A Short and Incomplete Introduction to Python

Part 0: Introduction

Riccardo Murri <riccardo.murri@uzh.ch>,
Sergio Maffioletti <sergio.maffioletti@uzh.ch>
S3IT: Services and Support for Science IT,
University of Zurich

January 25–26, 2018

Welcome!

Prerequisites

This course assumes a basic experience with computer programming.

Any language should do, as long as you are already familiar with the concepts of variables and functions.

Python 2 *vs* Python 3

There are currently two major versions of Python available, with slightly different syntax and features.

Python 2.7 is the last release in the 2.x series.

Python 3.x has a more polished syntax, removing inconsistencies and some historical baggage.

In this course we will use **Py3 syntax**.

Watch a debate between "Pro" and "Contra" advocates:
http://www.physik.uzh.ch/~nchiapol/webm/3_1_Python3.webm

Explore the key differences:
http://tinyurl.com/py2-and-py3-key-differences

Talk outline

1. Python basics
2. NumPy and plotting
3. Workflows with GC3Pie

Next steps

The course will be structured as a mixture of slides and hands-on sessions for practicing Python programming.

So, the very first step is making sure you can access the Jupyter/IPython server for running the exercise notebooks.

How to run Python code

The Python shell, I

Python is an *interpreted* language.

It also features an interactive "shell" for evaluating expressions and statements immediately.

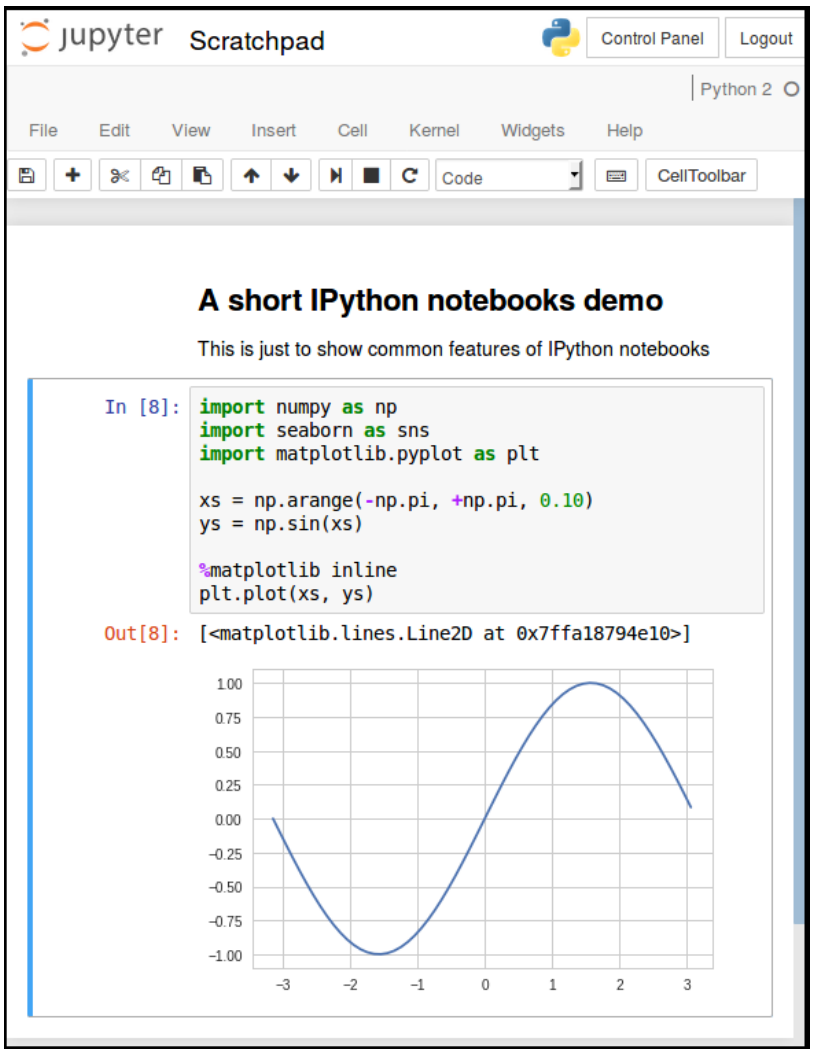
The IPython shell is started by invoking the command `ipython` in a terminal window.

```
$ ipython
Python 2.7.13 |Anaconda 4.3.0 (64-bit)| (default, Dec 20 2016, 23:09:15)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 - An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: ← here is where you enter commands
```

The IPython notebook, I



A more appealing way of interacting with Python is through the IPython notebooks.

Notebooks are made of “cells”, which come in two flavors:

- documentation cells, containing text formatted according to the Markdown conventions;
- code cells, containing arbitrary Python code

The IPython notebook, II

To run Python code in the notebook:

- Type your code in a cell besides the `In [ ]:` (multiple lines are allowed)
- Press **Ctrl+Enter** to evaluate the cell (prompt changes to `In [*]:`) — or press **Alt+Enter** to evaluate the code *and* open a new code cell.
- When the Python kernel has done computing, the result appears *under* the code cell marked with a `Out [ ]:` label.

The Python shell, II

Expressions can be entered at the Python shell prompt; they are evaluated and the result is printed:

```
In [1]: 2+2
Out[1]: 4
```

Note that the classic Python shell uses '>>>' as a prompt; expression evaluation works exactly the same, though:

```
>>> 2+2
4
```

Throughout these slides, all Python code marked with either 'In [*]' or '>>>' can also be entered and evaluated in the IPython notebook cells.