

Introduction à Python

Python est un langage inventé par Guido van Rossum (NL) en 1989 et accessible au public dès 1991.

La version 3 est la plus récente de Python. La version 2 est obsolète.

Les avantages de Python sont nombreux : son utilisation est multiplateforme (Windows, macOS, Linux, Android, iOS), le logiciel est gratuit, le langage est relativement facile à apprendre, il est orienté objet (pour tout type de développement), et Python est un langage interprété par un logiciel (l'interpréteur Python). Ce qui signifie que les fichiers Python ne doivent pas être compilés pour être lus. Ils sont directement exécutés par un logiciel qui interprète les instructions données en fonction de leur syntaxe. Tous ces avantages font de Python le langage le plus utilisé au monde.

Pour écrire des fichiers (scripts) Python, un éditeur de texte (IDE : notepad++, VSCode, Sublime Text) est nécessaire.

Depuis l'IDE on peut accéder à deux terminaux. Le terminal système (PowerShell) et le terminal Python. Le terminal système apparaît par défaut sous Windows dans VSCode. Powershell est un interpréteur de commandes système, c'est-à-dire qu'il permet d'interagir avec l'ordinateur en fonction de certaines commandes.

Depuis le terminal système on accède au terminal Python en tapant « python » à la suite de l'invite de commande en forme de chemin d'accès (PS C:\Users\...). Le signe >>> apparaît alors pour indiquer que le terminal est prêt à recevoir du code Python. Le terminal Python permet d'écrire des programmes qui sont directement exécutés après interprétation en fonction de leur syntaxe. Pour revenir à l'environnement système depuis Python, taper « quit() » ou « exit() » à la suite du signe >>>.

Python est un interpréteur de codes tandis que PowerShell un interpréteur de commandes.

Chaque instruction donnée à l'interpréteur Python entraîne une réponse de l'interpréteur qui s'affiche à la ligne qui suit l'instruction. Soit l'interpréteur attend une instruction supplémentaire à la suite de l'instruction reçue et il affiche ... pour indiquer qu'il attend la suite de l'instruction pour l'exécuter, soit il exécute l'instruction et affiche >>> pour indiquer qu'il est prêt à recevoir d'autres instructions. Tout le travail de codage dans Python repose sur le principe général qu'un input (instruction donnée à l'interpréteur) entraîne un output (attente de la suite d'une instruction ou d'un résultat).

Dans le cas particulier des codes qui utilisent des boucles ou des testeurs logiques (bool), la séquence input/output est mise en évidence par le principe de l'indentation. La ligne qui reçoit le bloc d'instructions à exécuter juste après une boucle ou un testeur logique est décalée vers la droite pour le distinguer des instructions qui n'en font pas partie et qui s'écrivent sans décalage vers la droite. C'est le principe de l'indentation, il permet de reconnaître qu'une instruction ou un bloc d'instructions fait partie d'une boucle ou d'un test logique.

Une bonne habitude consiste à respecter une indentation de 4 espaces.

Python permet d'introduire des commentaires dans son code. Tout ce qui est écrit après le signe # est ignoré par Python qui l'interprète comme un commentaire. Un commentaire est la documentation d'un code. Il doit expliquer en langage humain ce que fait un code pour qu'il soit compris par un autre programmeur. La présence de commentaires dans un code fait partie des bonnes pratiques en programmation.