# Selenium

## An Introduction to Python and the Selenium Module

Patrick Messina

Department of Mathematics
University of Houston, Harris County
Houston, Texas 77004

pmessina@math.uh.edu

July 20, 2016

# Selenium

- Selenium is a webdriver; web automation framework.
- An API designed to write functional/acceptance tests.
- Can be used with the following languages: Python, Csharp, Java, PHP, Ruby, and Javascript.

# Python

- Python is a high-level object-oriented interpreted language.
- Really easy to use third party modules such as Selenium.
- Most popular is CPython which is written in C, but IronPython is written in Csharp.

# Installing Python

- Python has two versions Python 2.7 and Python 3.X
- Either is fine, but 2.7 is what we will be using here.
- Python is pretty easy to install, just follow the instructions here: `https://www.python.org/`
- On Linux it is even easier. Python is already installed.

# Installing Selenium

- With Python 2.7.9 and Python 3.2, Pip now comes with Python. Pip allows us to install different modules easily. In the command line: *pip install selenium*
- In Visual Studio go to Tools $\rightarrow$ Python Tools $\rightarrow$ Python Enviornments, and you can download packages through pip here.
- Linux users should set up a virtual enviorment to prevent packages from interfering with your system.

# User Input

We first need to add the modules we are using.

```
1 import time, getpass
2 from selenium import webdriver
```

Then we get the username and password.

```
1 username = raw_input("Username: ") # Gets raw string.
2 password = getpass.getpass() # getpass prevents echo.
```

# Opening the Browser

We first open up the browser. I used Chrome, but we can use other browsers. See documentation. We then go to the chosen url.

```
1 browser = webdriver.Chrome('LOCATION OF CHROME DRIVER')
2 browser.get('https://www.casa.uh.edu/CourseWare2008/Root/
      Pages/Account/Login.aspx?ReturnUrl=%2fCourseWare2008%2
      fRoot%2fPages%2fLogin.aspx')
3 # browser.get('URL')
```

# Finding Login Entries

We are going to first find the field entries. We do this by inspecting the element by right clicking the element and selecting inspect. Then we find the id.

```
<input name="userNameInput" type="text" id=
"userNameInput" class="cw-inline-block"> == $0
```

We now find the entries using Selenium.

```
1 username_field = browser.find_element_by_id("userNameInput")
2 password_field = browser.find_element_by_id("passwordInput")
```

# Sending Keys

We now send keys.

```
1 username_field.send_keys(username)
2 password_field.send_keys(password)
```

# Actually Logging In

We can also find code by class name. I have also included how to find an element by using XPath, that is by using XML. We also submit the form.

```
1 login_attempt = browser.find_element_by_class_name("
    loginButton")
2 # login_attempt = browser.find_element_by_xpath("//*[@type='
    submit' and @id='loginButton']")
3 login_attempt.submit()
```

# Useful Code

The print statement is self explanatory.

```
print("This works in Python 2.7 and Python 3.X")
print "This only works in python 2.7"
```

The quit fucntion will close the browser.

```
browser.quit()
```

The sleep function from the time module is useful to slow the program down to see what is happening.

```
time.sleep(1) # sleep for 1 second
```

The click function simulates a mouse click. Add to the end of an element to click.

```
browser.SOME_ELEMENT_FOUND_BY_SELENIUM.click()
```

# Projects

- Try opening a browser and doing a google search, finding the link you want, and then going to that webpage.
- Try logging onto your email, sending an email, and then logging out.

# Resources

- Python: Automate the Boring Stuff With Python is free and has a section on using Selenium. It also covers many different areas of Python while teaching the language. It is free online (link) and has an accompanying youtube playlist (link).
- Python: Python Crash Course is a more in depth coverage of the Python language.
- Selenium: The unoffical Python documentation is great (link).

# Other Modules

- Beautiful Soup: Used to pull data from HTML and XML files.
- urllib2: Used to navigate the web.
- requests: Also used to navigate the web.

An sample project could be to use requests to download json files from a weather site, parsing the json file (using *json*), and then send weather notifications via email or text.