# Combined logoPython 3

## Définition

Python est un langage de programmation inte[[1]](#footnote-1)rprété, polyvalent et de haut niveau. Il a été créé par Guido van Rossum et publié pour la première fois en 1991. Python se distingue par sa syntaxe simple et lisible, ce qui en fait un langage facile à apprendre et à comprendre pour les débutants. Il favorise une approche claire et expressive du code, mettant l'accent sur la lisibilité et la facilité de compréhension.

Python prend en charge plusieurs paradigmes de programmation, tels que la programmation impérative, la programmation orientée objet et la programmation fonctionnelle. Il offre une vaste bibliothèque standard qui couvre divers domaines, allant des opérations de base sur les chaînes de caractères et les listes aux fonctionnalités avancées telles que la gestion de fichiers, les opérations réseau, la manipulation de données, etc.

Python est utilisé dans de nombreux domaines tels que le développement web, l'analyse de données, l'apprentissage automatique, l'intelligence artificielle, l'automatisation de tâches, les sciences, la robotique et bien d'autres. Sa popularité est en grande partie due à sa polyvalence, sa facilité d'utilisation, sa communauté active et sa vaste gamme de bibliothèques et de frameworks disponibles.

L'interpréteur Python est disponible sur plusieurs plateformes, y compris Windows, macOS et différentes distributions Linux, ce qui permet aux développeurs de travailler sur divers environnements.

## Histoire

L'histoire du langage de programmation Python remonte aux années 1980. Guido van Rossum, un informaticien néerlandais, a créé Python en tant que projet personnel pendant son temps libre. Il souhaitait développer un langage de programmation facile à lire, élégant et efficace, adapté aux tâches quotidiennes de programmation.

La première version publique de Python, Python 0.9.0, a été publiée en février 1991. Le nom "Python" a été choisi en référence à la série télévisée britannique "Monty Python's Flying Circus", dont Guido van Rossum était fan. Depuis lors, la communauté Python a adopté de nombreux termes humoristiques inspirés de cette série.

Guido van Rossum a continué à développer Python et à coordonner les efforts de la communauté jusqu'en juillet 2018, date à laquelle il a annoncé sa retraite en tant que chef de projet. Depuis lors, le développement de Python est géré par une équipe de contributeurs et de développeurs actifs.

Au fil des ans, Python a gagné en popularité et est devenu l'un des langages de programmation les plus utilisés dans le monde. Sa simplicité syntaxique, sa polyvalence, sa grande bibliothèque standard et sa communauté active ont contribué à son adoption dans divers domaines, de l'informatique scientifique à la programmation Web en passant par l'intelligence artificielle et l'apprentissage automatique.

Python a connu plusieurs versions majeures, la plus récente étant Python 3, qui a été publiée en 2008. Python 3 a apporté de nombreuses améliorations et des changements de syntaxe par rapport à Python 2, tout en veillant à maintenir la compatibilité avec les versions précédentes. Bien que Python 2 ait été officiellement abandonné en janvier 2020, de nombreuses applications et bibliothèques ont été migrées vers Python 3 pour bénéficier de ses fonctionnalités et de ses améliorations.

## Installation

Voici la procédure complète pour installer Python 3 sur Windows 10 et Windows 11 :

1. Tout d'abord, vous devez accéder au site officiel de Python pour télécharger l'installateur. Ouvrez votre navigateur Web et rendez-vous sur <https://www.python.org/downloads/>.
2. Sur la page de téléchargement, vous verrez différentes versions de Python disponibles. Assurez-vous de sélectionner la dernière version de Python 3.x (par exemple, Python 3.9.x) pour bénéficier des dernières fonctionnalités et corrections de bugs.
3. Faites défiler la page jusqu'à ce que vous trouviez les téléchargements pour Windows. Vous verrez des options pour les versions 32 bits et 64 bits de Python. Si vous n'êtes pas sûr de la version de votre système d'exploitation, vous pouvez vérifier en allant dans "Paramètres" -> "Système" -> "À propos" et recherchez la section "Type de système".
4. Choisissez l'installateur correspondant à votre version de Windows (par exemple, "Windows x86-64 executable installer" pour Windows 64 bits) et cliquez dessus pour télécharger l'installateur.
5. Une fois le fichier téléchargé, ouvrez-le en double-cliquant dessus. Vous verrez une boîte de dialogue intitulée "Python Setup". Cochez la case "Add Python 3.x to PATH" (Ajouter Python 3.x au chemin d'accès) et assurez-vous que toutes les autres options sont sélectionnées selon vos préférences.
6. Cliquez sur le bouton "Install Now" (Installer maintenant) pour lancer l'installation. Vous pouvez également personnaliser l'emplacement d'installation en cliquant sur "Customize installation" (Personnaliser l'installation).
7. L'installateur Python extraira les fichiers nécessaires et configurera Python sur votre système. Attendez que le processus d'installation se termine.
8. Une fois l'installation terminée, vous verrez une fenêtre "Setup was successful" (L'installation s'est déroulée avec succès). Cochez la case "Disable path length limit" (Désactiver la limite de longueur du chemin d'accès) si vous souhaitez contourner les restrictions liées à la longueur des chemins d'accès.
9. Cliquez sur le bouton "Close" (Fermer) pour fermer l'installateur.
10. Pour vérifier si Python a été installé correctement, ouvrez l'invite de commandes en appuyant sur la touche Windows + R, tapez "cmd" et appuyez sur Entrée. Dans la fenêtre de l'invite de commandes, tapez "python --version" et appuyez sur Entrée. Vous devriez voir la version de Python installée affichée.

## Premier essai

Voici la procédure pour effectuer un premier test de Python dans l'invite de commandes Windows:

1. Ouvrez l'invite de commandes en appuyant sur la touche Windows + R, tapez "cmd" et appuyez sur Entrée. L'invite de commandes s'ouvrira.
2. Tapez "python" dans l'invite de commandes et appuyez sur Entrée. Cela lancera l'interpréteur Python et vous permettra d'entrer des commandes Python directement dans l'invite de commandes.
3. Pour effectuer un test simple, vous pouvez saisir une commande Python de base, par exemple:

print("Hello, World!")

Cette commande affiche le message "Hello, World!" à l'écran.

1. Appuyez sur Entrée pour exécuter la commande. Vous devriez voir le résultat affiché dans l'invite de commandes, comme ceci :

Hello, World!

1. Vous pouvez aussi faire une équation mathématique :

1 + 1

1. Affectez une valeur numérique à un variable et effectuez une opération mathématique avec la variable :

alex = 10

1. Pour terminé, affichez le contenu de la variable :

print(alex)

Vous pouvez également exécuter des scripts Python depuis l'invite de commandes en utilisant la commande "python chemin\_du\_script.py", où "chemin\_du\_script.py" est le chemin d'accès complet vers votre fichier Python. Assurez-vous d'avoir préalablement enregistré le script Python avec l'extension .py.

N'oubliez pas que pour exécuter Python dans l'invite de commandes, vous devez avoir Python installé et ajouté au chemin d'accès système. Si vous n'avez pas encore installé Python, vous pouvez suivre les instructions de l'une des réponses précédentes pour installer Python sur votre système Windows.

1. Créer un nouveau fichier avec l’extension .py

copy con test.py

1. Écrivez le texte suivant et sauvegardez le fichier avec CTRLL+Z

print("Hello, World!")

1. Executez le script python :

python test.py

# Une image contenant Graphique, graphisme, Police, logo Description générée automatiquementPyCharm

## Définition

PyCharm est un environnement de développement intégré (IDE) spécifiquement conçu pour le développement de logiciels en utilisant le langage de programmation Python. Il est développé par JetBrains, une société spécialisée dans les outils de développement logiciel.

PyCharm offre un large éventail de fonctionnalités pour faciliter le développement Python, qu'il s'agisse de projets petits ou grands. Voici quelques-unes des caractéristiques principales :

1. **Éditeur de code avancé** : PyCharm offre un éditeur de code puissant avec des fonctionnalités telles que la coloration syntaxique, l'achèvement automatique, la mise en forme du code, la navigation intelligente, la refactorisation, la gestion des imports, etc.
2. **Débogage intégré** : PyCharm permet le débogage de votre code Python, ce qui facilite la détection et la correction des erreurs. Il offre des fonctionnalités de débogage avancées telles que des points d'arrêt, l'inspection des variables, l'exécution pas à pas, etc.
3. **Gestion des projets** : PyCharm fournit une interface conviviale pour gérer vos projets Python. Il facilite la création de nouveaux projets, l'organisation des fichiers et des répertoires, la configuration des dépendances, la gestion des environnements virtuels, etc.
4. **Intégration avec les outils externes** : PyCharm s'intègre avec de nombreux outils et frameworks couramment utilisés dans l'écosystème Python. Il prend en charge l'intégration avec des outils de gestion de versions tels que Git, des outils de test comme pytest, des frameworks web comme Django et Flask, des outils d'analyse de code statique, etc.
5. **Productivité améliorée** : PyCharm propose des fonctionnalités pour améliorer la productivité du développeur. Il inclut des raccourcis clavier personnalisables, des modèles de code, des inspections de code pour détecter les erreurs potentielles, des suggestions de refactorisation, des outils d'analyse de performance, etc.
6. **Prise en charge des langages connexes** : Outre Python, PyCharm prend également en charge d'autres langages de programmation tels que HTML, CSS, JavaScript, SQL, etc. Cela facilite le développement d'applications Web complètes en utilisant des technologies front-end et back-end.

En résumé, PyCharm est un environnement de développement puissant et convivial pour les développeurs Python. Il fournit des fonctionnalités avancées, une intégration étroite avec les outils et frameworks populaires, et vise à améliorer la productivité et la qualité du code lors du développement de logiciels Python.

## Installation

Vous devez aller sur votre compte JetBrains : <https://account.jetbrains.com/login>

## Création d’un premier projet

Voici la procédure complète pour préparer un premier projet Python dans PyCharm[[2]](#footnote-2) :

* **Lancement de PyCharm** :
  + Une fois l'installation terminée, lancez PyCharm en cherchant son icône dans le menu de démarrage ou en cliquant sur son raccourci sur le bureau.
  + Connexion à votre compte JetBrains. Si vous avez ce message, cliquez sur ‘ACTIVATE’

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* **Création d'un nouveau projet** :
  + À l'ouverture de PyCharm, vous verrez l'écran de bienvenue. Cliquez sur "Create New Project" (Créer un nouveau projet) ou allez dans "File" (Fichier) -> "New Project" (Nouveau projet).
  + Dans la fenêtre "New Project", choisissez un emplacement pour votre projet en spécifiant le répertoire et le nom du projet.
  + Sélectionnez l'interpréteur Python que vous souhaitez utiliser pour le projet. Si vous n'avez pas encore configuré un interpréteur, vous pouvez en choisir un existant ou en créer un nouveau en cliquant sur le lien "..." à droite du champ de sélection.
  + Cliquez sur "Create" (Créer) pour créer le projet.
* **Configuration des paramètres du projet** :
  + Une fois le projet créé, vous verrez l'interface de PyCharm. Dans la partie supérieure droite, cliquez sur l'icône d'engrenage à côté du nom du projet pour accéder aux paramètres du projet.
  + Dans les paramètres, vous pouvez configurer diverses options telles que les interpréteurs Python, les environnements virtuels, les dépendances, les outils externes, etc. Vous pouvez également définir les préférences de l'éditeur selon vos besoins.
* **Création d'un fichier Python** :
  + Dans la barre de navigation à gauche de PyCharm, cliquez avec le bouton droit sur le nom du projet et sélectionnez "New" (Nouveau) -> "Python File" (Fichier Python).
  + Donnez un nom à votre fichier Python, par exemple "main.py", et cliquez sur "OK".
  + Le fichier Python sera créé et ouvert dans l'éditeur de code de PyCharm.
* **Écriture du code** :
  + Dans le fichier Python, vous pouvez commencer à écrire votre code Python. PyCharm propose des fonctionnalités d'achèvement automatique, de mise en forme du code, de débogage, etc., qui peuvent vous aider lors de l'écriture du code.
* **Exécution du code** :
  + Pour exécuter votre code Python, vous pouvez cliquer avec le bouton droit dans l'éditeur de code et sélectionner "Run" (Exécuter), ou appuyer sur la touche de raccourci appropriée (généralement Ctrl+Shift+F10).
  + Les résultats de l'exécution du code seront affichés dans la fenêtre "Run" en bas de l'interface de PyCharm

## Fenêtres principales

PyCharm, en tant qu'environnement de développement intégré (IDE) complet pour Python, présente plusieurs fenêtres principales qui offrent différentes fonctionnalités pour faciliter le développement. Voici une description des fenêtres principales couramment utilisées :

1. **Éditeur de code** : C'est la fenêtre principale où vous écrivez votre code Python. Elle occupe généralement la plus grande partie de l'interface. L'éditeur de code de PyCharm offre des fonctionnalités avancées telles que la coloration syntaxique, l'achèvement automatique, la mise en forme du code, la navigation intelligente, la refactorisation, la gestion des imports, etc.
2. **Explorateur de projets** : Cette fenêtre affiche l'arborescence des fichiers et répertoires de votre projet. Vous pouvez y naviguer pour accéder aux différents fichiers de votre projet, créer de nouveaux fichiers, organiser les fichiers et les dossiers, etc. L'explorateur de projets permet également de rechercher et de filtrer les fichiers.
3. **Fenêtre de terminal** : PyCharm intègre un terminal dans son interface, ce qui vous permet d'exécuter des commandes Python ou des commandes système directement depuis l'IDE. La fenêtre du terminal vous permet d'interagir avec votre projet, d'exécuter des scripts Python, de gérer des dépendances, etc.
4. **Fenêtre de résultats (Run)** : Lorsque vous exécutez votre code Python, les résultats sont affichés dans la fenêtre de résultats. Elle affiche les sorties du programme, les erreurs éventuelles, les messages de débogage, etc. Vous pouvez également utiliser cette fenêtre pour analyser les résultats et effectuer des opérations supplémentaires lors de l'exécution de votre code.
5. **Fenêtre de débogage (Debug)** : Si vous utilisez le débogueur intégré de PyCharm, la fenêtre de débogage s'affiche lorsque vous exécutez votre code en mode de débogage. Elle affiche les variables, les points d'arrêt, les piles d'appels, les étapes de débogage, etc. Vous pouvez utiliser cette fenêtre pour inspecter et surveiller l'état de votre programme pendant le débogage.
6. **Fenêtre de contrôle de version** : Si vous utilisez un système de contrôle de version tel que Git, PyCharm offre une fenêtre dédiée pour gérer les opérations de versionnement. Cette fenêtre vous permet de voir les modifications apportées aux fichiers, d'effectuer des validations, des mises à jour, des branches, des fusions, etc.
7. **Fenêtre des outils externes** : PyCharm propose une intégration avec divers outils externes tels que des bases de données, des serveurs d'application, des outils d'analyse de code, etc. Vous pouvez ouvrir des fenêtres spécifiques à ces outils pour effectuer des opérations liées à ces domaines.

Ces fenêtres principales de PyCharm peuvent être personnalisées et redimensionnées en fonction de vos préférences. Vous pouvez également les arranger en différentes dispositions d'interface pour optimiser votre flux de travail.

# Références

* <https://python.sdv.univ-paris-diderot.fr/01_introduction/#12-conseils-pour-installer-et-configurer-python>

1. <https://www.docstring.fr/blog/python-est-il-un-langage-interprete/#:~:text=Un%20langage%20interpr%C3%A9t%C3%A9%20est%20un,a%20besoin%20de%20langage%20machine> [↑](#footnote-ref-1)
2. <https://www.jetbrains.com/help/pycharm/creating-empty-project.html> [↑](#footnote-ref-2)