

Introduction à Python

Historique du langage

Python est un langage de programmation de haut niveau, interprété et polyvalent, reconnu pour sa syntaxe claire et sa lisibilité. La genèse de Python remonte à la fin des années 1980, lorsque **Guido van Rossum**, alors chercheur au CWI (Centre pour les mathématiques et l'informatique) aux Pays-Bas, travaillait sur le langage pédagogique ABC. Cherchant à concevoir un langage plus puissant et flexible tout en conservant la simplicité d'ABC, il entame le développement de Python en décembre 1989. La première version publique (Python 0.9.0) est ainsi publiée en février 1991. Quant au nom Python, il ne provient pas du serpent homonyme, mais de la célèbre troupe d'humour britannique Monty Python, dont van Rossum était fan.

Objectifs initiaux de Python

Dès l'origine, Python a été guidé par des objectifs de simplicité, de lisibilité et de productivité. L'ambition première de Guido van Rossum était de proposer un langage de programmation « convivial et facile à comprendre » pour les développeurs. Pour cela, Python adopte une syntaxe épurée et explicite : le code s'écrit de manière presque naturelle, et l'indentation (les retraits) sert à délimiter les blocs d'instructions au lieu d'accolades ou de mots-clés, ce qui encourage un code bien structuré et lisible. Ce choix de conception – une syntaxe claire privilégiant la lisibilité – a pour corollaire une meilleure productivité des programmeurs, qui peuvent écrire et maintenir leur code plus efficacement. En d'autres termes, Python cherche à réduire la complexité « accidentelle » du code pour que les développeurs se concentrent sur la logique métier du programme.

Caractéristiques fondamentales du langage

Sur le plan technique, Python présente plusieurs caractéristiques majeures découlant de sa philosophie initiale. Tout d'abord, c'est un langage interprété : le code source n'a pas besoin d'être compilé à l'avance, il est exécuté directement ligne par ligne par un interpréteur. Cela facilite le développement interactif et la détection rapide des erreurs, au prix d'une exécution moins rapide que les langages compilés. Ensuite, Python est multiparadigme, supportant pleinement la programmation impérative structurée, la programmation orientée objet, ainsi que la programmation fonctionnelle. Ce n'est pas un langage lié à un seul paradigme : il permet par exemple d'écrire aussi bien des scripts procéduraux que de définir des classes et objets pour structurer un programme, ou d'utiliser des fonctions de premier ordre et des lambdas comme en programmation fonctionnelle. Par ailleurs, Python est doté d'un typage dynamique et fort : les variables n'ont pas de type statique déclaré dans le code, leur type est déterminé automatiquement à l'exécution (dynamique), et le langage s'assure qu'on ne combine pas de types incompatibles dans les opérations (fort). De plus, la gestion de la mémoire y est automatisée par un ramasse-poussières (garbage collector), et Python offre un système robuste de gestion d'exceptions qui simplifie le traitement des erreurs. Enfin, il convient de noter que Python est multiplateforme et distribué sous une licence libre, ce qui le rend disponible sur la plupart des systèmes d'exploitation (Windows, macOS, Linux) sans modification du code source.

Usages contemporains de Python

En un peu plus de trente ans, Python est passé du statut de projet confidentiel à celui de langage incontournable de l'informatique moderne. Il figure aujourd'hui parmi les langages les plus populaires et les plus enseignés, porté par une vaste communauté qui enrichit en permanence son écosystème. Python dispose en effet d'une bibliothèque standard très fournie ainsi que d'un grand nombre de bibliothèques spécialisées ou de frameworks développés par la communauté, ce qui facilite son utilisation dans de nombreux domaines. On le retrouve ainsi largement utilisé dans le développement web (principalement côté serveur, souvent en s'appuyant sur des frameworks comme Django ou Flask), dans la science des données (DataScience) et l'analyse statistique (où sa simplicité d'utilisation alliée à des bibliothèques comme NumPy, pandas ou matplotlib en fait un outil de choix pour les data scientists), ainsi qu'en intelligence artificielle et machine learning – domaines dans lesquels Python s'est imposé grâce à des frameworks de haut niveau tels que **TensorFlow** ou **PyTorch**. Python sert également énormément à l'automatisation des tâches informatiques et à l'écriture de scripts, qu'il s'agisse de prototyper rapidement une application, de réaliser des opérations répétitives (tests, déploiement, extraction de données en ligne, etc.) ou de piloter des systèmes complexes. Sa polyvalence s'étend enfin au domaine éducatif : la clarté de sa syntaxe et son faible "barrier to entry" (barrière d'entrée) expliquent qu'il soit fréquemment choisi comme premier langage d'apprentissage en programmation, permettant une initiation aisée aux concepts de base sans s'encombrer de détails techniques superflus.

Contenu de la formation

Le programme prévoit l'apprentissage progressif des structures de données principales du langage (listes, dictionnaires, etc.), de la définition et de l'utilisation de fonctions, des bases de la programmation orientée objet ainsi que de la manipulation de fichiers, avec une mise en pratique de l'ensemble de ces connaissances assurée au travers de deux projets applicatifs (travaux pratiques). En d'autres termes, les apprenants seront amenés à maîtriser les fondations de la programmation en Python – depuis la gestion des données en mémoire jusqu'à l'organisation du code en fonctions et en objets, en passant par l'interaction avec le système de fichiers – et à mobiliser ces compétences dans des projets concrets. Cette approche pédagogique par la pratique, centrée sur Python, vise à consolider les acquis théoriques tout en développant le savoir-faire nécessaire pour concevoir des programmes clairs, efficaces et fonctionnels.