

Selenium

Công cụ kiểm thử tự động

Trần Duy Hoàng – tdhoang@fit.hcmus.edu.vn

Nội dung

- ▶ **Tổng quan Selenium**
- ▶ Selenium IDE
- ▶ Selenium WebDriver

Tổng quan Selenium

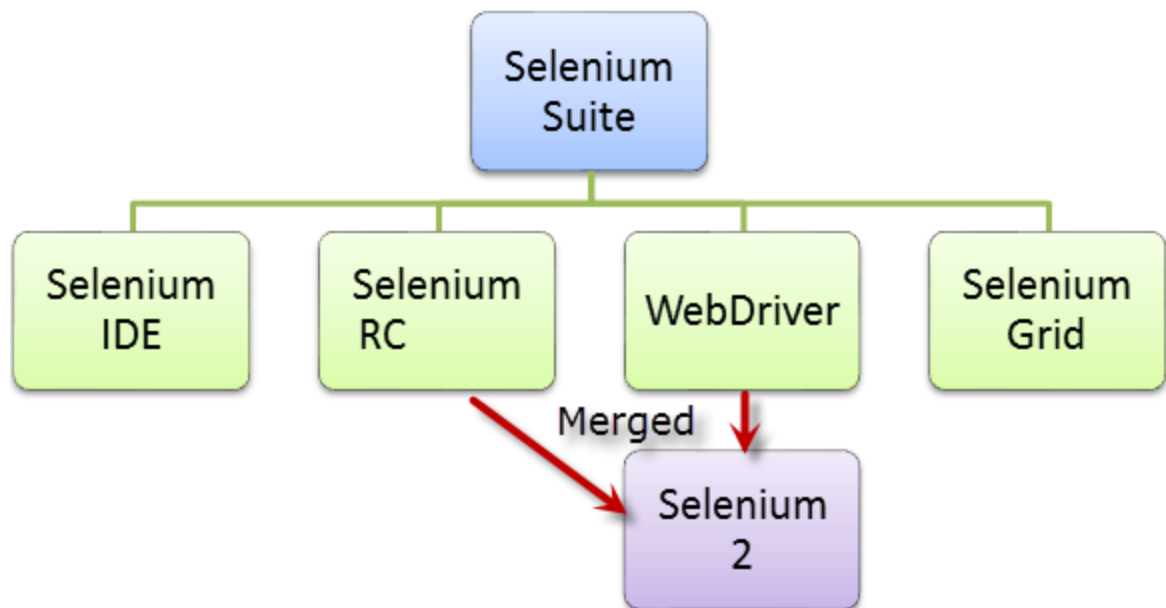
- ▶ Selenium là công cụ
 - ▶ Mã nguồn mở
 - ▶ Kiểm thử tự động
 - ▶ Ứng dụng website



Tổng quan Selenium

- ▶ Selenium có 4 thành phần

- ▶ Selenium IDE
- ▶ Selenium RC
- ▶ Selenium WebDriver
- ▶ Selenium Grid



Nội dung

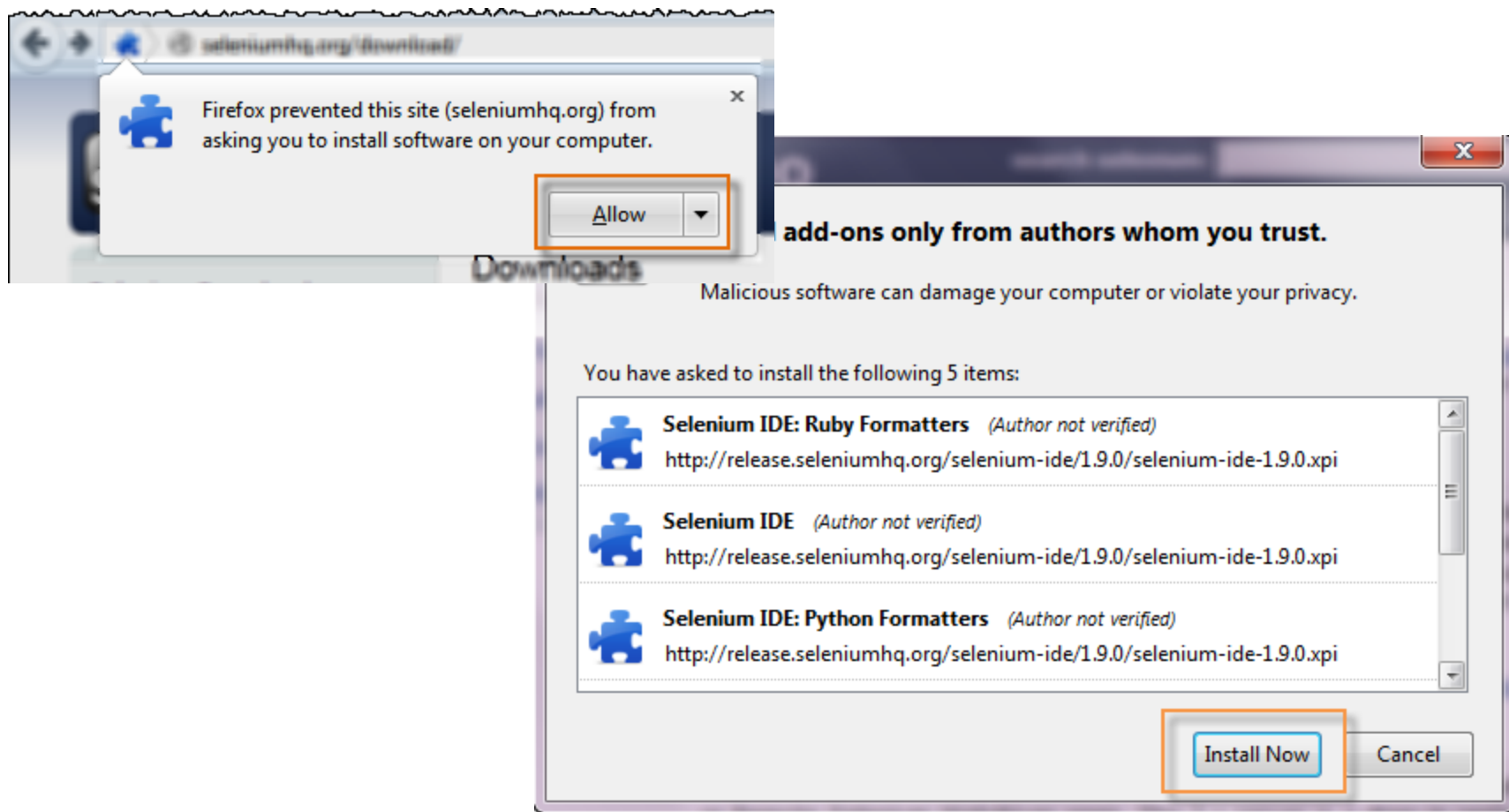
- ▶ Tổng quan Selenium
- ▶ **Selenium IDE**
- ▶ Selenium WebDriver

Selenium IDE

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

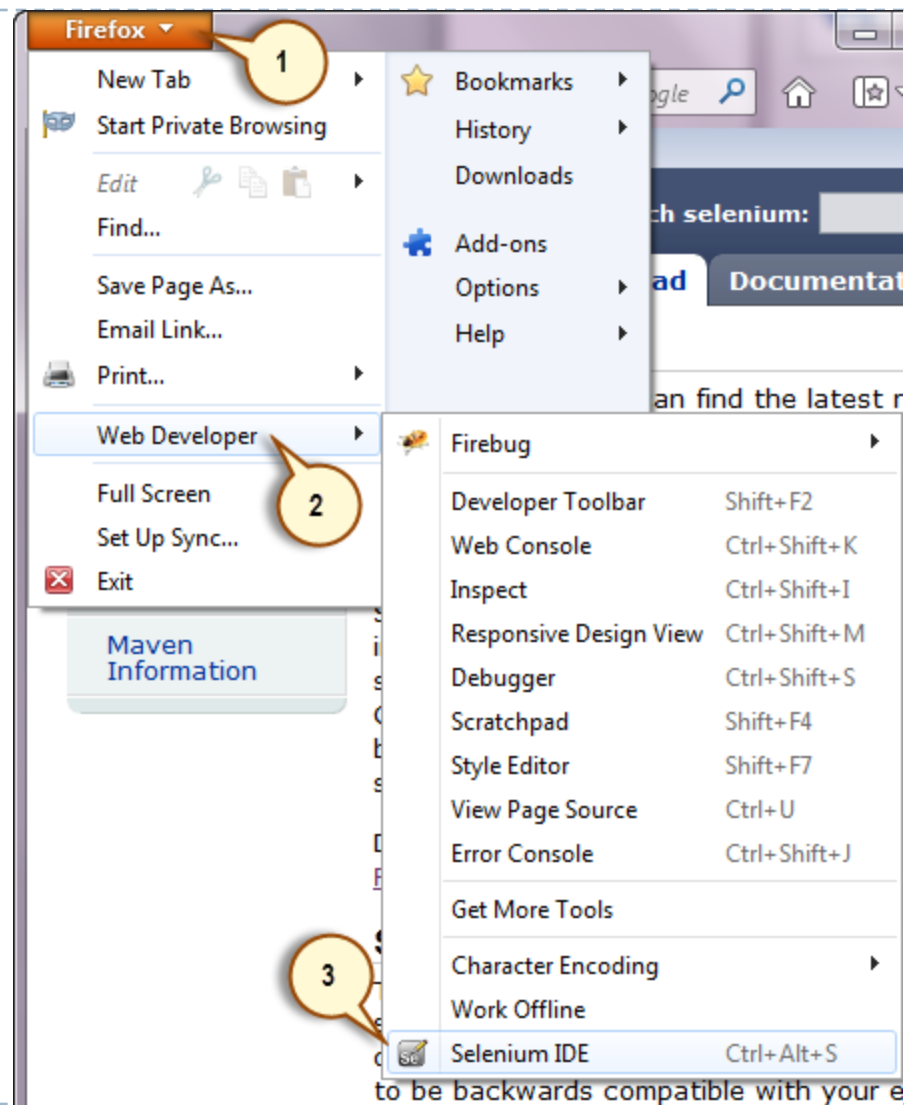
Selenium IDE

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



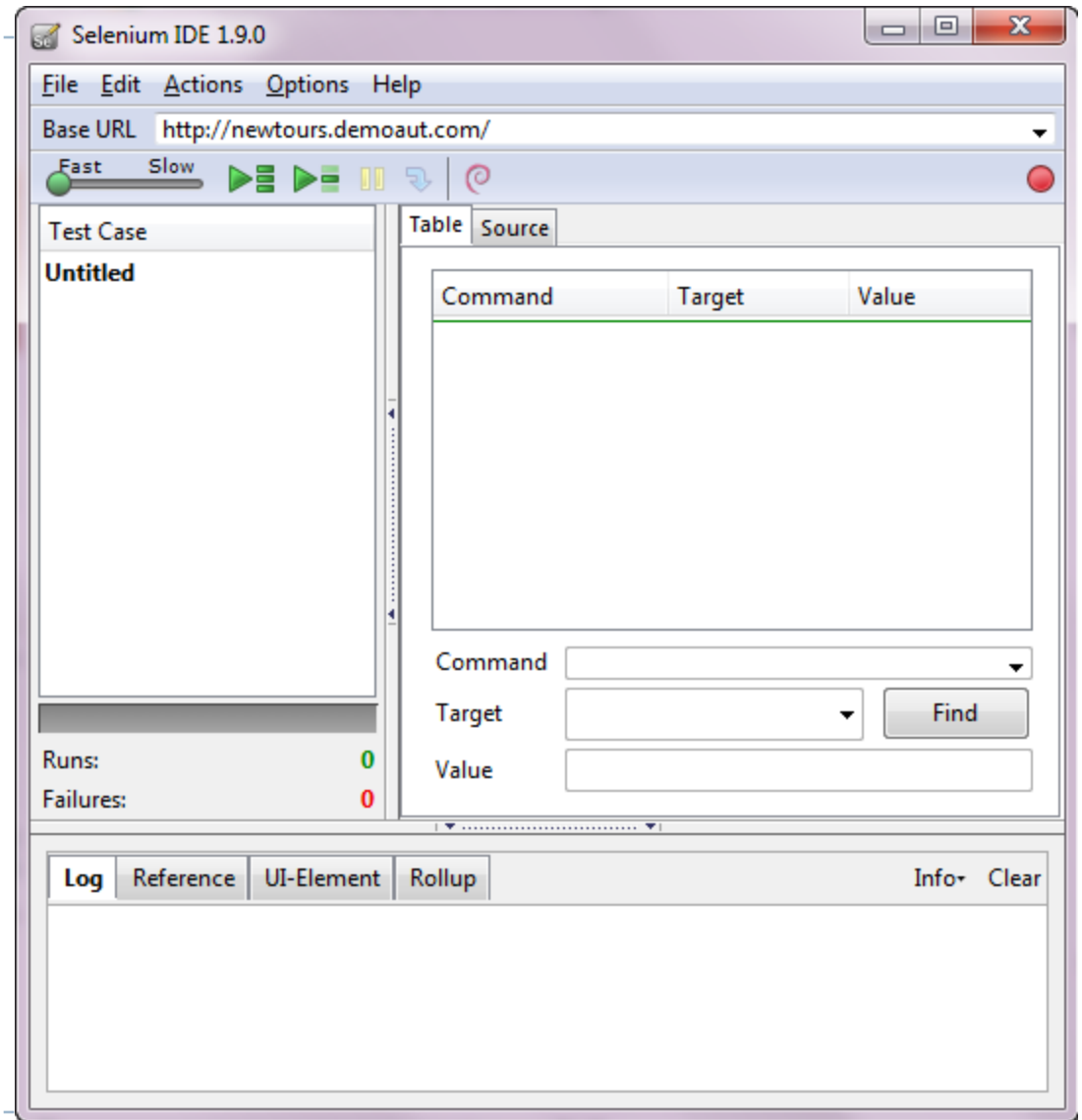
Selenium IDE

► 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

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: | |

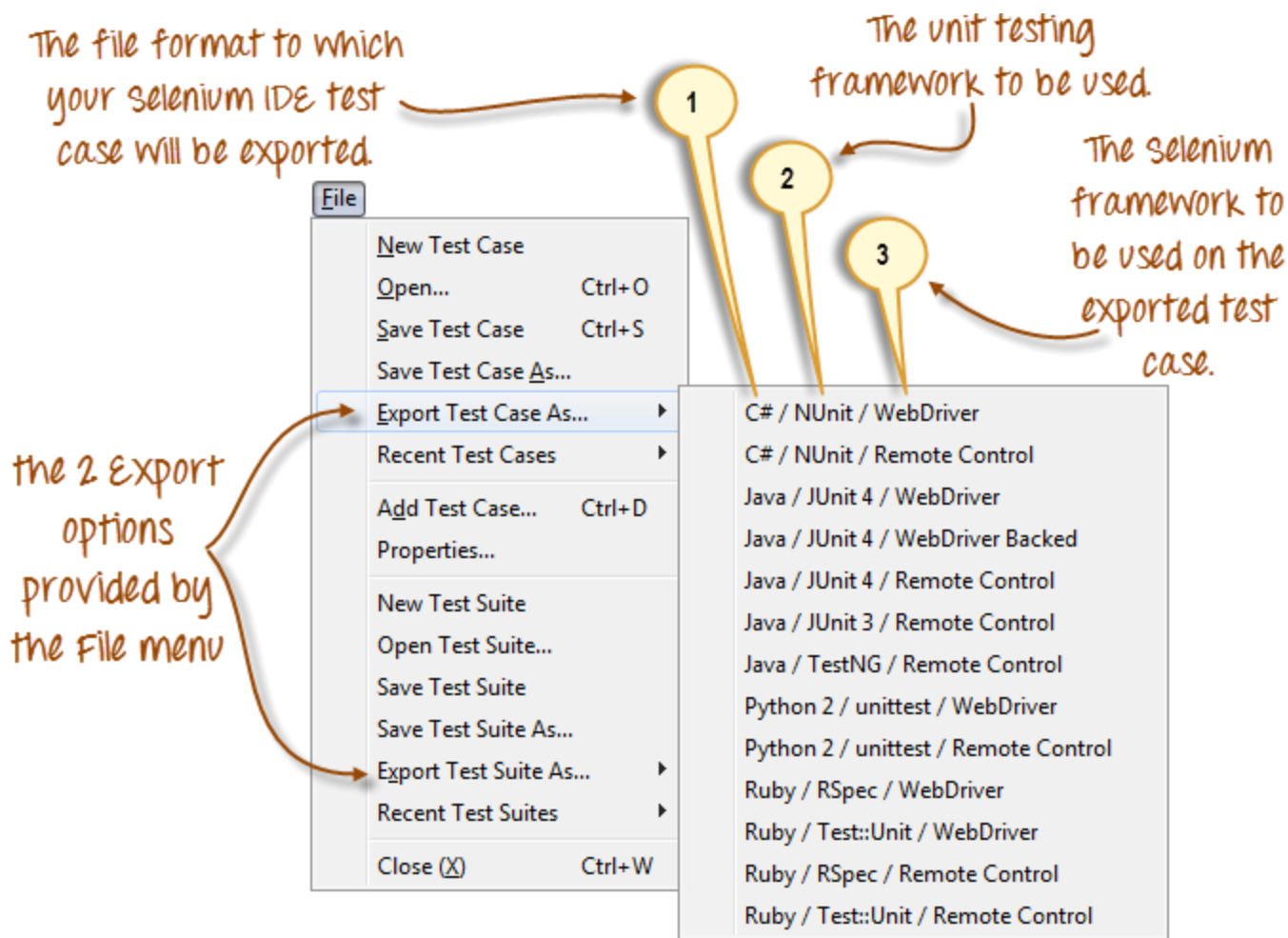
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'

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

[error] Element link=SIGN-OFF not found

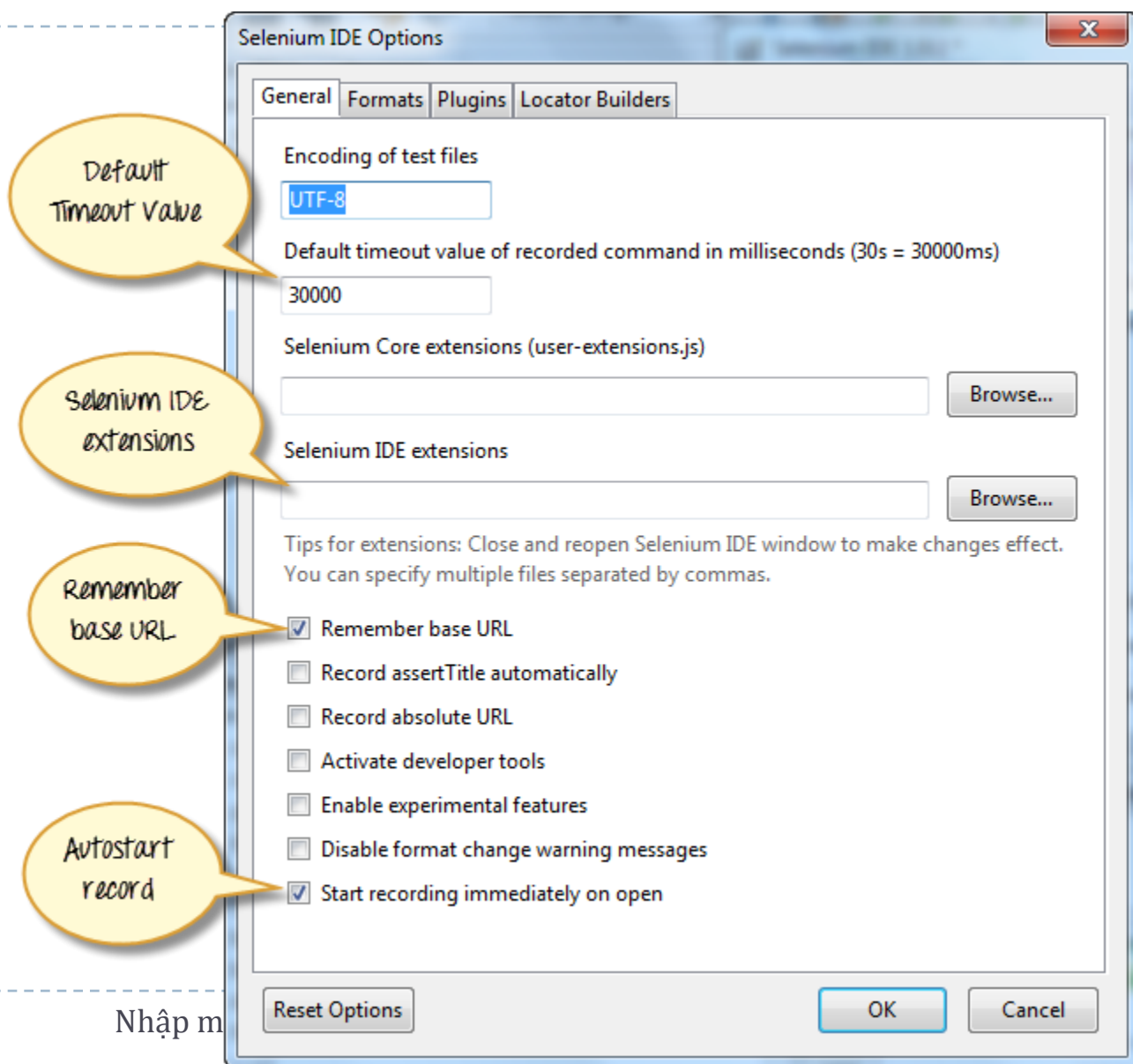
Selenium IDE

► Menu



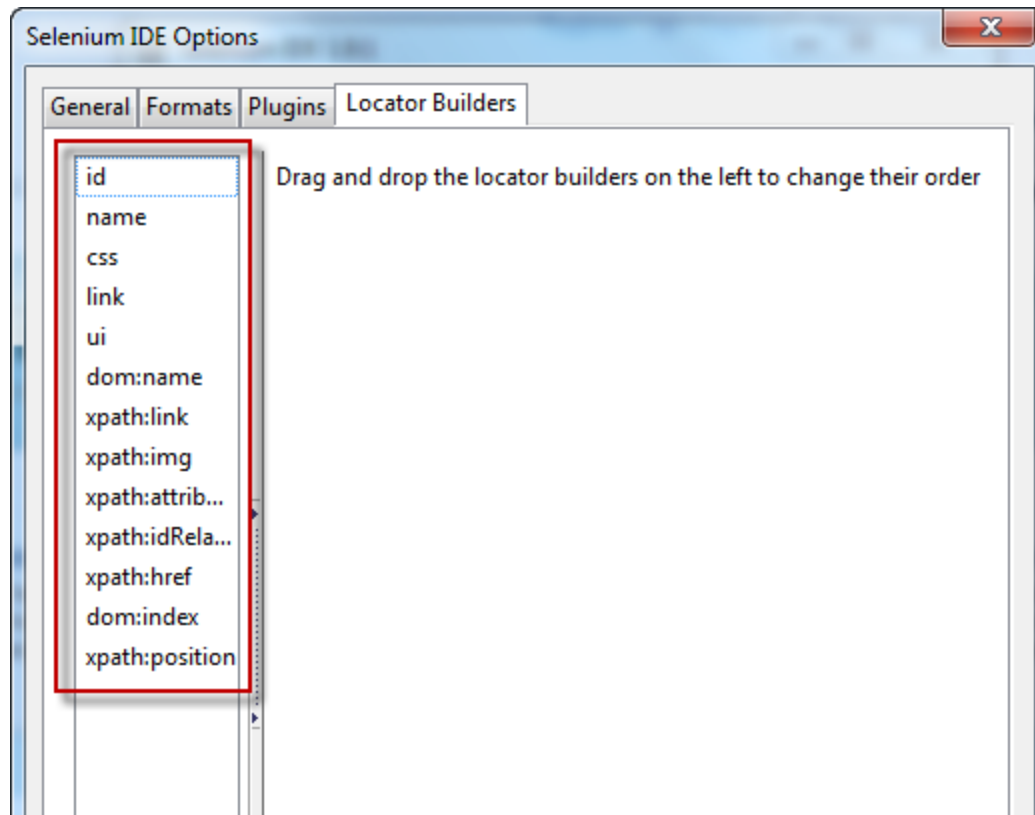
Selenium IDE

► Cấu hình



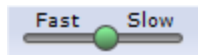
Selenium IDE

► Thứ tự ưu tiên Locator



Selenium IDE

► 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



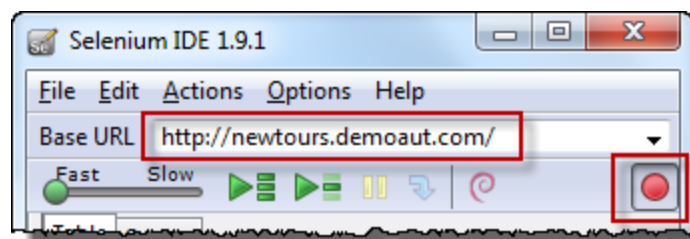
Nhóm các bước lại

Selenium IDE

▶ Ví dụ Record and Playback

▶ Bước 1:

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



▶ Bước 2:

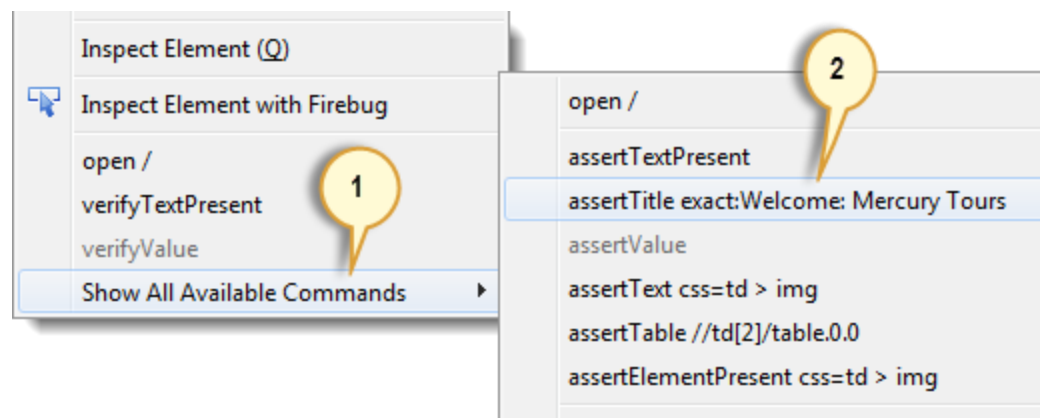
- ▶ Trong Firefox nhập Url <http://newtours.demoaut.com/>

Selenium IDE

▶ 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	

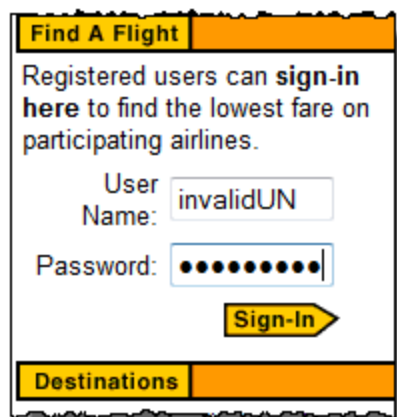
After clicking on the assertTitle context menu option, your Selenium IDE Editor pane should now contain the following commands

Selenium IDE

▶ Ví dụ Record and Playback

▶ Bước 4:

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



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”

Selenium IDE

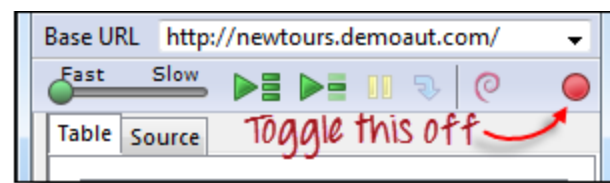
▶ Ví dụ Record and Playback

▶ 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

Selenium IDE

► 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

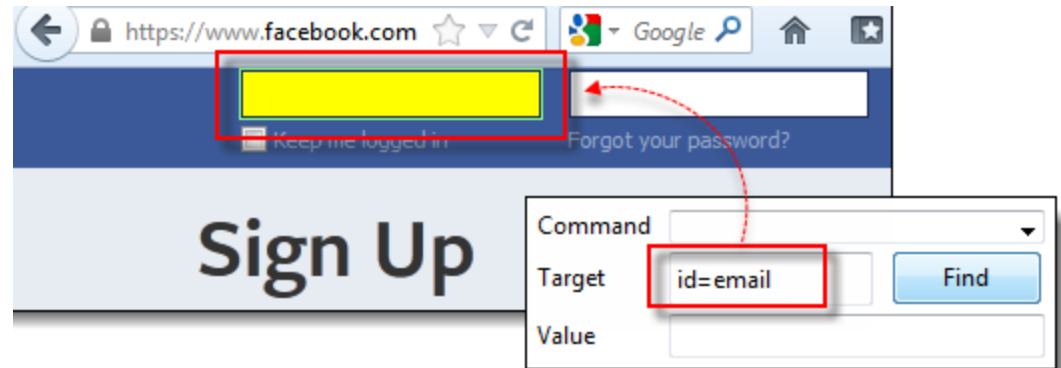
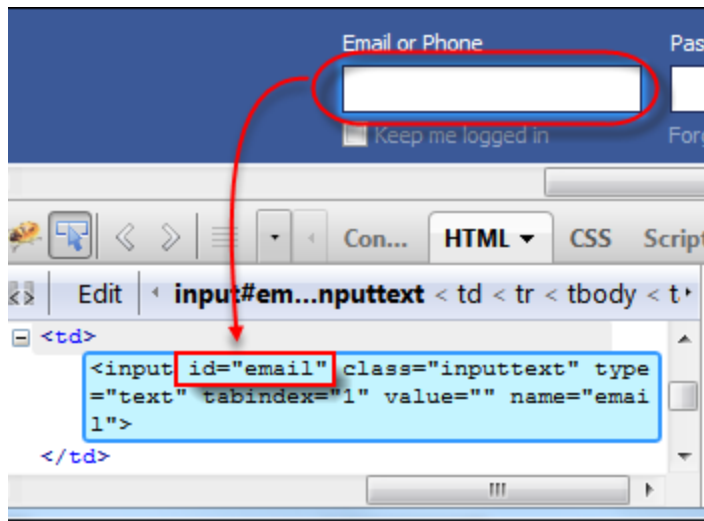
Selenium IDE

▶ 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

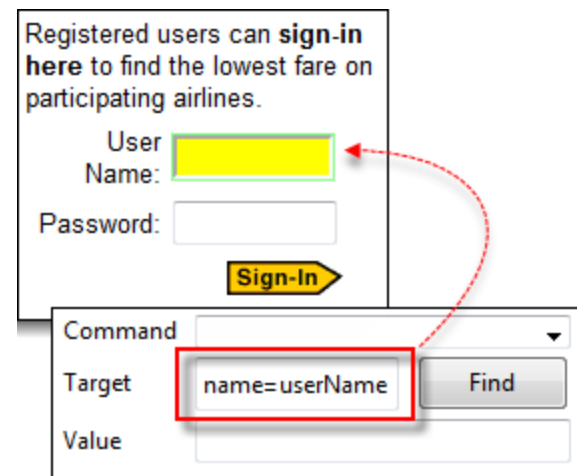
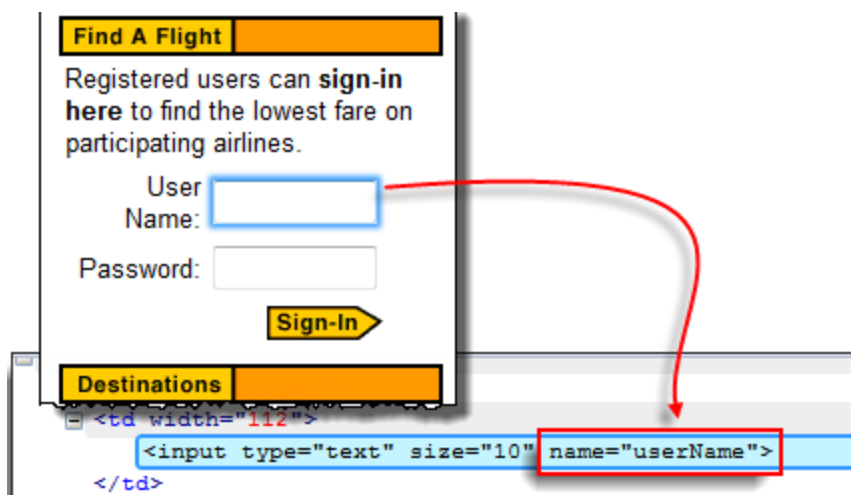
Selenium IDE

- ▶ 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: 1 ▼

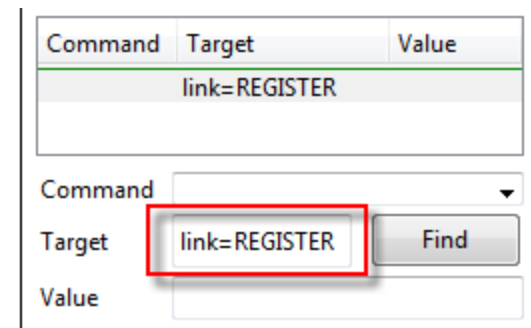
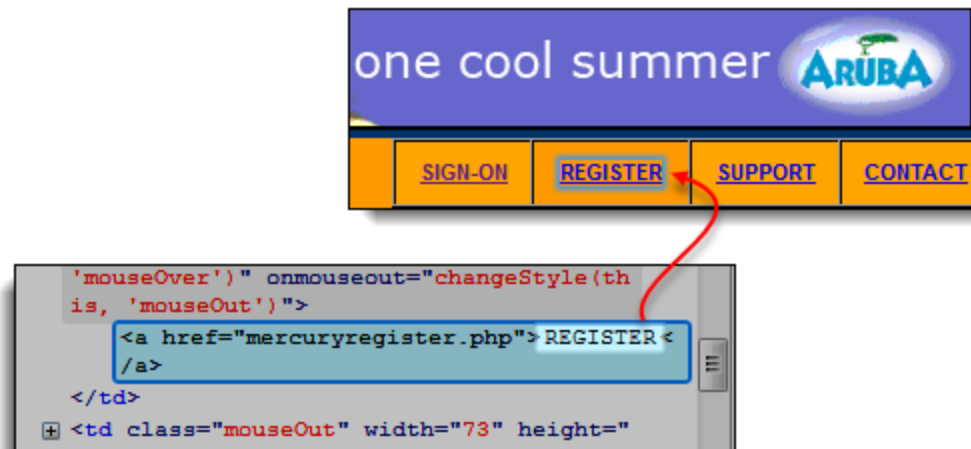
Departing From: Acapulco ▼

Mercury Tours Flight Finder page

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

Selenium IDE

- ▶ Xác định bằng Link Text
 - ▶ Target: link=link_text



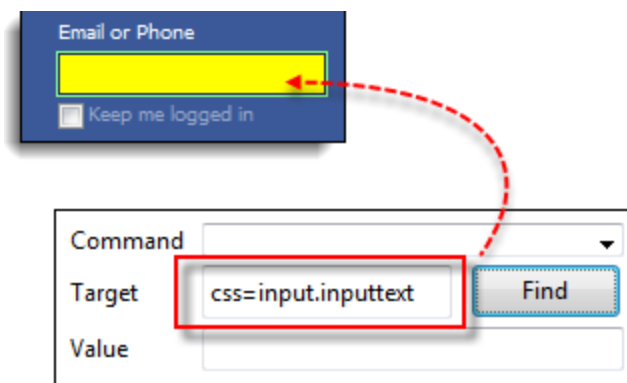
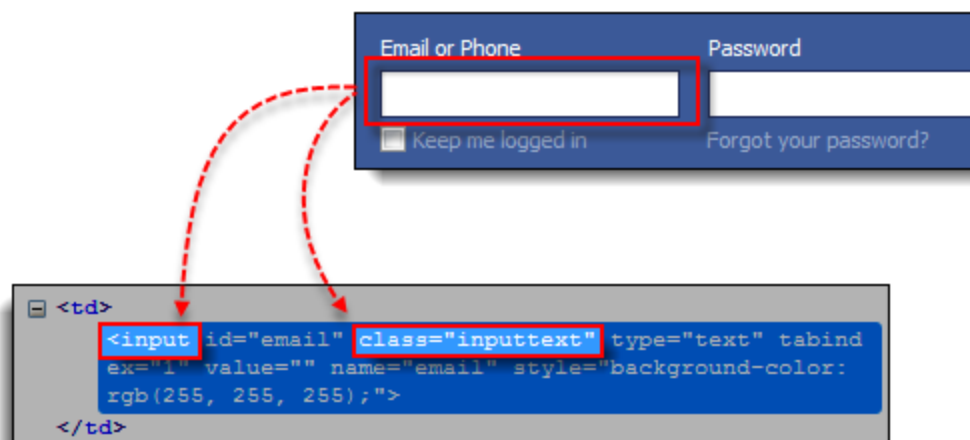
Selenium IDE

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

The diagram illustrates the process of identifying a web element for Selenium IDE using CSS. It 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 code, which is highlighted in a blue box: `<input id="email" class="inputtext" type="text" tabindex="1" value="" name="email">`. Another red dashed arrow points from the "Email or Phone" input field to the "Target" field in the Selenium IDE Command window, which contains the value `css=input#email`.

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")`

The diagram illustrates the process of selecting a target in Selenium IDE based on its inner text using CSS. It consists of three main parts:

- HTML Source:** A code snippet showing an HTML element:

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

 The text "Password:" is highlighted with a red box.
- Handwritten Note:** The text "This is the inner text" is written in blue, with a red dashed arrow pointing from it to the "Password:" text in the HTML source.
- Form Element:** A screenshot of a web form with two input fields: "User Name:" and "Password:". The "Password:" label is highlighted with a yellow box.
- Selenium IDE Command:** A screenshot of the Selenium IDE command window. The "Target" field contains the CSS selector `css=font:contains("Password:")`, which is highlighted with a red box. A red dashed arrow points from this box to the "Password:" label in the form element above.

Selenium IDE

► 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=<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</i>.<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	

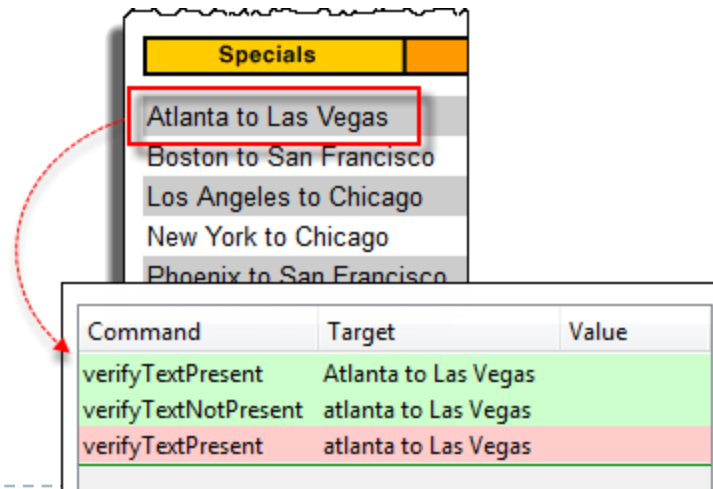
Selenium IDE

▶ verifyTextPresent

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

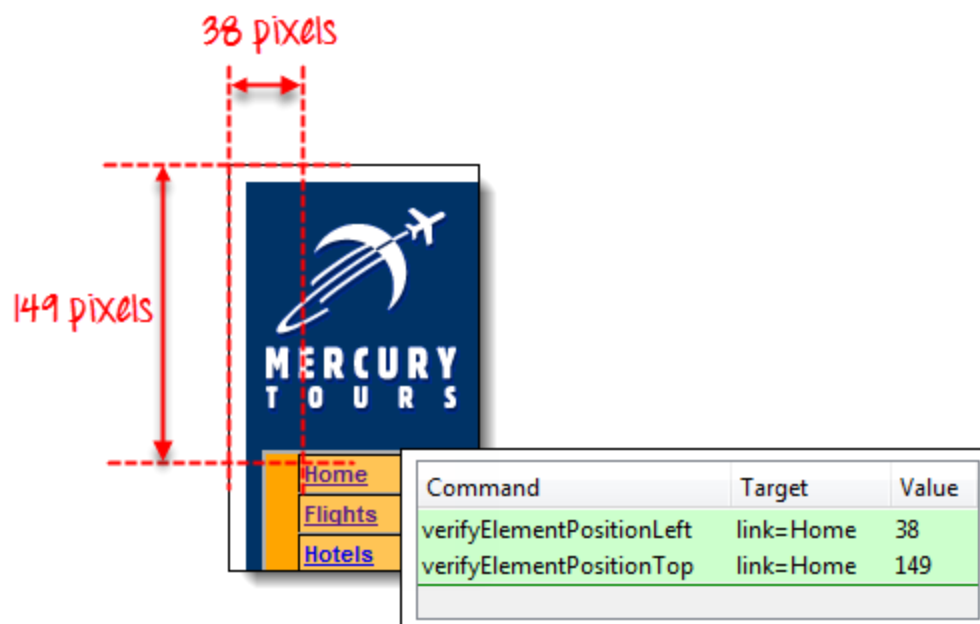
▶ verifyTextNotPresent

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



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.

Selenium IDE

- ▶ **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 a Facebook sign-up form. On the left, a modal box with a red border contains text about providing a birthday and a 'Not creating a personal account?' section. On the right, the 'Birthday:' section has dropdowns for Month, Day, and Year, radio buttons for Female and Male, and a 'Sign Up' button. A blue box highlights the link 'Why do I need to provide my birthday?' which is pointed to by a blue dashed arrow from the text above. A red dashed arrow points from the same text to the modal box.

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

Selenium IDE

► Lưu giá trị vào biến

value of the variable

Command	Target	Value
store	tutorial	myVariable

name of the variable

User Name:

Password:

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

Selenium IDE

► 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

- ▶ 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



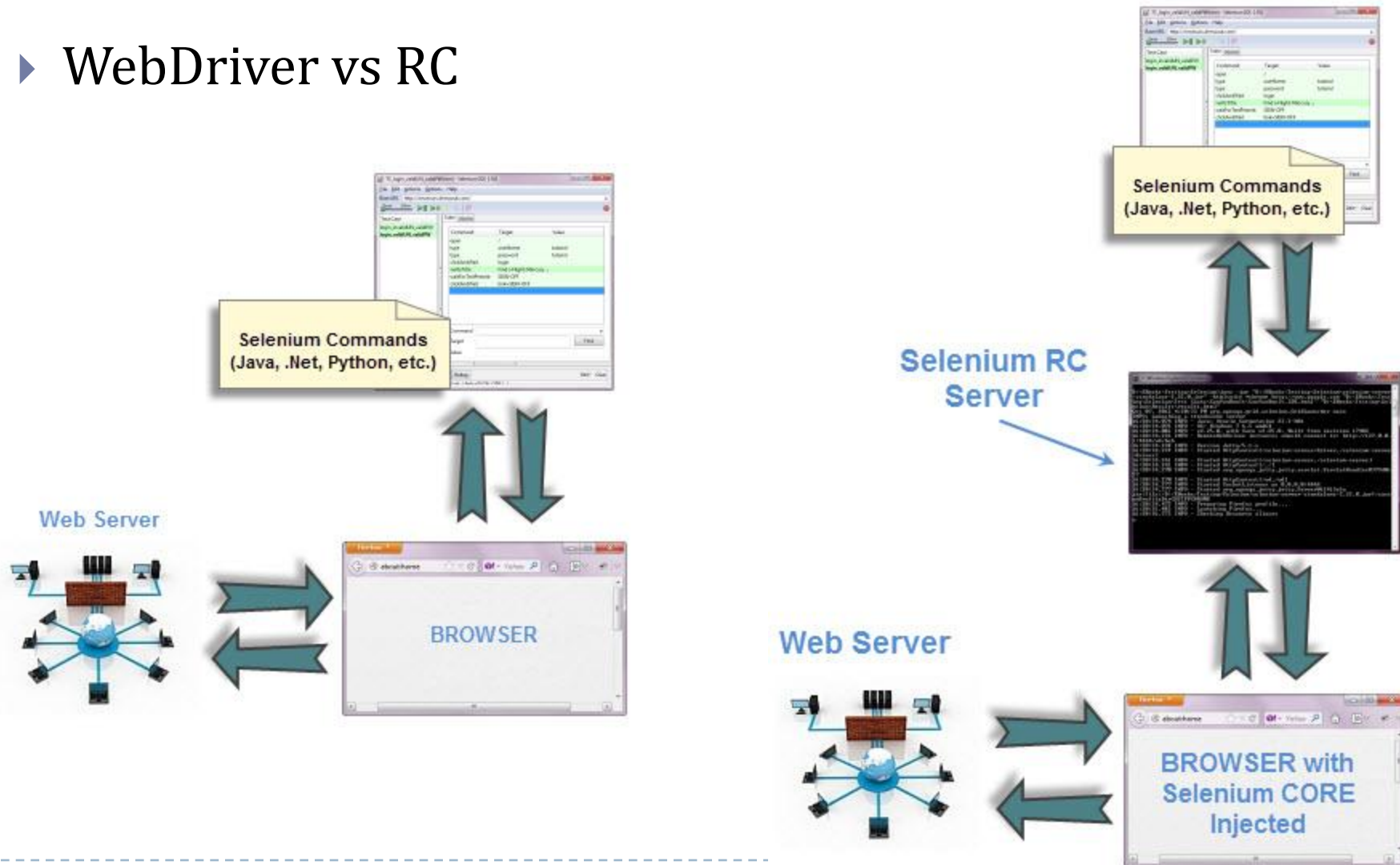
Selenium WebDriver

► 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	Download	Change log	Javadoc
C#	2.32.0	2013-04-09	Download	Change log	API docs
Ruby	2.32.0	2013-04-09	Download	Change log	API docs
Python	2.32.0	2013-04-10	Download	Change log	API docs

- ▶ 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 / WebDriver

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

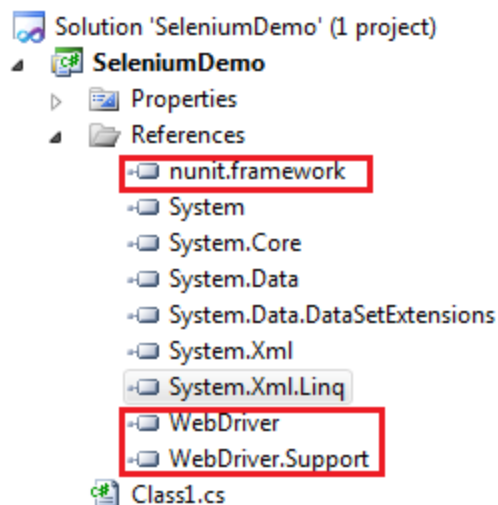
        [SetUp]
        public void SetupTest()
        {
            driver = new FirefoxDriver();
            baseURL = "http://newtours.demoaut.com/";
            verificationErrors = new StringBuilder();
        }
    }
}
```

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



Selenium WebDriver

- ▶ 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);
}
```

Selenium WebDriver

- ▶ 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

► 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

