Introduction to Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

In *Processament de So i Música* course, we use Python version 2.7 with a set of additional modules. This document helps you to get started with installing python and the necessary modules on linux/windows/mac operating systems. Besides the material we provide for learning python as needed in the course, there are few resources listed as well to give you a head start before our first seminar and lab for introducing python.

We support and help you with installations on ubuntu 12.04 using python 2.7. If you have any other linux, that should be fine as well! We do provide some helpful instructions for windows and mac os x operating systems, but if you should face an issue somewhere, you are on your own. If you want to use the recommended ubuntu 12.04, but have a windows/mac system with you, you can install ubuntu 12.04 on virtualbox.

1 Installing Python 2.7

1.1 Linux

On most linux operating systems, python should be installed by default! To check, open a terminal and type (without the dollar/$ sign, which stands for command prompt):

```
$ python --version
```

It should show you the python version installed. If it says python 2.7.*, it's perfect. You can skip to the next section for installing required modules. If it shows python 3.* or gives an error that the command was not found, follow the instructions below to install python 2.7:

Ubuntu/debian distributions

```
$ sudo apt-get install python2.7
```

Fedora/rpm-based distributions

```
$ sudo yum install python27
```

Alternatively, you can also use the graphical version of your distribution's package manager (like synaptic/yumex) and use it to install python 2.7.

1.2 Windows/Mac

We suggest using Anaconda, a collection of core python with all the required modules. Once you install it (making it the default python interpreter on your system), you can access it from the start/application menu.

2 Installing modules

2.1 Linux

Python has a very nice tool to install and manage the additional modules/packages, called pip. We install it once and use it to further install the necessary modules. As some of the modules need python header files to be compiled when being installed through pip, we also install python-dev package.

That said, all the linux distributions have most of the modules we use in their packages repositories. So, we install some of them using the distribution's installation tool (apt-get/yum). And the others (in particular the ones we want the latest versions of), we install them using pip.

Ubuntu/debian distributions

```
$ sudo apt-get install python-pip python-dev python-scipy
python-numpy python-matplotlib python-pyaudio python-pyside
cython
```

Fedora/rpm-based distributions

```
$ sudo yum install python-pip python-dev scipy numpy python-
matplotlib pyaudio pyside Cython
```

Using pip to install other modules on <u>all distributions</u>

```
$ sudo pip install --upgrade distribute
```

```
$ sudo pip install --upgrade ipython pyzmq jinja2 tornado
```

2.2 Windows/Mac

If you have installed Anacoda as mentioned in the previous section, then all the modules will have already been installed. If you have your own setup of python, you can refer to the following webpages for the installation of various modules:

- numpy and scipy: http://www.scipy.org/install.html
- matplotlib: http://matplotlib.org/downloads.html
- ipython & ipython notebook: http://ipython.org/ipython-doc/stable/install/install.html#installnotebook
- cython: http://docs.cython.org/src/quickstart/install.html
- pyaudio: http://people.csail.mit.edu/hubert/pyaudio/
- pyside: http://qt-project.org/wiki/Category:LanguageBindings::PySide::Downloads

3 *Basics of python programming*

3.1 Ipython notebook

In this document, we get you started with using ipython notebook, which we'll use in the first lab for a hands-on session on python. Basically, there are two ways to use python interpreter: a) interactive mode, b) script mode. The interactive mode can be started by running:

```
$ python
```

It opens up a console like this one:

```
[gopal@rayalu ~]$ python
Python 2.7.3 (default, Aug  9 2012, 17:23:57)
[GCC 4.7.1 20120720 (Red Hat 4.7.1-5)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This default console is quite basic when it comes to productivity enhancing features such as tab completion, code pasting, history etc. So, we use a popular alternative, known as ipython. If you followed the instructions to install all the modules, you can start it by running:

```
$ ipython
```

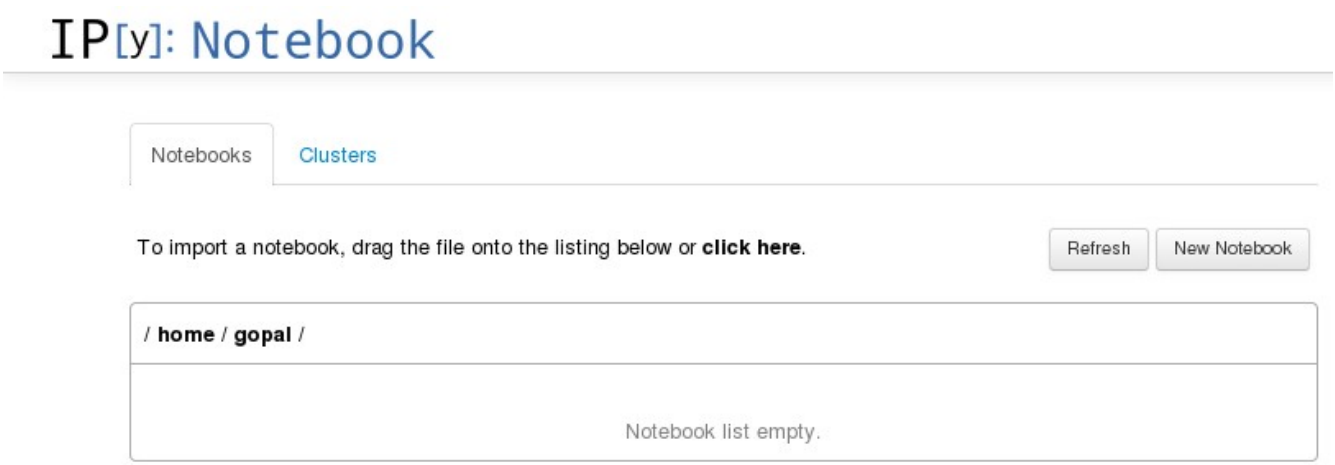
This opens up a console like this one:

```
[gopal@rayalu ~]$ ipython
Python 2.7.3 (default, Aug  9 2012, 17:23:57)
Type "copyright", "credits" or "license" for more information.

IPython 1.1.0 -- 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]:
```

Ipython comes with many productivity features that make life much happier. It also has a notebook that is well suited for doing things interactively and managing the code blocks intuitively. Go to a directory where you want to save those notebooks, and start the notebook server by runing:

```
$ ipython notebook
```

This will start the server and opens the localhost address in a browser where you can create/access notebooks (like the figure below):



For this course, we have prepared few notebooks to give a quickstart to python and the modules we will use in the course. Download them from the moodle (they are placed with Lab 1), place them in a directory and start the `ipython notebook` in the same directory. In the browser window that opens, click on the first notebook to get started!

4 More tutorials to acquire basics of python

- Code Academy's interactive venue to learn python (recommended):
  http://www.codecademy.com/tracks/python

- A nicely written basic tutorial from Google developer education materials:
  https://developers.google.com/edu/python/

  This has a set of basic exercises as well to test your understanding. It also has an elaborate set of videos if you want to learn more.

- This is the tutorial from the official python website:
  http://docs.python.org/2/tutorial/introduction.html

  A bit lenghty, but covers all the basics of python, and all the standard libraries/modules!