Intro to Python and Webscraping

Robert Vesco

May 26, 2014

Class Objectives

- Introduce basic python and webscraping
- Provide skills & knowledge not in online tutorials
- Tools that can be used with any programming language

Plan

- 9 - 9:15: Setup issues
- 9:15 - 9:30 Python in Scientific Computing
- 9:30 - 9:45 Anaconda & Spyder
- 9:45 - 10:30 Command line basics
- 10:30 - 12:00 Python Basics
- 12:00 - 12:30 Lunch
- 12:30 - 3:00 Python Webscraping
- 3:00 - 4:30 Practice with your own site
- 4:30 - 5:00 Other Tools, Development Environment

Abbreviated/Opinionated History of Programming Languages

- C, C++
- Awk, Sed & shell scripts
- Practical Extraction and Reporting (perl)
- S (R precursor)
- Java
- Ruby (perl 2.0)
- R
- Python
- Julia (R 2.0)

Python and Stats

Python and Jobs

Python Considerations

Support For

- Readability & Consistency (pythonic)
- Fairly fast
- Not Java
- Used in biz ops & domains

Support Against

- Backward compatibility
- Fragile package dependencies
- Fragmentation
- Complementary Assets for Science

The many faces and versions of Python

- Cython (main)
- IronPython (.net)
- PyPy (JIT)
- Jython (compiles to java)
- Ipython (scientific and interactive)

Anaconda and Spydyer

- Anaconda is a pre-packaged python distribution for scientists
- Spyder is an IDE (Integrated Development Environment)
- Open a terminal or click spyder

```
1 anaconda/bin/spyder
```

- Open terminal within spyder

Why Terminals and Command Line Programs?

- Troubleshooting python programs
- Managing programs and files
- Right tool for some jobs

Shells vs Terminals

- Shells are programs (like python) that help you interact computer.
  - csh (c shell, mostly seen on older servers)
  - bash (most common)
  - zsh (most convenient)
- Terminals are wrappers around shells (iterm2 for macs)
- .bashrc, .cshrc, .zshrc are configuration files for shells

Top Aligned Blocks

Code

Cool Lots of Stuf
To talk
about

Result

pretty nice!

Inline math

Beamer: Animated Bullets

- Trouble Shooting

Beamer: Animated Bullets

- Trouble Shooting
- A framework for thinking about programming

Beamer Columns

Stuff

- Truth is ephemeral

- What is right?
- What is Wrong?

How to use virtualenv & pip

```
1  ## run this on the command line
2  ## assuming you are in your projects folder, create a new
       folder
3  mkdir projects1
4
5  cd projects1
6
7  ## now create your virtualenv environment
8  ## this will create a folder called "env".
9  ## this will house a local version of python.
10 virtualenv env
11
12 ## IMPORTANT.
13 ## Now you need to activate your environment.
14 source env/bin/activate
15
16 ## now you will be using a local version of python instead
       of your
17 ## system's python
18
19 ## to deactivate, simply type
20 deactivate
```

How to Share Ipython Notebooks

How to share your vagrant box

Testing Python Output

```
1 a = ('b', 200)
2 b = ('x', 10)
3 c = ('q', -42)
4 return (a, b, c)
```

b	200
x	10
q	-42

Python Output

```
a = ('b', 200)
b = ('x', 10)
c = ('q', -42)
return (a, b, c)
```

By removing the :exports both, you can export just the code and not the output. By replaceing it with :exports results, you can export the output without the source.

Using pip once virtualenv is activated

```
## again, these should be run on the command line.
## first, let's activate your virtual environment, if you haven't
## already
source env/bin/activate

## first, let's inspect what command are available in pip
pip help

## from this, we see that there are a number of commands we will
## find useful
pip list # this shows what programs are already installed
pip search numpy # this searches for packages named "numpy"
pip install numpy # this installs the numpy package.

## if you have many packages you want to install, you can
## create a requirements list
## this will create a file with a list of modules to install
## you can use your editor of choice to install this.
echo "numpy\nbeautifulsoup" > requirements.txt

## this will install all the packages in the text file.
## NOTE: you can specify the versions of module too
```