
STP
SOFTWARE
TEST PROFESSIONALS
CONFERENCE FALL 2017
Demystifying Selenium
Peter Kim
#STPCon

INTRO

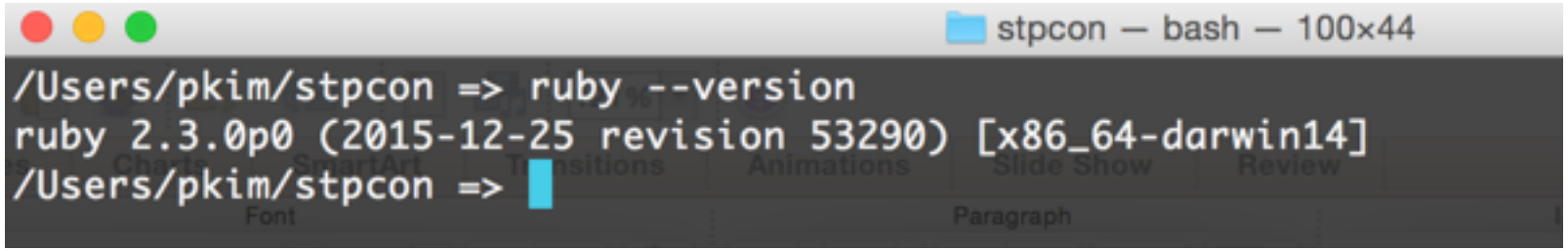
- Software Engineer
- Quality Engineer
- Test Manager
- 7+ Start-ups
- Fortune 10
- Open source enthusiast
- Co-organizer for DCAST (DC Agile Software Testing – dcast.io)
- Consultant

Objectives

- Install and setup environment
  - Ruby with Selenium/WebDriver dependencies
  - Nodejs with Selenium/WebDriver dependencies
- What is Selenium, JSON Wire Protocol, WebDriver?
- Basic programming concepts
  - Functional programming
  - Using “classes”
  - PageObjects
  - Using “irb” (Interactive Ruby)
- Test Automation
  - Selenium/WebDriver
    - APIs
    - Exercises
    - Playground
  - Test Frameworks (Rspec)
  - Playground
- Next Steps

Installing Ruby – Mac OS X

Should already be pre-installed.



Installing Ruby - Windows

Download and install from here - https://rubyinstaller.org/

Windows – Post Installation

- Open Ruby terminal
- gem install selenium-webdriver
- gem install rspec

Chromedriver

- Download and install

https://sites.google.com/a/chromium.org/chromedriver/downloads

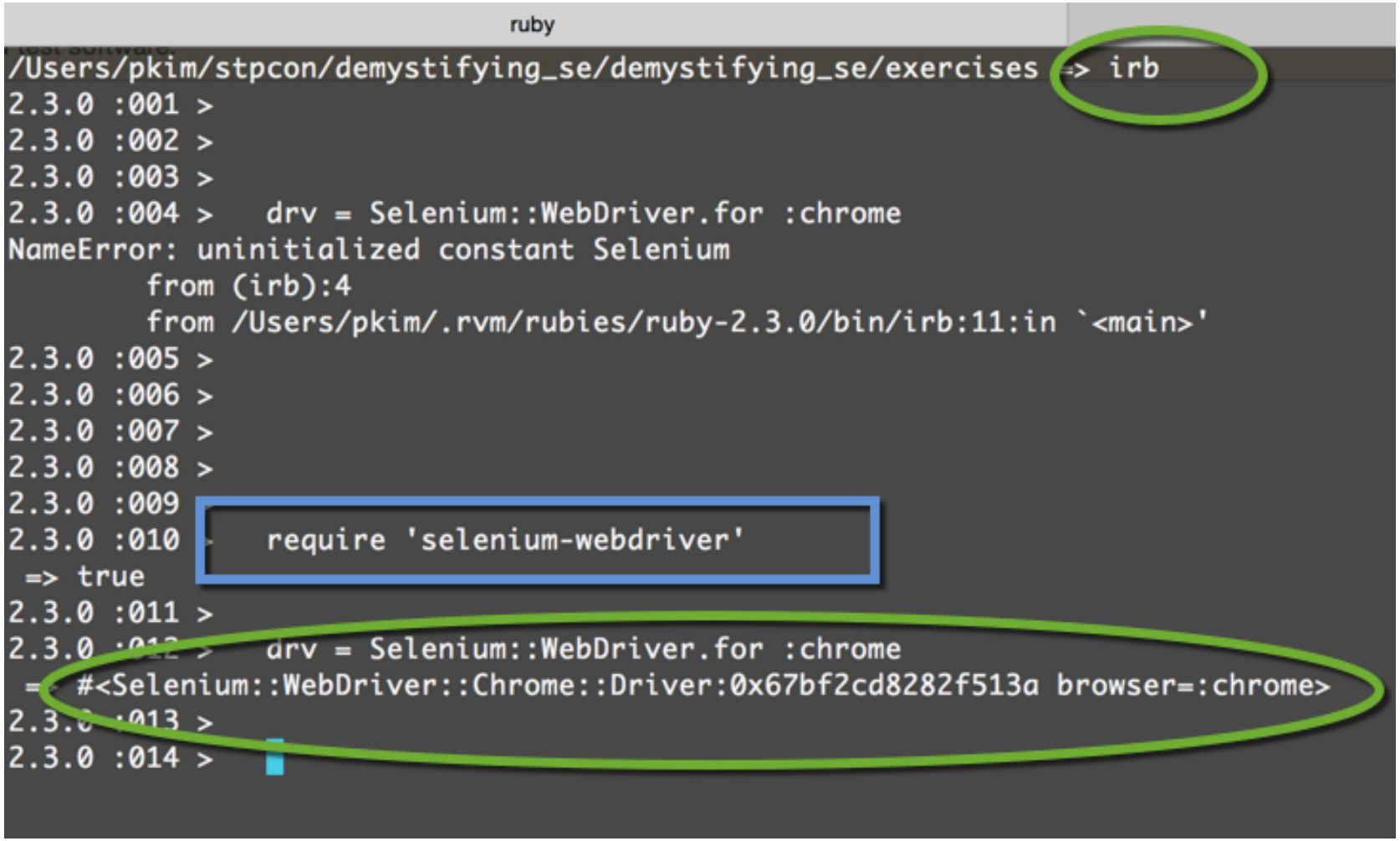
Basic Programming Concepts

- “require” external libraries
  - selenium-webdriver
- “require_relative” libraries by relative path (local)
- Classes (Objects)
  - attributes
  - methods (functions)

require and require_relative

require ‘selenium-webdriver’
require_relative ‘../common/utils’

Example: require/require_relative



Example: require/require_relative

```
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises => cat common/mylib.rb | nl



     1  def echo(s)
     2     puts "[ECHO]: " + s
     3  end

/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises =>
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises =>
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises => irb
2.3.0 :001 >
2.3.0 :002 >   echo("STPCon")
NoMethodError: undefined method `echo' for main:Object
        from (irb):2
        from /Users/pkim/.rvm/rubies/ruby-2.3.0/bin/irb:11:in `<main>'
2.3.0 :003 >
2.3.0 :004 >
2.3.0 :005 >   require_relative './common/mylib'
 => true
2.3.0 :006 > echo("STPCon")
[ECHO]: STPCon
 => nil
2.3.0 :007 >
```

Class

- Objects are your “friends” with specific characteristics and services
  - Manages data (e.g. “state” data)
  - Provides services to do things (e.g. methods/functions)

- Benefits
  - Object Oriented Design (Encapsulation, scalability)
  - Avoid static “sequential” code

Class:
Example

```
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises/basics => cat vehicle.rb
class Vehicle

   attr_accessor :engineType
   attr_accessor :wheels

   def initialize(des = 'TBD')
     puts __FILE__ + (__LINE__).to_s +  "  [Vehicle]: initialize"

     # Place any initialization code here
     @description = des
     @wheels = 0
     @totalPassengers = 0
   end

   def dump()
     puts __FILE__ + (__LINE__).to_s + "  [Vehicle]:dump"
     puts "o Description: " + @description
     puts "o Wheels     : " +  @wheels.to_s
     puts "o Passengers : " + @totalPassengers.to_s
  end

end
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises/basics => irb
2.3.0 :001 > require_relative 'vehicle'
 => true
2.3.0 :002 > m3 = Vehicle.new('My M3')
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises/basics/vehicle.rb7  [Vehicle]: initialize
 => #<Vehicle:0x007fef409ac6a8 @description="My M3", @wheels=0, @totalPassengers=0>
2.3.0 :003 >
2.3.0 :004 > m3.dump
/Users/pkim/stpcon/demystifying_se/demystifying_se/exercises/basics/vehicle.rb16  [Vehicle]:dump
o Description: My M3
o Wheels     : 0
o Passengers : 0
 => nil
2.3.0 :005 >
```

Selenium-WebDriver

require ‘selenium-webdriver’

Selenium/WebDriver – what Is it ?

- RESTFul web service, leveraging JSON, over HTTP.
- Selenium RC (Server/Core) – JS Injection (JS App Running inside the browser)
- WebDriver
  - ".. Should do what a user can do"
  - Tiny API (More readable code)
  - No JS Injection

- W3C Standard for browser automation
  - https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol

Selenium/WebDriver References

- https://w3c.github.io/webdriver/webdriver-spec.html
- http://www.rubydoc.info/gems/selenium-webdriver/Selenium/WebDriver
- https://github.com/SeleniumHQ/selenium/wiki/Ruby-Bindings

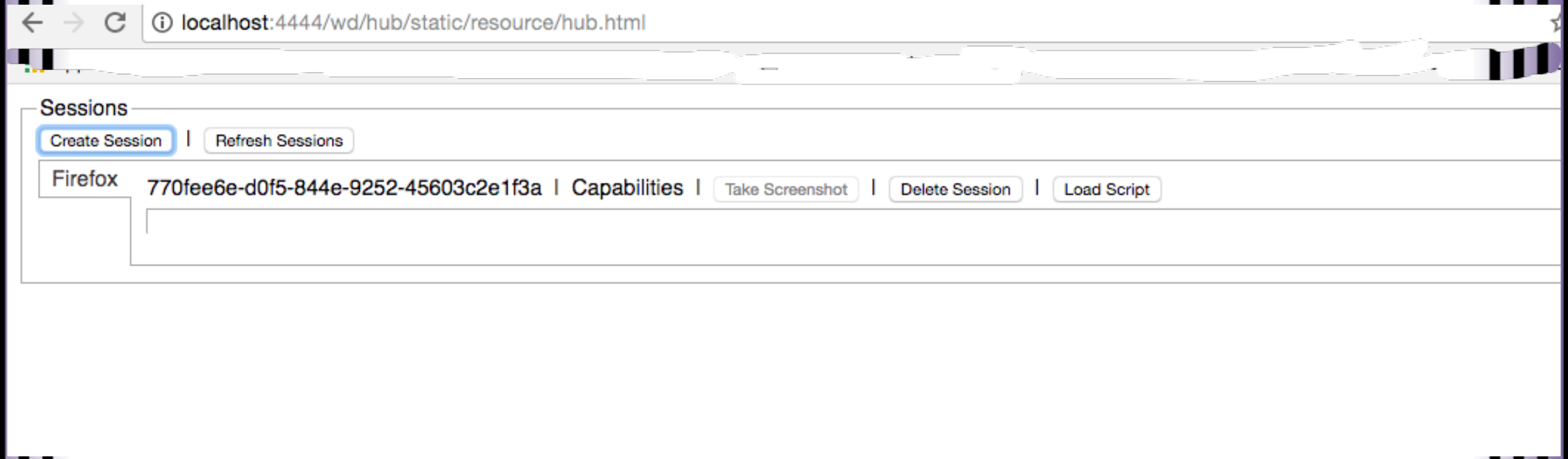
Selenium RC Server

- Download latest Selenium Standalone Server
  - http://docs.seleniumhq.org/download/
- Run Selenium Standalone Server (Jar file)
  - java –jar selenium-server-standalone-3.5.3.jar

Selenium RC Server

```
/Users/pkim/bin => java -jar selenium-server-standalone-3.5.3.jar
15:11:39.613 INFO - Selenium build info: version: '3.5.3', revision: 'a88d25fe6b'
15:11:39.614 INFO - Launching a standalone Selenium Server
2017-09-24 15:11:39.643:INFO::main: Logging initialized @250ms to org.seleniumhq.jetty9.util.log.StdErrLog
15:11:39.726 INFO - Driver class not found: com.opera.core.systems.OperaDriver
15:11:39.766 INFO - Driver provider class org.openqa.selenium.ie.InternetExplorerDriver registration is skipped:
 registration capabilities Capabilities [{ensureCleanSession=true, browserName=internet explorer, version=, platform=WINDOWS}]
 does not match the current platform MAC
15:11:39.766 INFO - Driver provider class org.openqa.selenium.edge.EdgeDriver registration is skipped:
 registration capabilities Capabilities [{browserName=MicrosoftEdge, version=, platform=WINDOWS}] does not match the current p
latform MAC
15:11:39.804 INFO - Using the passthrough mode handler
2017-09-24 15:11:39.839:INFO:osjs.Server:main: jetty-9.4.5.v20170502
2017-09-24 15:11:39.863:WARN:osjs.SecurityHandler:main: ServletContext@o.s.j.s.ServletContextHandler@3e9b1010{/,null,STARTING}
 has uncovered http methods for path: /
2017-09-24 15:11:39.867:INFO:osjsh.ContextHandler:main: Started o.s.j.s.ServletContextHandler@3e9b1010{/,null,AVAILABLE}
2017-09-24 15:11:39.893:INFO:osjs.AbstractConnector:main: Started ServerConnector@a7e666{HTTP/1.1,[http/1.1]}{0.0.0.0:4444}
2017-09-24 15:11:39.894:INFO:osjs.Server:main: Started @501ms
15:11:39.894 INFO - Selenium Server is up and running
```

Selenium RC Server



Selenium RC Server – Console

- Demo

  *exercise0

Demystify Selenium/WebDriver by sending requests directly to the browser

- Create/open a browser
- Find the sessionID
- Verify that the sessionID exists using Selenium (JSON Wire)
- Run Selenium commands from Selenium test script
- Run commands from JSON Wire

Selenium – JSON Wire Protocol

- irb
  - Starts the interactive Ruby session
- require 'selenium-webdriver'
  - Loads library
- b=Selenium::WebDriver.for :chrome
  - Opens Chrome browser (creates a sessionID)
- b.send(:bridge).session_id
  - Get the session_id – needed for our HTTP Requests
- Open a browser and goto 'http://localhost:[port]'
  - Get the [port] by 'ps –ef | grep 'port'
  - 606319619 80306 79899 0 12:33AM ttys000 0:00.05 /Users/pkim/.rvm/gems/ruby-2.3.0/bin/chromedriver --port=**9515**
  - **From browser, goto http://localhost:9515/session/[sessionid]**

Selenium – JSON Wire Protocol

```
stpcon — ruby — 100×44
2.3.0 :002 > require 'selenium-webdriver'
 => true
2.3.0 :003 > b=Selenium::WebDriver.for :chrome
 => #<Selenium::WebDriver::Chrome::Driver:0x6daa5e10f282a356 browser=:chrome>
2.3.0 :004 > b.send(:bridge).session_id
 => "2ab5dc10802b74a83bfd1d41c625b3ac"
2.3.0 :005 > b.send(:bridge).http
 => #<Selenium::WebDriver::Remote::Http::Default:0x007fa0433007b0 @open_timeout=nil, @read_timeout=n
il, @server_url=#<URI::HTTP http://127.0.0.1:9515/>>
2.3.0 :006 >
```

www.rubydoc.info/gems/selenium-webdriver/Selenium/WebDriver/Remote/Bridge

Apps | Concur - VM Stuff | Concur - Release a... | Concur - ScoutUI a... | React and Nodejs | Concur - Git H20Dr... | Concur - NUI

Class List

Classes | Methods | Files

Search:

- Navigation < Object
- Options < Object
- PhantomJS
- Platform
- Point < Struct
- PointerActions
- PortProber < Object
- ProfileHelper
- Proxy < Object
- Rectangle < Struct
- Remote
  - Bridge < Object
  - Capabilities < Object
  - Driver
  - Http
  - OSS
  - Response < Object
  - W3C
- Safari
- SearchContext
- Service < Object
- SocketLock < Object
- SocketPoller < Object
- Support
- TargetLocator < Object
- Timeouts < Object

Libraries » selenium-webdriver (3.4.3) » Index (B) » Selenium » WebDriver » Remote » Bridge

Class: Selenium::WebDriver::Remote::Bridge

Inherits: Object	show all
Includes: Atoms, BridgeHelper	
Defined in: lib/selenium/webdriver/remote/bridge.rb	

Direct Known Subclasses

OSS::Bridge, W3C::Bridge

Constant Summary

COMMANDS =

```
{
  new_session: [:post, 'session'.freeze]
}.freeze
```

Instance Attribute Summary

`#capabilities ⇒ Object` readonly

Returns the value of attribute capabilities.

`#context ⇒ Object`

Returns the value of attribute context.

`#dialect ⇒ Object` readonly

Returns the value of attribute dialect.

`#file_detector ⇒ Object`

Returns the value of attribute file_detector.

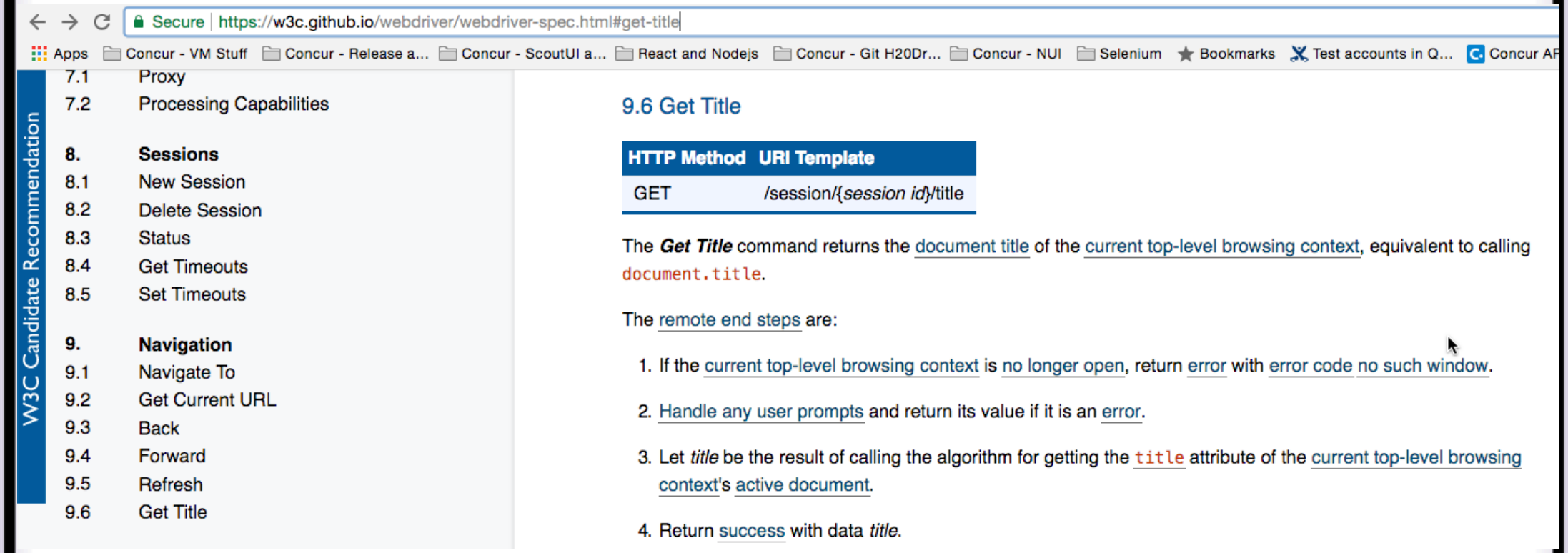
`#http ⇒ Object`

Returns the value of attribute http.

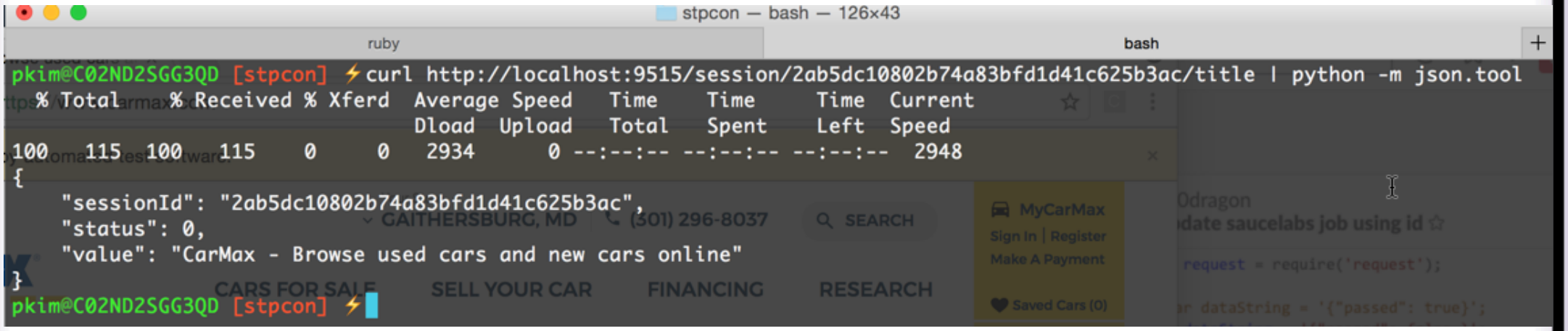
localhost:9515/session/2ab5dc10802b74a83bfd1d41c625b3ac


```
{
   sessionId: "2ab5dc10802b74a83bfd1d41c625b3ac",
   status: 0,
 - value: {
       acceptSslCerts: true,
       applicationCacheEnabled: false,
       browserConnectionEnabled: false,
       browserName: "chrome",
     - chrome: {
           chromedriverVersion: "2.31.488774 (7e15618d1bf16df8bf0ecf2914ed1964a387ba0b)",
           userDataDir: "/var/folders/mh/jv80zk713tbdzsr8w61t27wwk27d03/T/.org.chromium.Chromium.7JbzeE"
       },
       cssSelectorsEnabled: true,
       databaseEnabled: false,
       handlesAlerts: true,
       hasTouchScreen: false,
       javascriptEnabled: true,
       locationContextEnabled: true,
       mobileEmulationEnabled: false,
       nativeEvents: true,
       networkConnectionEnabled: false,
       pageLoadStrategy: "normal",
       platform: "Mac OS X",
       rotatable: false,
       setWindowRect: true,
       takesHeapSnapshot: true,
       takesScreenshot: true,
       unexpectedAlertBehaviour: "",
       version: "61.0.3163.100",
       webStorageEnabled: true
   }
}
```

Selenium – JSON Wire Protocol Command Example



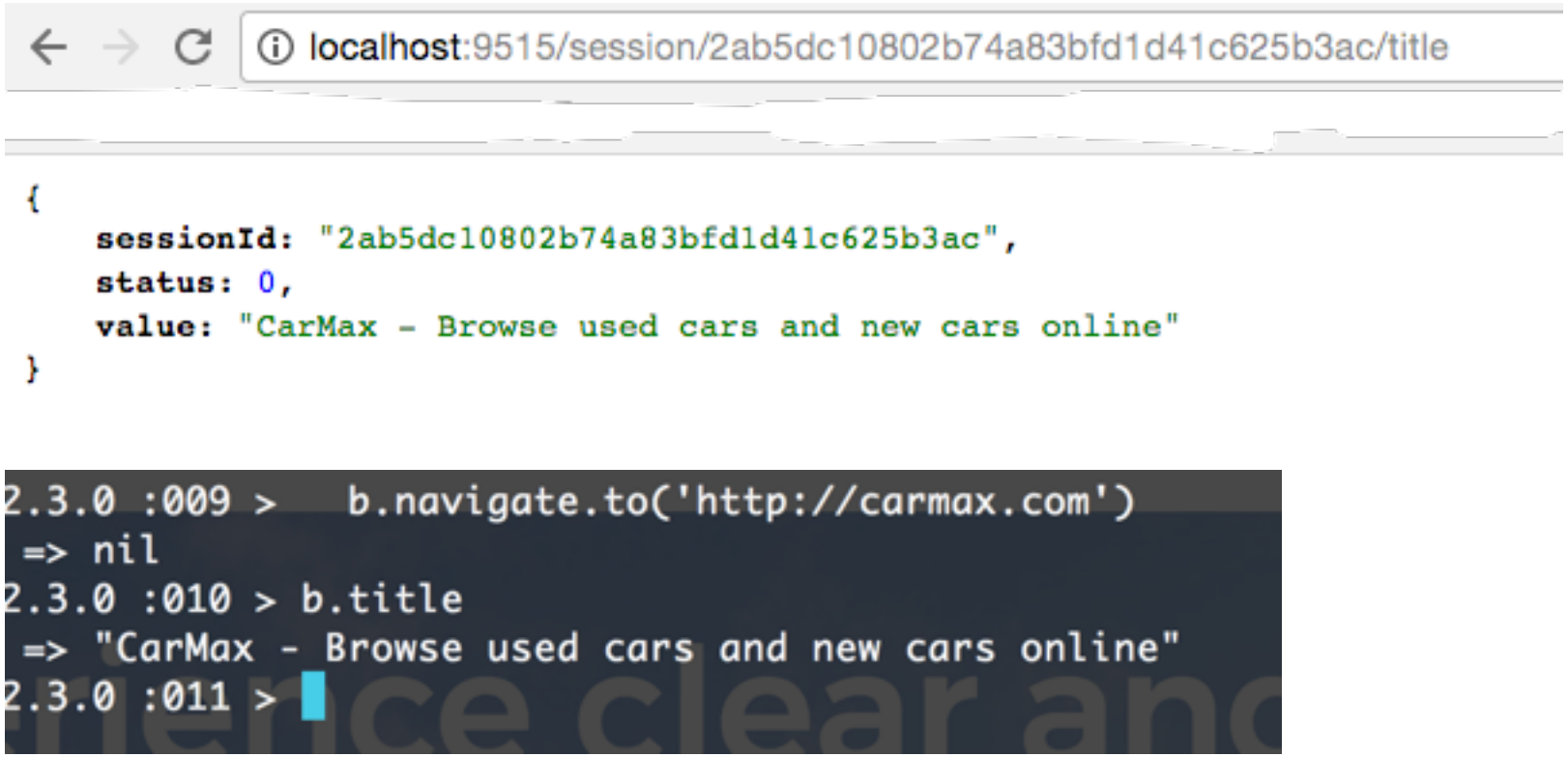
Selenium – JSON Wire Protocol Command Example



Selenium – JSON Wire Protocol Command Example

```
2.3.0 :009 >   b.navigate.to('http://carmax.com')
 => nil
2.3.0 :010 > b.title
 => "CarMax - Browse used cars and new cars online"
2.3.0 :011 >
```

Selenium – JSON Wire Protocol Command Example



Selenium – JSON Wire Protocol Command Reference

https://github.com/SeleniumHQ/selenium/blob/master/rb/lib/selenium/webdriver/remote/oss/commands.rb

GitHub, Inc. [US] | https://github.com/SeleniumHQ/selenium/blob/master/rb/lib/selenium/webdriver/remote/oss/commands.rb

Concur - VM Stuff | Concur - Release a... | Concur - ScoutUI a... | React and Nodejs | Concur - Git H20Dr... | Concur - NUI | Selenium | Bc

```
21    module WebDriver
22      module Remote
23        module OSS
24
25          #
26          # https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol#command-reference
27          # @api private
28          #
29
30          class Bridge
31
32            COMMANDS = {
33              get_capabilities: [:get, 'session/:session_id'.freeze],
34              status: [:get, 'status'.freeze],
35
36              #
37              # basic driver
38              #
39
40              get_current_url: [:get, 'session/:session_id/url'.freeze],
41              get: [:post, 'session/:session_id/url'.freeze],
42              go_forward: [:post, 'session/:session_id/forward'.freeze],
43              go_back: [:post, 'session/:session_id/back'.freeze],
44              refresh: [:post, 'session/:session_id/refresh'.freeze],
45              quit: [:delete, 'session/:session_id'.freeze],
46              close: [:delete, 'session/:session_id/window'.freeze],
47              get_page_source: [:get, 'session/:session_id/source'.freeze],
48              get_title: [:get, 'session/:session_id/title'.freeze],
49              find_element: [:post, 'session/:session_id/element'.freeze],
50              find_elements: [:post, 'session/:session_id/elements'.freeze],
51              get_active_element: [:post, 'session/:session_id/element/active'.freeze],
```

Getting Comfortable with “irb”

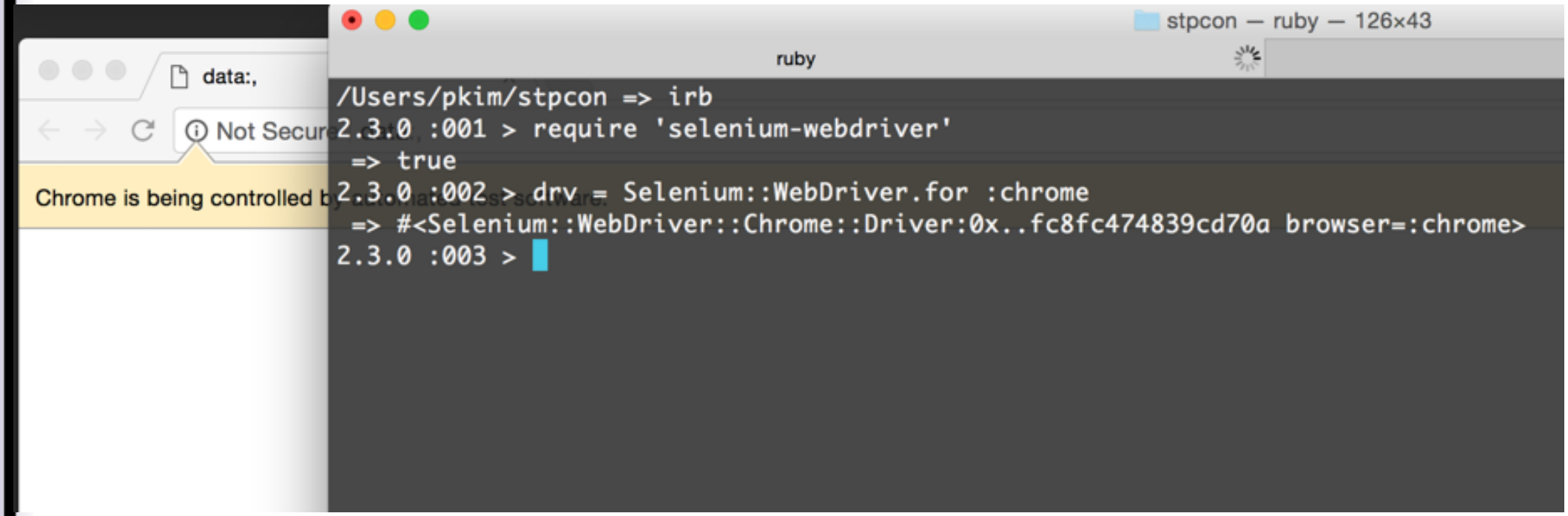
Start “irb” session.

Interactively enter statements.

e.g.

```
require ‘selenium-webdriver’
x = 100
puts “Hello World”
puts  “My value is “ + x
myHash = { :name => ‘Elvis’,  :car => ‘Cadillac’ }
puts “My hash #{myHash}”
```

Access Selenium WebDriver library



Start irb – “Interactive Ruby”

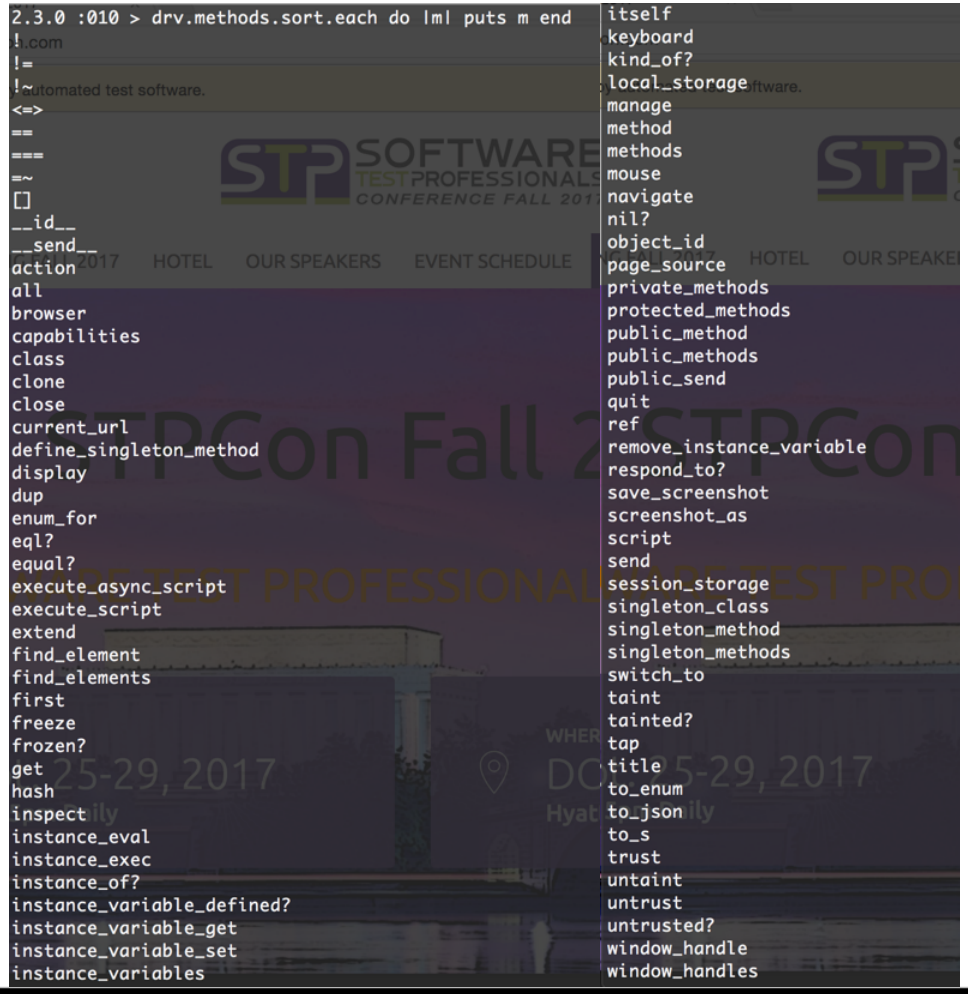
irb

require ‘selenium-webdriver’

```
/Users/pkim/stpcon => irb
2.3.0 :001 > require 'selenium-webdriver'
 => true
2.3.0 :002 >
```

What WebDriver services (methods) are available?

drv.methods.sort.each do |m| puts m end



Intro - WebDriver Commands

- get
- title
- current_url
- quit

Exercise 1. Create a browser object

- Objective: Open a chrome browser and navigate to a target website, while obtaining the title and current URL.

```
require 'selenium-webdriver'


drv = Selenium::WebDriver.for :chrome

puts "WebDriver object is " + drv.class.to_s


drv.navigate.to( url 'http://www.stpcon.com')

title = drv.title
url = drv.current_url


puts "Current Title is " + title
puts "Current URL is " + url


drv.quit
```

Selenium::WebDriver::Navigation

- back
- forward
- Refresh
- to(url)

http://www.rubydoc.info/gems/selenium-webdriver/Selenium/WebDriver/Navigation

Exercise 2. Navigation

- Objective: Open a chrome browser and **navigate** to a target website, while obtaining the **title**, **current URL**, and **refresh** the page.

```
require 'selenium-webdriver'

host = 'http://www.stpcon.com'

drv = Selenium::WebDriver.for :chrome

drv.navigate.to( url host)

puts "Title of #{host} is #{drv.title}"


drv.get('http://www.carmax.com')

puts "Title of Carmax site is " + drv.title

drv.navigate.back()

puts "After navigationg back, title is #{drv.title}"

drv.navigate.refresh()

drv.quit()
```

Selenium Locators

- A DSL/Language that gives you the ability to find web elements.

Selenium Locators

- A DSL/Language that gives you the ability to find web elements.

- This will consume most QE time/effort when automating.
  - UI changes
  - UI challenges in finding appropriate locator
    - Frames (within frames)
    - Alerts
    - Timing conditions and constraints
    - Refactoring of automation code

Selenium Locators

- A DSL/Language that gives you the ability to find web elements.
- This will consume most QE time/effort when automating.
  - UI changes
  - UI challenges in finding appropriate locator
    - Frames (within frames)
    - Alerts
    - Timing conditions and constraints
    - Refactoring of automation code
- You will get frustrated .. but as you gain more experience, you'll find that it becomes easier.

Selenium::WebDriver::SearchContext

http://www.rubydoc.info/gems/selenium-webdriver/Selenium/WebDriver/SearchContext

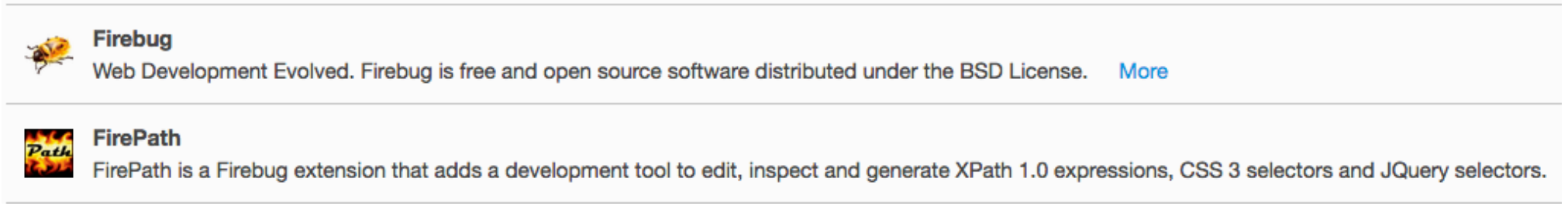
- FINDERS

```
class: 'class name',
class_name: 'class name',
css: 'css selector',
id: 'id',
link: 'link text',
link_text: 'link text',
name: 'name',
partial_link_text: 'partial link text',
tag_name: 'tag name',
xpath: 'xpath'
```

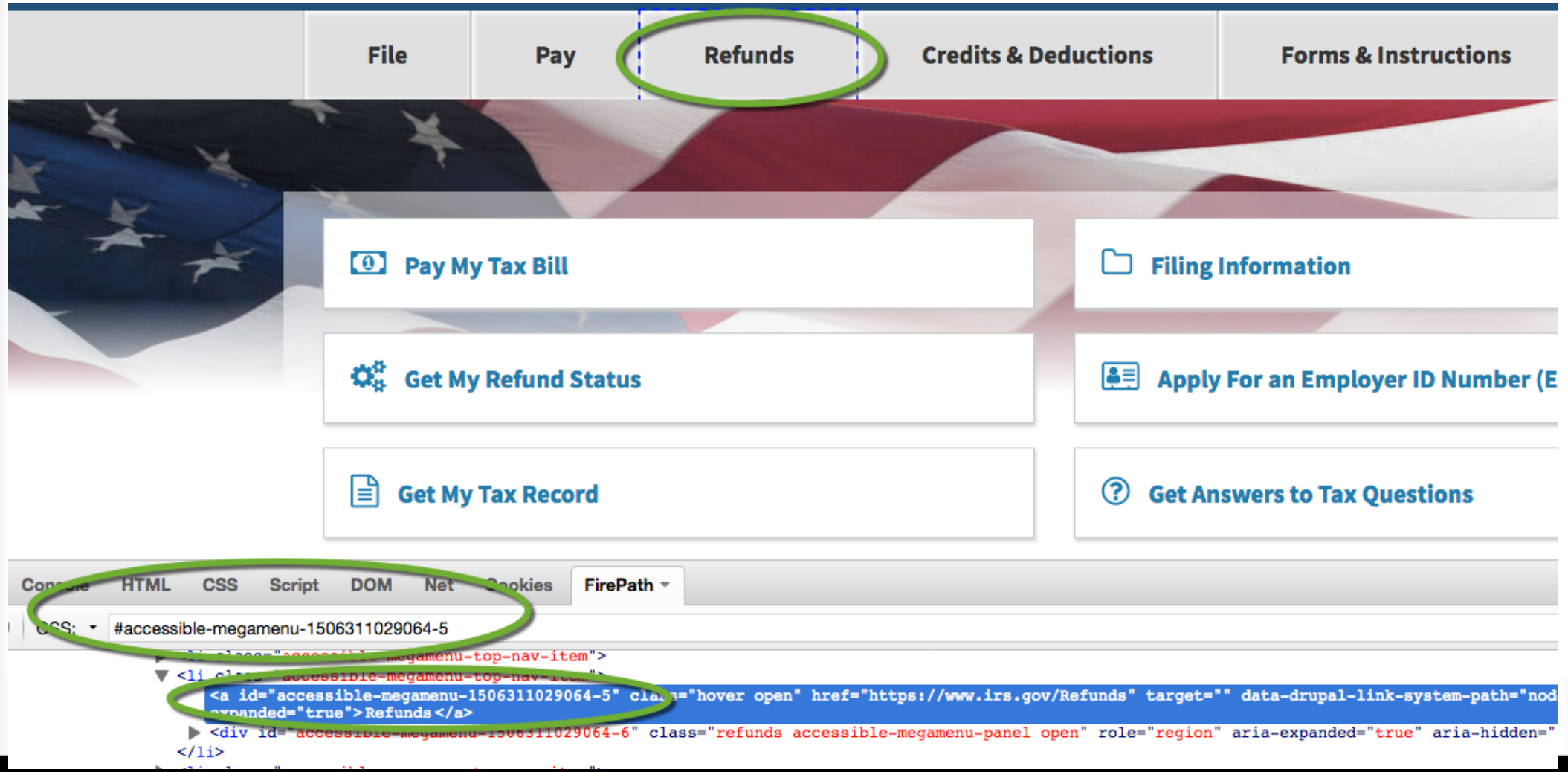
- find_element
- find_elements

Let's experiment with Locators

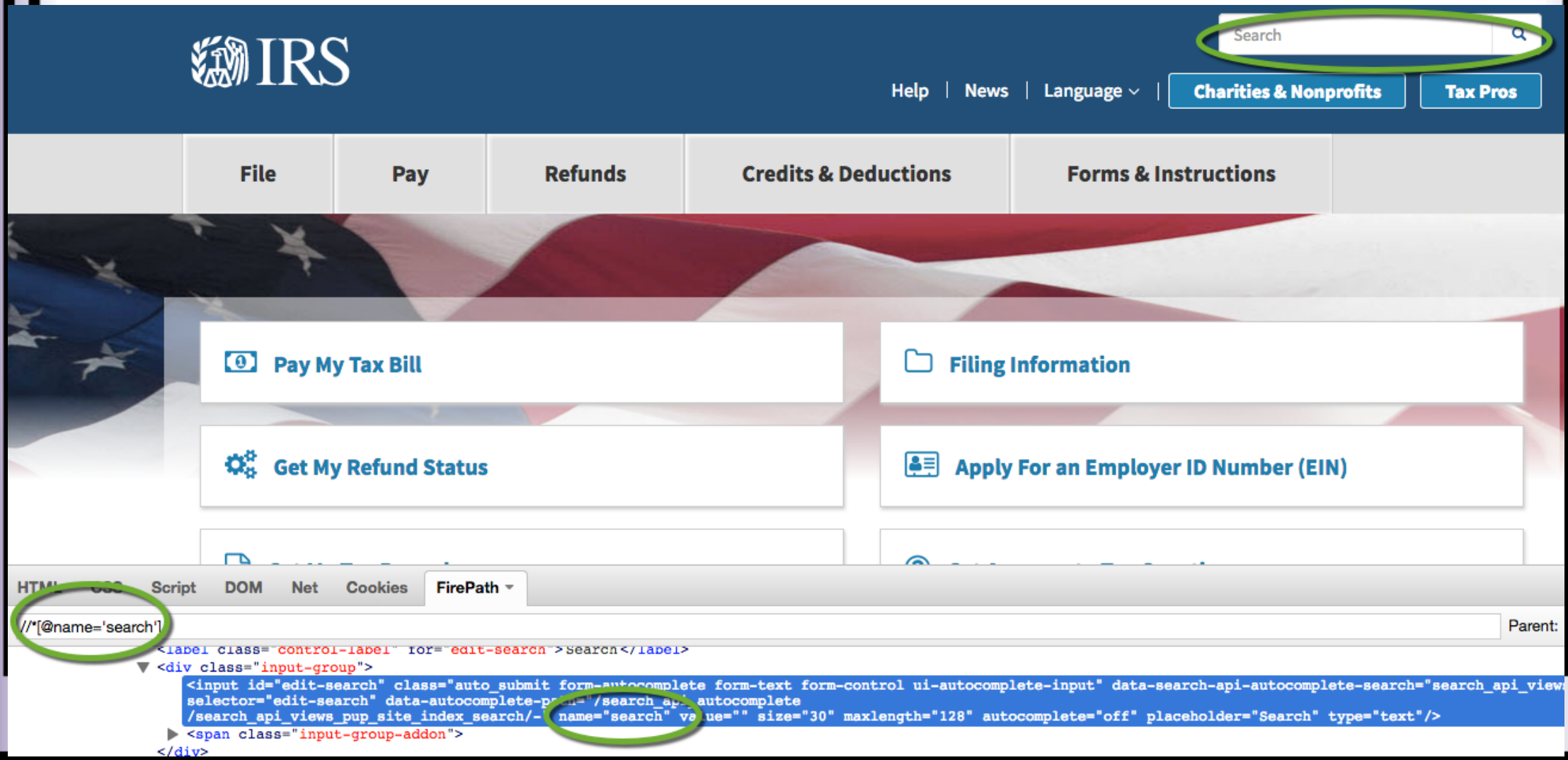
- Firefox browser
- Install "firebug" plugin
- Install "xpather" plugin



id

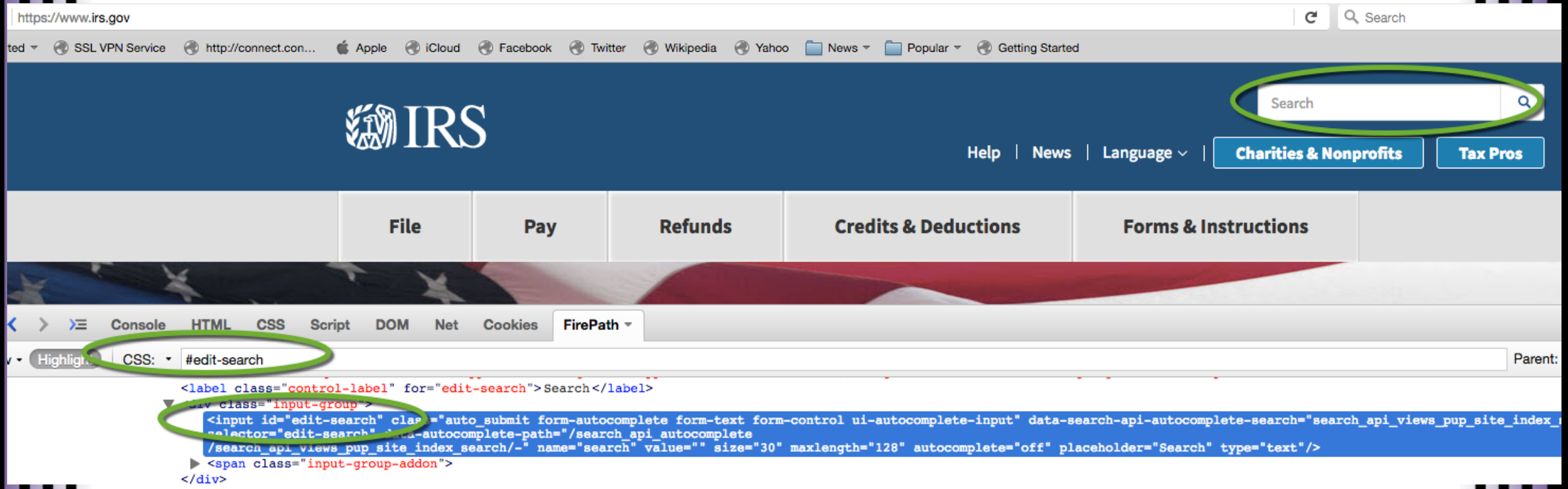

File
Pay
Refunds
Credits & Deductions
Forms & Instructions
Pay My Tax Bill
Filing Information
Get My Refund Status
Apply For an Employer ID Number (E
Get My Tax Record
Get Answers to Tax Questions
HTML
CSS
Script
DOM
Net
FirePath
#accessible-megamenu-1506311029064-5
<a id="accessible-megamenu-1506311029064-5" class="hover open" href="https://www.irs.gov/Refunds" target="" data-drupal-link-system-path="nod
expanded="true">Refunds
<div id="accessible-megamenu-1506311029064-6" class="refunds accessible-megamenu-panel open" role="region" aria-expanded="true" aria-hidden="

name

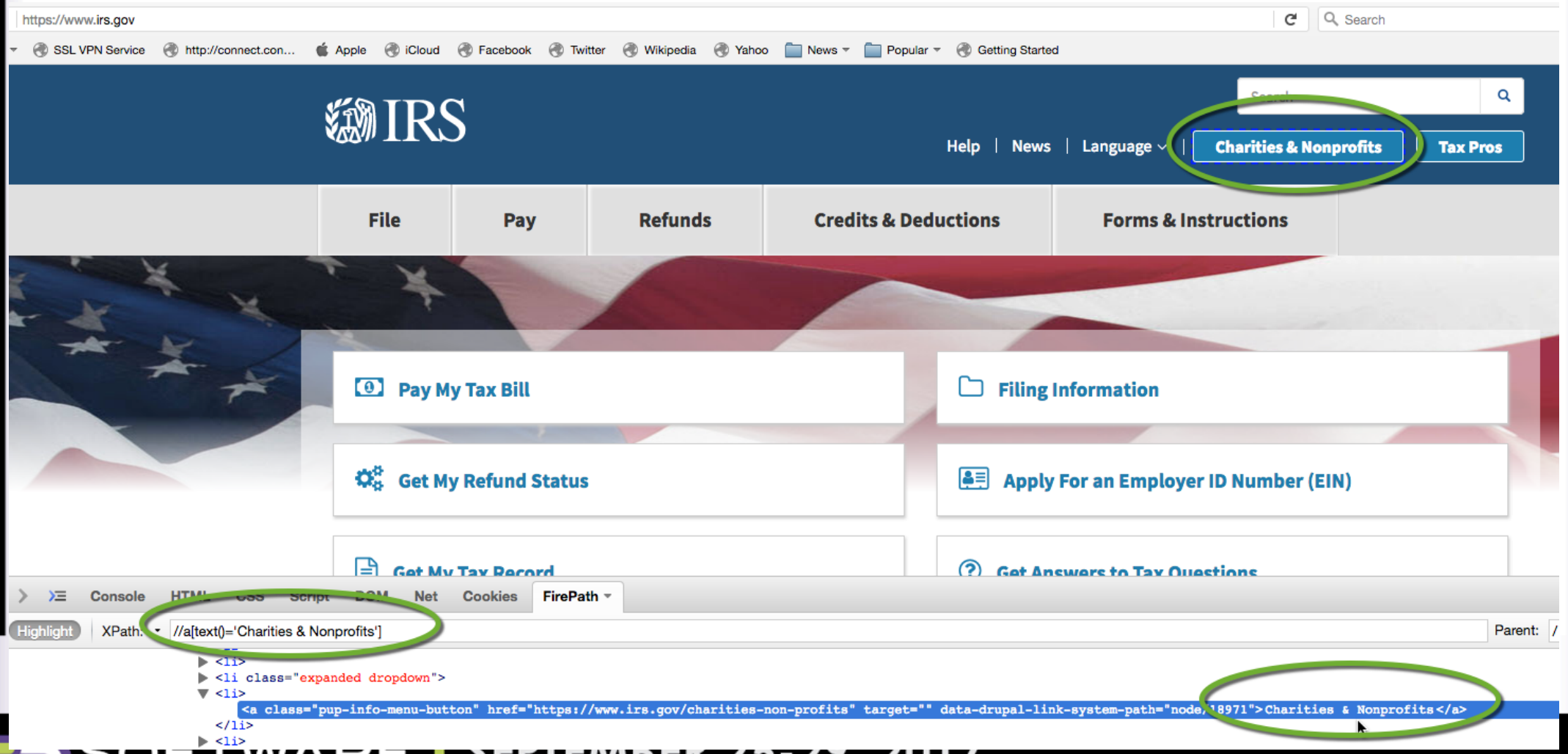

IRS
Search
Help | News | Language | Charities & Nonprofits | Tax Pros
File | Pay | Refunds | Credits & Deductions | Forms & Instructions
Pay My Tax Bill
Filing Information
Get My Refund Status
Apply For an Employer ID Number (EIN)
Script DOM Net Cookies FirePath
//*[@name='search']
Parent:
<label class="control-label" for="edit-search">Search</label>

<input id="edit-search" class="auto_submit form-autocomplete form-text form-control ui-autocomplete-input" data-search-api-autocomplete-search="search_api_view
selector="edit-search" data-autocomplete-p /search_a autocomplete
/search_api_views_pup_site_index_search/- name="search" va ue="" size="30" maxlength="128" autocomplete="off" placeholder="Search" type="text"/>

CSS


https://www.irs.gov
Search
SSL VPN Service
http://connect.con...
Apple
iCloud
Facebook
Twitter
Wikipedia
Yahoo
News
Popular
Getting Started
IRS
Search
Help
News
Language
Charities & Nonprofits
Tax Pros
File
Pay
Refunds
Credits & Deductions
Forms & Instructions
Console
HTML
CSS
Script
DOM
Net
Cookies
FirePath
CSS:
#edit-search
Parent:
<label class="control-label" for="edit-search">Search</label>
<input id="edit-search" class="auto_submit form-autocomplete form-text form-control ui-autocomplete-input" data-search-api-autocomplete-search="search_api_views_pup_site_index_
selector="edit-search" data-autocomplete-path="/search_api_autocomplete
/search_api_views_pup_site_index_search/-" name="search" value="" size="30" maxlength="128" autocomplete="off" placeholder="Search" type="text"/>

link_text


https://www.irs.gov
Search
SSL VPN Service
http://connect.con...
Apple
iCloud
Facebook
Twitter
Wikipedia
Yahoo
News
Popular
Getting Started
IRS
Help
News
Language
Charities & Nonprofits
Tax Pros
File
Pay
Refunds
Credits & Deductions
Forms & Instructions
Pay My Tax Bill
Filing Information
Get My Refund Status
Apply For an Employer ID Number (EIN)
Get My Tax Record
Get Answers to Tax Questions
Console
Net
Cookies
FirePath
Highlight
XPath:
//a[text()='Charities & Nonprofits']
Parent:

<li class="expanded dropdown">

Charities & Nonprofits

xpath – XML path language

https://www.w3schools.com/xml/xpath_syntax.asp

Selecting Nodes

XPath uses path expressions to select nodes in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

Expression	Description
nodename	Selects all nodes with the name "*nodename*"
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

xpath – XML path language

"//" - selects nodes in the document (page) that match the selection no matter where they are

"@" - Selects attributes

Examples:

//*[@id='username']

//input[@name='password']

xpath – XML path language

Operators:

“and”

Examples:

//*[@make=‘porsche’ and @model=‘carrera’]

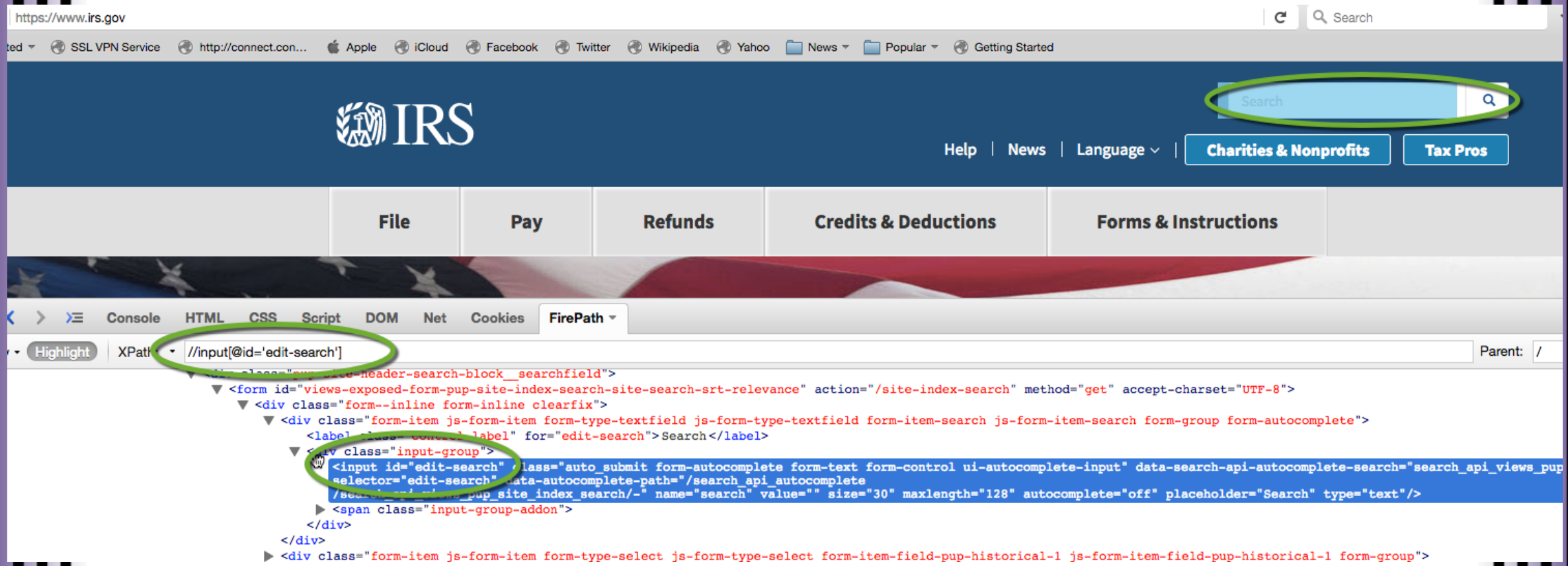
xpath regarding performance?

If a solid attribute such as “id” and “name” is available .. that is preferable.

If a solid CSS locator can be leveraged .. preferable.

However, ensuring that a reliable and scalable locator is used is more important.

xpath


https://www.irs.gov
Search
SSL VPN Service
http://connect.con...
Apple
iCloud
Facebook
Twitter
Wikipedia
Yahoo
News
Popular
Getting Started
IRS
Search
Help
News
Language
Charities & Nonprofits
Tax Pros
File
Pay
Refunds
Credits & Deductions
Forms & Instructions
Console
HTML
CSS
Script
DOM
Net
Cookies
FirePath
Highlight
XPath
//input[@id='edit-search']
Parent:
<form id="views-exposed-form-pup-site-index-search-site-search-srt-relevance" action="/site-index-search" method="get" accept-charset="UTF-8">

<label for="edit-search">Search</label>

<input id="edit-search" class="auto_submit form-autocomplete form-text form-control ui-autocomplete-input" data-search-api-autocomplete-search="search_api_views_pup
selector="edit-search" data-autocomplete-path="/search_api_autocomplete
name="search" value="" size="30" maxlength="128" autocomplete="off" placeholder="Search" type="text"/>

SOFTWARE TEST PROFESSIONALS CONFERENCE FALL 2017
SEPTEMBER 25-29, 2017
HYATT REGENCEY DULLES HOTEL
#STPCon

SearchContext – find_element

http://www.rubydoc.info/gems/selenium-webdriver/Selenium/WebDriver/SearchContext

```
#find_element(how, what) ⇒ Element
```

Parameters:

- **how** (Symbol, String) — The method to find the element by
- **what** (String) — The locator to use

```
class: 'class name',
class_name: 'class name',
css: 'css selector',
id: 'id',
link: 'link text',
link_text: 'link text',
name: 'name',
partial_link_text: 'partial link text',
tag_name: 'tag name',
xpath: 'xpath'
```

Exercise. Find a link and assign it to a variable using CSS.

Objective: Find a link with find_element

…

drv.get('https://stark-bastion-95510.herokuapp.com/playground')

comment = drv.find_element**(css: "#comment")**

comment.send_keys("STP Conference 2017")

…

Exercise. Find a link and assign it to a variable using XPATH.

Objective: Find a link with find_element

…

```
drv.get('https://stark-bastion-95510.herokuapp.com/playground')
comment = drv.find_element(xpath: "//textarea[@id='comment']")
comment = drv.find_element(xpath: "//textarea[@id='foo' or @id='comment']")
comment = drv.find_element(xpath: "//*[@id='comment']")

comment.send_keys("STP Conference 2017")
```

…

Exercise. Find all links and provide a simple report that displays their text and “alt” value.

Objective: Leverage find_elements and manage the result.

```
elements = drv.find_elements( [LOCATOR ] )
…
puts elements[0].attribute(‘text’)
puts elements[1].attribute(‘alt’)
```

Ref: exercise3 “find_elements”

Waiting on elements and conditions

Explicit Waits

An explicit wait is code you define to wait for a certain condition to occur before proceeding further in the code. The worst case of this is Thread.sleep(), which sets the condition to an exact time period to wait. There are some convenience methods provided that help you write code that will wait only as long as required. WebDriverWait in combination with ExpectedCondition is one way this can be accomplished.

Explicit waits

Use the Wait class to explicitly wait for some condition:

```
wait = Selenium::WebDriver::Wait.new(timeout: 3)
wait.until { driver.find_element(id: "cheese").displayed? }
```

http://www.seleniumhq.org/docs/04_webdriver_advanced.jsp

Implicit wait ..

Implicit Waits

An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available. The default setting is 0. Once set, the implicit wait is set for the life of the WebDriver object instance.

Implicit waits

WebDriver lets you configure implicit waits, so that a call to `#find_element` will wait for a specified amount of time before raising a `NoSuchElementError`:

```
driver = Selenium::WebDriver.for :firefox
driver.manage.timeouts.implicit_wait = 3 # seconds
```

Which should I use ?

WARNING: Do not mix implicit and explicit waits. Doing so can cause unpredictable wait times. For example setting an implicit wait of 10 seconds and an explicit wait of 15 seconds, could cause a timeout to occur after 20 seconds.

Use .. explicit

- Waits on elements based on conditions with respect to the timeout
- Handles AJAX elements better than implicit

Demo: exercise3

Working with Frames

To find an element, located inside a frame (e.g. <iframe>), you must be “switched” into the frame that contains that element.

E.g.
drv.switch_to.frame [id|name]

Ref: exercise4/5_frames*

Ruby RSpec

rspec-core - Test Runner, reporter, manage tests

rspec-expectations - express conditions (assertions)

http://rspec.info/documentation/

Rspec – Simple example

./basic/frameworks/

Page Objects

Ref: ./basics/pageObjects

Useful modules (gems)

- Browza – Simple automation library.

  gem install browza

- Benchmark

  gem install benchmark

```
2.3.0 :048 >   Benchmark.measure { b.find_element(xpath: "//*[@id='comment']") }
 => #<Benchmark::Tms:0x007fb21958a110 @label="", @real=0.008711349917575717, @cstime=0.0, @cutime=0.0, @stime=0.0, @utime=0.0, @total=0.0>
2.3.0 :049 >
```

Thank You

Peter Kim

h20dragon@outlook.com

Web: https://dcast.io
Join Slack!

Join DCAST – DC Agile Software Test Meetup (+1,000 members)