First Scripts in Python

Jamie Saxon

Introduction to Programming for Public Policy

September 26, 2016

Simple Scripts

Writing a Script

- Our ‘salaries’ command was getting hard to read (for humans).
- A script is a saved series of instructions for the computer.
- We can write that series in many languages: bash, python, perl, tcl, ...
- The command line language is ‘bash,’* and can be run via ‘source‘.

```
■ echo "echo hello" > hw
■ source hw
hello
```

- But ‘echo’ is not a great way to write our long command!

*Bourne Again SHell

Editing Text and Writing Code: Vim or Atom

- Computers 'interpret' your code, or run a 'compiled binary.'
  - **Interpreters** are computer programs that follow your instructions 'step-by-step.'
  - **Compilers** translate what you write into something the computer understands 'natively' (ones and zeros = binary).
- Files must contain exactly and only the code: nothing extraneous.
  - So, needless to say (?), Microsoft Word won't cut it.

- I propose to use **Atom** (gui) or **vim**: Vi IMproved (vi = visual).
  - The advantage of vi is that it's all from the command line.
  - Emacs (also command line) has many partisans, as well.
  - But emacs and vi have very high learning curves.
- Sometimes we'll use 'Jupyter' – a browser based environment.

Putting the Salaries Script Into a File

The long command: **tinyurl.com/z45357y**

```
#!/bin/bash
```
← Which language?

```
# SORT CHICAGO SALARIES
```
← # : comment

```
# start the party!!
echo "Top 10 Salaries in Chicago"
cat salaries.csv |
    grep '\$' |                 # keep only lines with dollar signs.
    sed 's/\$//g' |             # remove the dollar signs.
    sed "s/\"//g" |             # remove the quotes
    sort -t, -k 5 -n -r |       # sorting is the best.
    head -10 | column -s, -t    # clean it up
```

A First Python Script

Create a file `hello_world.py`, using vim Atom, TextEdit, etc.

Write `print("hello world")` in this file and save it.

Navigate to the directory that holds that file:
`cd /Users/jsaxon/Documents/...`

Type: `python hello_world.py`

- To go further, we need the rules and building blocks of Python...

chmod

- chmod allows you to change the permissions of a file
- Each file has separate 'permissions' for whether you (u), people in its 'group' (g), or anyone (o), can read (r), write (w), or execute/run (x) the file. You can add (+), remove (-), or set (=) permissions.
- Most often, use it to make a script executable, perhaps just for you:

```
■ cat hello_world.py
#!/usr/bin/env python
print("hello world")
■ chmod u+x hello_world.py
■ ./hello_world.py # don't need 'python'
hello world
```


python
TM

What is Python?

Python is a popular, high-level interpreted programming language.

Interpreted: computer 'runs your instructions,' so you can:

- Run **interactively**: execute one line of code at a time.
- Or **script**: write down and save all of your commands.
- (It actually compiles itself, 'just-in-time.')

High level: python hides a lot of the complexity from you.

- You don't have to worry about moving bits (1s and 0s) around.
- And you can accomplish a lot, with relatively little code.

Through the next two lecture, we'll discuss:

Types

Control

Assignment, Operators, and Methods

Some of them will be mixed together.

Launch python on your computer, or open a jupyter notebook:

Command line: 'python' (must be python 3).

Local Jupyter: Anaconda Navigator

Online Jupyter: tmpnb.org or try.jupter.org

Click 'New' in the upper right corner, then Python 3.