

Python Workshop - ANU (CBE)

Setup and Installation, Jupyter Notebooks, and Python Programming

Matthew McKay [matthew.mckay@anu.edu.au]

June 2017

Agenda ...

1. Python (Anaconda) Installation
2. Intro to Python
3. Tools for Writing Code
4. Jupyter Notebooks
5. Python Language Fundamentals

These notes are available here:

<https://github.com/mmcky/anu.python-cbephd.june-2017>

Note: The version number for Anaconda has been updated since this guide was put together. The current version is **4.4.0**

Anaconda

What is Anaconda (Python Distribution)?

1. Provides python along with many useful python packages
 - Python 3.6 = 485 Packages
 - Python 3.5 = 488 Packages
 - Python 2.7 = 530 Packages
2. Provides infrastructure for managing those packages

Has become the standard for data science and is very convenient

Installation Guides

The following installation guides to install Anaconda are available for:

1. Linux¹
2. OS X
3. Windows

on the Github site:

<https://github.com/mmcky/anu.python-cbephd.june-2017>.

¹Ubuntu and its derivatives are the most common. Linux Mint is used in the installation guide.

Some simple tests

Run these commands in a terminal.

1. Try updating conda by typing: `conda update conda`
2. Try updating the anaconda library by typing:
`conda update anaconda`
3. Open IPython Notebook by typing: `jupyter notebook`. Open a new notebook and try out a few python examples
4. Install QuantEcon library by typing in a terminal:
`pip install quantecon`. Next open an Jupyter notebook and try importing the library using: `import quantecon as qe` in one of the code blocks

Note: For Windows systems these should be run in a cmd or powershell terminal.

What is Python?

Python is a general purpose high level programming language²

Python has experienced rapid adoption in the last decade and is now one of the most popular programming languages.

<http://github.com/>

It is currently 3rd most popular on GitHub

²started in 1989 by Guido van Rossum

Why Python?

Python is:

1. free
2. a full programming environment
3. easier to learn than some other languages
4. highly productive
 - Large library of user contributed packages
 - high level language design
5. has a large and active community
6. cross platform
7. ...

Provides a powerful environment for scientific research and computation.

Language Comparison: Python

```
>>> print("Hello World")  
"Hello World"
```

Language Comparison: C

```
#include <stdio.h>

int main()
{
    printf("Hello world\n");
    return 0;
}
```

Run Program after compilation with gcc:

```
./a.out
"Hello World"
```

Language Comparison: Assembly

```
.file      "simple.c"
.section   .rodata

.LC0:
.string    "Hello world"
.text
.globl     main
.type      main, @function

main:
.LFB0:
.cfi_startproc
pushq      %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq       %rsp, %rbp
.cfi_def_cfa_register 6
movl       $.LC0, %edi
call       puts
movl       $0, %eax
popq       %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc

.LFE0:
.size      main, .-main
.ident     "GCC: (Ubuntu 4.8.4-2ubuntu1~14.04) 4.8.4"
.section   .note.GNU-stack,"",@progbits
```

Python Usage

Scientific Community

1. Machine Learning
2. Astronomy
3. Artificial Intelligence
4. Chemistry
5. Biology

But also used extensively to manage servers, computing clusters, websites etc. by many companies such as Amazon, Google,

Python Features

1. A high level programming language
2. Expansive library support
3. a multiparadigm language (procedural, object-oriented, scripting etc.)
4. Interpreted rather than compiled (Good and Bad)
5. Elegant Syntax (Easy to Read and Understand)
6. Lots of language features (iterators, generators) that allow the language to be highly expressive.
7. Concise

Python 2.7 or 3.6?

Python 2.7

- Pro
 - More packages are available in Python 2.7
 - A lot of examples are written in Python 2.7 syntax.
- Con
 - In maintenance mode - not getting new features as the language develops over time.

Python 3.6 (Best **default** selection)

- Pro
 - Newest version which is the long term future of Python
 - Most of the scientific stack has been ported to Python 3
- Con
 - Sometimes want to use a library which has not been migrated to Python 3 yet. (but can make use of conda environments if needed)

Ways to use Python

The main ways are:

1. `python` REPL³
2. `ipython` REPL
3. `jupyter`

Jupyter notebooks (formerly ipython notebooks) is a really good place to start and will be used extensively in this course.

³Read-Eval-Print Loop

Start python in your terminal: python

XKCD Cartoon:

```
>>> import antigravity
```

The language is named `python` in part because Guido's a big fan of Monty Python's Flying Circus.

The “Zen” of Python

```
>>> import this
```


IDE's

There are a couple of interesting IDE environments that can also make working with Python a little easier. The best is probably ...

Spyder IDE

You may wish to use it to start with as it provides a TextEditor, a python REPL, and some Data and Documentation Panes that are “integrated” together.

Spyder IDE also comes as part of the Anaconda distribution.

Demo

Ways to write Python Code

Down the track - you may wish to start writing your python code in a full text editor.

1. Sublime Text [My Favourite]
2. Atom
3. Emacs
4. Vim

Many others ...

Why use a Text Editor

Demo

1. Syntax highlighting
2. More productive (tab completion, auto-indentation)
3. Regex and pattern matching
4. ...

Warning: Choosing your “favourite” text editor can become time consuming and endless ... :)

Jupyter

Jupyter is an excellent interactive environment that is used extensively in the Data Science community

Learn more **here**

Supports the notion of executable documents ...

Live Jupyter Demo

Start Jupyter Notebook from the terminal,

```
bash \ $ jupyter notebook
```

Jupyter Notebook Topics

1. Notebook Basics
2. Modal Editing
3. Running Code
4. Text Editor Features (Syntax Highlighting etc.)
5. Tab Completion
6. Object Introspection
7. Working with the shell
8. Working with Files
9. First Python Program

Swap to “intro-to-jupyter-notebooks.ipynb” notebook

Intro to Python Topics

1. Introductory Example
2. Basic Structure of a Python Program
3. Assignment
4. Data Types

Swap to “intro-to-python.ipynb” notebook

Additional References

The main reference is:

https://lectures.quantecon.org/py/learning_python.html

Additional References:

1. “Think Python”, Allen B. Downey, Oreilly Media
2. “Data Science from Scratch”, Joel Grus, Oreilly Media
3. “Python for Data Analysis”, Wes McKinney, Oreilly Media