental features:** ☐ (Unchecked)
- Disable format change warning messages:** ☐ (Unchecked)
- Start recording immediately on open:** ☒ (Checked)

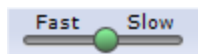
Buttons at the bottom: Reset Options, OK, Cancel.

# Selenium IDE

## □ Thứ tự ưu tiên Locator



## □ Thanh công cụ



Tốc độ chạy lại



Ghi / dừng ghi



Chạy toàn bộ Test suite



Chạy Test case hiện tại



Tạm dừng / chạy tiếp



Chạy từng bước



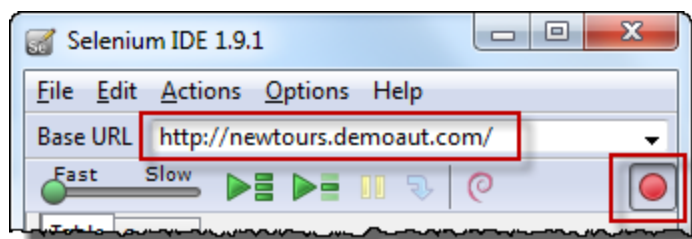
Nhóm các bước lại

# Selenium IDE

## □ Ví dụ Record and Playback

### □ Bước 1:

- Chạy Selenium IDE
- Nhập Base Url: <http://newtours.demoaut.com/>



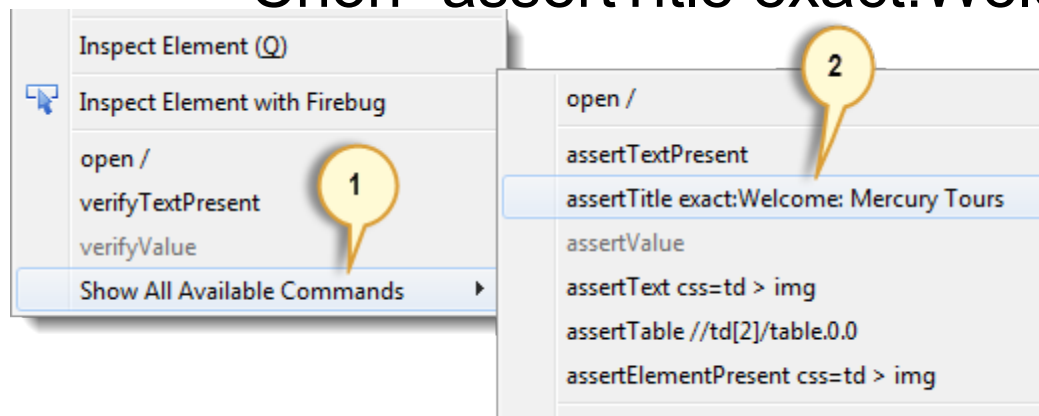
### □ Bước 2:

- Trong Firefox nhập Url  
<http://newtours.demoaut.com/>

## □ Ví dụ Record and Playback

### □ Bước 3:

- Phải chuột trên khoảng trống trong trang
- Chọn “Show Available Commands”
- Chọn “assertTitle exact:Welcome: Mercury Tours”



Comm...	Target	Value
open	/	
assertTitle	exact:Welcome: Mercury Tours	

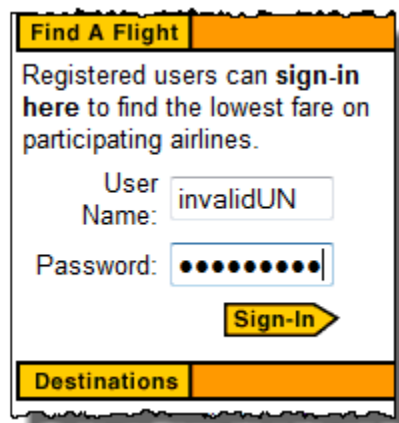
After clicking on the assertTitle context menu option, your Selenium IDE Editor pane should now contain the following commands



## □ Ví dụ Record and Playback

### □ Bước 4:

- Nhập “User name”: invalidUN
- Nhập “Password”: invalidUN



Find A Flight

Registered users can **sign-in here** to find the lowest fare on participating airlines.

User Name:

Password:

**Sign-In**

Destinations

Comm...	Target	Value
open	/	
assertTitle	exact>Welcome: Mercury Tours	
type	name=username	invalidUN
type	name=password	invalidPW

*Your Editor should now look like this*

### □ Bước 5:

- Bấm vào button “Sign-in”

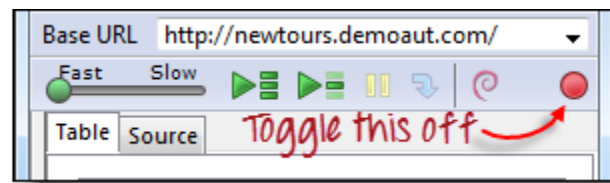
## □ Ví dụ Record and Playback

### □ Bước 6:

- Phải chuột trên khoảng trống trong trang
- Chọn “Show Available Commands”
- Chọn “assertTitle exact:Sign-on: Mercury Tours”

### □ Bước 7:

- Bấm nút dừng quay



### □ Bước 8:

- Bấm nút chạy lại test

# Selenium IDE

## □ 3 loại command

### □ Action

- Thao tác trực tiếp thành phần của trang web
- Ví dụ: click, type

### □ Accessors

- Cho phép lưu các giá trị vào các biến

### □ Assertions

- Kiểm tra một điều kiện được thỏa
- 3 loại
  - Assert: khi bị fail sẽ dừng test
  - Verify: khi bị fail sẽ ghi nhận và chạy tiếp
  - Waitfor: nếu điều kiện đúng trong thời gian chờ thì pass

## □ Các command thông dụng

Command	SL tham số	Giải thích
open	0 – 2	Mở 1 trang theo url
click/clickAndWait	1	Click vào 1 thành phần
type/typeKeys	2	Nhập 1 chuỗi ký tự
verifyTitle/assertTitle	1	Kiểm tra tiêu đề
verifyTextPresent	1	Kiểm tra có 1 đoạn text
verifyElementPresent	1	Kiểm tra có 1 thành phần
verifyTable	2	Kiểm tra 1 table
waitForPageToLoad	1	Chờ cho đến khi trang được tải xong
waitForElementPresent	1	Chờ cho đến khi 1 thành phần xuất hiện

# Selenium IDE

## □ Các loại Locator

### □ ID

### □ Name

### □ Link Text

### □ CSS Selector

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

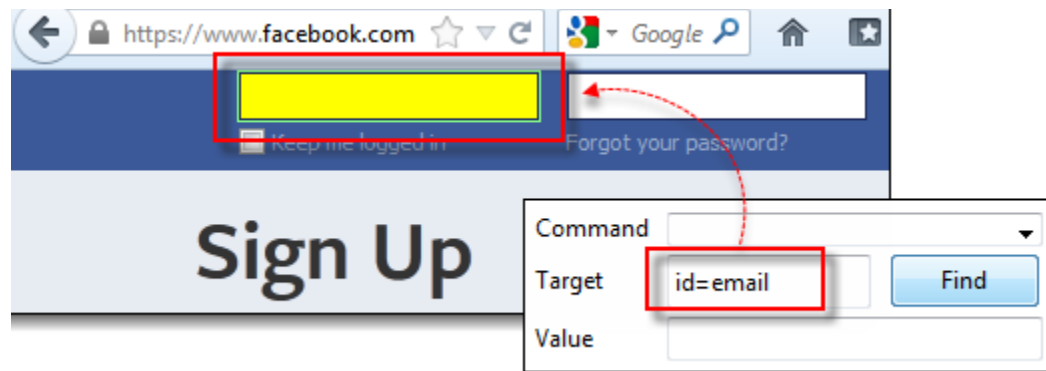
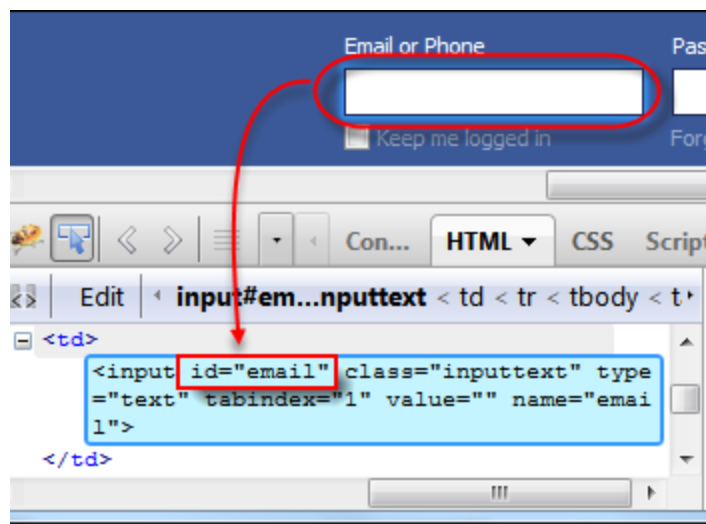
### □ DOM (Document Object Model)

- getElementById
- getElementsByName
- dom:name
- dom:index

### □ XPath

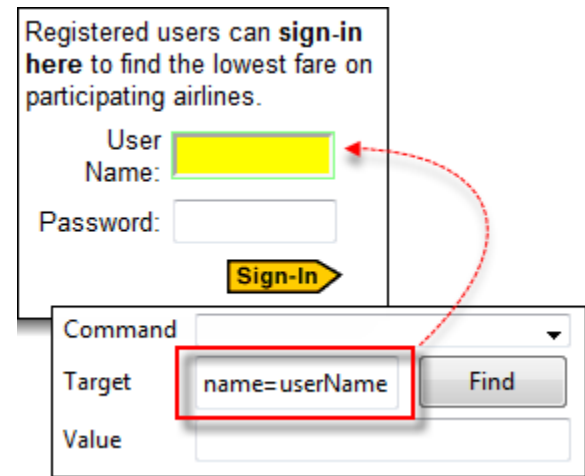
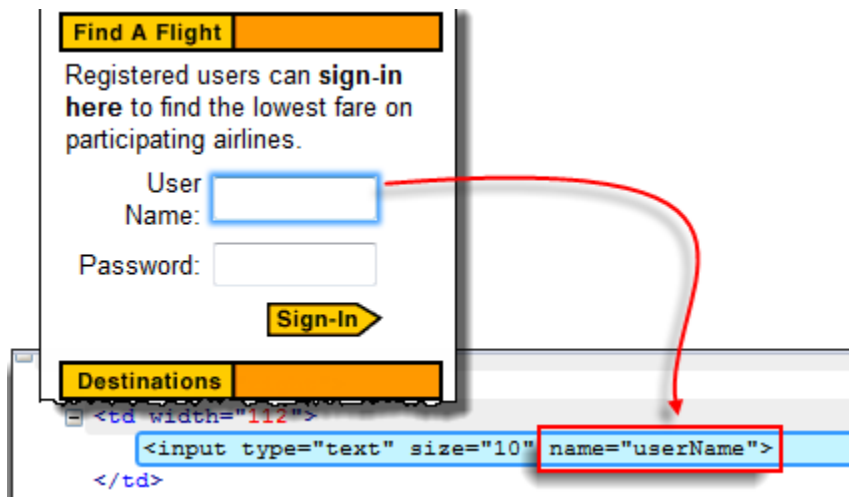
# Selenium IDE

- Xác định bằng ID
- Target: id=id of the element



# Selenium IDE

- Xác định bằng Name
  - Target: name=name of the element



# Selenium IDE

- Xác định bằng Name có filter
- Target: name=name\_of\_the\_element

filter value of filter

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:

Departing From:

Mercury Tours Flight Finder page

Command	Target	Value
click	name=tripType value=oneway	
Command	click	
Target	name=tripType value=oneway	Find
Value		



# Selenium IDE

- Xác định bằng Link Text
  - Target: link=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		

# Selenium IDE

- Xác định bằng CSS – Tag và ID
  - Target: css=tag#id

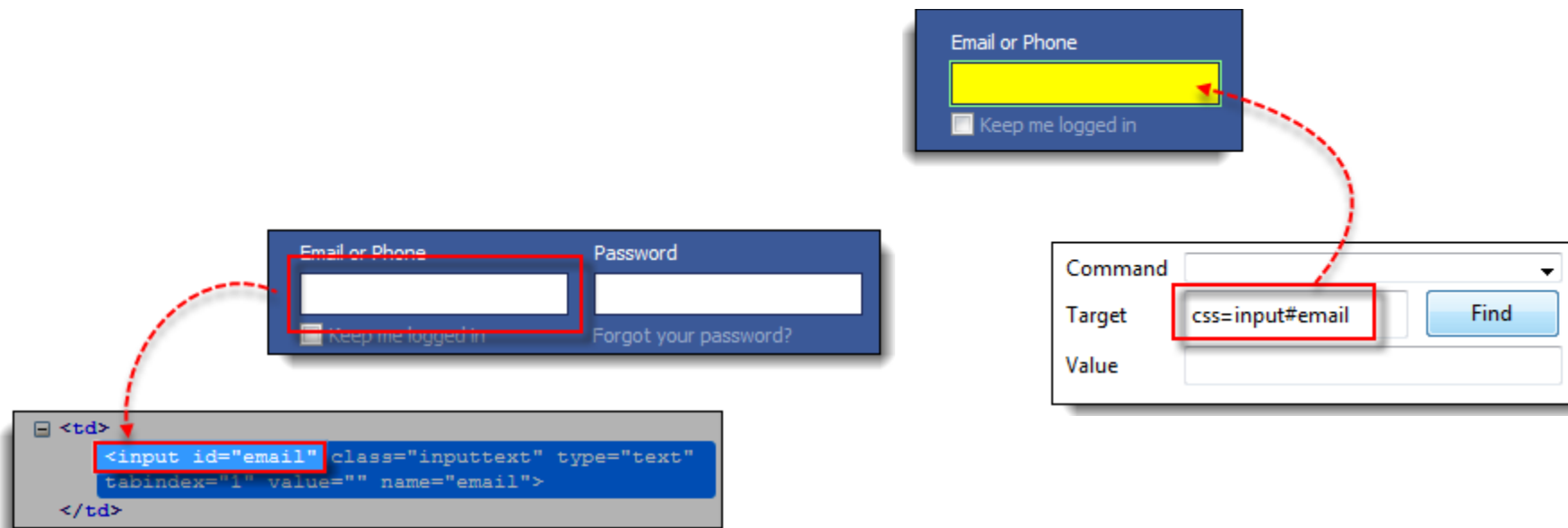


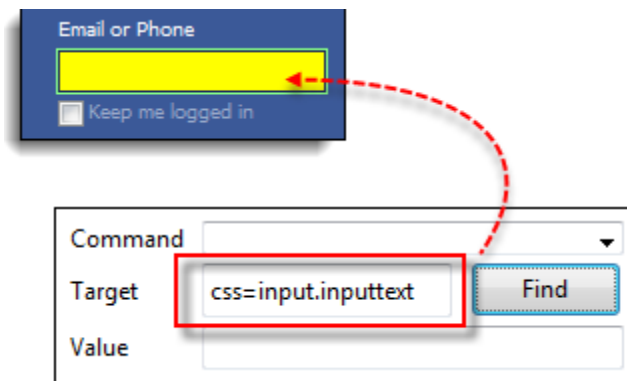
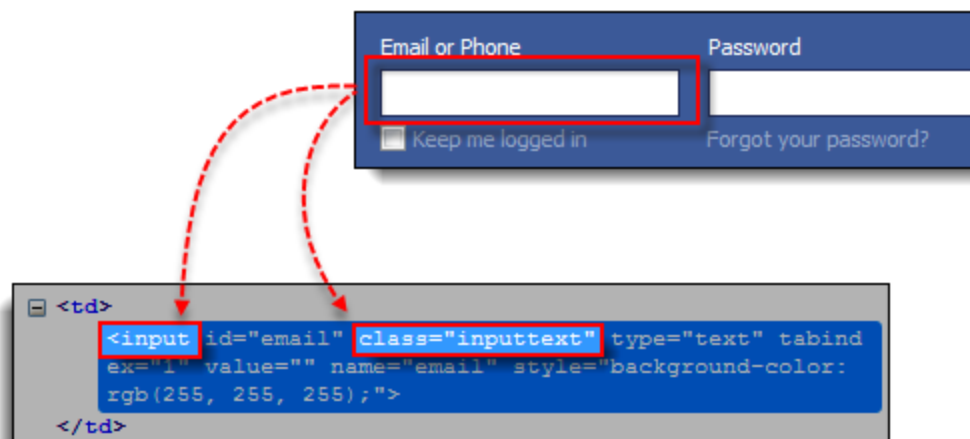
Diagram illustrating Selenium IDE target identification using CSS. The diagram shows a login form with two input fields: "Email or Phone" and "Password". A red dashed arrow points from the "Email or Phone" input field to its HTML source code, which is displayed in a box below. The source code is:

```
<td>  
<input id="email" class="inputtext" type="text"  
tabindex="1" value="" name="email">  
</td>
```

Another red dashed arrow points from the "Email or Phone" input field to the "Target" field in the Selenium IDE Command window. The "Target" field contains the CSS selector: `css=input#email`. The "Command" dropdown is set to "Find".

# Selenium IDE

- Xác định bằng CSS – Tag và Class
  - Target: tag.class



# Selenium IDE

- Xác định bằng CSS – Inner Text
  - Target: `css=tag:contains("inner text")`

```
<td align="right">  
  <font size="2" face="Arial, Helvetica,  
  sans-serif">Password:</font>  
</td>
```

This is the inner text

User

Name:

Password:

User

Name:

Password:

Command

Target `css=font:contains("Password:")` Find

Value

## □ Tóm tắt các Locator

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=tag#<i>id</i></code>	<code>css=input#email</code>
Tag and Class	<code>css=tag.<i>class</i></code>	<code>css=input.inputtext</code>
Tag and Attribute	<code>css=tag[<i>attribute=value</i>]</code>	<code>css=input[name=lastName]</code>
Tag, Class, and Attribute	<code>css=tag.<i>class</i>[<i>attribute=value</i>]</code>	<code>css=input.inputtext[tabindex=1]</code>

# Selenium IDE

- **verifyElementPresent**
  - True nếu thành phần có tồn tại
  - False nếu thành phần không tồn tại
- **verifyElementNotPresent**
  - True nếu thành phần không tồn tại
  - False nếu thành phần tồn tại

*This step failed thereby confirming that the element with name=firstname is not indeed present in the page*

Command	Target	Value
verifyElementPresent	name=username	
verifyElementNotPresent	name=firstname	
verifyElementPresent	name=firstname	

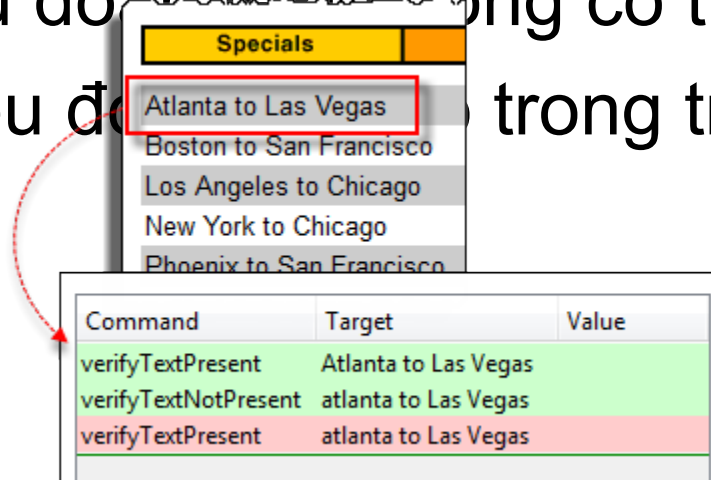
# Selenium IDE

## □ verifyTextPresent

- True nếu đoạn text có trong trang web
- False nếu đoạn text không có trong trang web

## □ verifyTextNotPresent

- True nếu đoạn text không có trong trang web
- False nếu đoạn text có trong trang web

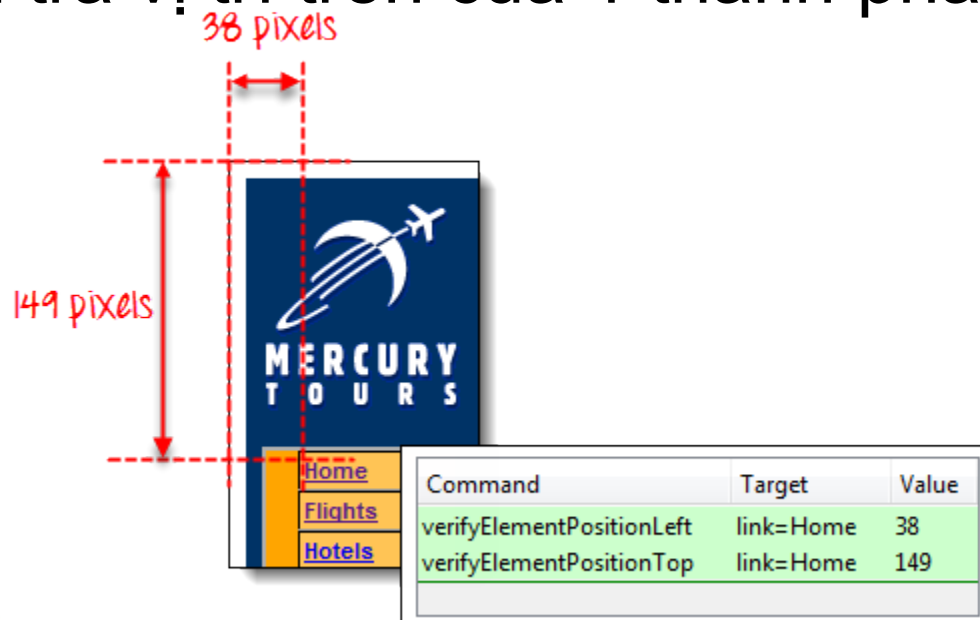


The screenshot shows a web application with a 'Specials' menu. The menu items are: Atlanta to Las Vegas, Boston to San Francisco, Los Angeles to Chicago, New York to Chicago, and Phoenix to San Francisco. The 'Atlanta to Las Vegas' item is highlighted with a red box. A red arrow points from this box to a table in the Selenium IDE command log.

Command	Target	Value
verifyTextPresent	Atlanta to Las Vegas	
verifyTextNotPresent	atlanta to Las Vegas	
verifyTextPresent	atlanta to Las Vegas	

# Selenium IDE

- `verifyElementPositionLeft`
  - Kiểm tra vị trí trái của 1 thành phần
- `verifyElementPositionTop`
  - Kiểm tra vị trí trên của 1 thành phần





# Selenium IDE

## □ andWait commands

### □ Ví dụ

- clickAndWait
- typeAndWait
- selectAndWait

### □ Chờ cho đến khi trang tải xong mới thực hiện bước tiếp theo

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.

## □ waitFor commands

### □ Ví dụ

- waitForTitle
- waitForTextPresent
- waitForAlert

### □ Chờ cho đến khi điều kiện được thực thi mới qua bước tiếp theo

# Selenium IDE

This box appears after the "why do I need to provide my birthday" link was clicked.



The screenshot shows the Facebook sign-up form. A red dashed arrow points from the text above to a red-bordered box on the left. A blue dashed arrow points from the text above to a blue-bordered box on the right.

**Providing your birthday** helps make sure you get the right Facebook experience for your age. You can choose to hide this info from your timeline later if you want. For more details, please visit our [Data Use Policy](#).

**Not creating a personal account?** If you're here to represent your band, business or product, please create a Facebook Page.

**Okay**

**Birthday:**

Month:  Day:  Year:

☐ Female ☐ Male

By clicking Sign Up, you agree to our [Terms](#) and that you have read our [Data Policy](#), including our [Cookie Use](#).

**Sign Up**

**Why do I need to provide my birthday?**

Note: The box appeared without reloading the page.

Command	Target	Value
open		
click	link=Why do I need to provide my birthday?	
waitForTextPresent	Providing your birthday	
verifyTextPresent	Providing your birthday	

PASSED

## □ Lưu giá trị vào biến

value of the variable

Command	Target	Value
store	tutorial	myVariable

name of the variable

User Name:

Password:

**Sign-In**

Command	Target	Value
open		
store	tutorial	myVariable
type	name=username	\${myVariable}

## □ Alerts và Pop-up

assertAlert assertNotAlert	Kiểm tra chuỗi trong alert
assertAlertPresent assertAlertNotPresent	Kiểm tra có alert hay không
storeAlert	Lưu chuỗi trong alert
storeAlertPresent	Trả về True khi có alert và False khi không có alert
verifyAlert verifyNotAlert	Kiểm tra chuỗi trong alert
verifyAlertPresent verifyAlertNotPresent	Kiểm tra có alert hay không

# Nội dung

- ☐ Kiểm thử tự động
- ☐ Tổng quan Selenium
- ☐ Selenium IDE
- ☐ **Selenium WebDriver**

# Selenium WebDriver

- ☐ Sử dụng ngôn ngữ lập trình để tạo các script
  - ☐ Sử dụng điều kiện rẽ nhánh
  - ☐ Sử dụng vòng lặp
- ☐ Các ngôn ngữ hỗ trợ
  - ☐ Java
  - ☐ .Net
  - ☐ PHP
  - ☐ Python
  - ☐ Perl
  - ☐ Ruby

# Selenium WebDriver

## □ Các trình duyệt hỗ trợ





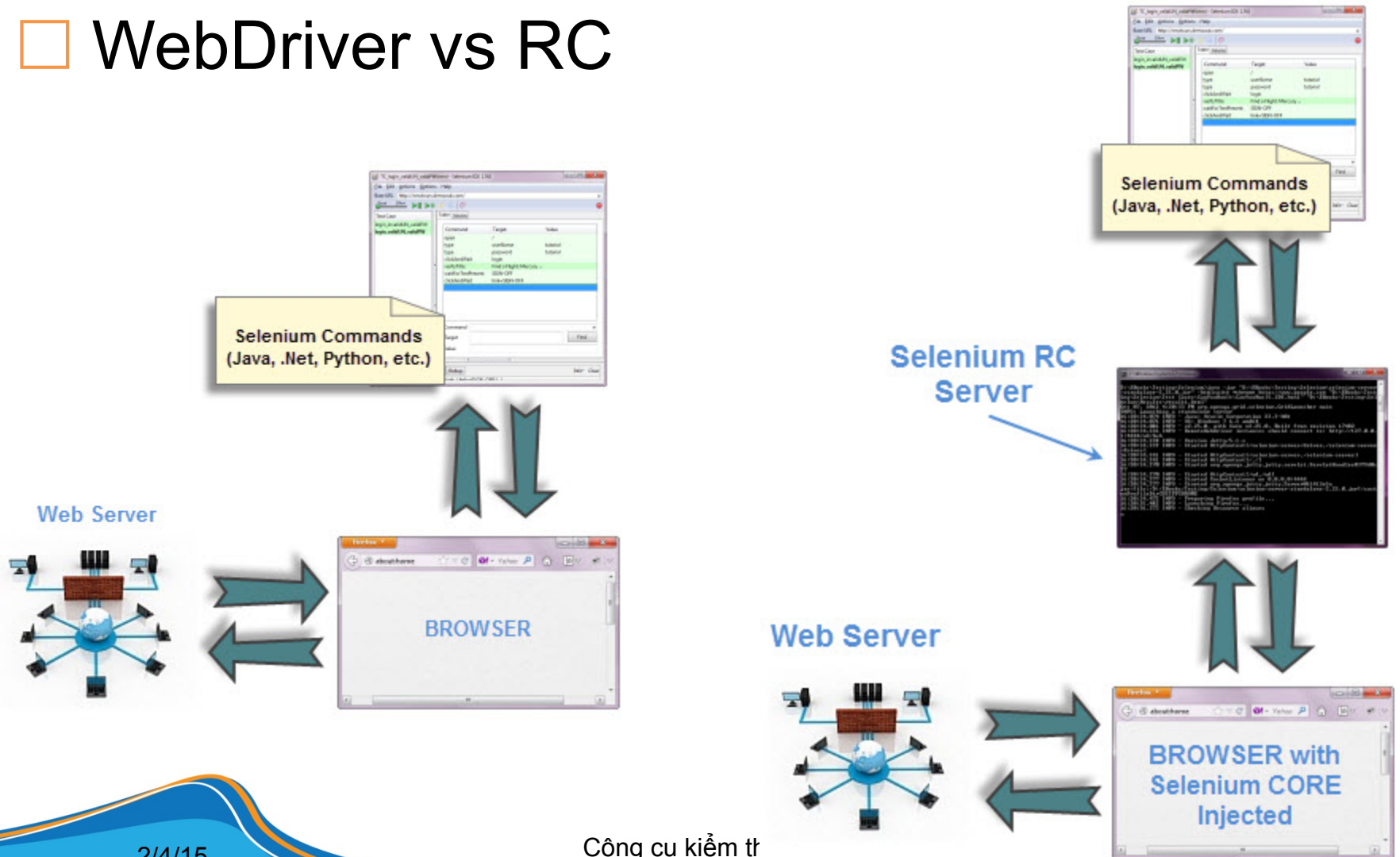
# Selenium WebDriver

## □ Các trình duyệt hỗ trợ

Trình duyệt	Tên Driver Server	Chi chú
HTMLUnit	(none)	Không cần driver
Firefox	(none)	Không cần driver
Internet Explorer	Internet Explorer Driver Server	Có 2 phiên bản 32 and 64-bit
Chrome	ChromeDriver	
Opera	OperaDriver	
PhantomJS	GhostDriver	
Safari	SafariDriver	

# Selenium WebDriver

## WebDriver vs RC



# Selenium WebDriver

## □ Ví dụ WebDriver với C#

- Bước 1: download Selenium C# Client Driver  
<http://docs.seleniumhq.org/download/>

Language	Client Version	Release Date			
Java	2.32.0	2013-04-09	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	2.32.0	2013-04-09	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	2.32.0	2013-04-09	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	2.32.0	2013-04-10	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

- Bước 2: cài đặt NUnit

# Selenium WebDriver

## □ Ví dụ WebDriver với C#

□ Bước 3: dùng Selenium ghi lại các bước test

□ Bước 4: Export Test Case As → C# / NUnit /

```
W namespace SeleniumTests
{
    [TestFixture]
    public class Testcase1
    {
        private IWebDriver driver;
        private StringBuilder verificationErrors;
        private string baseURL;
        private bool acceptNextAlert = true;

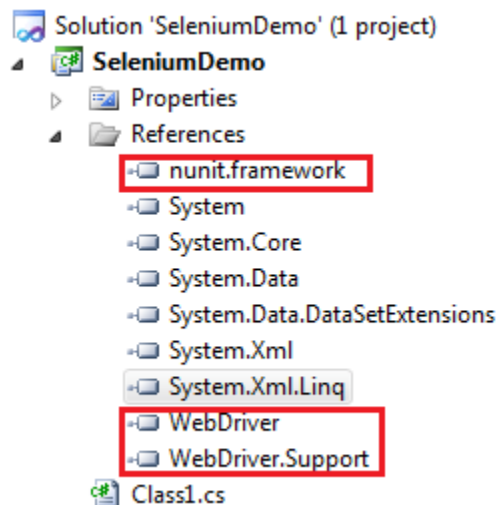
        [SetUp]
        public void SetupTest()
        {
            driver = new FirefoxDriver();
            baseURL = "http://newtours.demoaut.com/";
            verificationErrors = new StringBuilder();
        }
    }
}
```

# Selenium WebDriver

## □ Ví dụ WebDriver với C#

### □ Bước 5: tạo dự án

- Tạo dự án Class Library, ngôn ngữ C#
- Sao chép mã nguồn đã export vào dự án
- Thêm các Reference



## □ Ví dụ WebDriver với C#

### □ Lập trình test case

```
[Test]
public void TheCase1Test()
{
    driver.Navigate().GoToUrl(baseUrl + "/");
    driver.FindElement(By.Name("userName")).Clear();
    driver.FindElement(By.Name("userName")).SendKeys("abcd");
    driver.FindElement(By.Name("password")).Clear();
    driver.FindElement(By.Name("password")).SendKeys("123456");
    driver.FindElement(By.Name("login")).Click();
    Assert.AreEqual("Sign-on: Mercury Tours", driver.Title);
}
```

- Ví dụ WebDriver với C#
  - ▣ Chạy trên trình duyệt Firefox

```
using OpenQA.Selenium.Firefox;  
using OpenQA.Selenium.Support.UI;  
using OpenQA.Selenium.Chrome;
```

```
[SetUp]  
public void SetupTest()  
{  
    driver = new FirefoxDriver();  
    baseUrl = "http://newtours.demoaut.com/";  
    verificationErrors = new StringBuilder();  
}
```

# Selenium WebDriver

## □ Ví dụ WebDriver với C#

### ▣ Chạy trên trình duyệt Chrome

```
using OpenQA.Selenium.Firefox;  
using OpenQA.Selenium.Support.UI;  
using OpenQA.Selenium.Chrome;
```

```
[SetUp]  
public void SetupTest()  
{  
    driver = new ChromeDriver();  
    baseUrl = "http://newtours.demoaut.com/";  
    verificationErrors = new StringBuilder();  
}
```



# Selenium WebDriver

- Ví dụ WebDriver với C#
  - ▣ Chạy nhiều testcases trên cùng 1 script

```
[TestCase(12, 3, 4)]  
[TestCase(12, 2, 6)]  
[TestCase(12, 4, 3)]  
public void DivideTest(int n, int d, int q)  
{  
    Assert.AreEqual( q, n / d );  
}
```

# Selenium WebDriver

- Ví dụ WebDriver với C#
  - ▣ Chạy nhiều testcases trên cùng 1 script

```
[Test, TestCaseSource("DivideCases")]  
public void DivideTest(int n, int d, int q)  
{  
    Assert.AreEqual( q, n / d );  
}  
  
static object[] DivideCases =  
{  
    new object[] { 12, 3, 4 },  
    new object[] { 12, 2, 6 },  
    new object[] { 12, 4, 3 }  
};
```

# Selenium WebDriver

## □ Ví dụ WebDriver với C#

### ▣ Chạy nhiều testcases từ database

```
public struct TestCase
{
    public int a;
    public int b;
    public int tong;
}
[Test, TestCaseSource("GetFromExcel")]
public void Cong2So(TestCase testcase)
{
    Assert.AreEqual(testcase.a + testcase.b, testcase.tong);
}
public List<TestCase> GetFromExcel()
{
    List<TestCase> testcases = new List<TestCase>();
    // đọc từ file excel ...
    return testcases;
}
```

# Selenium WebDriver

## ☐ Download các browser

- ☐ Chrome driver: chromedriver\_win\_26.0.1383.0.zip

<https://code.google.com/p/chromedriver/downloads/list>

- ☐ IE driver: IEDriverServer\_Win32\_2.32.3.zip

<https://code.google.com/p/selenium/downloads/list>

- ☐ Opera driver

<https://code.google.com/p/selenium/wiki/OperaDriver>

- ☐ Safari driver

<https://code.google.com/p/selenium/wiki/SafariDriver>

# Thảo luận

