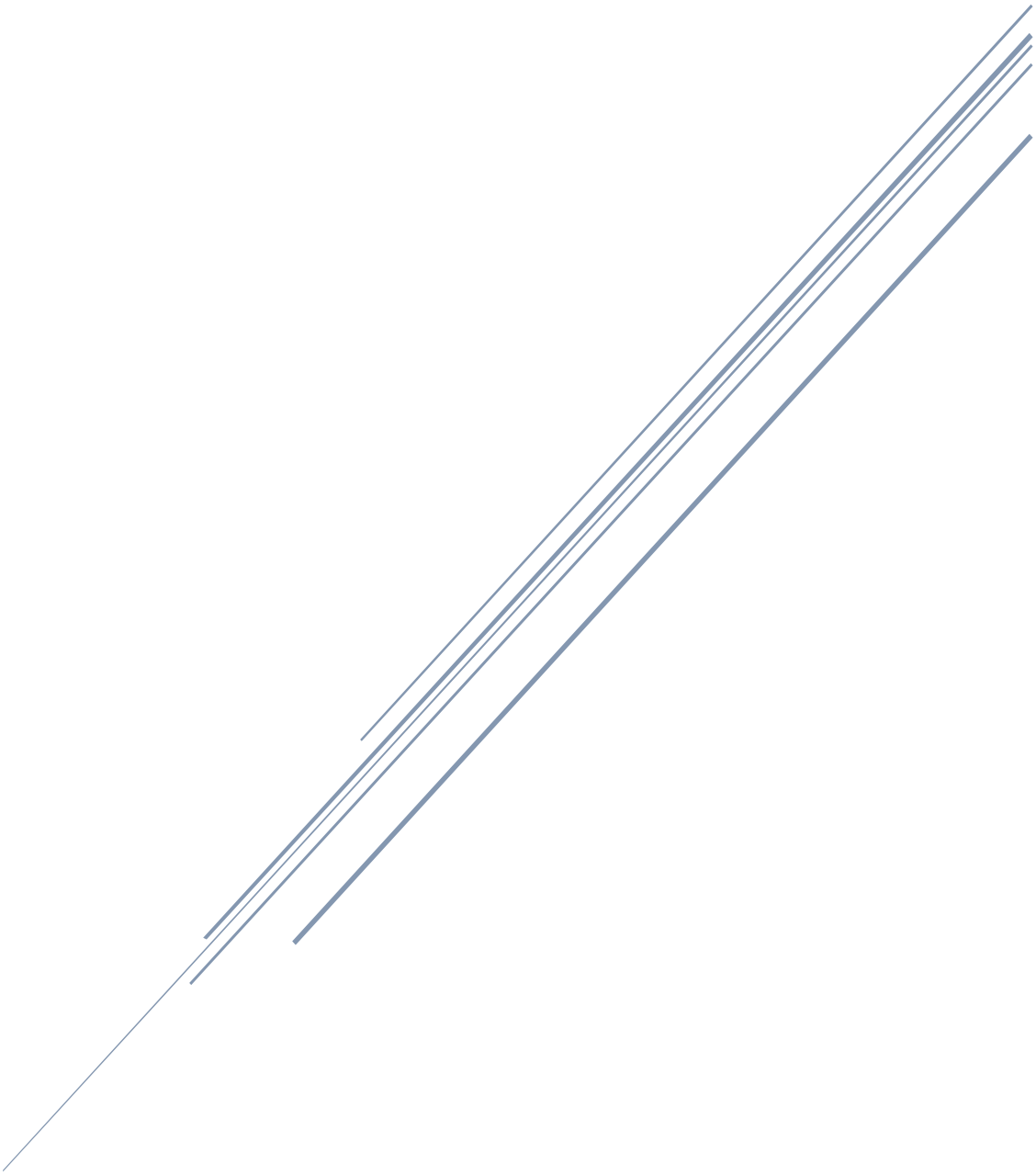
INTRODUCTION TO PYTHON

Installation and execution guide



Universitat Politècnica de Catalunya
Approaches to Machine Translation

What is Python?

Python is an **interpreted**[1] high-level programming language, with support for multiple programming paradigms, including object-oriented, imperative and functional programming.

Python stands out for its code readability and simplicity. It uses whitespace indentation, instead of keywords or curly braces, to delimit blocks.

There are currently two major versions of Python: Python 2 and Python 3[2]. Along the course, we will use Python 2 latest version, Python 2.7.

Installation

Windows

Although a broad set of Integrated Development Environments (IDEs) is available at the network, our *personal* recommendation is to use Canopy, as it provides easy installation of Python packages.

"Canopy is an easy way to install Python 2.7 with the tools you need for graphical analysis, visualization of data and script development on your computer, be it Windows, Mac or Linux." [3]

The standard version of Canopy is **free** and available to download at:

https://store.enthought.com

Once you have downloaded Canopy, you should proceed to its installation.

[1] An **interpreted language** is a programming language for which most of its implementations execute instructions directly, without previously compiling a program into machine-language instructions
[2] What's New in Python 3.0 https://docs.python.org/3/whatsnew/3.0.html
[3] Enthought Canopy: https://www.enthought.com/products/canopy

Linux / MacOS

Linux based platforms including Mac OSX distributions include a version of Python by default.

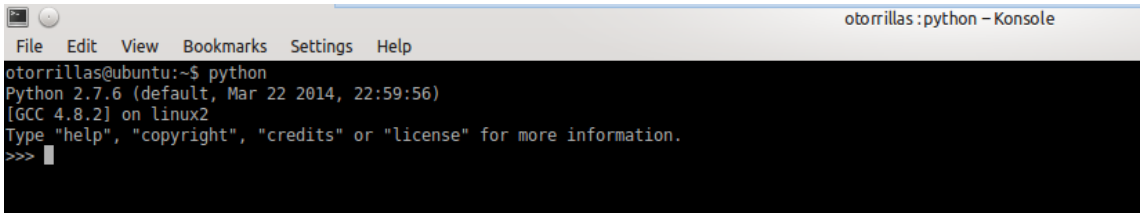
To check if you have Python installed on your system, open a terminal

- ◊ **Linux**: Applications -> System -> Terminal.
- ◊ **MacOs**: Applications -> Utilities -> Terminal.

and enter the command:

$ python

An output similar to the following should appear.



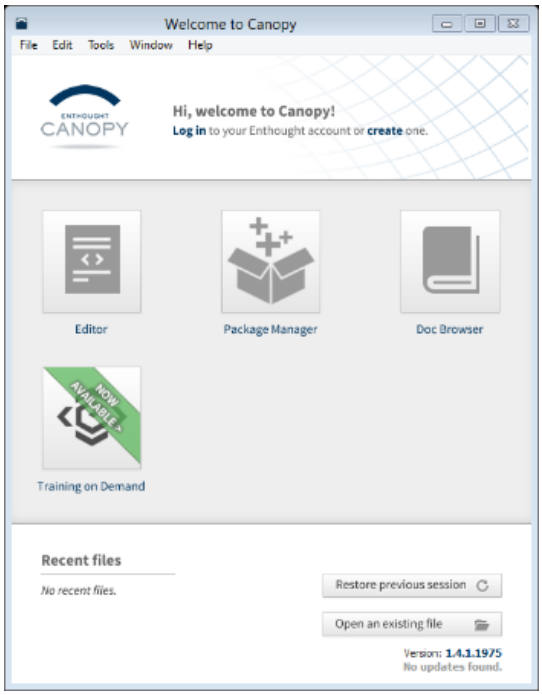
Execution example

As an introductory example, we would execute a Python script that outputs the sum two input arguments. The code is the following:

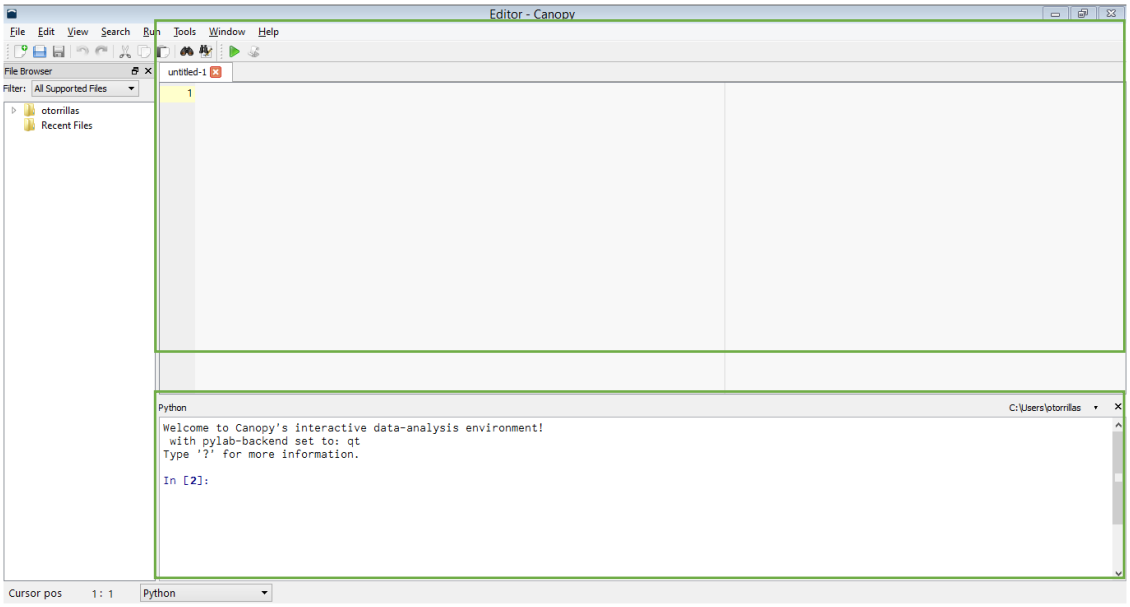
```
import sys

if __name__ == '__main__':
        if len(sys.argv) < 3:
                # In case we have provided less than program_name + 2 arguments
                print "Usage: python sum.py arg1 arg2 .. argn"
                print "Output: arg1+arg2+...+argn"
        else:
                # We avoid argv[0], which is the name of the program (sum.py)
                numbers = [int(arg) for arg in sys.argv[1:]]
                our_sum = sum(numbers)
                print "Our sum: %s" % our_sum
```

Using Canopy

If you have successfully installed Canopy, a window similar to the following should appear after launching it:



Then click on Editor button and create a new file. Now Canopy should look similar to the following snapshot:



Editor

Interactive environment

Paste the example code into editor window and save it as sum.py. Then try to execute the file pressing **Ctrl+R** or, in the menu, **Run -> Run File**. You will see that, as we have not provided any arguments, our previously written usage is printed.

```
In [2]: %run C:/Users/otorrillas/sum.py
Usage: python sum.py arg1 arg2 .. argn
Output: arg1+arg2+...+argn
```

To run Python scripts with Canopy, simply write at Canopy interactive environment:

%run sum.py *arg1 arg2 … argn* *# arg1, arg2…are the arguments that we want to sum.*

This will output something like:

```
In [4]: %run sum.py 2 3
Our sum: 5
```

Pipes in Canopy

To execute programs that use pipes, like you would do in Command line on Linux

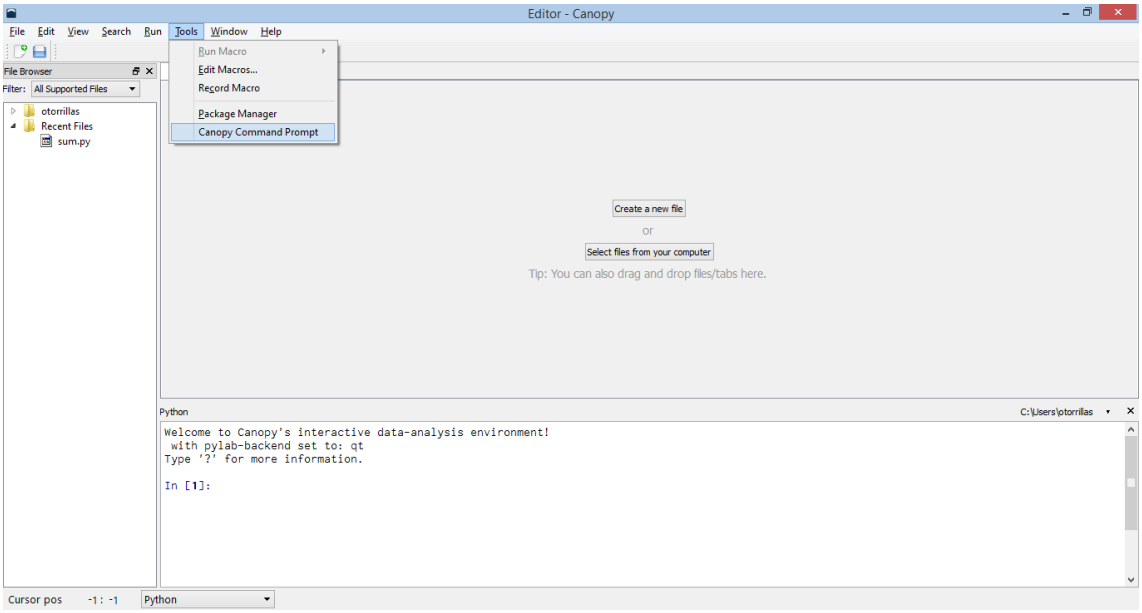
For example:

$python program1.py | python program2.py

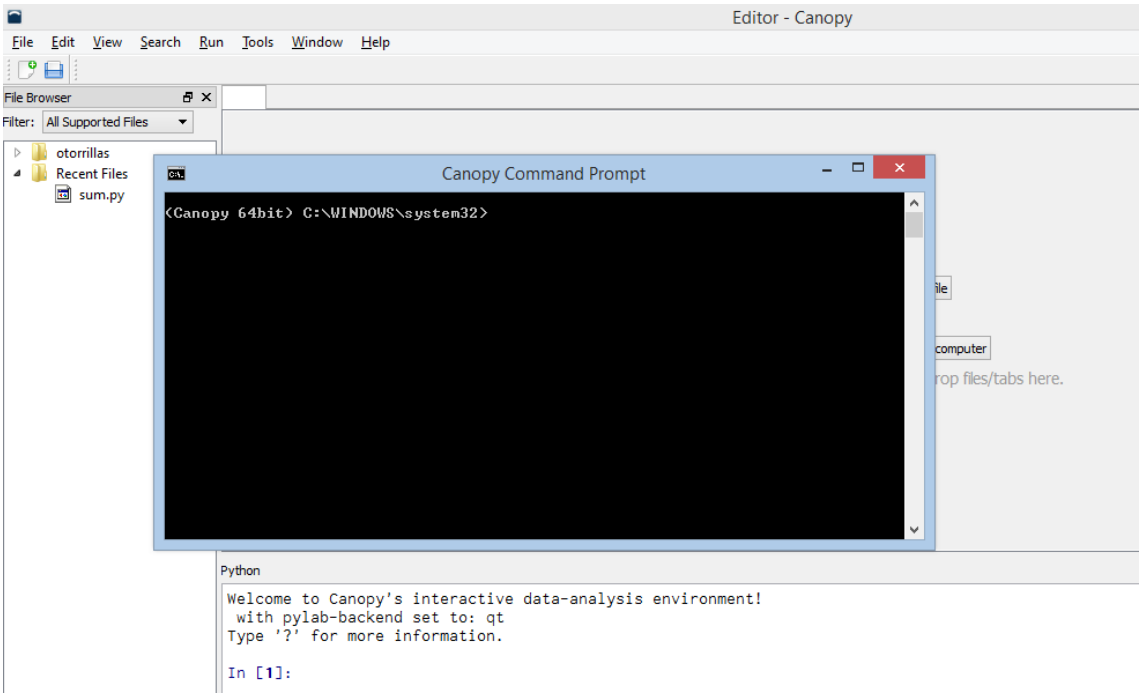
You have to:

1. Open Canopy Editor.

2. In the top bar, press Tools -> Canopy Command Prompt



3. A shell is opened and from now on, you can execute programs with pipes.



4. If you want to change the folder, you can simply type the traditional change directory command (cd) or you can drag and drop folders directly to the command prompt.

Using command line (Linux/MacOS)

Instead of using Canopy, we can directly execute a Python script from terminal. Firstly, you should create a file named **sum.py** with a text editor. Some *examples*:

- ◊ **Linux**: gedit, Kate, Vim.
- ◊ **MacOs**: Mac Vim, Text Edit.

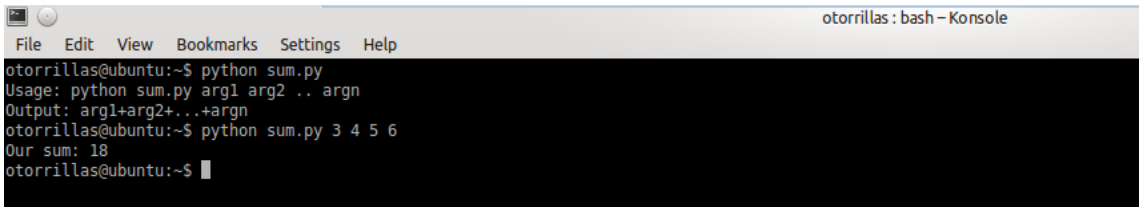
Once we have saved the file, for example in home directory, you should open a new terminal:

- ◊ **Linux**: Applications -> System -> Terminal.
- ◊ **MacOs**: Applications -> Utilities -> Terminal.

This will open a terminal in your home directory path[4]. Then, simply type in the terminal:

$ python sum.py *arg1 arg2 … argn* *# arg1, arg2…are the arguments that we want to sum.*

Outputs should be similar to the following:



[4] Note that if you have saved our **sum.py** somewhere else, we should firstly change the directory of the terminal via the cd command.

Numpy

Numpy[5] is a Python library for scientific computing. It contains sophisticated functions, a powerful N-dimensional array object, together with other mathematic utilities.

We will need the numpy library in our Alignment problem.

Installation

Canopy

If you have installed Canopy, there is no need to install anything more. Numpy comes by default in Canopy standard version.

Linux/MacOS[6]

For the installation of numpy, there are a varied possibilities of installing the numpy utilities. Although we are going to describe both methods, we recommend working with Python virtualenv, a tool to install and remove Python packages easily, without affecting the global Python installation. Moreover, with virtualenv you can have and work with more than one Virtual Environment at the same time, isolating one environment installation from another.

Traditional installation

Linux

According to your distribution, you can install numpy globally, typing the following command:

- **Ubuntu**: sudo apt-get install python-numpy
- **OpenSuse**: sudo zipper install python-numpy
- **Fedora/CentOS**: sudo yum install python-numpy

[5] Numpy official website – www.numpy.org

[6] For a complete tutorial of setting a complete Mac OS X python development scenario we recommend the following tutorial:
https://jakevdp.github.io/blog/2013/02/02/setting-up-a-mac-for-python-development/

MacOs

There are two ways of installing numpy:

- Installing via pip package manager, as specified at: https://pypi.python.org/pypi/pip

- Via the macports, running:
  - sudo port install py-numpy[7]

Our recommendation: Virtualenv

You can install virtualenv via the repositories:

1. Open a terminal
   - ◊ **Linux**: Applications -> System -> Terminal.
   - ◊ **MacOs**: Applications -> Utilities -> Terminal.
2. Type in the following command, according to your OS/distribution
   - ◊ **Ubuntu**: sudo apt-get install virtualenv
   - ◊ **OpenSuse**: sudo zipper install virtualenv
   - ◊ **Fedora/CentOS**: sudo yum install virtualenv
   - ◊ **MacOs**: sudo port install py-virtualenv
3. Once installed, create a virtualenv named *myenv* in one folder you remember. (for example, home), typing the following command:
   - virtualenv *myenv*
4. Activate your created environment:
   - myenv/bin/activate
5. From now on, until you close **this** terminal or execute $deactivate, you'll be working in your virtual environment. The next, and last, step is to install numpy. You just have to type:
   - pip install python-numpy

Important note:

[7] To download MacPorts: https://www.macports.org/

The installation of python-numpy may fail if you do not have python-devel installed. If it is the case, install it via the following command according to your OS/distribution:

- **Ubuntu**: sudo apt-get install python-devel
- **OpenSuse**: sudo zipper install python-devel
- **Fedora/CentOS**: sudo yum install python-devel
- **MacOs**: sudo port install py