PreWS Day 2: Basic Python

David Yu

July 8, 2014

Goals for today

- Have a working python installation with numpy, scipy, matplotlib, scikit-learn, and ipython.
- Learn python syntax, and write a small script.
- Manipulate built-in python data structures: lists, dictionaries, tuples, and sets.
- Read and write data from file.

Python!

```
>> import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one– and preferably only one –obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea – let's do more of those!

Why Python?

- You will find out!
- Programming is fast and easy.
- Code is readable ("executable pseudocode").
- Large community.
  - Many packages available - we'll use some of the most popular modules for scientific python.

Which Python?

- Anaconda: All-in-one python installation with centralized extension management. Comes with a good text editor (IDE).
- Enthought Canopy: Similar to Anaconda. Comes with a basic text editor.
- Built-in python: OK for Linux. Not recommended for OSX; tends to be a little out of date.
- Other (Macports, Homebrew, apt-get, yum): Feel free to use whatever you're comfortable with!

A note on versions: we'll use python 2.7 for the pre-workshop, rather than python 3.X. You might be able to make python 3.X work, though!

Essential Packages

- ipython: "A command shell for interactive computing."
- numpy: "The fundamental package for scientific computing with Python.""
- scipy: "Provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization."
- matplotlib: "2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.""
- sklearn: "An open source machine learning library."

Essential Packages

- Anaconda/Enthought Canopy: Distributions come with python, so you're all set.
- Otherwise: Install packages according to your setup, e.g.,

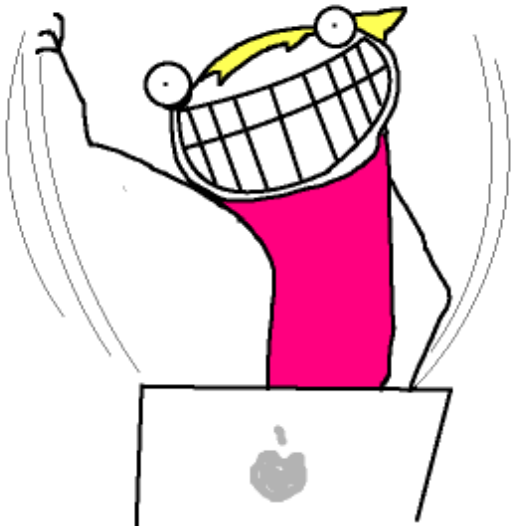
```
sudo pip install -U ipython, numpy, scipy, matplotlib, \
     scikit-learn

sudo port install py27-matplotlib py27-numpy py27-scipy \
     py27-ipython py27-scikit-learn
```

Text Editors

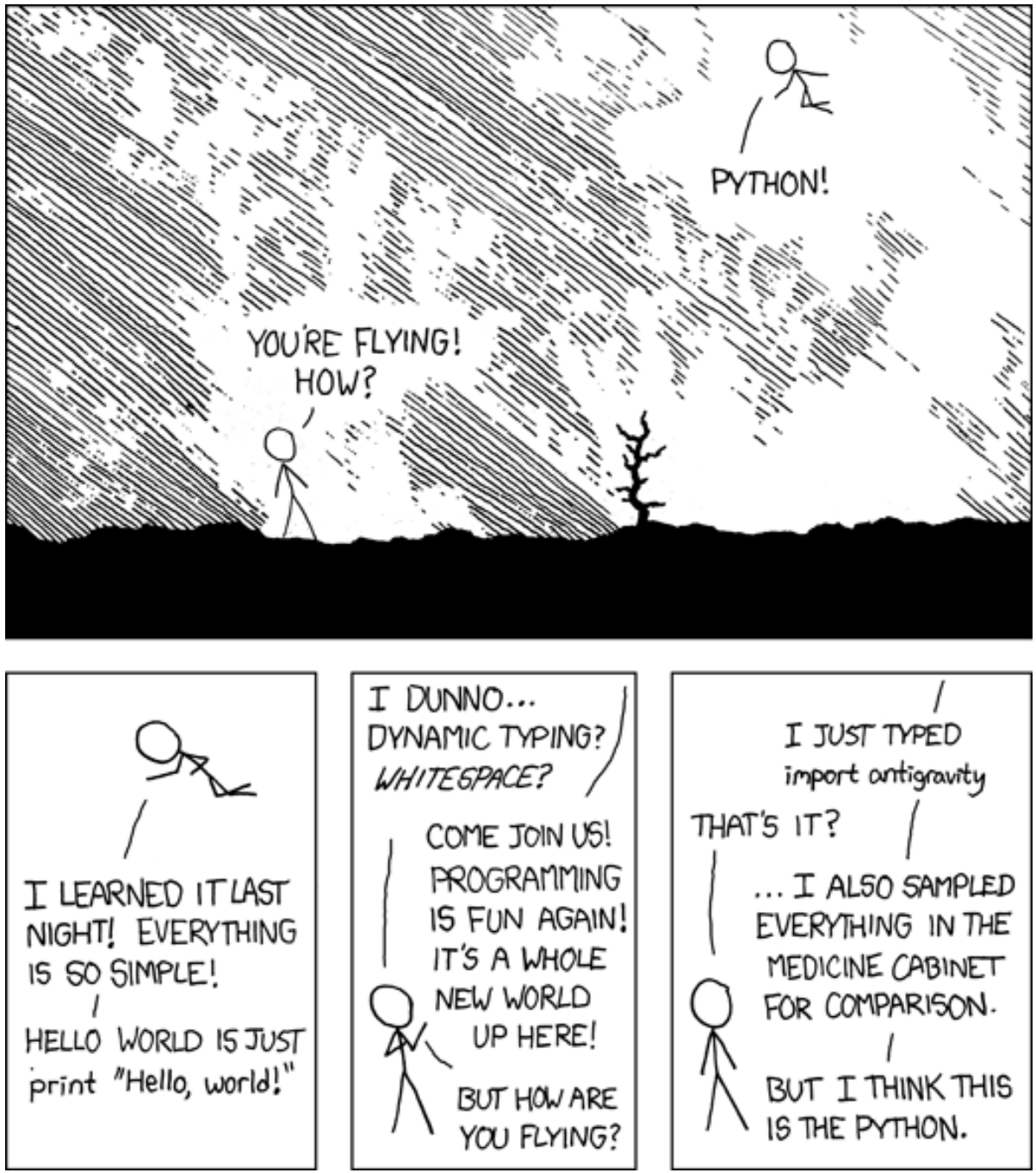
- Spyder: Manages files, intelligent auto-completion, execute code all in one window. Comes bundled with Anaconda.
- Sublime Text: Popular, decked-out text editor with many plugins.
- emacs/vim: Blah.
- Other: XCode, Coda, TextMate, Notepad++, TextWrangler...

IPython Notebooks

- The tutorials today are in IPython notebooks, Mathematica-like, interactive command shells.
- Start the notebook server in `Public.PreWS/Day2` by typing `ipython notebook` in a terminal, or by clicking on the "IPython Notebook" icon (Anaconda/Windows).
- The tutorials cover:
  - Python syntax and control flow.
  - Data structures.
  - File I/O.



Let's get started!


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Other Resources

- Python official documentation.
- Google's Python class and style guide.
- Dive Into Python.
- A giant list of modules.