Selenium 2 Workshop

Matt Stine

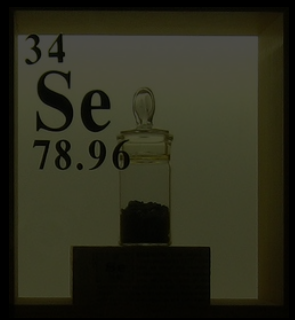
bash-3.2$ whoami

- Technical Architect - AutoZone
- Speaker (JavaOne, SpringOne/2GX, CodeMash, NFJS, RWX, PAX, UberConf)
- Author (GroovyMag, NFJS the Magazine, Selenium 2.0 Refcard)
- Founder of the Memphis/Mid-South Java User Group
- Former Agile Zone Leader @ DZone

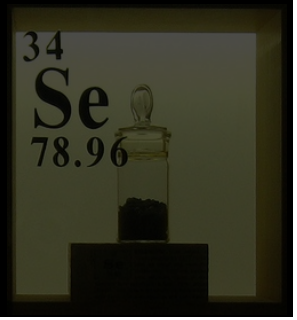
AutoZone

DZone AGILE ZONE
Software Methodologies for Development Managers

What is Selenium 2?



History



Selenium @ ThoughtWorks

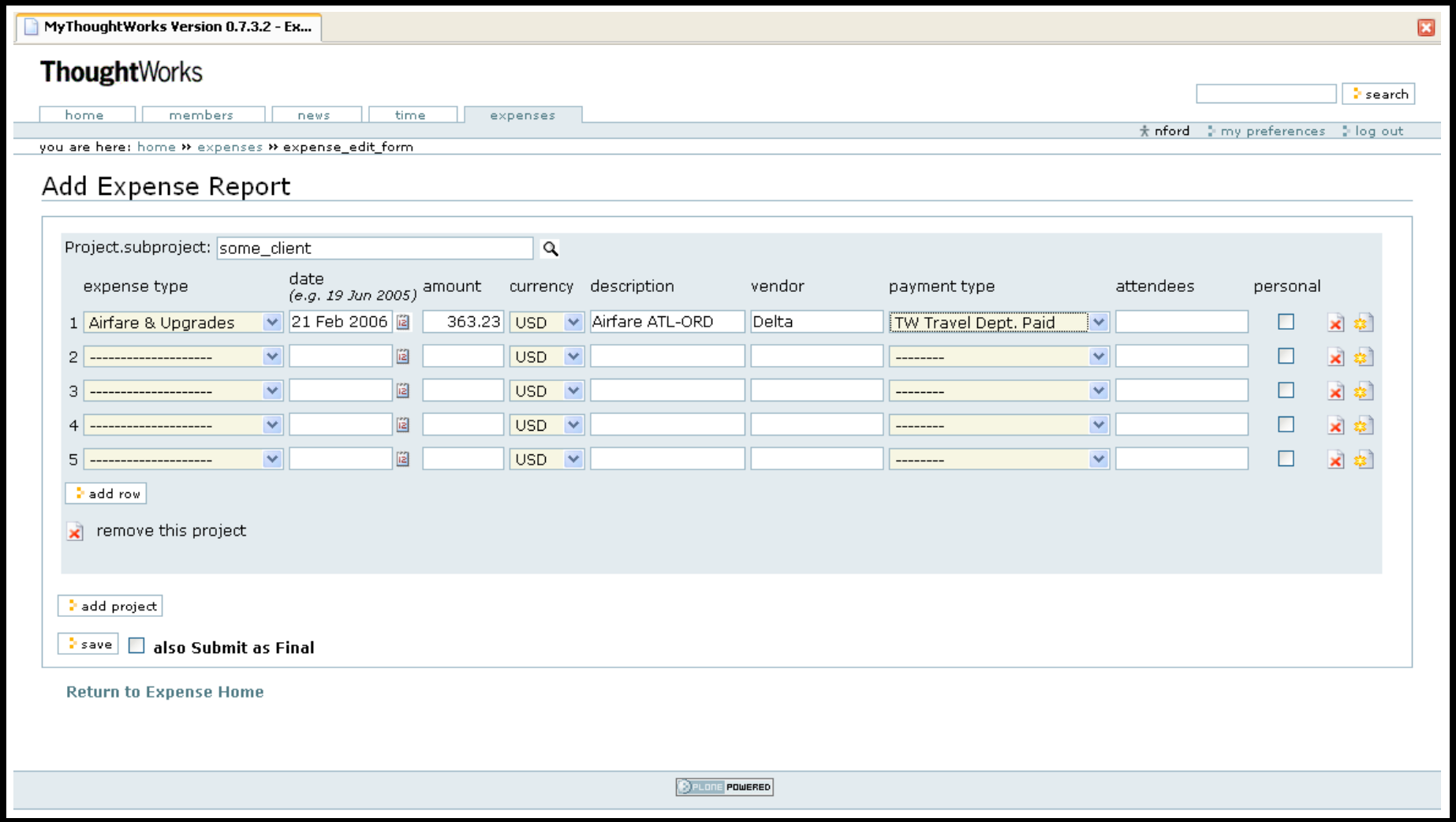


Image: Neal Ford, “Introdution to Selenium”


WARNING
Sudbury River Fish
Fish Contaminated With Mercury
DO NOT EAT
(Spanish)
Pescado Contaminado Con Mercurio
No Se Puede Comer
(Cambodian)
Massachusetts Department of Public Health
617-727-0201
Massachusetts Department of Environmental Protection
617-
5851
United States Enviro tal Protection Agency
617-565-2713
AHS


Se . 34
Selenium
(Grey)

Ecohealth. 2008 Dec;5(4):456-9. Epub 2009 Feb 6.

Mercury toxicity and the mitigating role of selenium.

Berry MJ, Ralston NV.

Department of Cell and Molecular Biology, John A. Burns School of Medicine, University of Hawaii at Manoa, Biomedical Sciences Building, Suite 222, 651 Ilalo Street, Honolulu, 96813 HI, USA.
mberry@hawaii.edu

Abstract

Mercury is a well-known environmental toxicant, particularly in its most common organic form, methylmercury. Consumption of fish and shellfish that contain methylmercury is a dominant source of mercury exposure in humans and piscivorous wildlife. Considerable efforts have focused on assessment of mercury and its attendant risks in the environment and food sources, including the studies reported in this issue. However, studies of mercury intoxication have frequently failed to consider the protective effects of the essential trace element, selenium. Mercury binds to selenium with extraordinarily high affinity, and high maternal exposures inhibit selenium-dependent enzyme activities in fetal brains. However, increased maternal dietary selenium intakes preserve these enzyme activities, thereby preventing the pathological effects that would otherwise arise in their absence. Recent evidence indicates that assessments of mercury exposure and tissue levels need to consider selenium intakes and tissue distributions in order to provide meaningful risk evaluations.

http://www.ncbi.nlm.nih.gov/pubmed/19198945

Google Open Source at Google

News about Google's Open Source projects and programs

Introducing WebDriver

Friday, May 8, 2009 | 4:45 PM

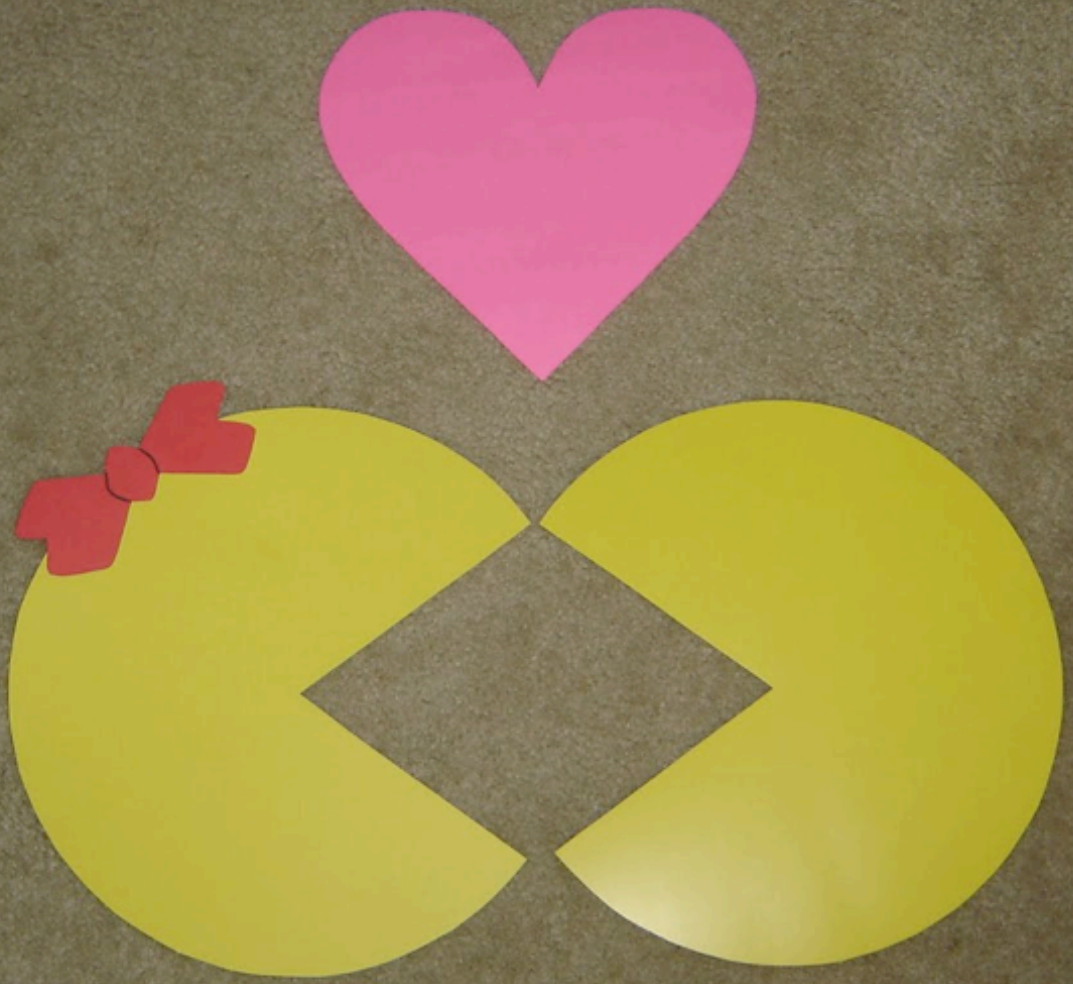
Labels: releases, selenium, webdriver

WebDriver is a clean, fast framework for automated testing of webapps. Why is it needed? And what problems does it solve that existing frameworks don't address?

For example, Selenium, a popular and well established testing framework is a wonderful tool that provides a handy unified interface that works with a large number of browsers, and allows you to write your tests in almost every language you can imagine (from Java or C# through PHP to Erlang!). It was one of the first Open Source projects to bring browser-based testing to the masses, and because it's written in JavaScript it's possible to quickly add support for new browsers that might be released

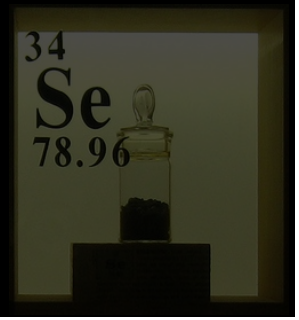
Like every large project, it's not perfect. Selenium is written in JavaScript which causes a significant weakness: browsers impose a pretty strict security model on any JavaScript that they execute in order to protect a user from malicious scripts. Examples of where this security model makes testing

August 2009: They Meet

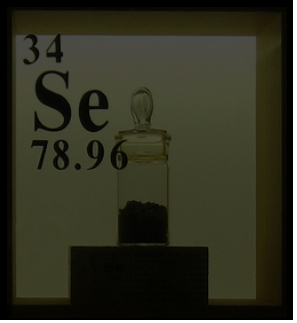


The Prerelease Years

- December 2009 - Alpha 1
- December 2010 - Beta 1

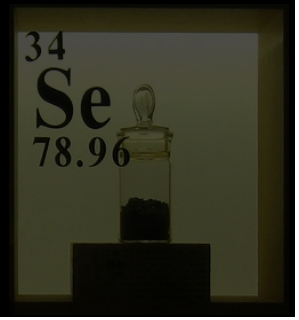


JULY 2011:
It's Alive!

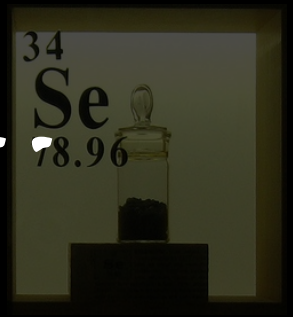


Differences from Original Selenium

- Object-based API for DOM Interaction
- Browser Driver Implementations
- Advanced User Interactions

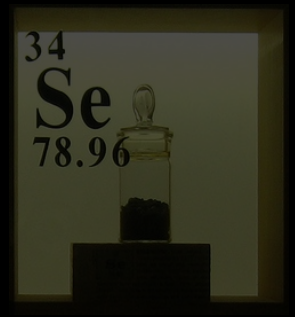


But you haven't lost an old friend...



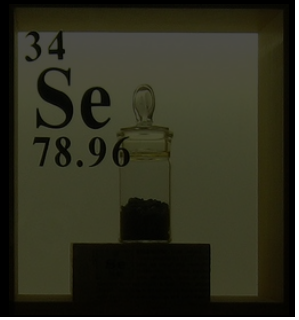
Language Support

- Python
- Ruby
- Java (JVM)
- C#



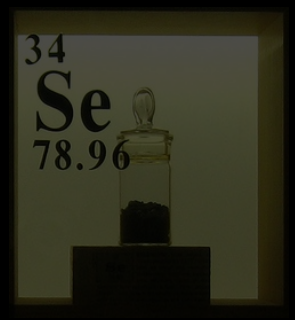
Browser Support

- Firefox
- Chrome
- Internet Explorer
- Opera
- Headless (HtmlUnit)



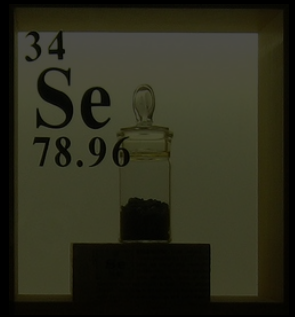
Mobile Support!

- iOS
- Android

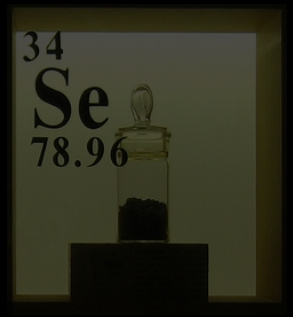


Tool Chain

- Selenium IDE
- Browser Drivers
- RemoteWebDriver
- Selenium Server (Selenium RC/ WebDriver)
- Selenium Grid

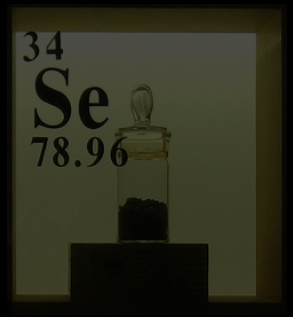


Drivers



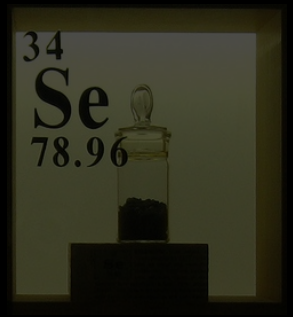
HtmlUnit Driver

- Fastest, most lightweight
- Headless (no browser GUI)
- Pure Java solution
- Supports (*but emulates behavior of*) JavaScript



Firefox Driver

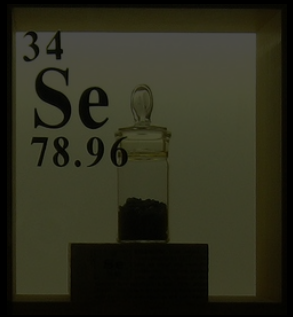
- Controls FF using a plugin
- Profile stripped down to only WebDriver.xpi
- Win/Mac/Linux - 3,3.6,5,6,7,8
- Faster than IE, slower than HtmlUnit



Internet Explorer Driver

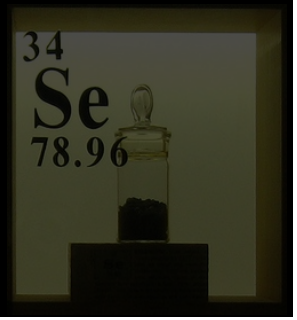
- Controlled via .dll (Windows only!)
- Windows XP - 6,7,8
- Windows 7 - 9
- Comparatively slow
- XPath via Sizzle in most versions
- CSS via Sizzle in 6 & 7
- CSS native in 8 & 9 (but CSS3 not fully supported)

http://sizzlejs.com



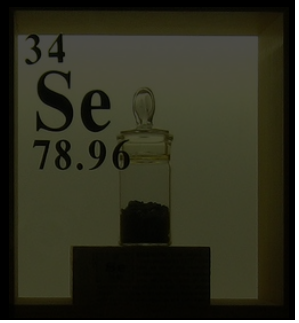
Chrome Driver

- Maintained by Chromium project (http://www.chromium.org/)
- Get ChromeDriverServer from http://www.chromium.org/developers/testing/webdriver-for-chrome
- Potentially useful for Safari compatibility as well (but remember Nitro vs. V8!)



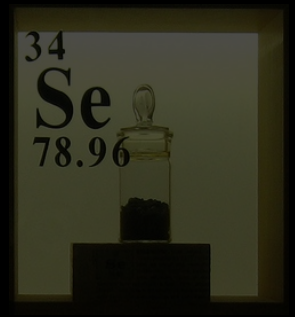
Opera Driver

- Developed/supported by Opera Software
- Communicates via “Scope Interface” directly from Java
- Opera 11.5+



iPhone Driver

- iOS application:
  - Implements JSON-based wire protocol
  - Drives UIWebView (WebKit browser embeddable in iOS applications)
- Need iOS SDK and provisioning profile (for testing on device)



iphoneWd.groovy

```groovy
@Grapes([
  @Grab("org.seleniumhq.selenium:selenium:latest.release")
])
import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.remote.RemoteWebDriver
import org.openqa.selenium.remote.DesiredCapabilities

def driver = new RemoteWebDriver(new URL("http://localhost:3001/hub"),
                    DesiredCapabilities.iphone())

driver.get("http://www.google.com")
def element = driver.findElement(By.name("q"))

element.sendKeys("Cheese!")
element.submit()

println("Page title is: " + driver.getTitle())
```

Running "iphoneWd.groovy"...

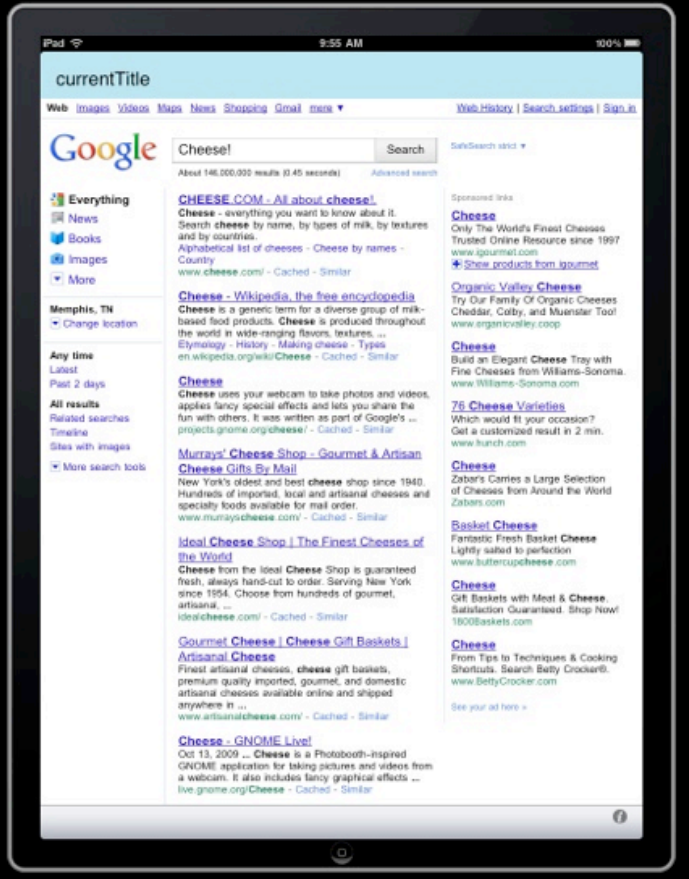
Running "iphoneWd.groovy"...

Groovy Version: 1.7.2 JVM: 1.6.0_20

```
Page title is: Cheese! - Google Search
```

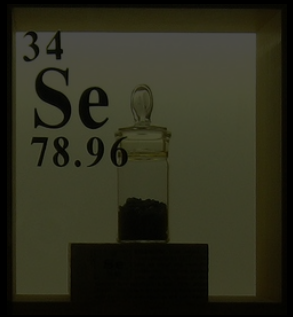
Program exited with code #0 after 9.14 seconds. copy output

iPhone Simulator



Android Driver

- Android application:
  - Implements JSON-based wire protocol
  - Drives Android WebView (WebKit browser embeddable in iOS applications)
- Uses native touch and key events
- Current: Honeycomb SDK 3.1 onwards through ICS
- Can use RemoteWebDriver server or Android test framework



androidWd.groovy

```groovy
import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.openqa.selenium.WebElement
import org.openqa.selenium.android.AndroidDriver

def driver = new AndroidDriver()

driver.get("http://www.google.com")
def element = driver.findElement(By.name("q"))

element.sendKeys("Cheese!")
element.submit()

println("Page title is: " + driver.getTitle())
```

Running "androidWd.groovy"...

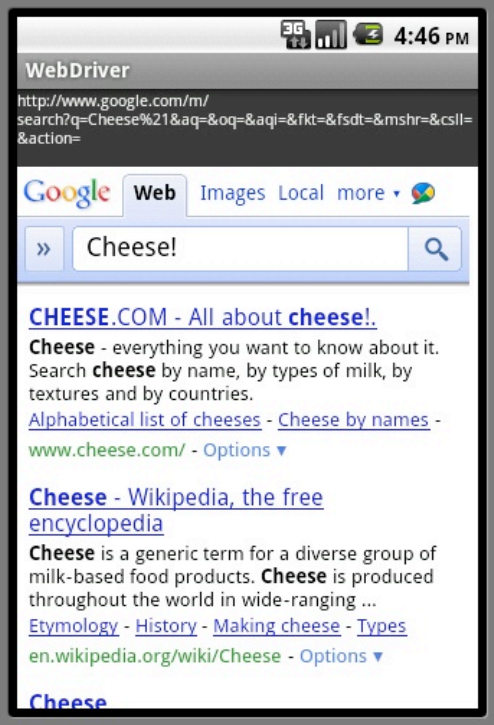
Running "androidWd.groovy"...

Groovy Version: 1.7.2 JVM: 1.6.0_20

```
Page title is: Cheese! - Google Search
```

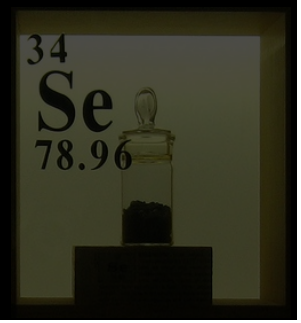
Program exited with code #0 after 29.42 seconds. copy output

5554:my_android

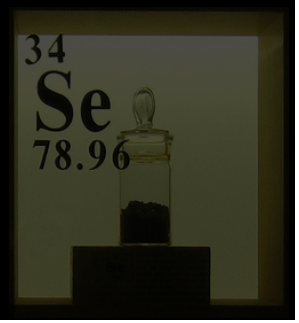


Selenium RC Emulation

```
WebDriver driver = new FirefoxDriver();

String baseUrl = "http://downforeveryoneorjustme.com/";
Selenium selenium = new WebDriverBackedSelenium(driver, baseUrl);

selenium.open("http://downforeveryoneorjustme.com/");
selenium.type("domain_input", "google.com");
selenium.click("//div[@id='container']/form/a");

assertTrue(selenium.isTextPresent("It's just you"));
```

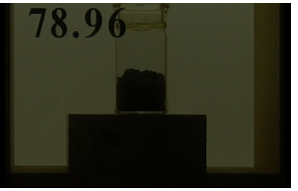


Page Interaction API



Locators

Locator	Example (Java)
id attribute	By.id("myElementId")
name attribute	By.name("myElementName")
XPATH	By.xpath("//input[@id='myElementId']")
Class name	By.className("even-table-row")
CSS Selector	By.cssSelector("h1[title]")
Link Text	By.linkText("Click Me!") By.partialLinkText("ck M")
Tag Name	By.tagName("td")



WebElement

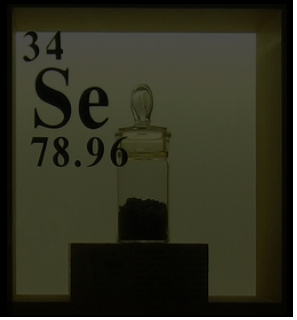
Method	Purpose
clear()	Clears all of the contents if the element is a text entity.
click()	Simulates a mouse click on the element.
getAttribute(String name)	Returns the value associated with the provided attribute name (if present) or null (if not present).
getCssValue(String propertyName)	Returns the value of the provided property.
getLocation()	Returns a Selenium Point (copy of the java.awt.Point API) representing the top left-hand corner of the element.
getSize()	Returns a Selenium Dimension (copy of the java.awt.Dimension API) representing the width and height of the element.
getTagName()	Returns the tag name for this element.

WebElement

Method	Purpose
getText()	Returns the visible text contained within this element (including subelements) if not hidden via CSS.
isDisplayed()	Returns true if the element is currently displayed; otherwise false.
isEnabled()	Returns true for input elements that are currently enabled; otherwise false.
isSelected()	Returns true if the element (radio buttons, options within a select, and checkboxes) is currently selected; otherwise false.
sendKeys(CharSequence... keysToSend)	Simulates typing into an element.
submit()	Submits the same block if the element if the element is a form (or contained within a form). Blocks until new page is loaded.

WebElement

Method	Purpose
findElement(By by)	Finds the first element located by the provided method (see Locators table).
findElements(By by)	Finds all elements located by the provided method.

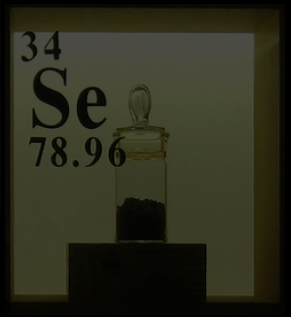


Select Support Class

Method	Purpose
selectByIndex(int index)/ deselectByIndex(int index)	Selects/deselects the option at the given index.
selectByValue(String value)/ deselectByValue(String value)	Selects/deselects the option(s) that has a value matching the argument.
selectByVisibleText(String text)/ deselectByVisibleTest(String text)	Selects/deselects the option(s) that displays text matching the argument.
deselectAll()	Deselects all options.
getAllSelectedOptions()	Returns a List<WebElement> of all selected options.
getFirstSelectedOption()	Returns a WebElement representing the first selected option.
getOptions()	Returns a List<WebElement> of all options.
isMultiple()	Returns true if this is a multi-select list; false otherwise.

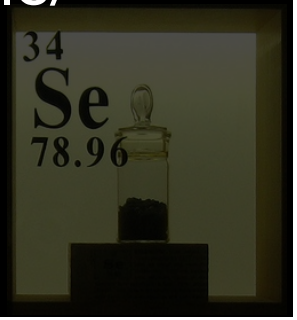
The Page Object Pattern

http://www.flickr.com/photos/essjay/3227454514/

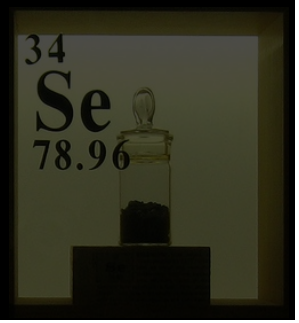


what / why

- Two orthogonal concerns:
  - Logical application functionality
  - Mechanics of DOM interaction
- “An API to an HTML page...”
- Promotes DRY tests and reuse
- “Peanut Brittle Insurance”
- Also known as “Window Driver Pattern” (Humble/Farley, Continuous Delivery, p.201-204)



PageFactory



```
public class GoogleHomePage {
  private final WebDriver driver;

  public GoogleHomePage(WebDriver driver) {
    this.driver = driver;

    //Check that we're actually on the Google Home Page.
    if (!"Google".equals(driver.getTitle())) {
      throw new IllegalStateException("This isn't Google's Home Page!");
    }
  }

  public GoogleResultsPage search(String searchTerms) {
    driver.findElement(By.name("q")).sendKeys(searchTerms);
    driver.findElement(By.name("btnG")).submit();

    return new GoogleResultsPage(driver);
  }

  public UnknownTopHitPage imFeelingLucky(String searchTerms) {
    driver.findElement(By.name("q")).sendKeys(searchTerms);
    driver.findElement(By.name("btnI")).submit();

    return new UnknownTopHitPage(driver);
  }
}
```

```java
public class GoogleHomePage {
  private final WebDriver driver;

  private WebElement q;
  private WebElement btnG;
  private WebElement btnI;

  public GoogleHomePage(WebDriver driver) {
    this.driver = driver;

    //Check that we're actually on the Google Home Page.
    if (!"Google".equals(driver.getTitle())) {
      throw new IllegalStateException("This isn't Google's Home Page!");
    }
  }

  public GoogleResultsPage search(String searchTerms) {
    q.sendKeys(searchTerms);
    btnG.submit();
    return new GoogleResultsPage(driver);
  }

  public UnknownTopHitPage imFeelingLucky(String searchTerms) {
    q.sendKeys(searchTerms);
    btnI.submit();
    return new UnknownTopHitPage(driver);
  }
}
```

```
WebDriver driver = new HtmlUnitDriver();
driver.get("http://www.google.com");
GoogleHomePage page = PageFactory.initElements(driver,
                                               GoogleHomePage.class);
page.search("WebDriver");
```

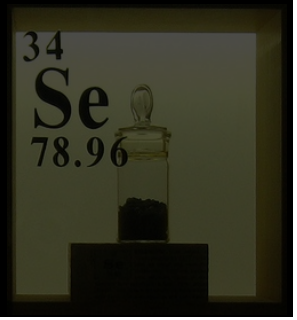
```
AnotherPageObject page = AnotherPageObject("testing",
                                           1, 2, 3, driver);
PageFactory.initElements(driver, page);
```

Annotations

Locator	Examples
id attribute	@FindBy(how = How.ID, using = "myElementId") @FindBy(id = "myElementId)
name attribute	@FindBy(how = How.NAME, using = "myElementName") @FindBy(name = "myElementName")
id or name attribute	@FindBy(how = How.ID_OR_NAME, using = "myElement")
XPATH	@FindBy(how = How.XPATH, using = "//input[@id='myElementId']") @FindBy(xpath = "//input[@id='myElementId']")
Class name	@FindBy(how = How.CLASS_NAME, using="even-table-row") @FindBy(className = "even-table-row")
Link Text	@FindBy(how = How.LINK_TEXT, using="Click Me!") @FindBy(linkText = "Click Me!")
Partial Link Text	@FindBy(how = How.PARTIAL_LINK_TEXT, using="ck M") @FindBy(partialLinkText = "ck M")
Tag Name	@FindBy(how = How.TAG_NAME, using="td") @FindBy(tagName = "td")

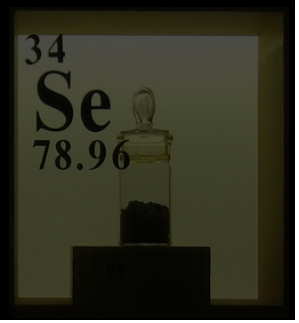
```java
public class GoogleHomePage {
  private final WebDriver driver;

  @FindBy(name = "q") private WebElement searchBox;
  @FindBy(name = "btnG") private WebElement searchButton;
  @FindBy(name = "btnI") private WebElement imFeelingLuckyButton;

  public GoogleHomePage(WebDriver driver) {
    this.driver = driver;

    //Check that we're actually on the Google Home Page.
    if (!"Google".equals(driver.getTitle())) {
      throw new IllegalStateException("This isn't Google's Home Page!");
    }
  }

  public GoogleResultsPage search(String searchTerms) {
    searchBox.sendKeys(searchTerms);
    searchButton.submit();
    return new GoogleResultsPage(driver);
  }

  public UnknownTopHitPage imFeelingLucky(String searchTerms) {
    searchBox.sendKeys(searchTerms);
    imFeelingLuckyButton.submit();
    return new UnknownTopHitPage(driver);
  }
}
```

LoadableComponent



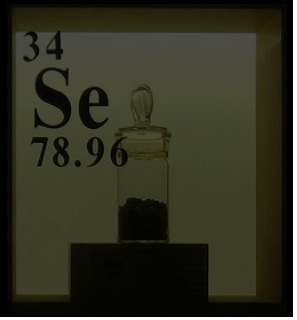
```java
public class HomePage extends LoadableComponent<HomePage> {

    private WebElement findAFluffbox;

    private WebDriver driver;
    private String baseUrl;

    public HomePage(WebDriver driver, String baseUrl) {
        this.driver = driver;
        this.baseUrl = baseUrl;
        PageFactory.initElements(driver, this);
    }

    @Override
    protected void load() {
        driver.get(baseUrl + "/fluffbox-rwx/");
    }

    @Override
    protected void isLoaded() throws Error {
        Assert.assertEquals("Fluffbox", driver.getTitle());
    }
}
```

Explicit/Implicit Waits



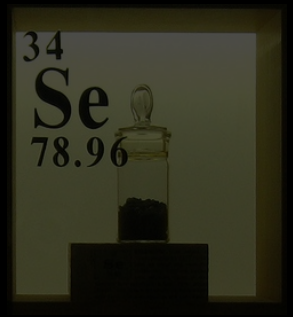
Replace This...

```
WebElement findSpeakersHereLink = null;

for (int second = 0;; second++) {
  if (second >= 60)
    fail("timeout");
  try {
     findSpeakersHereLink = driver.findElement
            (By.linkText("Find Speakers Here"));
     break;
  } catch (Exception e) {}
  Thread.sleep(1000);
}

findSpeakersHereLink.click();
```



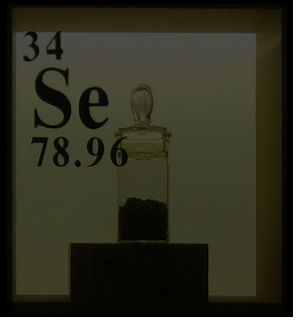
...with this...

```
WebElement findSpeakersHereLink = (new WebDriverWait(driver, 60, 1))
  .until(new ExpectedCondition<WebElement>(){
    @Override
    public WebElement apply(WebDriver d) {
      return d.findElement(By.linkText("Find Speakers Here"));
    }});

findSpeakersHereLink.click();
```

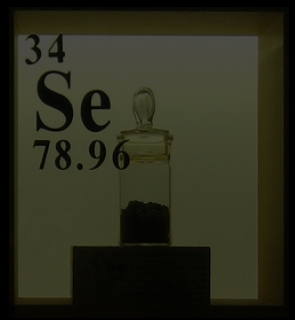


...or this.

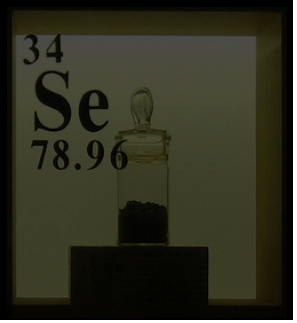
```
driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
WebElement findSpeakersHereLink = driver.findElement
  (By.linkText("Find Speakers Here"));

findSpeakersHereLink.click();
```



Advanced User Interactions

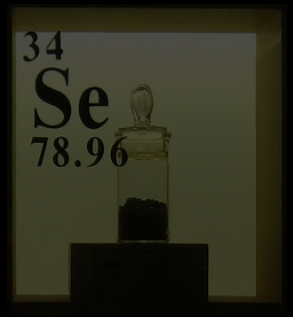


Actions Generator

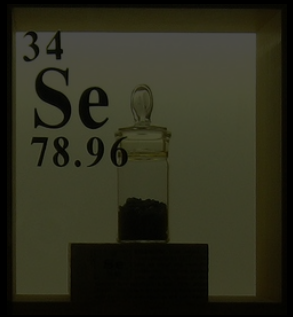


Keyboard Interactions

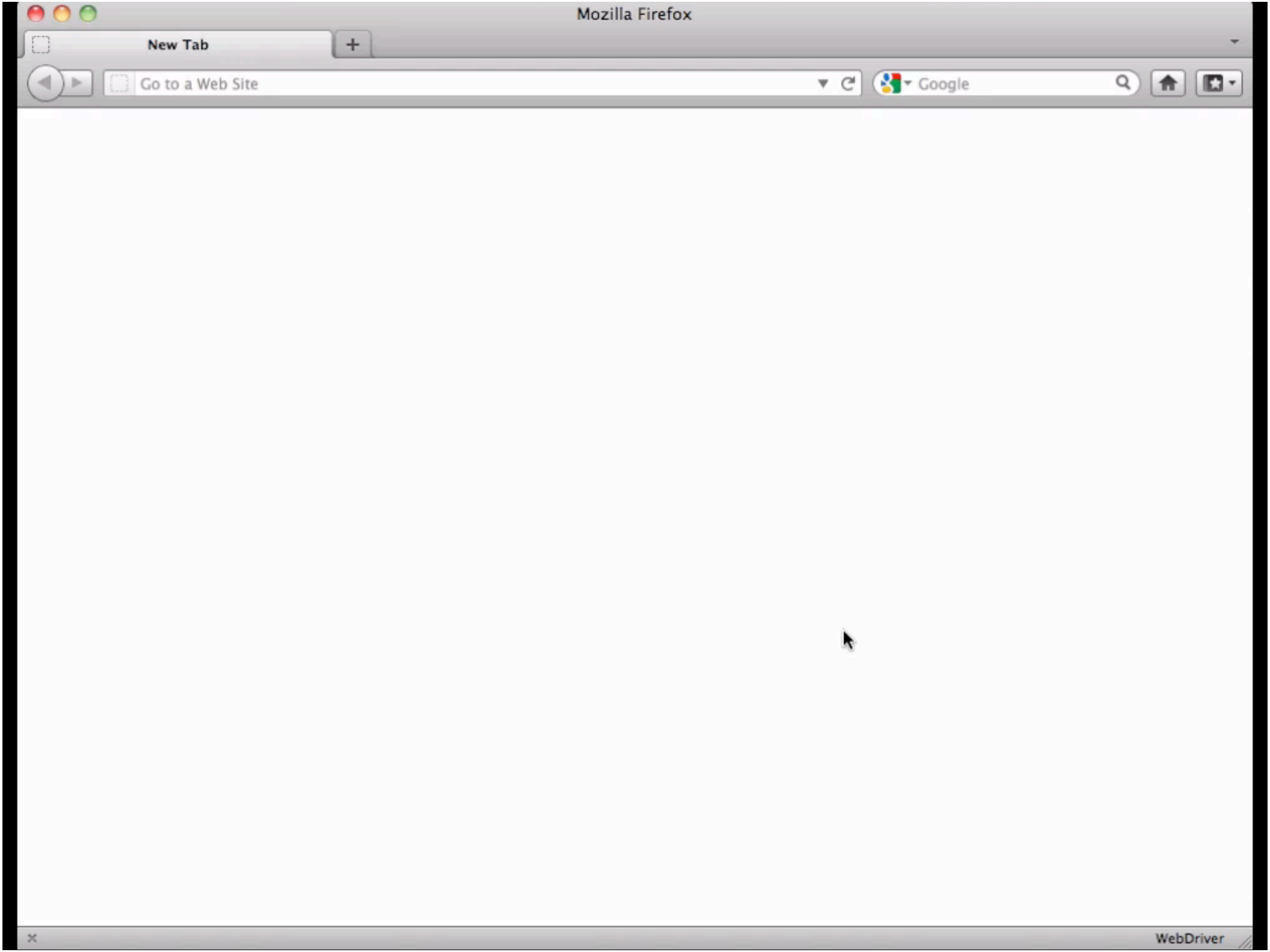
```
Actions builder = new Actions(driver);

builder.keyDown(Keys.CONTROL)
   .click(someElement)
   .click(someOtherElement)
   .keyUp(Keys.CONTROL);

Action selectMultiple = builder.build();

selectMultiple.perform();
```



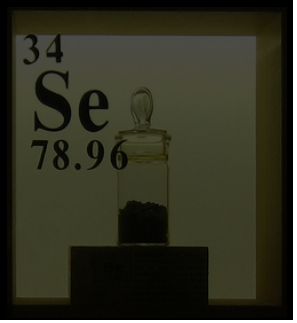
Drag and Drop

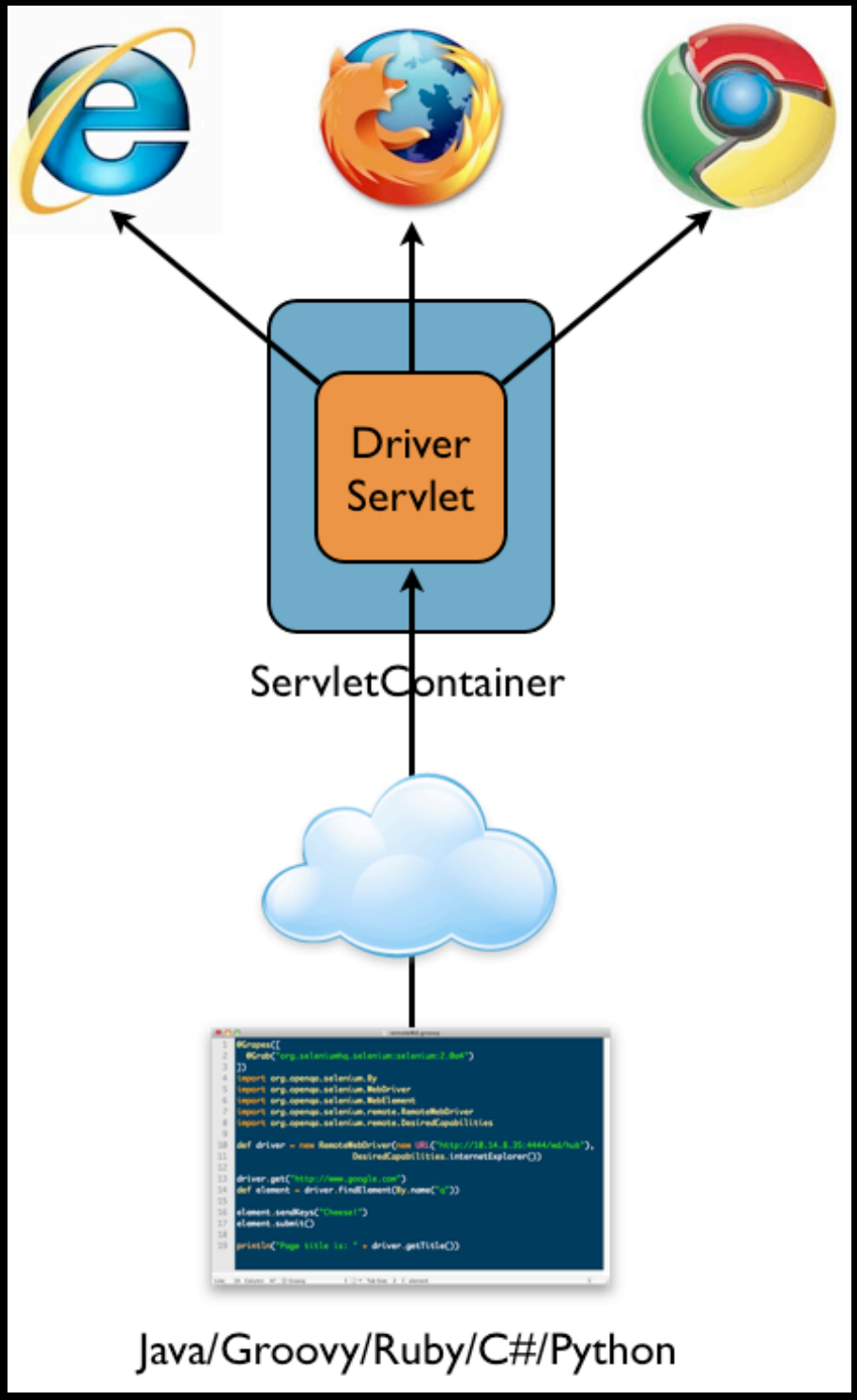


```
WebDriver driver = new FirefoxDriver();

driver.get("file:///Users/mstine/Projects/" +
  "se-2-workshop-examples/src/main/webapp/dnd.html");

WebElement element = driver.findElement(By.id("draggable_demo"));
WebElement target = driver.findElement(By.id("droppable_demo"));

for (int i = 0; i < 5; i ++) {
  (new Actions(driver)).dragAndDrop(element, target).perform();
  Thread.sleep(5000);
}

driver.quit();
```


Mozilla Firefox
New Tab
Go to a Web Site
Google
WebDriver

RemoteWebDriver



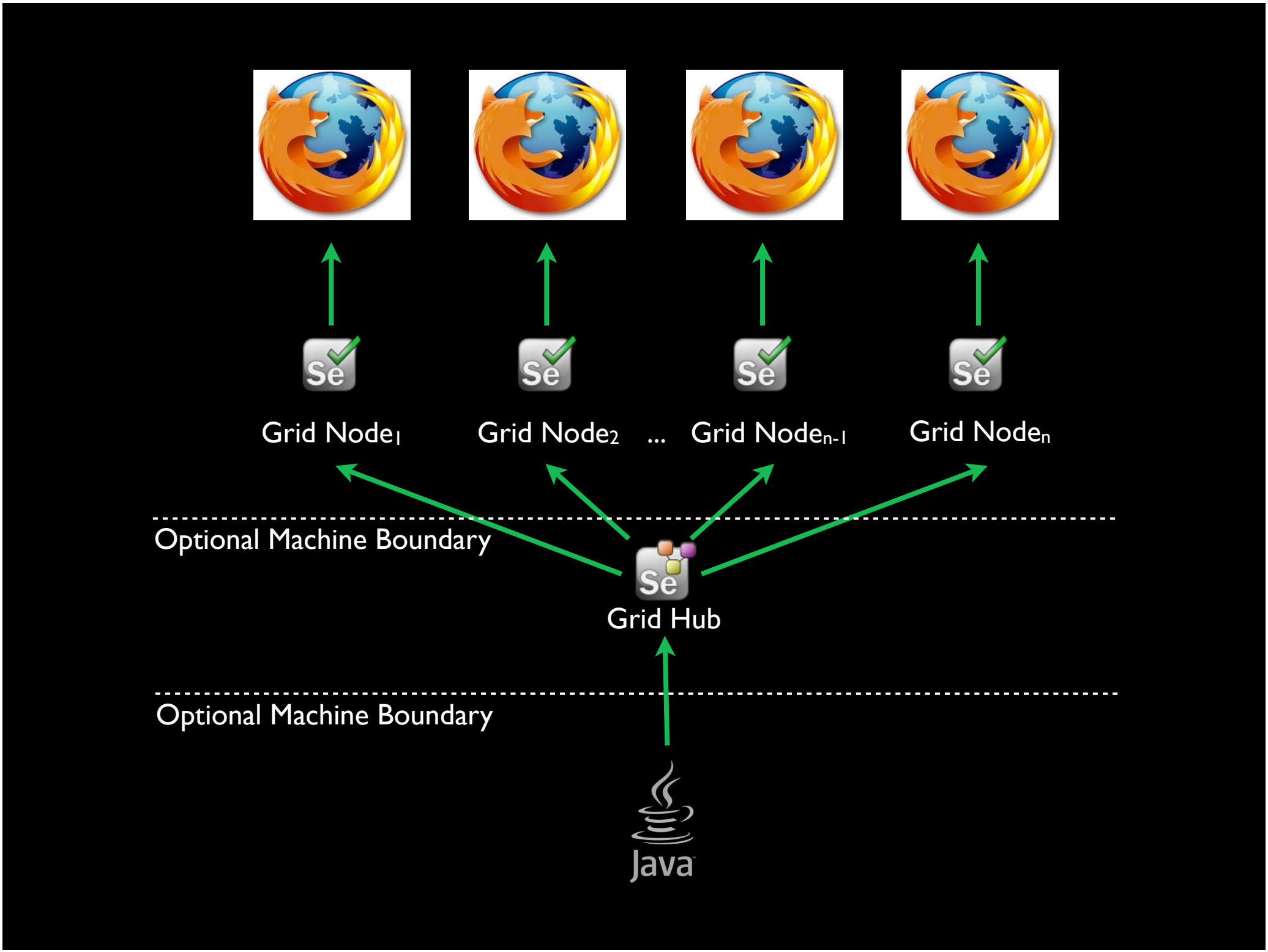

Driver
Servlet
ServletContainer
Java/Groovy/Ruby/C#/Python

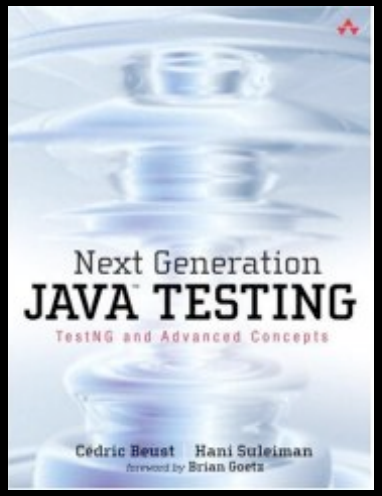
Selenium tests are slow...

...my test suite is growing...

...my build is getting slower...


Selenium
GRID


Grid Node1
Grid Node2
...
Grid Noden-1
Grid Noden
Optional Machine Boundary
Grid Hub
Optional Machine Boundary
Java



TEST PARALLELIZATION

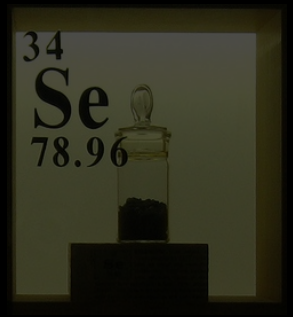
```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Serial Suite for Se Grid">
  <test name="Serial Test for Se Grid">
    <packages>
      <package name="com.deepsouthsoftware.seworkshop.testng" />
    </packages>
  </test>
</suite>
```

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="3" name="Parallel Suite for Se Grid" parallel="classes">
  <test name="Parallel Test for Se Grid">
    <packages>
      <package name="com.deepsouthsoftware.seworkshop.testng"/>
    </packages>
  </test>
</suite>
```

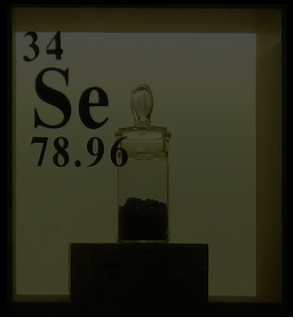
```
package com.deepsouthsoftware.seworkshop.testng;

import com.thoughtworks.selenium.*;
import org.testng.annotations.*;

public class TestFindSpeakerFlowNG extends SeleneseTestBase {
  @BeforeClass
  public void setUp() throws Exception {
    selenium = new DefaultSelenium("localhost", 4444, "*firefox",
                                   "http://localhost:8080/");
    selenium.start();
  }

  @Test
  public void testFindSpeakerFlow() throws Exception {
    selenium.open("/fluffbox-rwx/");
    selenium.click("//a[contains(@href, '/fluffbox-rwx/speaker/find')]");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=Matt Stine");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=Find this Speaker");
```

```
apply plugin:'java'

repositories {
  mavenCentral()
}

dependencies {
  testCompile 'junit:junit:4.8.2'
  testCompile 'org.seleniumhq.selenium:selenium-java:2.13.0'
  testCompile 'org.testng:testng:5.14.1'
}

task seGridParallel(type: Test) {
  useTestNG() {
    suites 'src/test/resources/parallel-suite.xml'
  }
}

task seGridSerial(type: Test) {
  useTestNG() {
    suites 'src/test/resources/serial-suite.xml'
  }
}
```

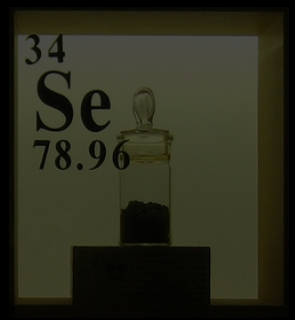
END LECTURE



PAIR UP!

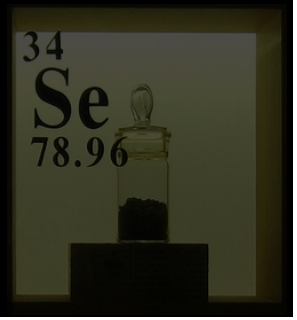


Getting Started



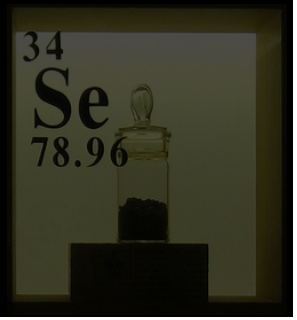
Lab #1

Get familiar with Selenium IDE record/playback features. Finish with record/playback of the “Find a Speaker” flow.



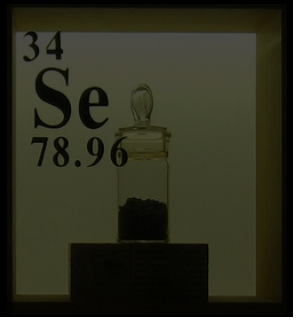
Lab #2

- Export to JUnit 4 (WebDriver)
- Run test using Gradle



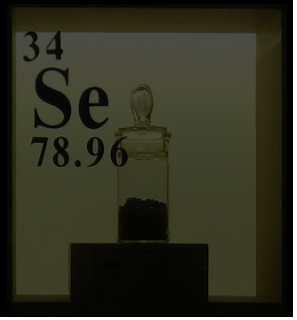
Lab #3

Refactor your exported test case to use Page Objects via the PageFactory (code along).



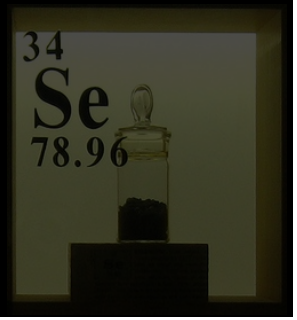
Lab #4

Leveraging existing page objects, develop from scratch a test for the “Find Kiosk Flow.”



Lab #5

Parallelize your test suite using Selenium Grid.




Please fill out
your evaluations!
Matt Stine
matt.stine@gmail.com
http://www.mattstine.com
Twitter: mstine