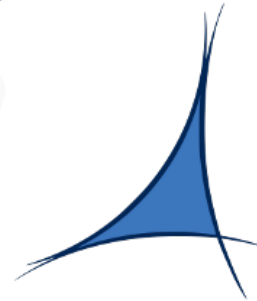




Barcelona Institute of  
Science and Technology

ICE



PIC  
port d'informació  
científica



python<sup>TM</sup>  
The Language

# A bit of history

- ▼ Python was conceived in 1980 and started being implemented in 1989 by Guido van Rossum at CWI (Netherlands), entitled BDFL (Benevolent Dictator For Life) by the community
- ▼ Focus on code readability, providing a syntax that allows more concepts in fewer lines of code
- ▼ Fully open-source with a extremely active and wide community

Python



# A bit of history – Versions

The first main community backed version:



- ▼ Python 2.0, October 16, 2000
  - ▼ Cycle-detecting garbage collector for memory management
  - ▼ Unicode support

# A bit of history – Versions

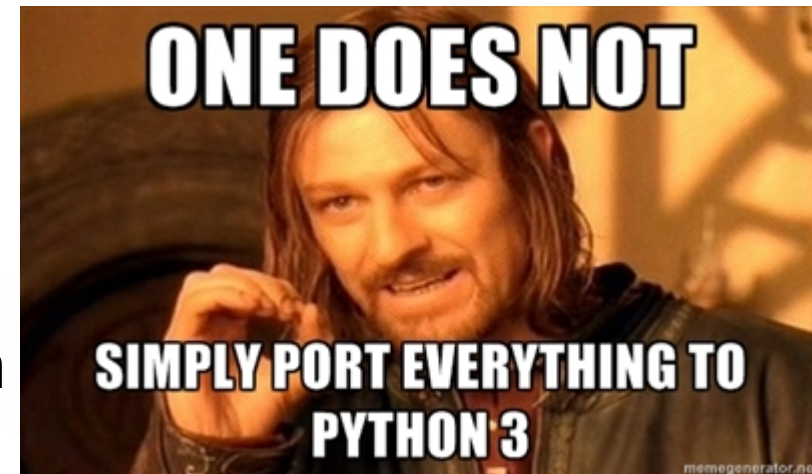
The first main community backed version:

- ▼ Python 2.0, October 16, 2000
  - ▼ Cycle-detecting garbage collector for memory management
  - ▼ Unicode support
- ▼ Python 3.0, December 3, 2008
  - ▼ Backwards-**incompatible**
  - ▼ Major features backported to python 2.6 & 2.7

python



(In 2005)



# A bit of history – Use of python

Python is currently being used for “**everything**”



Large-scale servers  
working 24/7



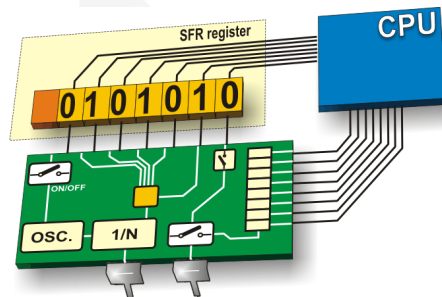
Astronomy



Teaching Kids



Throw-away  
scripts



Microcontrollers



Databases

TODO:  
MORE?  
ANY IDEAS?



# A bit of history – Use of python

(Some) Companies currently massively using python

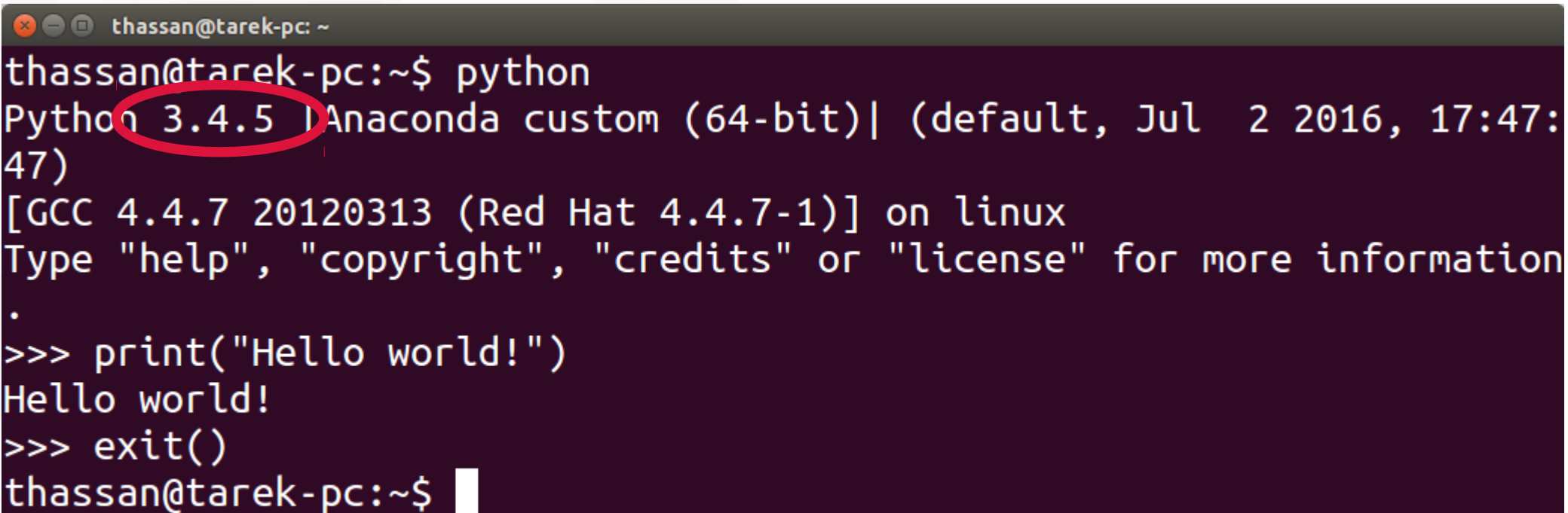


# A bit of history – Python 3

- ▼ Python 3 was designed to correct some fundamental design flaws (see [https://en.wikipedia.org/wiki/History\\_of\\_Python#Features](https://en.wikipedia.org/wiki/History_of_Python#Features))
  - ▼ Not possible to retain full backwards compatibility
- ▼ What does this mean?
  - ▼ If you start now with python 3, you will learn the “good way”
  - ▼ Some packages may still be not fully functional in python 3 (give them some time!)
  - ▼ You can check the version of python you are using simply by:  
`import sys; print(sys.version)`

# Getting started with python

- ▼ Open a terminal and type “python”

A terminal window with a dark purple background and white text. The window title bar shows 'thassan@tarek-pc: ~'. The terminal output shows the command 'python' being executed, followed by the Python version '3.4.5' which is circled in red. The output continues with 'Anaconda custom (64-bit)| (default, Jul 2 2016, 17:47:47)', compiler information '[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux', and instructions to type 'help', 'copyright', 'credits', or 'license' for more information. The user then enters '>>> print("Hello world!")' and the output 'Hello world!' is displayed. Finally, the user enters '>>> exit()' and the prompt returns to 'thassan@tarek-pc:~\$'.

```
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ python  
Python 3.4.5 |Anaconda custom (64-bit)| (default, Jul 2 2016, 17:47:47)  
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux  
Type "help", "copyright", "credits" or "license" for more information  
.br/>>>> print("Hello world!")  
Hello world!  
>>> exit()  
thassan@tarek-pc:~$
```



# Getting started with python

- ▼ To use an external library, you need to “import” it:

```
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ python  
Python 3.4.5 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:47:47)  
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import math  
>>> print (math.pi)  
3.141592653589793  
>>> 
```

- ▼ Python standard library: <https://docs.python.org/3/library/index.html>

# Getting started with python

- ▼ You have more interactive options, like “ipython”:

```
IPython: home/thassan
thassan@tarek-pc:~$ ipython
Python 3.4.5 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:47:47)
Type "copyright", "credits" or "license" for more information.

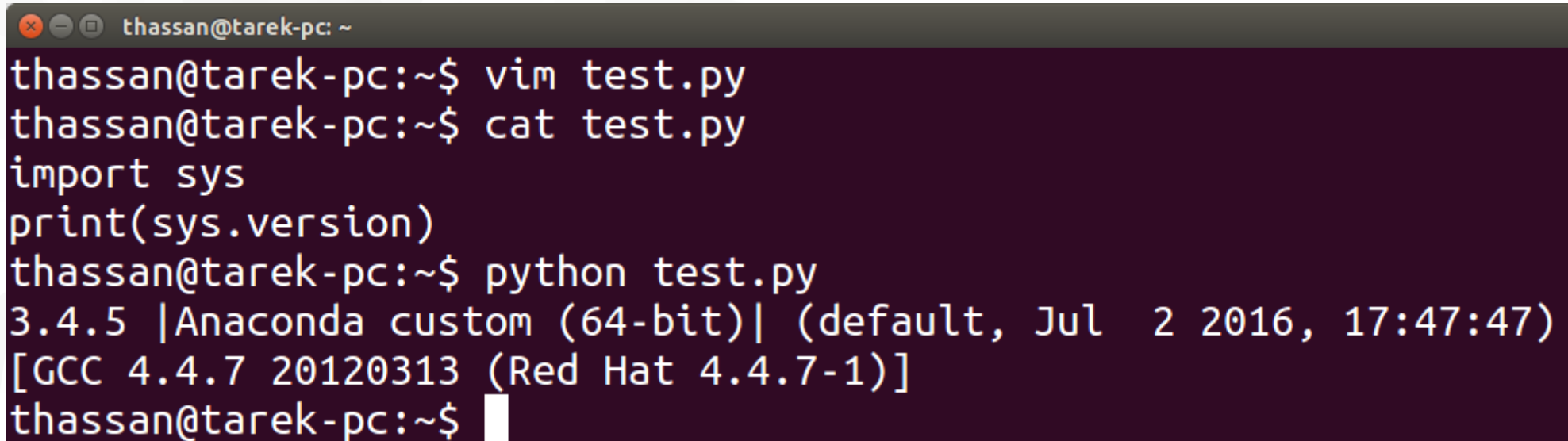
IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: import math

In [2]: math.p
      math.pi
      math.pow
```

# Getting started with python

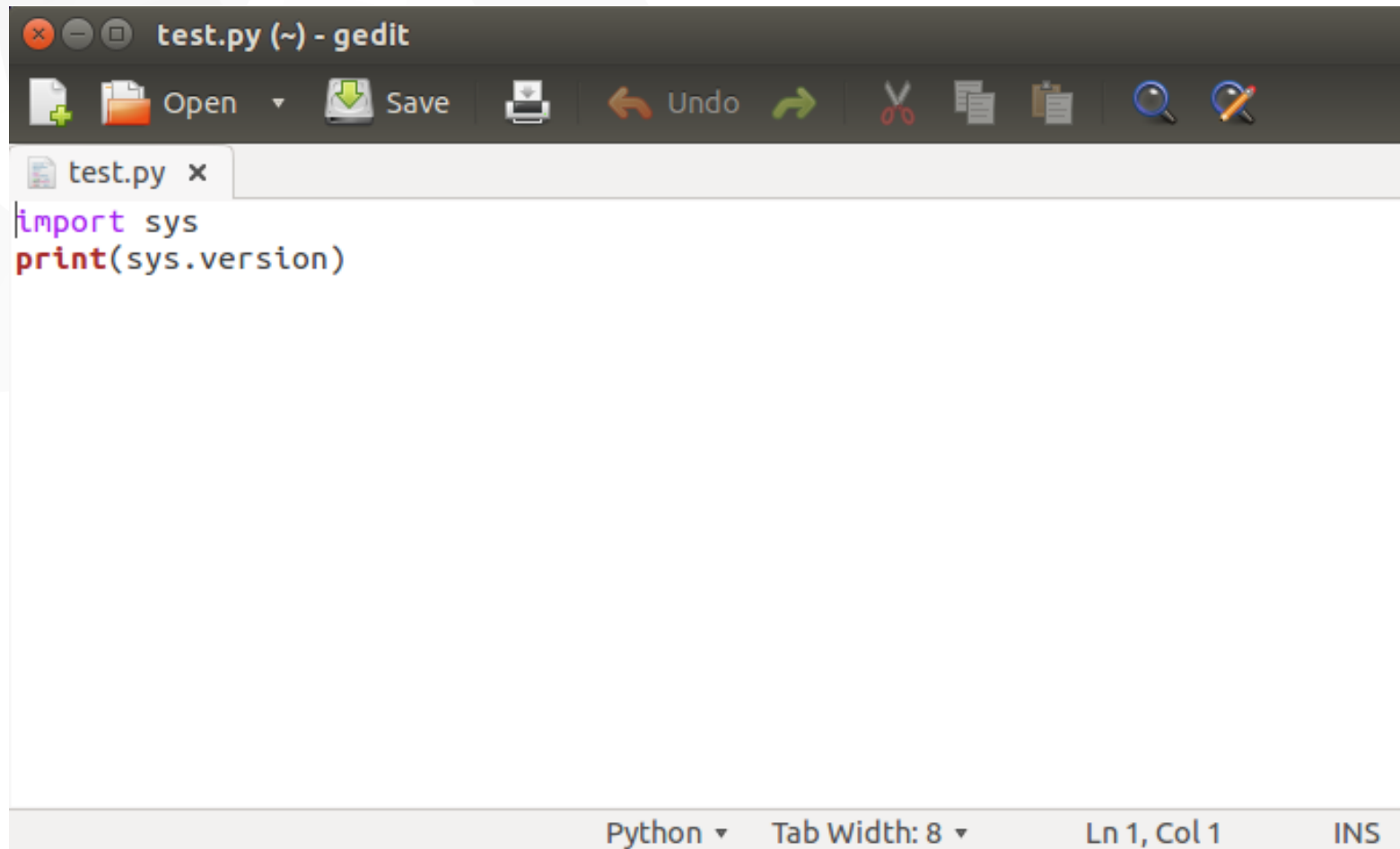
- ▼ For writing scripts, you can use any plain text editor:



```
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ vim test.py  
thassan@tarek-pc:~$ cat test.py  
import sys  
print(sys.version)  
thassan@tarek-pc:~$ python test.py  
3.4.5 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:47:47)  
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]  
thassan@tarek-pc:~$
```

# Getting started with python

- ▼ For writing scripts, you can use any “plain” text editor:



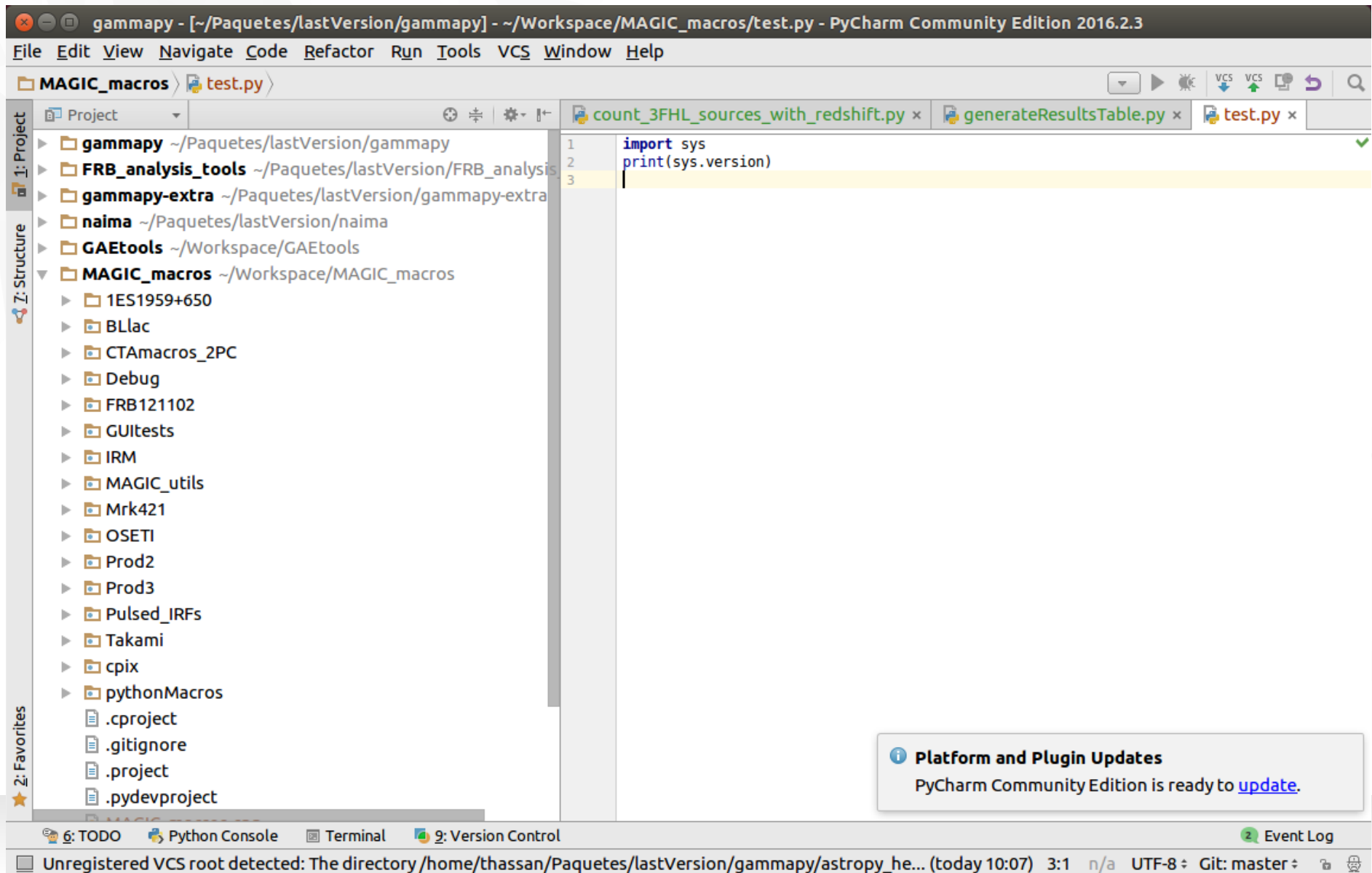
The image shows a screenshot of a text editor window titled "test.py (~) - gedit". The window has a dark-themed toolbar with icons for file operations (Open, Save, Print), editing (Undo, Cut, Copy, Paste), and search. Below the toolbar is a tab labeled "test.py x". The main editing area contains the following Python code:

```
import sys
print(sys.version)
```

At the bottom of the window, a status bar displays "Python", "Tab Width: 8", "Ln 1, Col 1", and "INS".

# Getting started with python

- ▼ Or use an IDE: (e. g. pyCharm, eclipse, vim, wing, spyder...)



# How python works: binaries

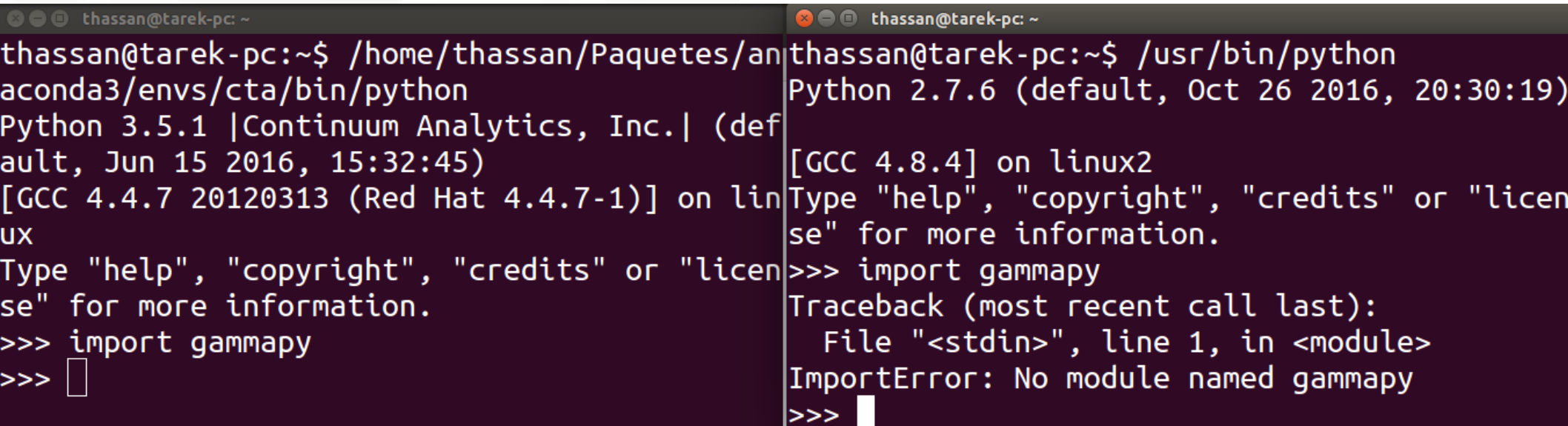
- ▼ As it is possible (and usual) to have several “pythons” installed, some clarifications are useful
  - ▼ When you enter the python console, you execute a python binary (you may have several installed!)

```
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ which python  
/home/thassan/Paquetes/anaconda3/bin/python  
thassan@tarek-pc:~$ python  
Python 3.4.5 |Anaconda custom (64-bit)| (default, Jul  2 2016, 17:47:47)  
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> exit()  
thassan@tarek-pc:~$ /usr/bin/python  
Python 2.7.6 (default, Oct 26 2016, 20:30:19)  
[GCC 4.8.4] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```



# How python works: binaries & libraries

- Some libraries may be installed only in some python environments:



```
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ /home/thassan/Paquetes/anaconda3/envs/cta/bin/python  
Python 3.5.1 |Continuum Analytics, Inc.| (default, Jun 15 2016, 15:32:45)  
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> import gammapy  
>>>   
thassan@tarek-pc: ~  
thassan@tarek-pc:~$ /usr/bin/python  
Python 2.7.6 (default, Oct 26 2016, 20:30:19)  
[GCC 4.8.4] on linux2  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> import gammapy  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
ImportError: No module named gammapy  
>>> 
```

# How python works: binaries & libraries

- ▼ To install a library:
  - ▼ You may do it manually (usually not encouraged):
    - ▼ Download package, “compile” and install

# How python works: binaries & libraries

- ▼ To install a library:
  - ▼ You may do it manually (usually not encouraged):
    - ▼ Download package, “compile” and install
  - ▼ You may use a package manager such as “pip”:
    - ▼ “pip install <package\_name>”
    - ▼ This will install all dependencies, in the version required by the package
    - ▼ Unless if you are a developer of an specific package, you will probably always use this method (or analog ones)