ESOpy v3.0

ESO - Vitacura

15-16-17 April 2019

R. Thomas,
I. Munoz, E. Sedagathi, A. Razza, F. Vogt

First Day (e.g. What is Python Day)

1- Introduction (Romain)

2-3-4 Basics of Python programming (Ivan)

5-Reading/writing ascii files in Python (Romain)

6-Functions in Python (Elyar)

7-Introduction to the Numpy array (Romain)

Let’s get started: What is Python?

Python is an interpreted, object-oriented, high-level programming language

No Need to compile

The notion of ‘object’ is central in Python (See Elyar’s talk tomorrow)

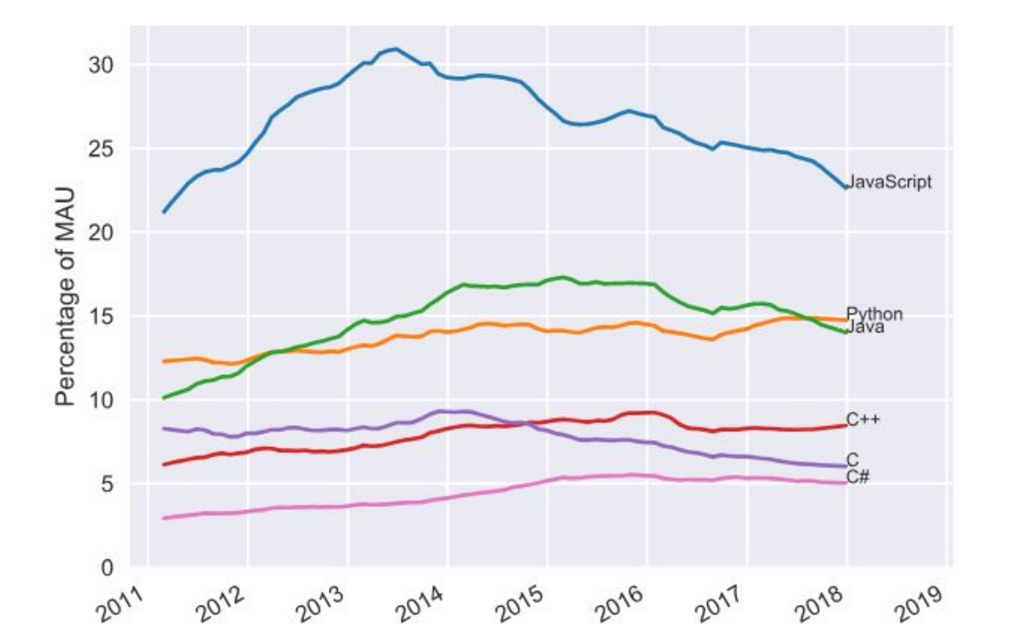
User-Friendly → Close to human language (english)

Python was invented in late 80’s/Early 90’s by Guido Van Rossum in the Netherlands and it is names after the Monthy Python.

- First Version v0.9.0 released in 02/1991
- v1.0 released in 01/1994.
- v2.0 released in 10/2000 → v2.7 is still used by a lot of people
- v3.0 released in 12/2008 → We use v3.7 it in the workshop
- v4.0 released in ?????????

Python is distributed under the General Public Licence (GPL)

How Popular is it?



Ranking Programming Languages by GitHub Users

(MAU = Monthly Active User)

Python is one of the most popular language in the world

→ Two important consequences

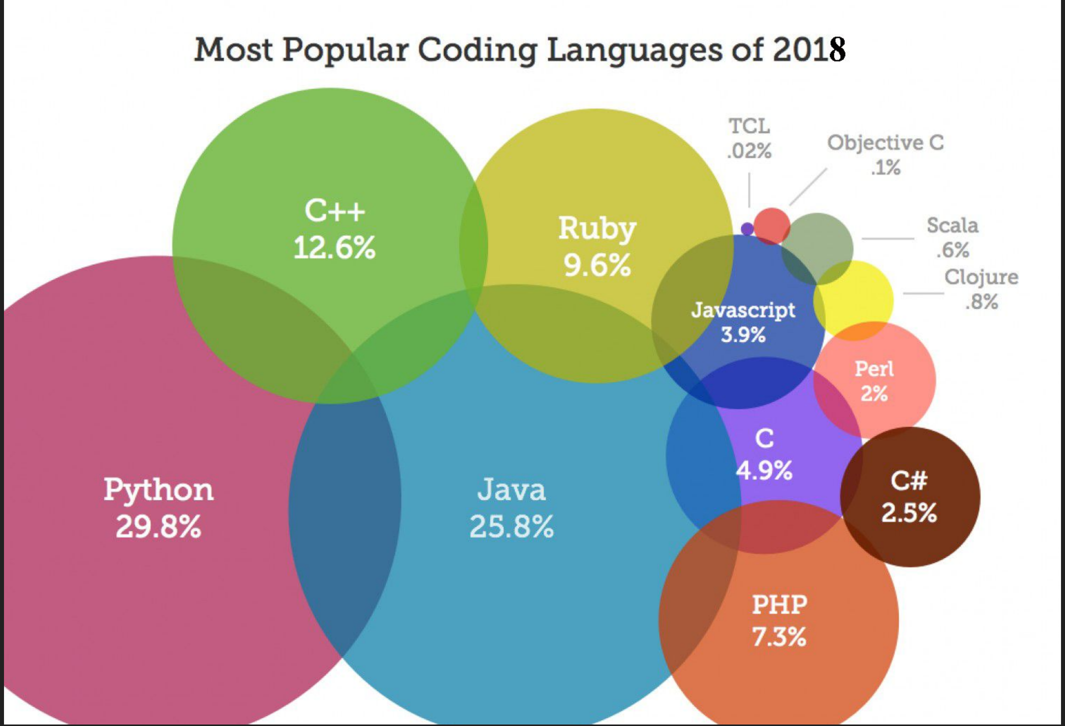
- Huge community

→ You can find help everywhere

- Countless piece of codes/libraries

→ Somebody already did what you want to do

How Popular is it?



Python is one of the most popular language in the world

→ Two important consequences

- Huge community

→ You can find help everywhere

- Countless piece of codes/libraries

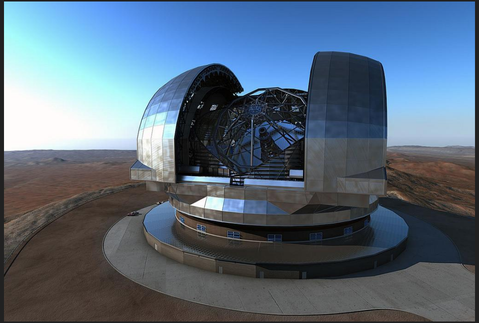
→ Somebody already did what you want to do

Where is Python used?

Where is Python used? → Science

Major Programming language in Astronomy!

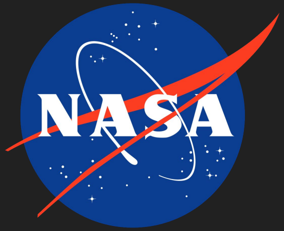
Huge community of scientist writing codes / libraries so you can do your job more efficiently!



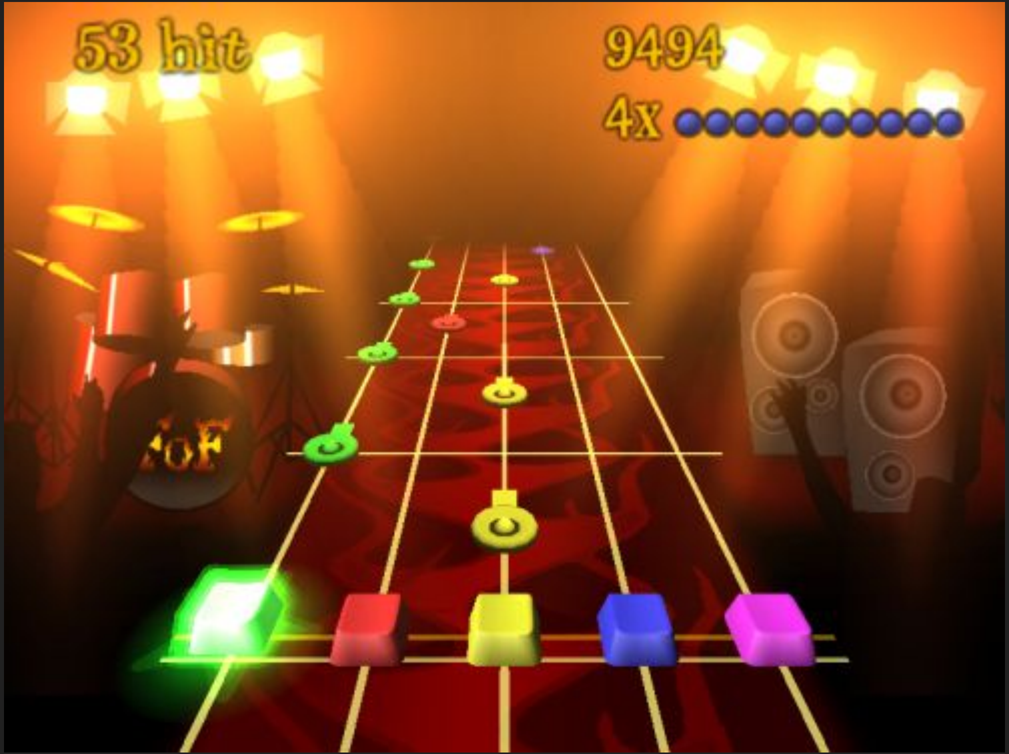
Where is Python used?

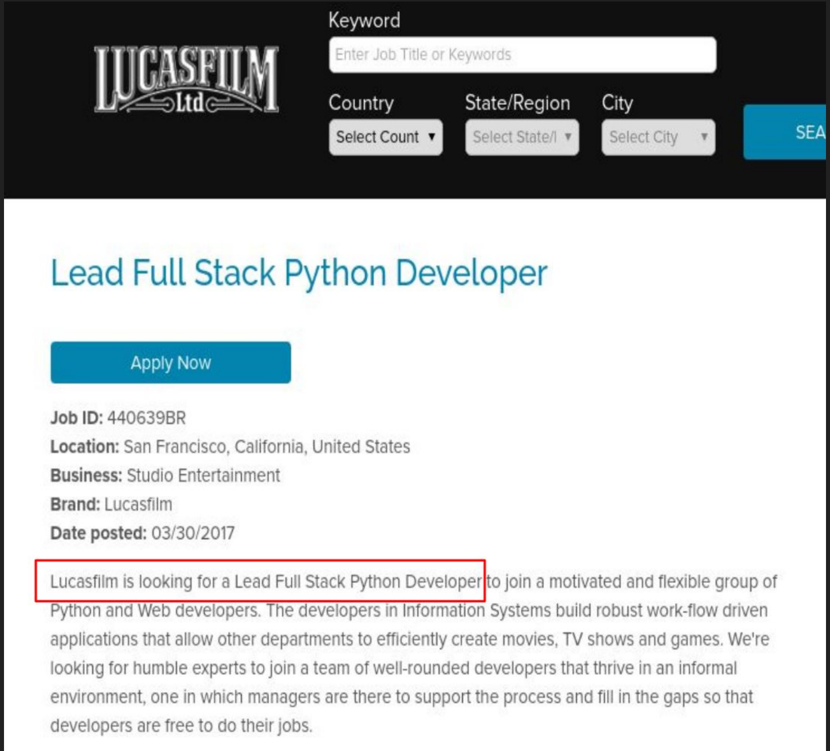
Build website!

Python + Django



Where is Python used? → Games and movies!





Lead Full Stack Python Developer

Job ID: 440639BR
Location: San Francisco, California, United States
Business: Studio Entertainment
Brand: Lucasfilm
Date posted: 03/30/2017

Lucasfilm is looking for a Lead Full Stack Python Developer to join a motivated and flexible group of Python and Web developers. The developers in Information Systems build robust work-flow driven applications that allow other departments to efficiently create movies, TV shows and games. We're looking for humble experts to join a team of well-rounded developers that thrive in an informal environment, one in which managers are there to support the process and fill in the gaps so that developers are free to do their jobs.

Learning Python…OK..But which one??

Python 2.X

Released in 2000

+A lot of material are still in python 2

-No more bug fixes or security update

-Support for a lot of libraries will stop soon

End of Life: 2020

Python 3.X

Release in 2008 with some compatibility problem

+Most of libraries are Py3.X compatible

+New version of libraries are always going to Py3

+This is the current state of the language

We used py3.7 for the camp!

Learning Python...OK...But How long it takes?

FOREVER!*

As any language you learn, it is long and requires practice! And you never stop learning…

The best Way: Take a project and do it in python!

That's the point of this workshop!
→ We will make you code, code, code, code

*Sorry :)

Learning Python...OK...But How long it takes?

FOREVER!*

As any language you learn, it is long and requires practice! And you never stop learning…

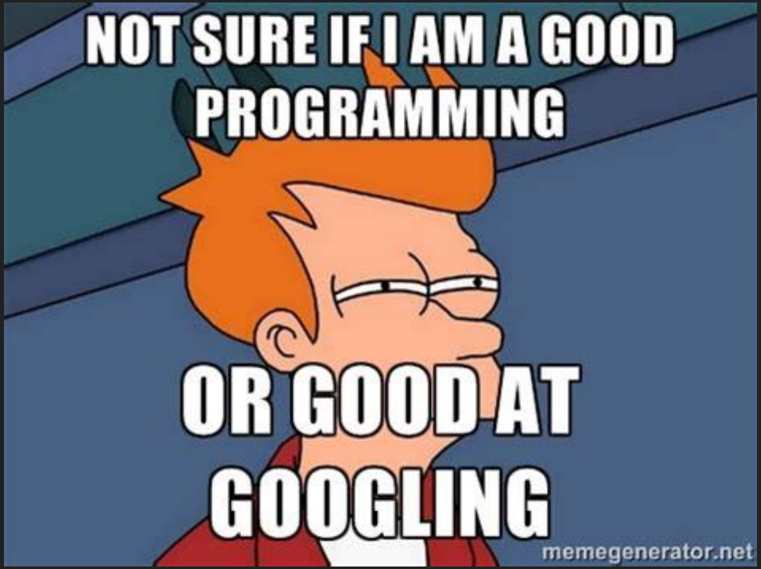
The best Way: Take a project and do it in python!

That's the point of this workshop!

→ We will make you code, code, code, code

You will discover yourself a new best friend:

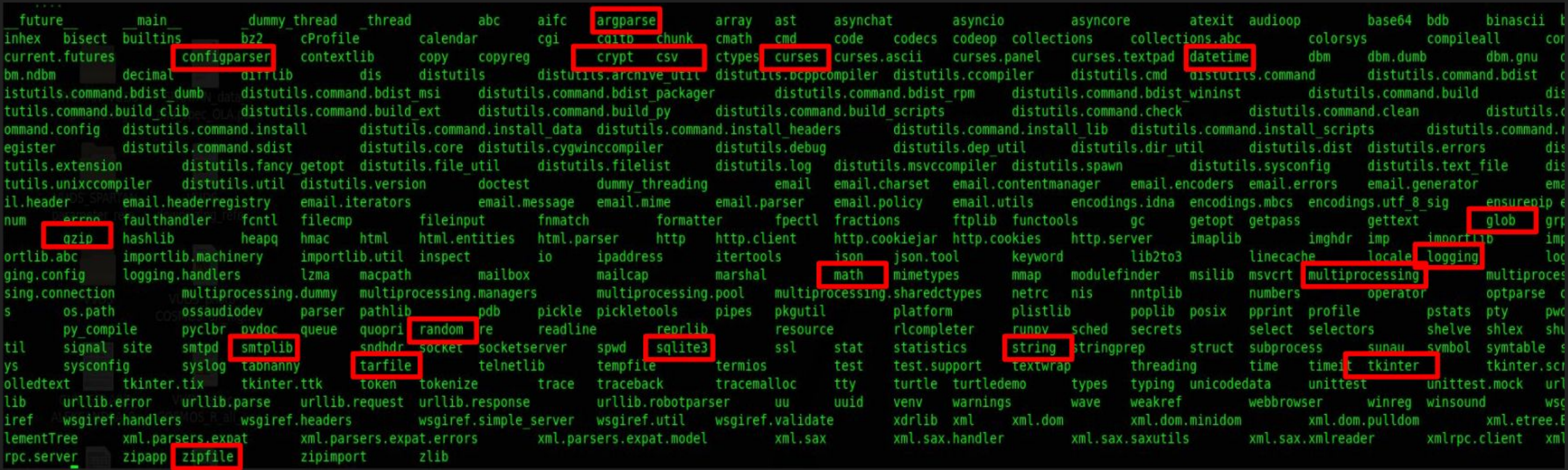
stackoverflow



*Sorry :)

The Python Standard Library:

Natively available Python modules
→They come with your installation



```
__future__      __main__        _dummy_thread   _thread         abc     aifc    argparse        array   ast     asynchat        asyncio         asyncore        atexit  audioop         base64  bdb     binascii b
inhex   bisect  builtins        bz2     cProfile        calendar        cgi     cgitb   chunk   cmath   cmd     code    codecs  codeop  collections     collections.abc         colorsys        compileall      co
current.futures         configparser    contextlib      copy    copyreg         crypt   csv     ctypes  curses  curses.ascii    curses.panel    curses.textpad  datetime        dbm     dbm.dumb        dbm.gnu  d
bm.ndbm         decimal         difflib         dis     distutils       distutils.archive_util  distutils.bcppcompiler  distutils.ccompiler     distutils.cmd   distutils.command       distutils.command.bdist  d
istutils.command.bdist_dumb     distutils.command.bdist_msi     distutils.command.bdist_packager        distutils.command.bdist_rpm     distutils.command.bdist_wininst         distutils.command.build         dis
tutils.command.build_clib       distutils.command.build_ext     distutils.command.build_py      distutils.command.build_scripts         distutils.command.check         distutils.command.clean         distutils.c
ommand.config   distutils.command.install       distutils.command.install_data  distutils.command.install_headers       distutils.command.install_lib   distutils.command.install_scripts       distutils.command.r
egister         distutils.command.sdist         distutils.core  distutils.cygwinccompiler       distutils.debug         distutils.dep_util      distutils.dir_util      distutils.dist  distutils.errors        dis
tutils.extension        distutils.fancy_getopt  distutils.file_util     distutils.filelist      distutils.log   distutils.msvccompiler  distutils.spawn         distutils.sysconfig     distutils.text_file     dis
tutils.unixccompiler    distutils.util  distutils.version       doctest         dummy_threading         email   email.charset   email.contentmanager    email.encoders  email.errors    email.generator         ema
il.header       email.headerregistry    email.iterators         email.message   email.mime      email.parser    email.policy    email.utils     encodings.idna  encodings.mbcs  encodings.utf_8_sig     ensurepip e
num     errno   faulthandler    fcntl   filecmp         fileinput       fnmatch         formatter       fpectl  fractions       ftplib  functools       gc      getopt  getpass         gettext         glob    grp
        gzip    hashlib         heapq   hmac    html    html.entities   html.parser     http    http.client     http.cookiejar  http.cookies    http.server     imaplib         imghdr  imp     importlib       imp
ortlib.abc      importlib.machinery     importlib.util  inspect         io      ipaddress       itertools       json    json.tool       keyword         lib2to3         linecache       locale  logging         log
ging.config     logging.handlers        lzma    macpath         mailbox         mailcap         marshal         math    mimetypes       mmap    modulefinder    msilib  msvcrt  multiprocessing         multiproces
sing.connection         multiprocessing.dummy   multiprocessing.managers        multiprocessing.pool    multiprocessing.sharedctypes    netrc   nis     nntplib         numbers         operator        optparse o
s       os.path         ossaudiodev     parser  pathlib         pdb     pickle  pickletools     pipes   pkgutil         platform        plistlib        poplib  posix   pprint  profile         pstats  pty     pwd
        py_compile      pyclbr  pydoc   queue   quopri  random  re      readline        reprlib         resource        rlcompleter     runpy   sched   secrets         select  selectors       shelve  shlex   shu
til     signal  site    smtpd   smtplib         sndhdr  socket  socketserver    spwd    sqlite3         ssl     stat    statistics      string  stringprep      struct  subprocess      sunau   symbol  symtable s
ys      sysconfig       syslog  tabnanny        tarfile         telnetlib       tempfile        termios         test    test.support    textwrap        threading       time    timeit  tkinter         tkinter.scr
olledtext       tkinter.tix     tkinter.ttk     token   tokenize        trace   traceback       tracemalloc     tty     turtle  turtledemo      types   typing  unicodedata     unittest        unittest.mock   url
lib     urllib.error    urllib.parse    urllib.request  urllib.response         urllib.robotparser      uu      uuid    venv    warnings        wave    weakref         webbrowser      winreg  winsound        wsg
iref    wsgiref.handlers        wsgiref.headers         wsgiref.simple_server   wsgiref.util    wsgiref.validate        xdrlib  xml     xml.dom         xml.dom.minidom         xml.dom.pulldom         xml.etree.E
lementTree      xml.parsers.expat       xml.parsers.expat.errors        xml.parsers.expat.model         xml.sax         xml.sax.handler         xml.sax.saxutils        xml.sax.xmlreader       xmlrpc.client   xml
rpc.server      zipapp  zipfile         zipimport       zlib
```

Adding extra packages:

From the Python Package Index [PyPi]:

Contains more than 100,000 packages

→ Four useful commands:

- pip search + modul

will display a list of package with the word you are giving

- pip install + modul [+ ‘--user’]

will install the package in your python distribution

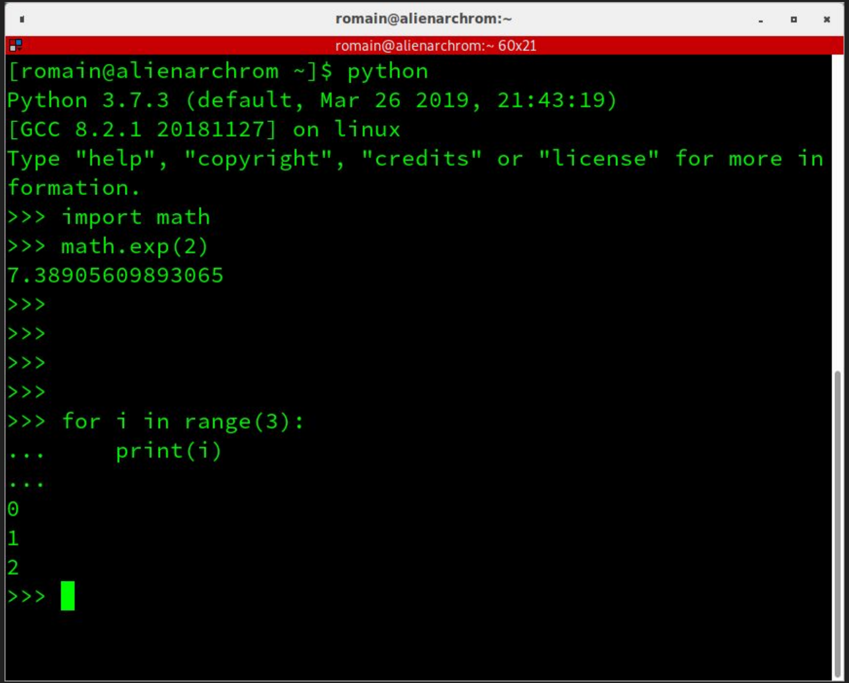
- pip uninstall + modul

will remove the package from your distribution

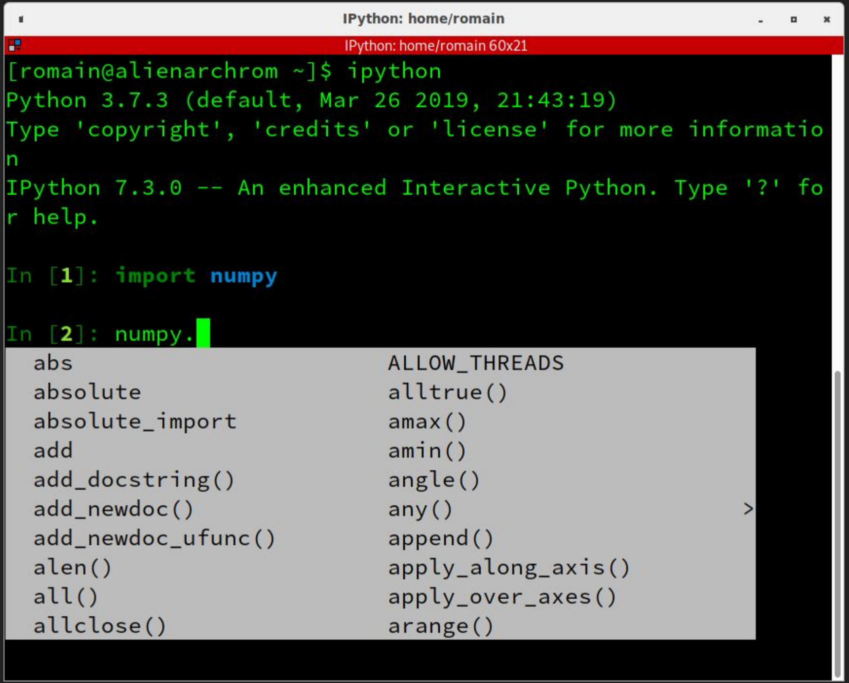
- pip freeze

will list all the pip-installed packages in your distribution with the version number

The Python Interpreter:



```
[romain@alienarchrom ~]$ python
Python 3.7.3 (default, Mar 26 2019, 21:43:19)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more in
formation.
>>> import math
>>> math.exp(2)
7.38905609893065
>>>
>>>
>>>
>>>
>>> for i in range(3):
...     print(i)
...
0
1
2
>>>
```



```
[romain@alienarchrom ~]$ ipython
Python 3.7.3 (default, Mar 26 2019, 21:43:19)
Type 'copyright', 'credits' or 'license' for more informatio
n
IPython 7.3.0 -- An enhanced Interactive Python. Type '?' fo
r help.

In [1]: import numpy

In [2]: numpy.
  abs                                ALLOW_THREADS
  absolute                           alltrue()
  absolute_import                    amax()
  add                                amin()
  add_docstring()                    angle()
  add_newdoc()                       any()                      >
  add_newdoc_ufunc()                 append()
  alen()                             apply_along_axis()
  all()                              apply_over_axes()
  allclose()                         arange()
```

Where do I write python code?

You have multiple choices!

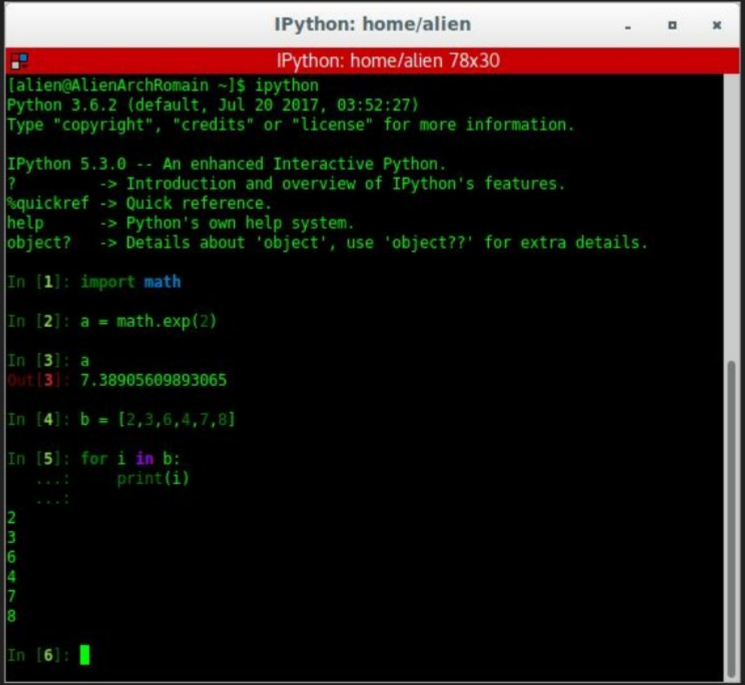
In the python interpreter
(with python or ipython)

-Easy to use (you write your line of code and press enter)

- Nice environment to test

- You get the help of the functions directly accessible

-Does not save your work in an external file



```
[alien@AlienArchRomain ~]$ ipython
Python 3.6.2 (default, Jul 20 2017, 03:52:27)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- 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]: a = math.exp(2)

In [3]: a
Out[3]: 7.38905609893065

In [4]: b = [2,3,6,4,7,8]

In [5]: for i in b:
   ...:     print(i)
   ...:
2
3
6
4
7
8

In [6]: 
```

Where do I write python code?

In an external file, with your favorite text editor or python IDE
→ You will create a *.py file

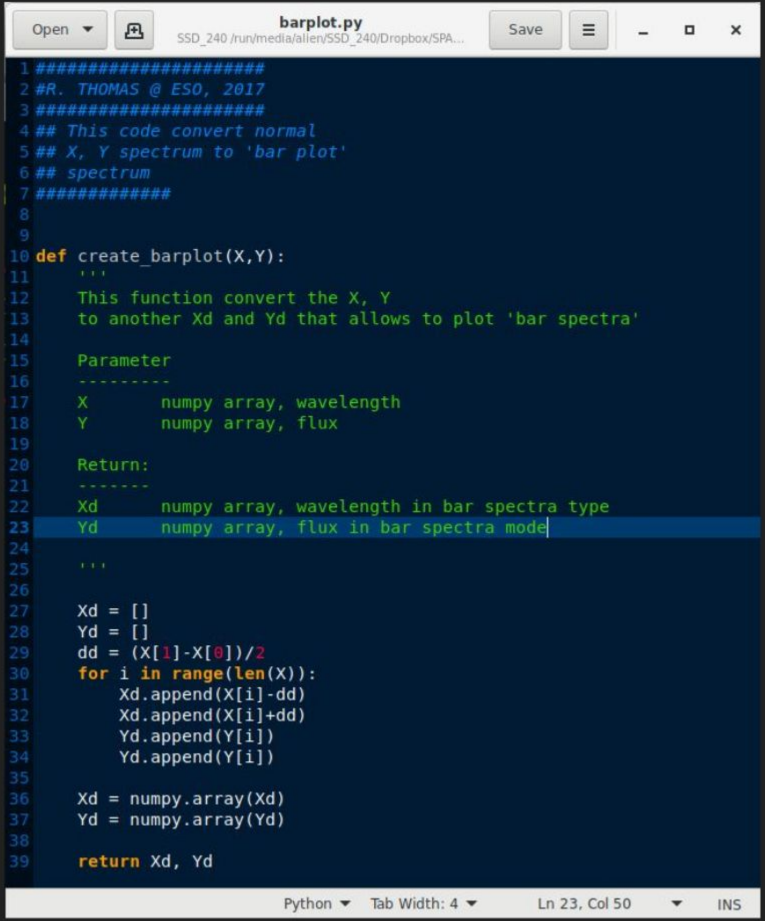
-You keep your code! (so you can share it!)

-You can create reusable modules

-Makes you organize your code and make it clearer

- Require a more organised work and thinking

You have multiple choices!



```python
######################
#R. THOMAS @ ESO, 2017
######################
## This code convert normal
## X, Y spectrum to 'bar plot'
## spectrum
#############


def create_barplot(X,Y):
    '''
    This function convert the X, Y
    to another Xd and Yd that allows to plot 'bar spectra'

    Parameter
    ---------
    X       numpy array, wavelength
    Y       numpy array, flux

    Return:
    -------
    Xd      numpy array, wavelength in bar spectra type
    Yd      numpy array, flux in bar spectra mode

    '''

    Xd = []
    Yd = []
    dd = (X[1]-X[0])/2
    for i in range(len(X)):
        Xd.append(X[i]-dd)
        Xd.append(X[i]+dd)
        Yd.append(Y[i])
        Yd.append(Y[i])

    Xd = numpy.array(Xd)
    Yd = numpy.array(Yd)

    return Xd, Yd
```

Where do I write python code?

You have multiple choices!

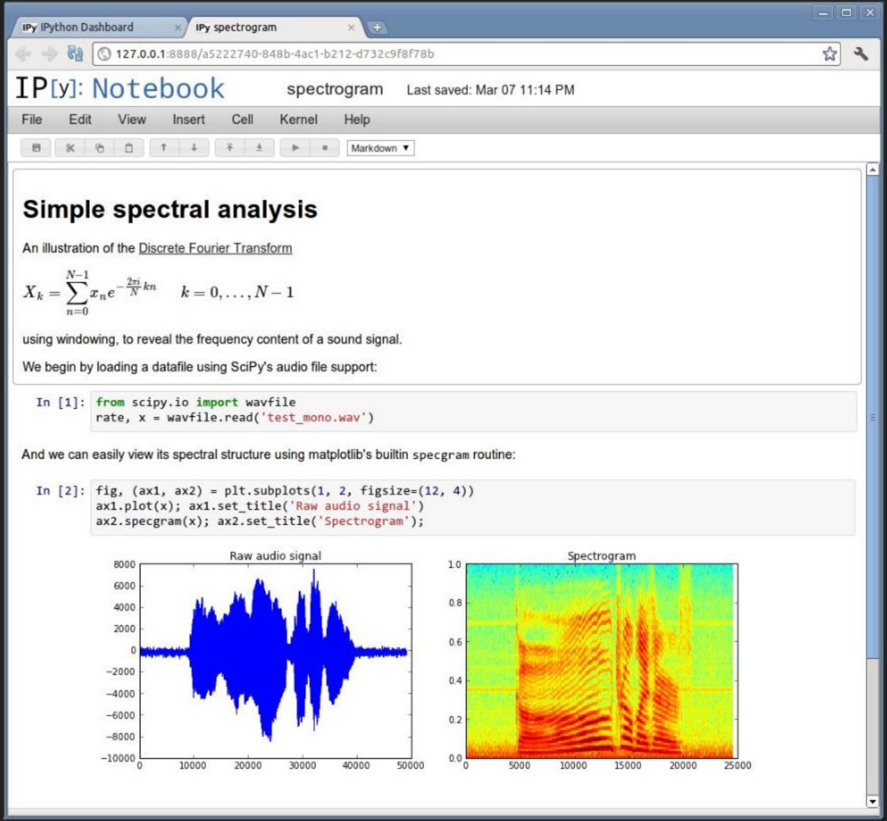
In a python notebook
→ you will create a *.ipynb file

-Web interface

-It is an *advanced* ipython

-You keep your code! (so you can share it!)

-Especially suitable for exercices and demonstration



Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \dots, N-1$$

using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Writing and *commenting* and *documenting* code:

Python comes with rules about how to write a good code. They are described in the PEP8 and PEP257 documents (*Python Enhancement Proposal*). If you ignore them your code will work (don't worry) but following them will make your code more readable.

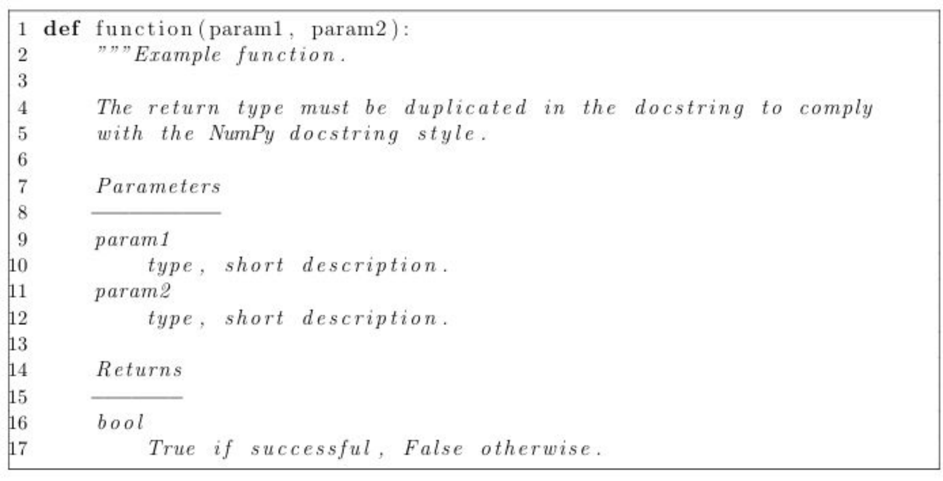
- Python is indentation sensitive
  → one indentation = 4 spaces

- Imports
  → one import per line, no wildcard imports

- Avoid too long lines
  → (max 80-100 characters / line)

- ***A code is read much more often than it is written*** - Guido van Rossum
  → Comment your code!!!!!!

- doctsrings: A format to document fct/classes/modules. Ex: numpydoc



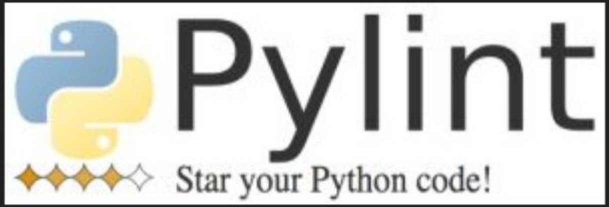
```
def function(param1, param2):
    """Example function.

    The return type must be duplicated in the docstring to comply
    with the NumPy docstring style.

    Parameters
    ----------
    param1
        type, short description.
    param2
        type, short description.

    Returns
    -------
    bool
        True if successful, False otherwise.
```

Useful tools and resources



supports a number of features, from coding standards to error detection, and it also helps with refactoring (by detecting duplicated code).

If you want to look at advanced codes you can go there →

Spyder & PyCharm are widely used development environment for python

Something you do not understand?
Something you do not know how to do?

I will not propose books or websites here with python lectures.
You can just type 'python lectures' , 'python for beginners' in google and you will find thousands of solutions (do not forget youtube, there are plenty of videos with python classes)

Open a python shell and type : import this

<u>Enjoy the three days of coding!</u>

And Biiiiig Thanks to the Speakers:

Ivan, Elyar, Alessandro, Frederic