

# Python

*batteries included*

## Présentation générale

- Guido van Rossum – 1990
- Versions ...1.4, 1.5, 1.6, 2.0, 2.1, 2.2, 2.3, 2.4...
- Python Software Foundation
- Certification OSI
- Copyright – license très libre, compatible GPL
- Language de script – pseudo-code + modules natifs
- Ecrit en C
- Multi-plateformes

# Pourquoi utiliser Python ?

- Haut niveau
- Syntaxe simple et claire
- Bibliothèque standard très riche (modules)
- Orienté Objet, typage dynamique
- Types évolués (listes, dict...)
- Comptage de références
- Extensible (C, C++, Fortran...)
- Langage d'extension (Gimp, Blender)
- Vivant – développement rapide mais cohérent
- Rapidité de développement
- Facilité de maintenance

## Domaine d'application

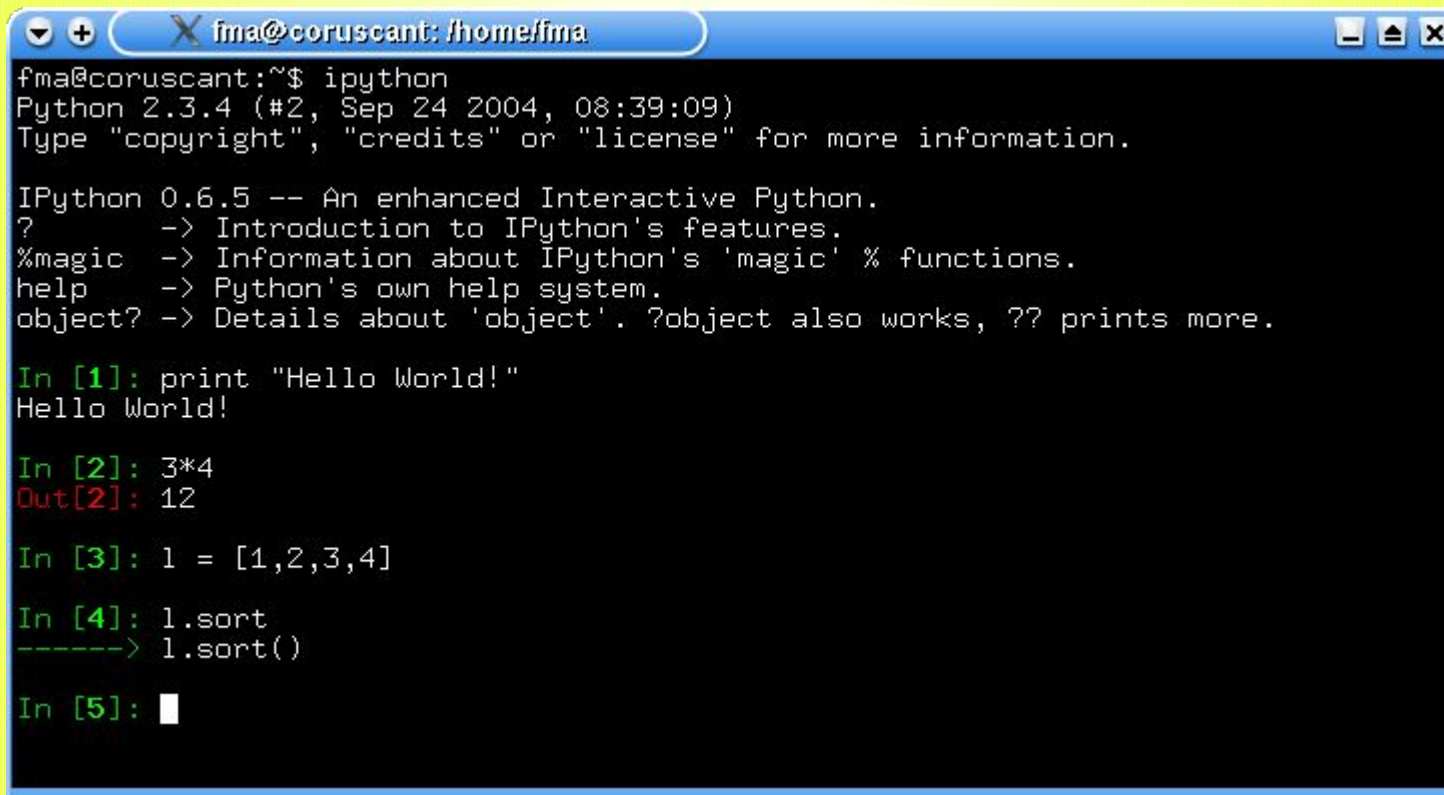
- Apprentissage de la programmation
- Calcul scientifique et imagerie
- Prototypage
- Administration système
- Applications *web*
- Multimédia
- Embarqué
- Jeux

## Applications en python

- Zope (serveur applicatif web)
- SpamBayes (anti-spam bayésien)
- Mailman (robot de listes)
- Sketch (dessin vectoriel)
- Admin RedHat
- Moteur de recherche Google
- Imagerie (PIL)
- Plugins :
  - Gimp
  - Blender
  - Scribus

## Programmation python

- Shell standard
- Shell évolué (ipython)



```
fma@coruscant: /home/fma
fma@coruscant:~$ ipython
Python 2.3.4 (#2, Sep 24 2004, 08:39:09)
Type "copyright", "credits" or "license" for more information.

IPython 0.6.5 -- An enhanced Interactive Python.
?      -> Introduction to IPython's features.
%magic -> Information about IPython's 'magic' % functions.
help   -> Python's own help system.
object? -> Details about 'object'. ?object also works, ?? prints more.

In [1]: print "Hello World!"
Hello World!

In [2]: 3*4
Out[2]: 12

In [3]: l = [1,2,3,4]

In [4]: l.sort
-----> l.sort()

In [5]: █
```

## Syntaxe

- Pas de caractères inutiles (; { })
- Casse significative
- Choix du jeu de caractères *ISO...*
- Identificateurs : `toto`, `_variable`, `__special__`
- Commentaires : `# iter of the file`
- Blocs définis par indentation
  - ```
if x != 0:
    print "x n'est pas nul"
    print 1/x
```

## Opérations

- Affectation

- `a = 3`
- `b = "Hello world!"`

- Arithmétique

- `+` `-` `*` `/` `%` `**`

- Comparaison

- `<` `>` `<=` `>=` `==` `!=` `<>` `!`

- Logique

- `or` `and` `not`

- Bits

- `<<` `>>` `|` `&`



## Types de base

- Numériques

- int, long
- float, Decimal
- bool
- complex

```
a = 10
b = 1./5. (0.20000000000000001)
c = True
d = (1-4j)
```

- Conteneurs

- tuple, list
- dict, Set
- buffer
- str

```
m = range(10); n = [1, 2, 3]
d = {'a':1, 'b':2}

s = "Salut les copains"
t = 'Hello'
u = """Autre"""
```

- Itérateurs

```
xrange(4)
```

## Structures de contrôle

- Conditions

- `if`
- `elif`
- `else`

- Boucles

- `for`
- `while`
- `break`
- `continue`

- *Compréhension de liste*

- `aList = [sin(x/100*pi) for x in xrange(100)]`
- `anotherList = [v for v in aList if v<0.5]`

## Fonctions

- Récursivité
- Passage d'arguments par référence
  - `def func(a, myList):`
- Paramètres optionnels; nombre variable
  - `def func(a="10", b=None, c=3):`
  - `def func(a, *args, **kwargs):`
  - `func(2, "a", b, toto=6, titi={'t':(1,2)})`
  - `func("in", *toto, **titi)`
- *Lambda*
  - `(lambda x: x**2)`
- Documentation intégrée
  - `def func(a):`
    - `""" This is a docstring.`
    - `Here we can explain how to use this function`
    - `"""`

## Exceptions

- Syntaxe
  - `try:`
  - `except:`
  - `else:`
  - `finally:`
- Affichage du *traceback*
- Manipulation du *traceback*
- Lever ses propres exceptions
  - `raise`

## Classes

- Héritage multiple et répété
- Les classes sont des objets
- Tout attribut est publique
- Surcharge d'opérateurs
- Intégration d'itérateur
- Méta-classes

## Modules

- Mécanisme d'*import*
- Installation facile (distutils)
- Indispensables intégrés (*batteries included*)
  - os, sys
  - re
  - db
  - xml
  - tkinter
  - threading
  - httplib, ftplib, email
- Autres
  - pyopengl
  - numarray, Numeric, Scientific, MLab
  - pygame
  - pyro, OmniORB

## Modules

`__builtin__`  
`__future__`  
`__main__`  
`aifc`  
`anydbm`  
`array`  
`asynchat`  
`asyncore`  
`atexit`  
`audioop`  
`base64`  
`BaseHTTPServer`  
`Bastion`  
`binascii`  
`binhex`  
`bisect`  
`bsddb (Unix, Windows)`  
`bz2`  
`calendar`  
`cgi`  
`CGIHTTPServer`  
`cgilib`  
`chunk`  
`cmath`  
`cmd`  
`code`  
`codecs`  
`codeop`  
`collections`  
`colorsys`  
`commands (Unix)`  
`compileall`  
`compiler`  
`compiler.ast`  
`compiler.visitor`  
`ConfigParser`  
`Cookie`  
`cookielib`  
`copy`  
`copy_reg`  
`cPickle`  
`crypt (Unix)`  
`cStringIO`  
`csv`

**curses**  
`curses.ascii`  
`curses.panel`  
`curses.textpad`  
`curses.wrapper`  
`datetime`  
`dbhash (Unix, Windows)`  
`dbm (Unix)`  
`decimal`  
`difflib`  
`dircache`  
`dis`  
`distutils`  
`distutils.archive_util`  
`distutils.bcppcompiler`  
`distutils.ccompiler`  
`distutils.cmd`  
`distutils.command`  
`distutils.command.bdist`  
`distutils.command.bdist_dumb`  
`distutils.command.bdist_packager`  
`distutils.command.bdist_rpm`  
`distutils.command.bdist_wininst`  
`distutils.command.build`  
`distutils.command.build_clib`  
`distutils.command.build_ext`  
`distutils.command.build_py`  
`distutils.command.build_scripts`  
`distutils.command.clean`  
`distutils.command.config`  
`distutils.command.install`  
`distutils.command.install_data`  
`distutils.command.install_headers`  
`distutils.command.install_lib`  
`distutils.command.install_scripts`  
`distutils.command.register`  
`distutils.command.sdist`  
`distutils.core`  
`distutils.cygwincompiler`  
`distutils.debug`  
`distutils.dep_util`  
`distutils.dir_util`  
`distutils.dist`  
`distutils.emxcompiler`

`distutils.errors`  
`distutils.extension`  
`distutils.fancy_getopt`  
`distutils.file_util`  
`distutils.filelist`  
`distutils.log`  
`distutils.msvccompiler`  
`distutils.mwerkscompiler`  
`distutils.spawn`  
`distutils.sysconfig`  
`distutils.text_file`  
`distutils.unixcompiler`  
`distutils.util`  
`distutils.version`  
`dl (Unix)`  
`doctest`  
`DocXMLRPCServer`  
`dumbdbm`  
`dummy_thread`  
`dummy_threading`  
`EasyDialogs (Mac)`  
**email**  
`email.Charset`  
`email.Encoders`  
`email.Errors`  
`email.Generator`  
`email.Header`  
`email.Iterators`  
`email.Message`  
`email.Parser`  
`email.Utils`  
`encodings.idna`  
`errno`  
`exceptions`  
`fcntl (Unix)`  
`filecmp`  
`fileinput`  
`fnmatch`  
`formatter`  
`fpectl (Unix)`  
`fpformat`  
**ftplib**  
`gc`  
`gdbm (Unix)`

`getopt`  
`getpass`  
`gettext`  
`glob`  
`gopherlib`  
`grp (Unix)`  
`gzip`  
`heapq`  
`hmac`  
`hotshot`  
`hotshot.stats`  
`htmlentitydefs`  
`htmlib`  
`HTML.Parser`  
`httplib`  
`imageop`  
`imaphib`  
`imgchr`  
`imp`  
`inspect`  
`itertools`  
`keyword`  
`linecache`  
`locale`  
`logging`  
`mailbox`  
`mailcap`  
`marshal`  
`math`  
`md5`  
`mhlib`  
`mimetools`  
`mimetypes`  
`MimeWriter`  
`mimify`  
`mmap`  
`multifile`  
`mutex`  
`netrc`  
`new`  
`nis (UNIX)`  
`nntplib`  
`operator`  
`optparse`

`os.path`  
`ossaudiodev (Linux, FreeBSD)`  
`parser`  
`pdb`  
`pickle`  
`pickletools`  
`pipes (Unix)`  
`pkutil`  
`platform`  
`popen2 (Unix, Windows)`  
`poplib`  
`posix (Unix)`  
`posixfile (Unix)`  
`pprint`  
`profile`  
`pstats`  
`pty (IRIX, Linux)`  
`pwd (Unix)`  
`py_compile`  
`pyclbr`  
`pydoc`  
`Queue`  
`quopri`  
`random`  
`re`  
`readline (Unix)`  
`repr`  
`resource (Unix)`  
`rexec`  
`rfc822`  
`rgbimg`  
`rlcompleter (Unix)`  
`robotparser`  
`sched`  
`ScrolledText (Tk)`  
`select`  
`sets`  
`sgmllib`  
`sha`  
`shelve`  
`shlex`  
`shutil`

`signal`  
`SimpleHTTPServer`  
`SimpleXMLRPCServer`  
`site`  
`smtplib`  
`smtplib`  
`sndhdr`  
`socket`  
`SocketServer`  
`stat`  
`statcache`  
`statvfs`  
`string`  
`StringIO`  
`stringprep`  
`struct`  
`subprocess`  
`sunau`  
`symbol`  
`sys`  
`syslog (Unix)`  
`tabnanny`  
`tarfile`  
`telnetlib`  
`tempfile`  
`termios (Unix)`  
`test`  
`test.test_support`  
`textwrap`  
`thread`  
`threading`  
`time`  
`timeit`  
`Tix`  
`Tkinter`  
`token`  
`tokenize`  
`traceback`  
`tty (Unix)`  
`turtle (Tk)`  
`types`  
`unicodedata`  
`unittest`  
`urllib`

`urllib2`  
`urlparse`  
`user`  
`UserDict`  
`UserList`  
`UserString`  
`uu`  
`W (Mac)`  
`warnings`  
`wave`  
`weakref`  
`webbrowser`  
`whichdb`  
`whrandom`  
`xdrlib`  
`xml.dom`  
`xml.dom.minidom`  
`xml.dom.pulldom`  
`xml.parsers.expat`  
`xml.sax`  
`xml.sax.handler`  
`xml.sax.saxutils`  
`xml.sax.xmlreader`  
`xmllib`  
`xmlrpclib`  
`zipfile`  
`zlib`

# Outils de développement

- Nombreux IDE
  - eric
  - boa
  - leo
  - *wingIDE*
- Pychecker, Pyreverse
- *Design Pattern*
- ipython



## Liens utiles

- Site officiel
  - <http://www.python.org>
- Liste francophone
  - [python@aful.org](mailto:python@aful.org)
- Wiki francophone
  - <http://wikipython.flibuste.net>
- Cours débutant
  - <http://www.ulg.ac.be/cifen/inforef/swi/python.htm>
- Dive Into Python
  - <http://fr.diveintopython.org>

**Le Python**  
*c'est bon, mangez-en !*