

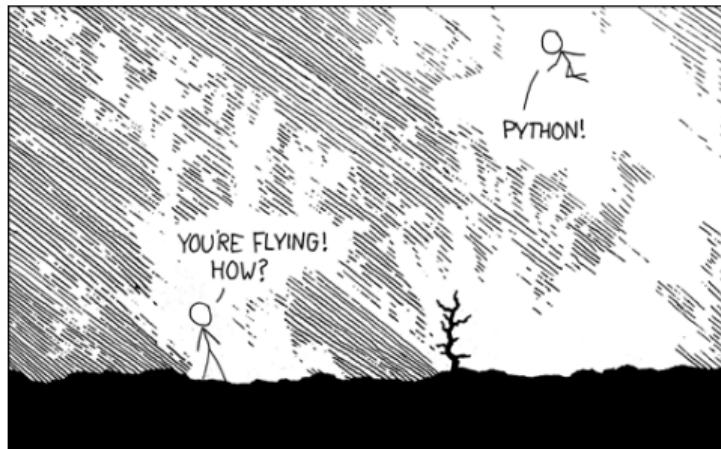
Option « Programmation en Python »

## Présentation & Organisation de l'option

Xavier Garrido  
Laboratoire de l'Accélérateur Linéaire, Orsay

“ Python est un langage de programmation, dont la première version est sortie en **1991**. Créé par **Guido van Rossum**, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au Centrum voor Wiskunde en Informatica aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la **Python Software Foundation** ↗, créée en 2001. Ce langage a été baptisé ainsi en hommage à la troupe de comiques les « **Monty Python** ». ”

# Langage Python : pourquoi ?



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!  
/ HELLO WORLD IS JUST  
print "Hello, world!"

I DUNNO...  
DYNAMIC TYPING?  
WHITESPACE?  
COME JOIN US!  
PROGRAMMING IS FUN AGAIN!  
IT'S A WHOLE NEW WORLD UP HERE!  
/ BUT HOW ARE YOU FLYING?

I JUST TYPED  
import antigravity  
THAT'S IT? /  
/ ... I ALSO SAMPLED  
EVERYTHING IN THE  
MEDICINE CABINET  
FOR COMPARISON.  
/ BUT I THINK THIS  
IS THE PYTHON.

† xkcd comic ↗

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

- ▶ **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

- ▶ Détails techniques :

- ▶ Python est un langage interprété, ce qui signifie qu'il n'a pas besoin d'être compilé avant d'être exécuté. Cela facilite la lecture et l'écriture du code.
- ▶ Python a une syntaxe très simple et intuitive, utilisant des balises de balise et des accolades pour délimiter les blocs de code, ce qui rend le code plus facile à lire et à comprendre.
- ▶ Python a une grande communauté d'utilisateurs et de contributeurs, ce qui signifie qu'il existe de nombreux tutoriels, livres et ressources en ligne pour apprendre et utiliser le langage.
- ▶ Python a une grande variété d'outils et de bibliothèques disponibles, ce qui permet aux développeurs de créer des applications et des scripts pour une grande variété de domaines.
- ▶ Python est un langage de programmation universel, ce qui signifie qu'il peut être utilisé pour développer des applications web, mobiles, desktop, et même des systèmes embarqués.
- ▶ Python a une grande communauté d'utilisateurs et de contributeurs, ce qui signifie qu'il existe de nombreux tutoriels, livres et ressources en ligne pour apprendre et utiliser le langage.

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

## ► **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

## ► **Double extension**

- ▶ Python est un langage interprété, ce qui signifie qu'il n'a pas d'extension native.
- ▶ Cependant, il existe une extension native pour Python, nommée Python C API, qui permet de développer des extensions en C ou C++.
- ▶ Ces extensions peuvent être utilisées pour accéder à des fonctionnalités natives, telles que la gestion de fichiers, la communication réseau, la manipulation de données, etc.
- ▶ Les extensions peuvent également être utilisées pour ajouter de nouvelles fonctionnalités à Python, telles que de nouvelles bibliothèques ou de nouveaux modules.
- ▶ Les extensions peuvent être utilisées pour optimiser le code Python, en évitant les conversions entre Python et C, et en utilisant les meilleures pratiques de programmation C.

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

- ▶ **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

- ▶ Détails techniques :

- ▶ **Type dynamique** : pas besoin de définir le type de variables, les arguments de fonction ou les types de retour.
- ▶ **Gestion automatique de la mémoire** : pas besoin d'allouer et de désallouer explicitement la mémoire pour les variables et les tableaux de données. Pas de bugs de fuite de mémoire.
- ▶ **Langage interprété** : Pas besoin de compiler le code. L'interpréteur Python lit et exécute directement le code python.

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

- ▶ **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

- ▶ Détails techniques :

- ▶ **Type dynamique** : pas besoin de définir le type de variables, les arguments de fonction ou les types de retour.
- ▶ **Gestion automatique de la mémoire** : pas besoin d'allouer et de désallouer explicitement la mémoire pour les variables et les tableaux de données. Pas de bugs de fuite de mémoire.
- ▶ **Langage interprété** : Pas besoin de compiler le code. L'interpréteur Python lit et exécute directement le code python.

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

- ▶ **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

- ▶ Détails techniques :

- ▶ **Type dynamique** : pas besoin de définir le type de variables, les arguments de fonction ou les types de retour.
- ▶ **Gestion automatique de la mémoire** : pas besoin d'allouer et de désallouer explicitement la mémoire pour les variables et les tableaux de données. Pas de bugs de fuite de mémoire.
- ▶ **Langage interprété** : Pas besoin de compiler le code. L'interpréteur Python lit et exécute directement le code python.

# Langage Python : pourquoi ?

Pour tout un tas de raisons que vous découvrirez durant cette option parmi lesquelles

- ▶ **Un langage simple et clair**

- ▶ code facile à lire et intuitif,
- ▶ visuellement épuré,
- ▶ **syntaxe minimalist**e facile à apprendre,
- ▶ moins de lignes de code, moins de bugs, plus facile à entretenir.

- ▶ Détails techniques :

- ▶ **Type dynamique** : pas besoin de définir le type de variables, les arguments de fonction ou les types de retour.
- ▶ **Gestion automatique de la mémoire** : pas besoin d'allouer et de désallouer explicitement la mémoire pour les variables et les tableaux de données. Pas de bugs de fuite de mémoire.
- ▶ **Langage interprété** : Pas besoin de compiler le code. L'interpréteur Python lit et exécute directement le code python.

# Langage Python : pros & cons

- ▶ Avantages:

- ▶ Le principal avantage est la facilité de programmation, minimisant le temps nécessaire pour développer, déboguer et maintenir le code.
- ▶ **Programmation modulaire et orientée objet**, système performant de *packaging* et de réutilisation du code.
- ▶ **Documentation** étroitement intégrée au code.
- ▶ Une vaste bibliothèque standard et **une grande collection de modules complémentaires**, scientifiques et autres (serveur web,...).
- ▶ **multi-plateformes** : Python est facilement accessible et utilisable que ce soit sur Windows, MacOS X ou **Linux**

- ▶ Désavantages:

- ▶ langage de programmation interprété et typé dynamiquement → **exécution du code python plus lente** par rapport aux langages de programmation compilés statiquement, tels que C et Fortran.

# Langage Python : pros & cons

## ► Avantages:

- ▶ Le principal avantage est la facilité de programmation, minimisant le temps nécessaire pour développer, déboguer et maintenir le code.
- ▶ **Programmation modulaire et orientée objet**, système performant de *packaging* et de réutilisation du code.
- ▶ **Documentation** étroitement intégrée au code.
- ▶ Une vaste bibliothèque standard et **une grande collection de modules complémentaires**, scientifiques et autres (serveur web,...).
- ▶ **multi-plateformes** : Python est facilement accessible et utilisable que ce soit sur Windows, MacOS X ou **Linux**

## ► Désavantages:

- ▶ langage de programmation interprété et typé dynamiquement → **exécution du code python plus lente** par rapport aux langages de programmation compilés statiquement, tels que C et Fortran.

# Option « Programmation en Python »

## ► À qui s'adresse cette option ?

- ▶ à tout étudiant de niveau académique suffisant (min. L3), **sans pré-requis en programmation et/ou algorithmie**
- ▶ pas aux *geeks* férus d'informatique ayant déjà 20 ans de pratique du Python

## ► Qu'est ce que cette option n'est pas ?

- ▶ pas un cours de **programmation numérique/algorithmie** → cours de C du L3 Physique Fondamentale
- ▶ pas un cours de **méthodes numériques** → option « Méthodes numériques » du L3 de Physique Fondamentale
- ▶ **une simple initiation** à la programmation Python

### ⚠ Python 2 ou Python 3 ?

En 2008, Python 3 a été officiellement lancé. Quelques (vieilles) librairies scientifiques ne fonctionnent pas encore sous Python 3 mais c'est de plus en plus l'exception. Au cours de cette option, nous utiliserons **Python 3.5**

# Option « Programmation en Python »

- ▶ À qui s'adresse cette option ?
  - ▶ à tout étudiant de niveau académique suffisant (min. L3), **sans pré-requis en programmation et/ou algorithmie**
  - ▶ pas aux geeks férus d'informatique ayant déjà 20 ans de pratique du Python
- ▶ Qu'est ce que cette option n'est pas ?
  - ▶ pas un cours de **programmation numérique/algorithmie** → cours de C du L3 Physique Fondamentale
  - ▶ pas un cours de **méthodes numériques** → option « Méthodes numériques » du L3 de Physique Fondamentale
  - ▶ **une simple initiation** à la programmation Python

## ⚠ Python 2 ou Python 3 ?

En 2008, Python 3 a été officiellement lancé. Quelques (vieilles) librairies scientifiques ne fonctionnent pas encore sous Python 3 mais c'est de plus en plus l'exception. Au cours de cette option, nous utiliserons **Python 3.5**

# Option « Programmation en Python »

- ▶ À qui s'adresse cette option ?
  - ▶ à tout étudiant de niveau académique suffisant (min. L3), **sans pré-requis en programmation et/ou algorithmie**
  - ▶ pas aux geeks férus d'informatique ayant déjà 20 ans de pratique du Python
- ▶ Qu'est ce que cette option n'est pas ?
  - ▶ pas un cours de **programmation numérique/algorithmie** → cours de C du L3 Physique Fondamentale
  - ▶ pas un cours de **méthodes numériques** → option « Méthodes numériques » du L3 de Physique Fondamentale
  - ▶ **une simple initiation** à la programmation Python

## ⚠ Python 2 ou Python 3 ?

En 2008, Python 3 a été officiellement lancé. Quelques (vieilles) librairies scientifiques ne fonctionnent pas encore sous Python 3 mais c'est de plus en plus l'exception. Au cours de cette option, nous utiliserons **Python 3.5**

# Option « Programmation en Python »

## ► **Bases de la programmation en Python**

- ▶ environnement Python (venv, pip, ipython)
- ▶ type de données, *containers*, bloc conditionnel, boucles for
- ▶ fonctions, scripts & modules, initiation à la programmation orientée objet
- ▶ gestion & manipulation données numériques (numpy)

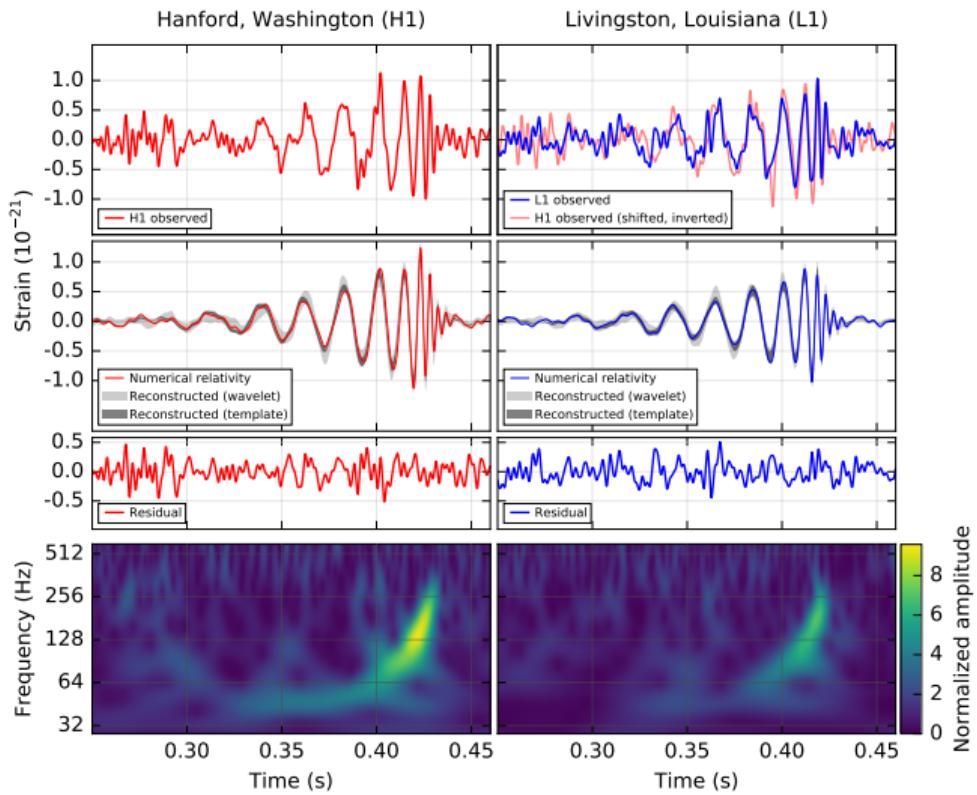
# Option « Programmation en Python »

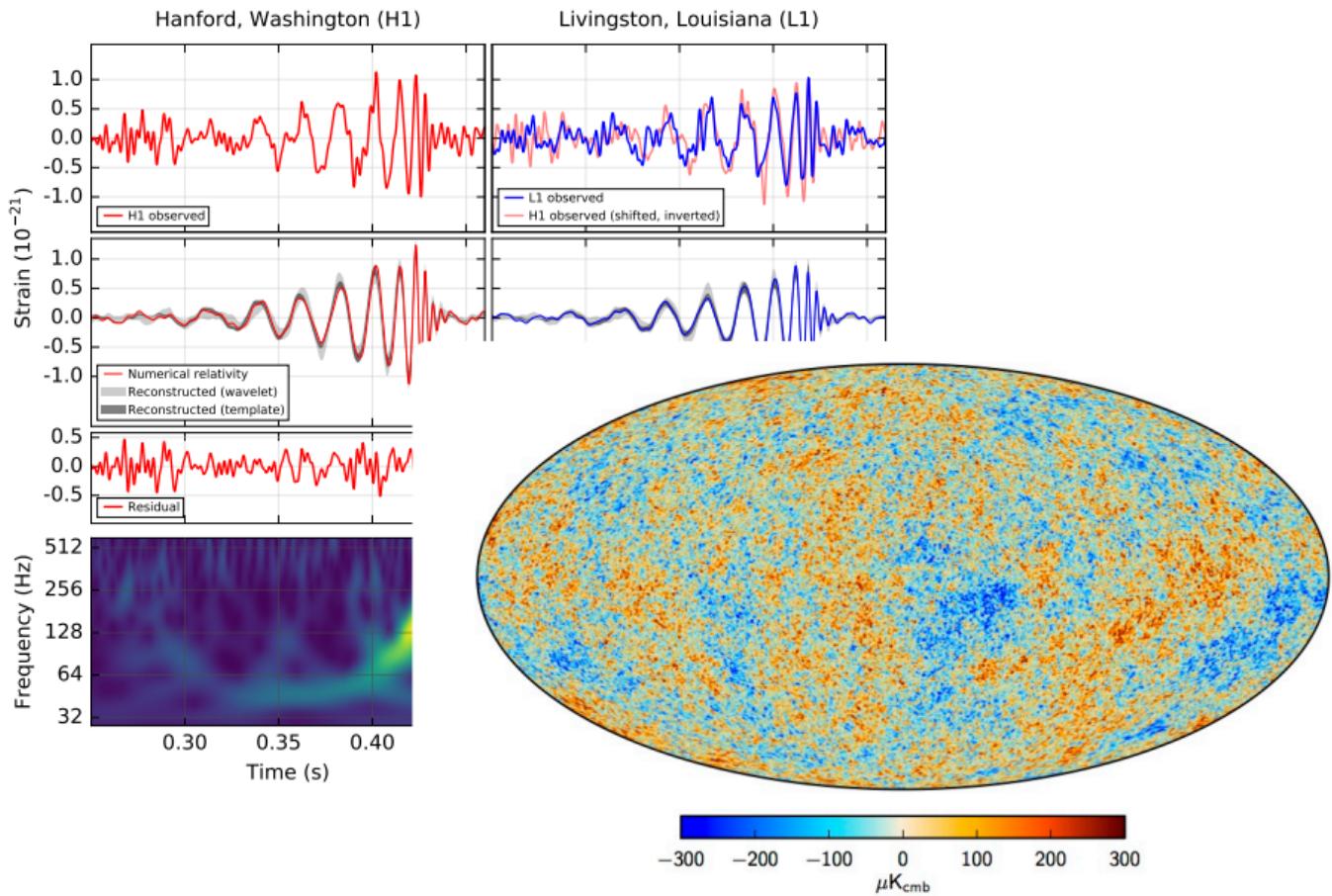
## ► **Bases de la programmation en Python**

- ▶ environnement Python (venv, pip, ipython)
- ▶ type de données, *containers*, bloc conditionnel, boucles for
- ▶ fonctions, scripts & modules, initiation à la programmation orientée objet
- ▶ gestion & manipulation données numériques (numpy)

## ► **Représentation graphique avec matplotlib**

- ▶ création de figures, *subplots*, axes, légendes
- ▶ histogramme, courbe de niveau, 3D





# Option « Programmation en Python »

## ► **Bases de la programmation en Python**

- ▶ environnement Python (venv, pip, ipython)
- ▶ type de données, *containers*, bloc conditionnel, boucles for
- ▶ fonctions, scripts & modules, initiation à la programmation orientée objet
- ▶ gestion & manipulation données numériques (numpy)

## ► **Représentation graphique avec matplotlib**

- ▶ création de figures, *subplots*, axes, légendes
- ▶ histogramme, courbe de niveau, 3D

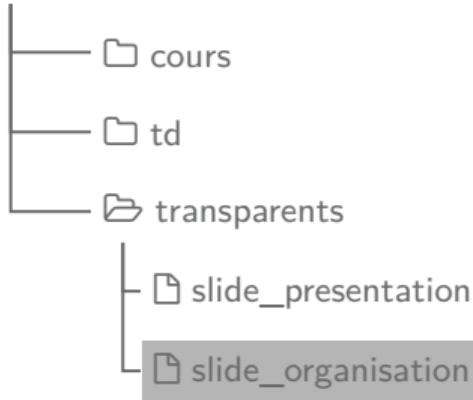
## ► **Interface graphique**

- ▶ utilisation des *widgets* de matplotlib
- ▶ applications directes aux TD Méca. Quan., EM, Phys. Stat.

## ► **Scipy : bibliothèques scientifiques**

- ▶ Algèbre linéaire
- ▶ Transformé de Fourier
- ▶ Optimisation & ajustement de fonction
- ▶ Outils statistiques & Générateurs aléatoires
- ▶ Interpolation, intégration numérique, traitement du signal...

## Option – Python



Ressources disponibles sur dokeos  
<http://formation.u-psud.fr>

- ▶ *Open Classrooms* : Apprenez à programmer en Python 
- ▶ **Python Data Science Handbook** 
- ▶ **Scipy Lectures Notes** 
- ▶ **Scientific Python Lectures** 
- ▶ ...