

Cours de Python – 2 : Pourquoi Python ?



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 ».

1) Le langage machine

Le langage machine est le seul langage qu'un processeur puisse exécuter.

Le langage binaire est uniquement constitué de 0 et de 1.

« 01000010011011110110111001101010011011110111010101110010 », par exemple, signifie « Bonjour ».

→ $(01000010)_2 = 1 \times 2^6 + 1 \times 2^1 = 64 + 2 = (66)_{10}$ et dans la table ASCII : $(66)_{10}$ correspond à « B ».

→ $(01101111)_2 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 64 + 32 + 8 + 4 + 2 + 1 = (111)_{10}$ soit « o ».

Bref, autant vous dire que discuter en binaire avec un ordinateur peut être long (surtout pour vous).

→ en hexadécimal, « Bonjour » devient : 42 6F 6E 6A 6F 75 72, assez ardu à maîtriser en programmation

2) Les langages interprétés : Langages interprétés et langages compilés

Python est un langage de programmation interprété, c'est-à-dire que les instructions que vous lui envoyez sont « transcrites » en langage machine au fur et à mesure de leur lecture.

D'autres langages (comme le C / C++) sont appelés « **langages compilés** » car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine. On appelle cette étape la « **compilation** ». À chaque modification du code, il faut rappeler une étape de compilation.

Les avantages d'un langage interprété sont **la simplicité** (on ne passe pas par une étape de compilation avant d'exécuter son programme) et **la portabilité** (un langage tel que Python est censé fonctionner aussi bien sous Windows que sous Linux ou Mac OS, et on ne devrait avoir à effectuer aucun changement dans le code pour le passer d'un système à l'autre). Cela ne veut pas dire que les langages compilés ne sont pas portables, loin de là ! Mais on doit utiliser des compilateurs différents et, d'un système à l'autre, certaines instructions ne sont pas compatibles, voire se comportent différemment.

En contrepartie, **un langage compilé se révélera bien plus rapide qu'un langage interprété** (la traduction à la volée de votre programme ralentit l'exécution), bien que cette différence tende à se faire de moins en moins sentir au fil des améliorations. De plus, **il faudra installer Python sur le système d'exploitation que vous utilisez** pour que l'ordinateur puisse comprendre votre code.

Langages assembleurs :

Assembleur, Fortran (1954), Lisp (1958), Cobol (1960) (applications gestions),

Basic (1964), C (1970), SQL (1978) (langage de requêtes),

C++ (1985), Python (1990), Java (1995), JavaScript (1995), PHP (1995), C# (2000), Swift (2014) (applications IOS).

Dans l'ordre de popularité : Java, C, C++, Python

Le classement TIOBE 2017 sur l'utilisation des langages

May 2017	May 2016	Change	Programming Language	Ratings	Change
1	1		Java	14.639%	-6.32%
2	2		C	7.002%	-6.22%
3	3		C++	4.751%	-1.95%
4	5	▲	Python	3.548%	-0.24%
5	4	▼	C#	3.457%	-1.02%
6	10	▲▲	Visual Basic .NET	3.391%	+1.07%
7	7		JavaScript	3.071%	+0.73%
8	12	▲▲	Assembly language	2.859%	+0.98%
9	6	▼	PHP	2.693%	-0.30%
10	9	▼	Perl	2.602%	+0.28%
11	8	▼	Ruby	2.429%	+0.09%
12	13	▲	Visual Basic	2.347%	+0.52%
13	15	▲	Swift	2.274%	+0.68%
14	16	▲	R	2.192%	+0.86%
15	14	▼	Objective-C	2.101%	+0.50%
16	42	▲▲	Go	2.080%	+1.83%
17	18	▲	MATLAB	2.063%	+0.78%
18	11	▼▼	Delphi/Object Pascal	2.038%	+0.03%
19	19		PL/SQL	1.676%	+0.47%

3) À quoi peut servir Python ?

Python est un langage interprété :

- les lignes sont interprétées les unes après les autres
- pas de compilation : une seule étape
- moindre performance que les langages compilés

Python est **polyvalent sur multi-plateforme** (linux, windows, mac os, android, ...)

Python est **libre, gratuit, open source**

Python possède de nombreuses bibliothèques comprenant beaucoup d'applications, les bibliothèques par défaut sont relativement complètes

Sa **syntaxe est légère et épurée**, ce qui rend le code assez lisible, véritable langage de programmation relativement simple (pas de gestion de la mémoire, contrairement au C...)

Les messages d'erreurs sont simples et compréhensibles

Il est utilisé dans l'industrie, utilisé par la communauté scientifique, utilisé dans l'éducation nationale pour l'apprentissage de l'algorithmique depuis quelques années

Simplicité de mise en œuvre, un simple éditeur de texte suffit

Débogage facilité par l'interpréteur, avec des messages compréhensibles

Python est orienté objet : il s'agit de définir des briques logicielles appelées objets et de définir les interactions et les communications entre ces objets.

Edupython au lycée, Pyzo en Prépa

A la différence d'AlgoBox, Python permet d'interpréter des chaînes de caractères.

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant où vous l'installez sur votre ordinateur, vous disposez de nombreuses fonctionnalités intégrées au langage que nous allons découvrir tout au long de ce livre.

Il est, en outre, très facile d'étendre les fonctionnalités existantes, comme nous allons le voir. Ainsi, il existe ce qu'on appelle des **bibliothèques** qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.

Concrètement, voilà ce qu'on peut faire avec Python :

- de petits programmes très simples, appelés **scripts**, chargés d'une mission très précise sur votre ordinateur ;
- des programmes complets, comme des jeux, des suites bureautiques, des logiciels multimédias, des clients de messagerie...
- des projets très complexes, comme des progiciels (ensemble de plusieurs logiciels pouvant fonctionner ensemble, principalement utilisés dans le monde professionnel).

Voici quelques-unes des fonctionnalités offertes par Python et ses bibliothèques :

- créer des interfaces graphiques ;
- faire circuler des informations au travers d'un réseau ;
- dialoguer d'une façon avancée avec votre système d'exploitation ;
- ... et j'en passe...