



# Python I

## Syllabus

Ce cours introduit les notions de base du langage python. L'objectif étant une prise en main rapide du langage et une présentation des concepts qui vous permettront de développer une autonomie considérable sur le reste de l'apprentissage. On y présentera la syntaxe globale de python, la philosophie du langage, de brèves comparaisons avec d'autres langages de bas niveau tels que le C++.

## Introduction

En allant sur la page Wikipédia de python, on tombe rapidement sur ceci

Certains langages tels que le **C++** sont **dits compilés**; c'est à dire que la traduction du code source en langage machine se fait une bonne fois pour toute au moment de la compilation. Avec Python, **il n'y a pas de phase de compilation**, la transformation en

langage machine se fait au moment de l'exécution du programme à la volée par un interpréteur.

## Naissance du langage

La première version du langage est développée par Guido van Rossum lors de l'hiver 1989 profitant d'une semaine de vacances durant les fêtes de Noël.



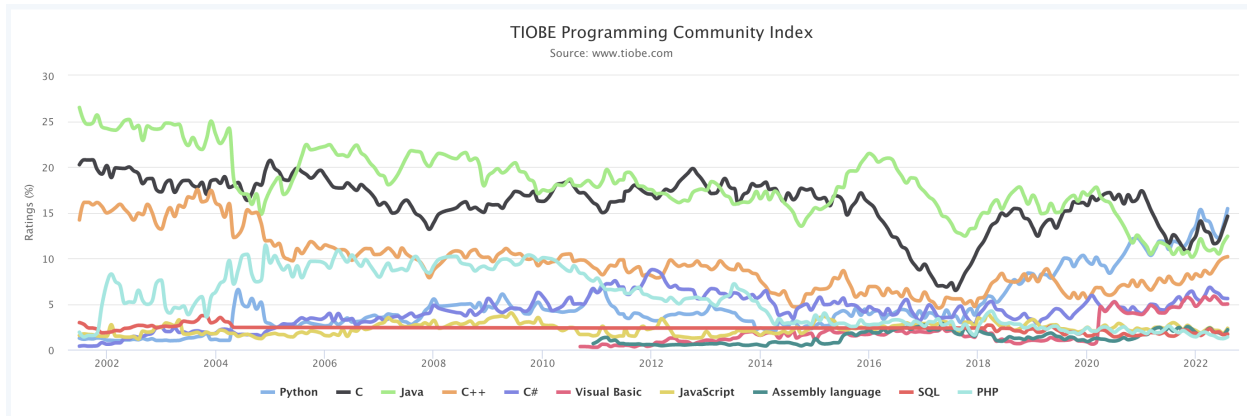
Guido van Rossum, [Hindustan times](#)

La première version publique est la 0.9.0 apparue en 1991. Depuis le code source est détenu par la Python Software Foundation fondée en 2001 pour promouvoir le langage et étendre sa communauté.

## Communauté et adoption

Comme toutes les bonnes choses, Python possède une horde de fans à travers le monde et pour s'en convaincre, il suffit de jeter un coup d'oeil à ces graphes (python -

courbe en bleu) provenant de TIOBE le site de référence pour suivre l'indice de popularité des langages de programmation.



## TIOBE

le classement.

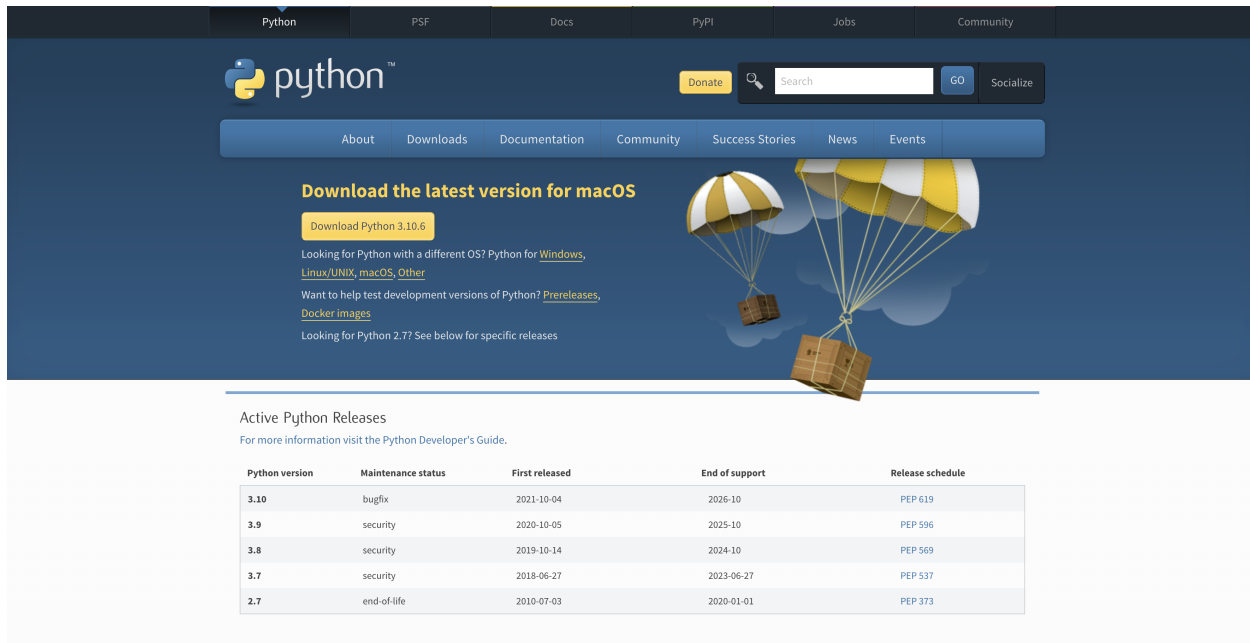
Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.42%	+3.56%
2	1	▼	 C	14.59%	+2.03%
3	3		 Java	12.40%	+1.96%
4	4		 C++	10.17%	+2.81%
5	5		 C#	5.59%	+0.45%
6	6		 Visual Basic	4.99%	+0.33%
7	7		 JavaScript	2.33%	-0.61%
8	9	▲	 Assembly language	2.17%	+0.14%
9	10	▲	 SQL	1.70%	+0.23%
10	8	▼	 PHP	1.39%	-0.80%

ça vous donne une bonne idée de la révérence qu'il convient de donner à ce langage.

Ceci s'explique en effet par une grande facilité d'apprentissage de python, une portabilité à toute épreuve et une grande communauté.

# Ressources

Pour suivre l'actualité python rendez vous sur [le site officiel](#)



The screenshot shows the Python.org website. At the top, there's a navigation bar with links to Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a search bar and a 'Donate' button. The main content area features a large banner for downloading the latest version of Python for macOS, with a 'Download Python 3.10.6' button. To the right of the banner is an illustration of two parachutes carrying boxes. Below the banner, there's a section titled 'Active Python Releases' with a link to the Python Developer's Guide. A table lists the active releases, showing the version, maintenance status, first release date, end of support date, and release schedule.

Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

et le [compte twitter du créateur](#)

## Langage

On introduit rapidement la syntaxe du langage en faisant abstraction de la mise en place d'un environnement de développement dédié. Les arcanes de la gestion d'environnement seront explicités dans une partie dédiée du cours. Ici, on travaillera dans un notebook anaconda.

Ce cours ne veut pas une présentation exhaustive de toutes les bases du langage mais constitue un bref rappel des notions de base de python.

## Introduction à la syntaxe du langage

Hello world

Ecrivons un premier programme qui permet d'afficher hello world sur la sortie de la cellule.

La syntaxe de base

```
# program that display hello world
print("Hello world, welcome to this world class course !")
```

```
# program that displays hello world
print("Hello World, welcome to this world class course")
✓ 0.2s
Hello World, welcome to this world class course
```

Ici le print est une fonction qui permet d'afficher quelque chose à l'écran, ici une chaîne de caractère.

## Opérations arithmétiques de base

Toujours sur votre notebook, vous pouvez lancer l'exécution de calculs arithmétiques de base.

```
2 + 3
✓ 0.3s
5
```

```
2 * 3
✓ 0.2s
6
```

```
2 / 3
✓ 0.2s
0.6666666666666666
```

```
3 // 2
✓ 0.2s
1
```

```
7 % 2
✓ 0.2s
1
```

## Affectation

L'opération d'affectation qui permet de passer une valeur à une variable.

```
a = 2
a
✓ 0.2s
2

b = a + 3
b
✓ 0.2s
5
```

## Affichage

La fonction print utilisée plus haut, elle permet d'afficher une chaîne de caractère sur l'IDLE.

```
print("bonjour")
✓ 0.2s
bonjour

a = 5
print(a)
✓ 0.2s
5
```

Elle peut aussi être utilisée avec deux arguments

```
a = 5
print("a vaut :", a)
✓ 0.2s
a vaut : 5
```

## Listes

Une liste est une suite d'éléments à laquelle on peut rajouter et enlever des items.

```
a = [1, 2, 8, 10]
a[0]
✓ 0.2s
1

a[-1]
✓ 0.2s
10

a.append(100)
a
✓ 0.2s
[1, 2, 8, 10, 100]
```

## la fonction len()

Cette fonction renvoie le nombre d'éléments d'un itérable (une liste par exemple).

```
a = [1, 2, 8, 10]
len(a)
✓ 0.3s
4
```

on reviendra plus en détails sur le concept d'itérable plus en détail sur le cours python II.

## Les commentaires

Un commentaire est une information générale mise dans le code afin de faciliter sa lecture ou d'expliquer certaines parties de ce dernier

```
# Ceci est un commentaire
✓ 0.1s
```

## Principaux types de données

Python étant un langage à typage dynamique, il n'est pas nécessaire de préciser le type de la variable lors de sa déclaration comme c'est le cas dans des langages tels que le C++. La valeur dépendant de la nature de la donnée qu'on manipule, le type du contenu d'une variable peut donc changer si on change sa valeur.

## La fonction type()

Elle permet de récupérer d'une variable.

```
a = 2
type(a)
✓ 0.2s
int

a = "i'm a and i am string"
type(a)
✓ 0.2s
str
```

## Le type int (entier)

Ce type est utilisé pour stocker un **entier** en anglais integer. Pour cette raison, on appelle ce type **int**.

```
a = 2
type(a)
int
```

## Le type float (flottant)

Ce type est utilisé pour stocker des **nombres à virgule flottante**, désignés en anglais par l'expression *floating point numbers*. Pour cette raison, on appelle ce type : **float**. En français, on parle de **flottant**.

```
a = 2.5
type(a)
✓ 0.3s
float
```

## Le type str (chaîne de caractères)

Sous Python, une donnée de type **str** est une suite quelconque de caractères délimitée soit par des apostrophes (simple quotes), soit par des guillemets (double quotes). **str** est l'abréviation de *string*, qui veut dire *chaîne* en français.



```
a = "i'm a and i am string"
type(a)
✓ 0.2s
str
```

## Le type bool (booléen)

Le type booléen prend uniquement deux valeurs (True ou False).

```
a = 2
a == 2
✓ 0.2s
True
```

## Le type list (liste)

Sous Python, on peut définir une liste comme une collection d'éléments.

```
a = [1, 2, 9]
isinstance(a, list)
✓ 0.2s
```

**isinstance(obj, type\_obj)** est une manière de dire que obj est du type type\_obj. elle renvoie un booléen.

En python les listes peuvent prendre des éléments de type différents.

```
a = [1, "hello world", 2.9]
a
✓ 0.2s
[1, 'hello world', 2.9]
```

# Les instructions de controle

## instruction if

Les instructions de controle permettent d'exécuter certaines parties d'un programme que sous certaines conditions.

La syntaxe de base est la suivante

```
if condition:
    instruction
```

ici condition est une expression booléenne; c'est à dire qui prend deux valeurs True ou False.

Un exemple.

```
x = 15
if x > 10:
    print(x, "est plus grand que 10")
✓ 0.2s
```

## instruction if else

```
if condition:
    Instruction A
else:
    Instruction B
```

Un exemple

```
a = 9
if a == 0:
    print("a equals to zero")
else:
    print("a do not equal to zero")
✓ 0.1s
a do not equal to zero
```

# Les boucles

## boucle for

Exemple

```
for i in [0, 1, 2, 3]:  
    print("i a pour valeur", i)  
✓ 0.2s  
i a pour valeur 0  
i a pour valeur 1  
i a pour valeur 2  
i a pour valeur 3
```

## boucle while

la syntaxe d'usage de la boucle while.

```
while condition:  
    instruction
```

## Exemple

```
a = 5  
while a > 1:  
    a = a - 1  
    print("a equals to", a)  
✓ 0.3s  
a equals to 4  
a equals to 3  
a equals to 2  
a equals to 1
```