

Programmation Orientée Objet avec pythonTM 3

Module PO-S6

Objectifs : L'objectif de ce module est de donner aux élèves ingénieurs les compétences nécessaires à la compréhension de la manipulation de la programmation objet.

Volume horaire : 60h (CM : 14h, TD : 12h TP : 34h)

Evaluation : Contrôle continu, Contrôle TP, Projet final

Programmation Orientée Objet

40h

Introduction - Prise en main

Les bases du langage Python

Programmation Orientée Objet

20h

Mini-projet


Mes sources :

Formations Python - Jean-Luc CHARLES et Eric DUCASSE



Cours PO S6 - 2022- Gilles GUETTE



Docs Python 

Cours Programmation Python – Laurent POINTAL



Cours Programmation Python – Éric BERTHOMIER



Diverses sources sur le web...

3 / 18



- ▶ Langage de programmation proposé dans les années 90 par [Guido van Rossum](#) qui a choisi le nom Python en hommage à la série *Monty Python*.
- ▶ Un langage **interprété**, puissant, compact, visuel.
- ▶ Multi paradigmes, supportant les principaux styles de programmation : [impératif](#), [procédural](#), [orienté objet](#)...
- ▶ Multi plateformes : GNU/Linux, Mac OS X, Windows...
- ▶ Libre et gratuit : logiciel *Open Source* (www.opensource.org) distribué sous la licence [PSF](#) (*Python Software Foundation*) compatible avec la [GPL](#) (*Gnu Public Licence*).

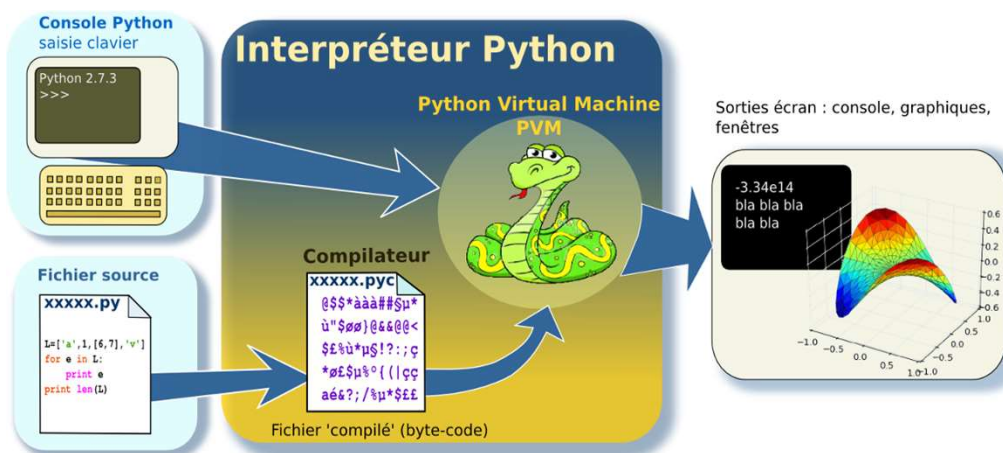
4 / 18



- ▶ Très polyvalent, grâce à de nombreux **modules** couvrant des domaines très variés :
 - numpy** algèbre linéaire, matrices, vecteurs, systèmes linéaires...
 - scipy** probabilité/statistiques, FFT 1D, 2D..., filtrage numérique, images;
 - matplotlib** tracé de courbes (look & feel MatLab 8);
 - os** manipulation des répertoires et des fichiers...
- ▶ Disposant d'une communauté d'utilisateurs très active.
- ▶ Simple à prendre en main; de plus en plus utilisé en recherche, enseignement, industrie...
- ▶ Enseigné au lycée (programme 2009, classe de seconde), au programme des classes prépa (depuis la rentrée 2013).
- ▶ Utilisé par des acteurs majeurs du monde industriel : NASA, Google, CEA, Airbus, YouTube...

5 / 18

Python est un langage **interprété**



Les 2 outils pour travailler avec Python : **Interpréteur** (console Python) et **Éditeur**.

6 / 18

Documentation officielle : site [Python.org](https://docs.python.org) docs.python.org

Python » French » 3.11.1 » 3.11.1 Documentation »

Recherche rapide: Go | modules | index

Téléchargement
Téléchargement de ces documentations

Docs par version
 Python 3.12 (in development)
 Python 3.11 (stable)
 Python 3.10 (stable)
 Python 3.9 (security-fixes)
 Python 3.8 (security-fixes)
 Python 3.7 (security-fixes)
 Python 3.6 (EOL)
 Python 3.5 (EOL)
 Python 2.7 (EOL)
 Toutes les versions

Autres ressources
 Index des PEP
 Guide du débutant
 Liste de livres
 Documents multimédia
 Guide du développeur Python

Documentation Python 3.11.1

Bienvenue sur la documentation officielle de Python 3.11.1.

La documentation :

[Les nouveautés de Python 3.11](#)
ou toutes les nouveautés depuis la 2.0

[Tutoriel](#)
démarrez ici

[Référence de la bibliothèque](#)
gardez ça sous votre oreiller

[Référence du langage](#)
décrit la syntaxe et les éléments du langage

[Installation et utilisation de Python](#)
utilisation de Python sur différentes plateformes

[Les HOWTOs de Python](#)
documents explorant certains sujets en profondeur

[Installation de modules Python](#)
installation depuis le "Python Package Index" ou d'autres sources

[Distribuer des modules Python](#)
publier des modules pour que d'autres puissent les installer

[Amélioration et intégration](#)
tutoriel pour les développeurs C/C++

[Python/C API](#)
référence pour les développeurs C/C++

[FAQs](#)
foire aux questions (avec les réponses !)

7 / 18

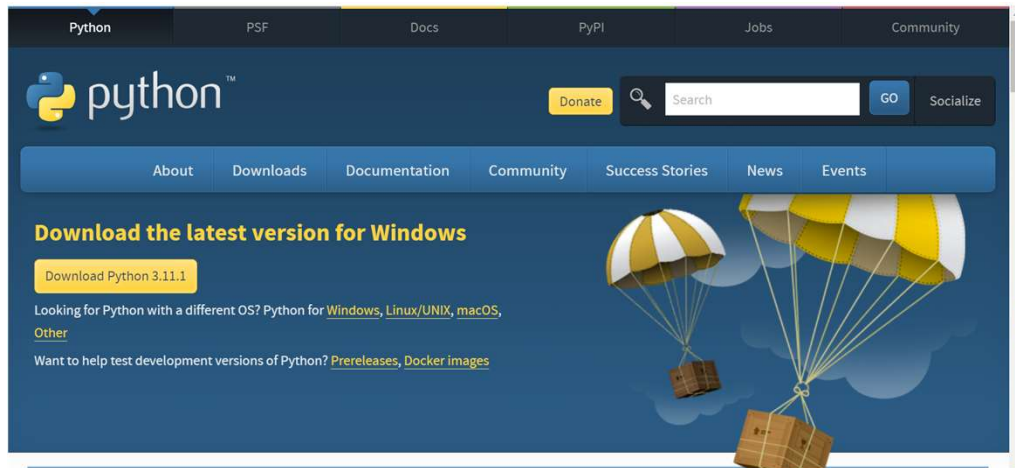
IP[y]: IPython Interactive Computing

- Interpréteur interactif **amélioré**
 - Plus d'interactivité avec les modules de tracés de courbes (**pylab**, **matplotlib**...) et le système d'exploitation (%magic).
 - Cellules question/réponse numérotées... (In[n]/ Out[n]).
 - Facilités pour charger le module pylab : graphiques, tracés de courbes interactifs... et bien plus encore!
 - Propose la déclinaison **notebook** : comme un *notebook Mathematica*, mais avec un "Python dans le moteur"!

8 / 18

Distributions du site [Python.org](https://www.python.org/downloads)

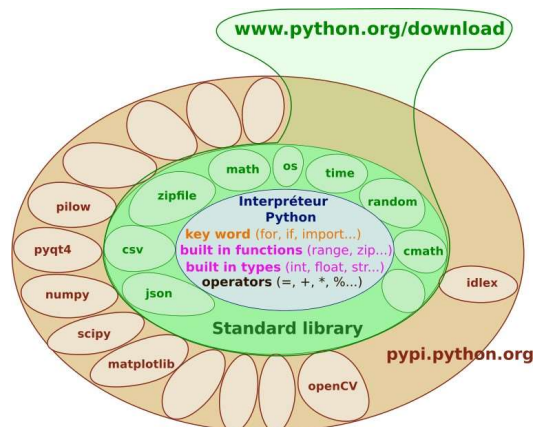
- Ne propose "que" : interpréteur Python + modules de la *standard library* pour les OS GNU/Linux, Mac OS X et Windows.



9 / 18

Distributions du site [Python.org](https://www.python.org/downloads)

- Le travail avec Python (calcul scientifique...) nécessite l'installation de modules supplémentaires !



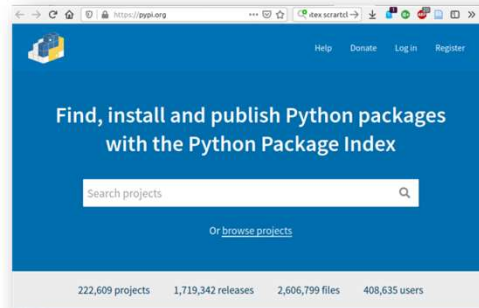
- Windows/Mac OS X : l'installation de paquets supplémentaires peut paraître difficile pour des utilisateurs habitués aux interfaces graphiques.

10 / 18

Ajout de modules

pypi.python.org

- Le site pypi.python.org est le dépôt officiel des modules Python :

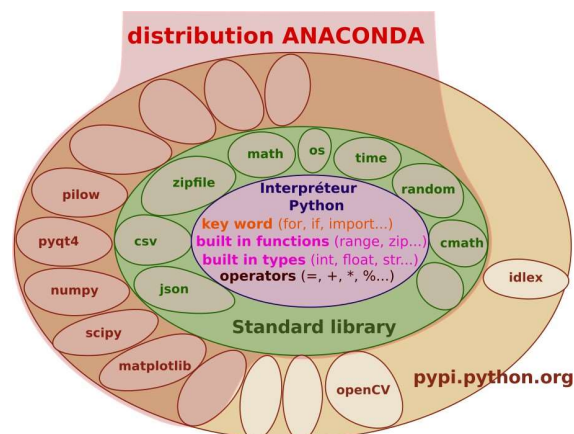


- L'installation d'un module du site pypi.python.org se fait en tapant dans un terminal :
 - `pip install nom_module` ~ installation *utilisateur*
 - `sudo pip install nom_module` ~ installation *administrateur* [Linux / Mac OS]
 - `conda install nom_module` ~ pour un Python installé avec Anaconda.

11 / 18

Distributions Python

- L'utilisation d'une **distribution** ([WinPython](#), [Canopy](#), [Anaconda](#)...) facilite le travail d'installation des modules :



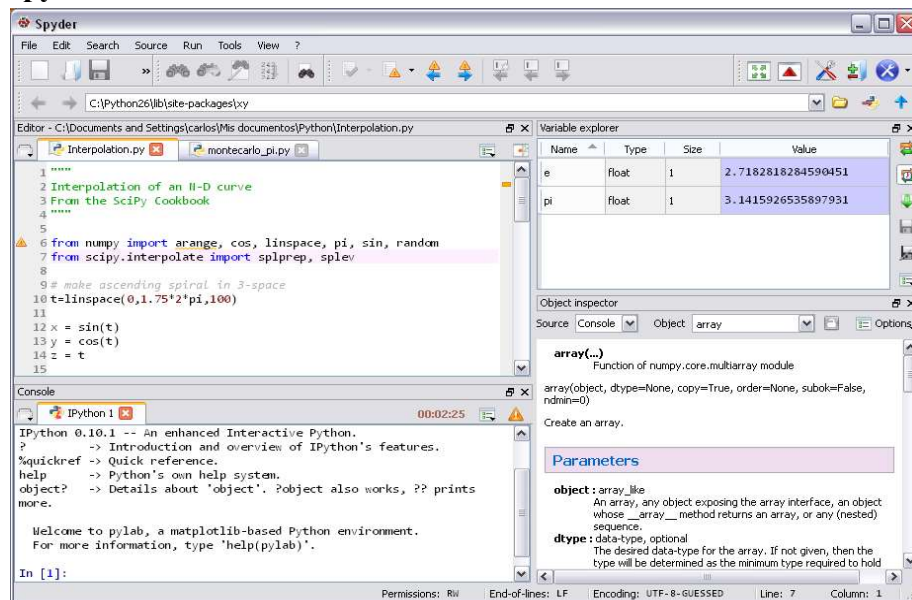
12 / 18

Spyder

- ▶ Un *MatLab-like*, écrit en Python! Très complet, convivial, efficace...
- ▶ Propose beaucoup de *goodies* : object inspector, help, debugger, Qt Designer...
- ▶ Propose un *workspace* analogue à celui de Matlab pour visualiser les données.
- ▶ Propose de l'aide interactive.
- ▶ Fournit les 2 outils fondamentaux : éditeur / interpréteur Python (ou IPython).

13 / 18

Spyder



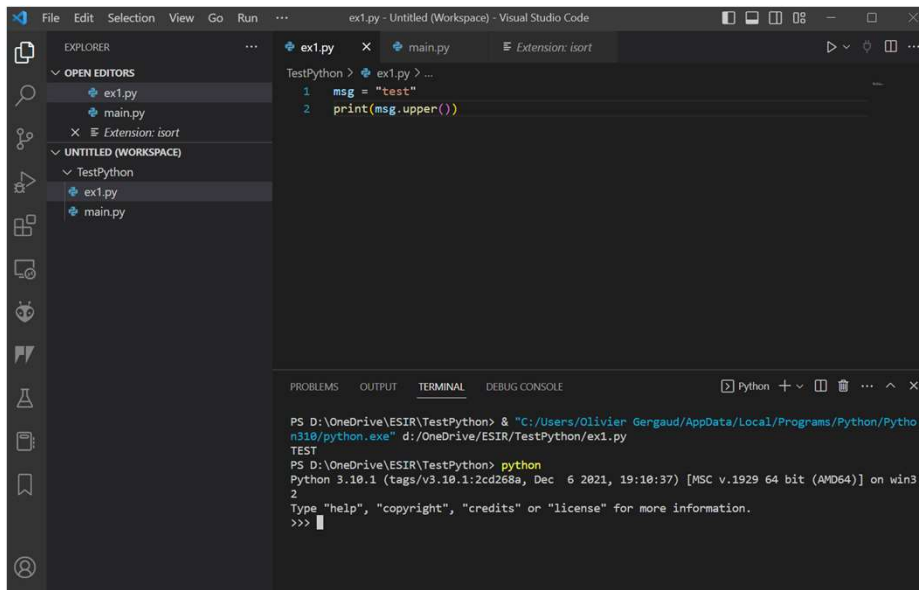
14 / 18

VS-code

- ▶ Un éditeur de code simplifié
- ▶ Prise en charge de plusieurs langage de programmation
- ▶ Nombreuses extensions pour faciliter le développement
- ▶ Commandes Git intégrés
- ▶ Fournit les 2 outils fondamentaux : éditeur / interpréteur Python (ou IPython).

15 / 18

VS-code



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a workspace named 'TestPython' containing 'ex1.py' and 'main.py'. The main editor area shows 'ex1.py' with the following code:

```
1 msg = "test"
2 print(msg.upper())
```

Below the editor, the TERMINAL panel is active, showing the execution of the script:

```
PS D:\OneDrive\ESIR\TestPython> "C:/Users/Olivier Gergaud/AppData/Local/Programs/Python/Python310/python.exe" d:/OneDrive/ESIR/TestPython/ex1.py
TEST
PS D:\OneDrive\ESIR\TestPython> python
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

16 / 18