



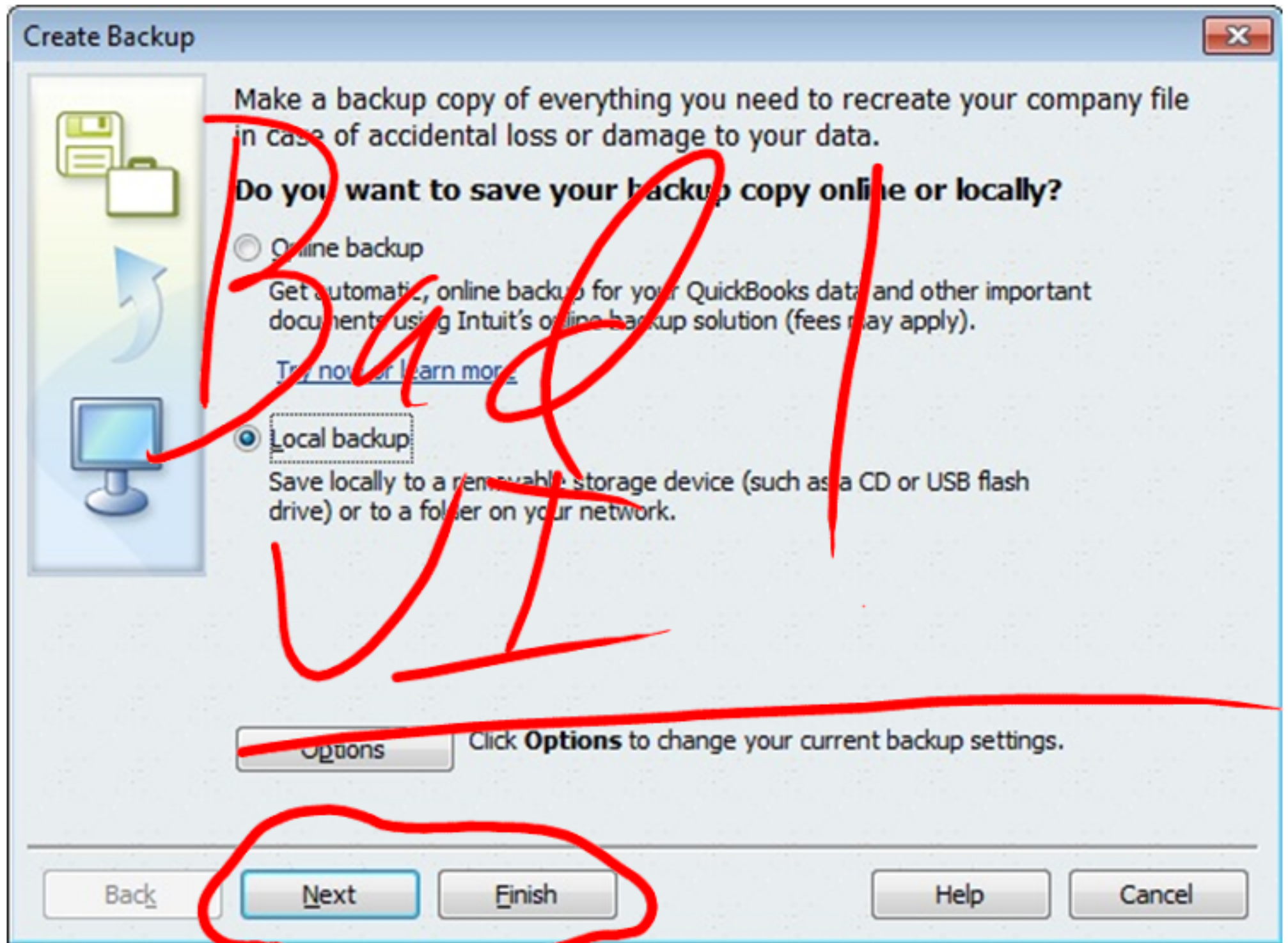
# Web Testing with Selenium

*VI testing*

## Lecture Objectives:

- 1) Construct a simple web test using the Selenium tool.
- 2) Execute test scripts using Selenium
- 3) Modify test scripts (if necessary) by editing the selenium generated HTML code

What is wrong with this screen?



# Why do we automate

- Manual testing is tedious & expensive
- Automate the process w/ regression test suites.

# Why do we automate testing?

- ➔ Reliable: Tests perform precisely the same operations each time they are run, thereby eliminating human error.
- Repeatable: You can test how the software reacts under repeated execution of the same operations.
- Reusable: You can reuse tests on different versions of an application, even if the user interface changes. ⇒ "Maybe"
- Speed: Run tests significantly faster than human users.
- Cost Reduction: As the number of resources for regression test are reduced. ⇒ Save \$
- Better Quality Software: Rapid feedback to developers during the development process as a result of frequent regression testing.

Maybe



Why might automation be

bad?

Over head w/ automated testing.

---

May not be able to detect everything that is wrong.

# Why might automation be

bad?

- Proficiency is required to write the automation test script.
- Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
- Test maintenance is costly in case of playback methods. Even though a minor changes occurs in the GUI, the test script has to be rewritten.
- Maintenance of test data files is difficult, if the test script tests more screens or web-pages.
- Short iteration or very tight deadline, there is not enough time to build test automation.

\$\$\$

# When not to automate?

([http://docs.seleniumhq.org/docs/01\\_introducing\\_selenium.jsp#to-automate-or-not-to-automate](http://docs.seleniumhq.org/docs/01_introducing_selenium.jsp#to-automate-or-not-to-automate))

- “It is not always advantageous to automate test cases. There are times when manual testing may be more appropriate. For instance, if the application’s user interface will change considerably in the near future, then any automation might need to be rewritten anyway. Also, sometimes there simply is not enough time to build test automation. For the short term, manual testing may be more effective. If an application has a very tight deadline, there is currently no test automation available, and it’s imperative that the testing get done within that time frame, then manual testing is the best solution.”



# Periodic Table of the Elements

The image shows a standard periodic table of elements. The element Selenium (Se) is highlighted in green and circled in yellow. It is located in the 4th period and 16th group. The table includes the following series:

- Lanthanide Series (57-71)
- Actinide Series (89-103)

Legend for element categories:

- Alkali Metal (Pink)
- Alkaline Earth (Light Blue)
- Transition Metal (Light Green)
- Base Metal (Light Orange)
- Semimetals (Light Yellow)
- Nonmetals (Light Purple)
- Halogens (Light Blue)
- Noble Gas (Light Green)
- Lanthanides (Light Orange)
- Actinides (Light Purple)

What is Selenium?

Atomic #34

Antidote for mercury poisoning







# What is Selenium

A set of tools that supports rapid development of test automation for web-based applications.

- Can be recorded and written as HTML
- Support for a number of programming languages: Java, C#, Perl, PHP, Python, Ruby *⇒ others*
- Cross browsers support
  - IE, Firefox, Opera, Safari and Google Chrome *Most mature*
- Cross platform support
  - Windows, Linux, and Macintosh.

# History

- Invented in 2004 by Jason R. Huggins and team. *"old tool"*
- Originally named JavaScript Functional Tester [JSFT]
- 100% Javascript and HTML
- Designed to make test writing easy *Firefox*
- Open source browser based integration test framework built originally by ThoughtWorks
- Selenium is open source software, released under the Apache 2.0 license and can be downloaded and used without charge.

# selenium Structure

Which part of Selenium is appropriate for me?



If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing

Then you want to use [Selenium IDE](#); a Firefox add-on that will do simple record-and-playback of interactions with the browser.

↑  
Selenium  
IDE



If you want to

- create robust, browser-based regression automation
- scale and distribute scripts across many environments

Then you want to use [Selenium WebDriver](#); a collection of language specific bindings to drive a browser -- the way it is meant to be driven.

Selenium WebDriver is the successor of [Selenium Remote Control](#) which has been officially deprecated. The Selenium Server (used by both WebDriver and Remote Control) now also includes built-in grid capabilities.

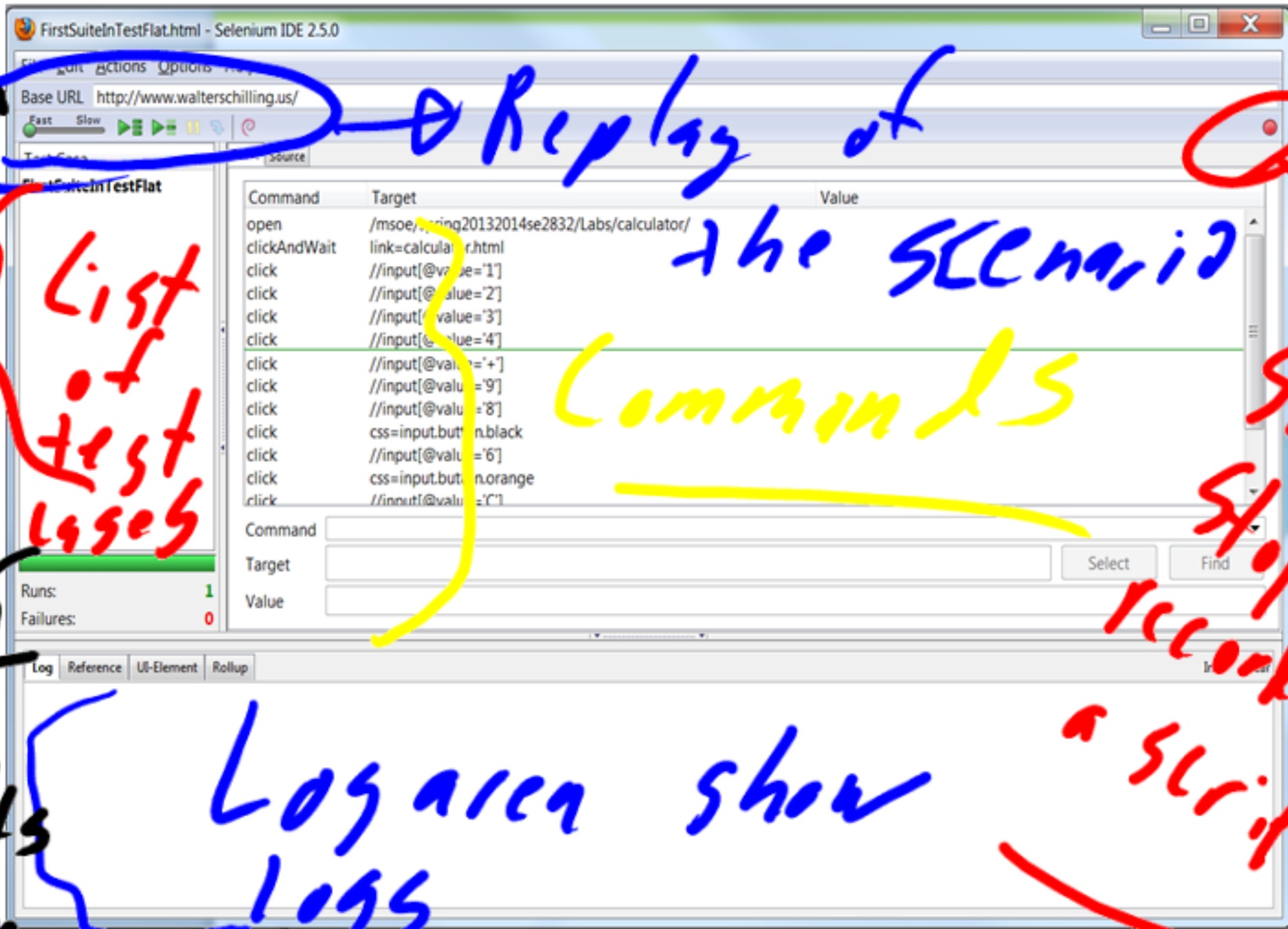
↑  
selenium webdriver

← Harder  
more complex





# Selenium IDE - UI



Base URL of the site to test

# Recording a test script

- Open Firefox that has the IDE installed ✓
- Open the base URL of the application to record. ✓
- Keep the application in a common base state. ✓
- Go To Tools → Selenium IDE and the IDE will be opened ✓
- Now perform the operations on the application as you are testing the application. }
- Once you are done with the recording click on the stop recording button and save the test case through the file menu. By default it will be saved as a selenese script (HTML format)

↳ HTML Comments

# Some commands that can

## occur

- clicking a link - *click* or *clickAndWait* commands
- entering values - *type* command *type in a dialog box*
- selecting options from a drop-down listbox - *select* command
- clicking checkboxes or radio buttons - *click* command



Lets take a look at an example

Record demo for website...

- Testing a basic web calculator

