

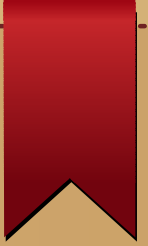
# Python



- Python is a widely used general-purpose, high-level programming language.
- Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in **fewer lines of code** than would be possible in languages such as C.
- The language provides constructs intended to enable **clear programs** on both a small and large scale.



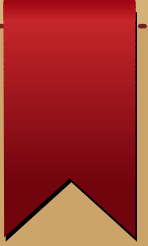
# Python



- Python supports multiple programming paradigms, including **object-oriented**, imperative (assignments, loops, tests) and functional programming or procedural styles.
- It features a dynamic type system and automatic **memory management** and has a large and comprehensive **standard library**.



# Python



- Like other dynamic languages, Python is often used as a **scripting language**, but is also used in a wide range of non-scripting contexts.
- Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into standalone executable programs.
- Python interpreters are available for **many operating systems**.



# Python very short history



Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC language.

Guido is now working for Dropbox, after a few years for Google.

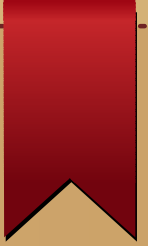
# Python very short history



- Python 2.0 was released on 16 October 2000, with many major new features including a full garbage collector and support for Unicode.
- With this release the development process was changed and became more transparent and community-backed.



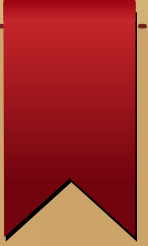
# Python 3



- Python 3.0 (also called Python 3000 or py3k), a major, backwards-incompatible release, was released on 3 December 2008 after a long period of testing.
- All the useful libraries are not all available under 3.x, so better still use 2.x, which becomes the standard for scientific use.
- But notice that today numpy and matplotlib have been updated to 3.x.



# Features and philosophy



Python is a multi-paradigm programming language: **object-oriented** programming and **structured** programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming (included by meta-programming and by magic methods).




# Compiled and interpreted languages

- C and Fortran programs are compiled before executed (fast, closer to the machine).
  - **Python is interpreted**, making it flexible: no declarations, on-the-fly changes (but a bytecode is produced (.pyc), making things a little faster.)
- Slower. **But C and Fortran program can be called** to speed up some part of a code (and are actually called by e.g. numpy and scipy functions).



# Compared to IDL



- Python seems to become since ~10 years the successor of IDL in the scientific mostly used programmatic language (STSCI).
  - It's also **I**nteractive (ipython), managing **D**ata and a high level **L**anguage.
  - **It's open source and gratis.**
  - It offers today more than IDL (more libraries), the community is very active and almost any question has already been asked and answered on developers forums (e. g. <http://stackoverflow.com>).
  - Same could be said comparing python with Matlab.
  - A huge number of manuals, how-to, tutorials on Internet.
- 

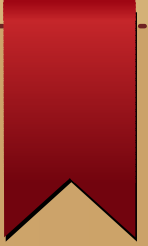
# Installation

- Most of the operating systems come with some python already available
- But to have easy access to numpy + matplotlib + scipy + others, it's better to install from one of the all-inclusive distributions
  - Entought Python Distribution (Canopy), with Academic licence
  - Anaconda
  - Python(x,y)
  - **Ureka, from STSCI (including IRAF, which may conflict with already installed IRAF, but easy to solve)** <http://ssb.stsci.edu/ureka/>

# ipython

- The interactive session of python. Python is already interactive, but this layer add functionality's that makes python so easy to use.
- Completion is available, recall lasts entries from previous sessions, some “magic” functions are defined: save sessions, timer, cat, basic linux commands (ls, cd)... just type “magic()” in ipython.
- This is what you want to use as your every day python interface.

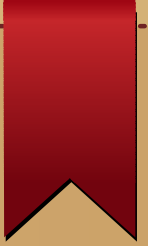
# Notebooks



- Facility of ipython.
- Very useful for sharing small scripts and for teaching.
- .ipynb extension.
- Plots inside the file.
- This is what I will use in this lecture.
- Started with *ipython notebook [--pylab] [inline]*
- A notebook can be view using the site <http://nbviewer.ipython.org/>



# Links



- <http://python-astro.blogspot.mx/>
- <https://docs.python.org/2.7/>
- <https://github.com/jrjohansson/scientific-python-lectures>
- <http://www.star.st-and.ac.uk/~pw31/CompAstro2015/IntroToPython.pdf>
- [https://www.kevinsheppard.com/images/0/09/Python\\_introduction.pdf](https://www.kevinsheppard.com/images/0/09/Python_introduction.pdf)

