



# Python - Installation - Démarrage

[Mise à jour le : 23/8/2022]

- **Sources**

- Python.org, [téléchargements](#) et [documentation](#).
- IDE [Visual Studio Code](#)
- MOOC "Python 3 : des fondamentaux aux concepts avancés du langage" sur [FUN](#)
- [Fonctions natives](#) (built-in)

- **Lectures connexes**

- [Python Program Lexical Structure](#)
- [Your Python Coding Environment on Windows: Setup Guide](#)

## Introduction

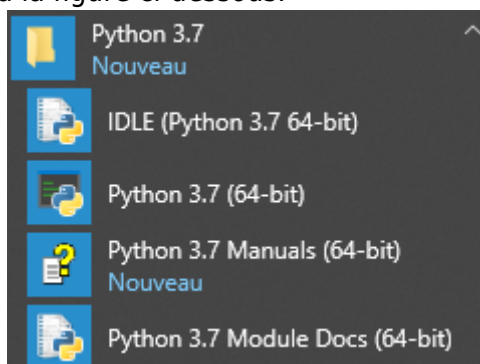
Après une brève présentation de la démarche à suivre pour installer Python sous W10 et supérieur et les extensions dans VSCode, cette page propose une première utilisation de l'interpréteur de commande (REPL) et de l'éditeur IDLE.

## 1. Installations

### 1.1 Python sur un PC sous Windows 10

- **Solution 1**

- **Télécharger** une version de Python (de préférence la dernière version 3) sur le site [Python.org](#).
- **Dézipper** le paquet et suivre les instructions d'installation. Le répertoire Python du menu démarrer doit ressembler à la figure ci-dessous.



- **Solution 2**

- **Installer le gestionnaire de paquets pour Windows** [Chocolatey](#) (mode administrateur). Celui-ci est construit sur l'infrastructure NuGet qui utilise PowerShell pour livrer des paquets à partir de la distribution.
- **Installer Python 3**

\*.ps

```
choco install python3
```




- **Solution 3**

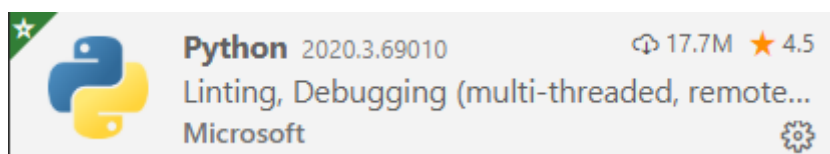
- **Installer** une distribution [Anaconda](#)

## 1.2 Python sous Linux

Python est déjà installé sur les distribution Linux courantes (**Ubuntu**, etc.)

## 1.3 L'extension "Python" dans VSCode

- Installer [VSCode](#) puis clic sur  et entrer "python" dans la barre de recherche. Installer l'extension ci-dessous.



## 1.4 Python Tutor

[Python Tutor](#) vous aide à apprendre **Python**, JavaScript, C, C++ et Java en visualisant l'exécution du code. Vous pouvez l'utiliser pour déboguer vos devoirs et en complément des didacticiels de codage en ligne.

## 2. Particularités du langage

Une spécificité du langage est la disposition du code. En C, C++, C# ou Java les instructions sont séparées par des “;” et les blocs d'instructions sont entourés par des “{ }”.

En Python, pour passer d'une instruction à la suivante, il suffit d'**aller à la ligne**. Pour créer un bloc d'instructions, on utilise l'**indentation (décalage de 4 espaces vers la droite)** et toutes les instructions du bloc doivent être alignées.

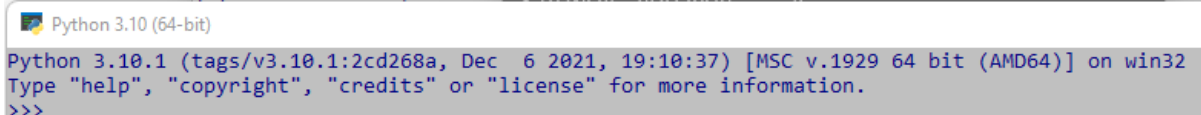
## 3. L'interpréteur de commandes (REPL)

Une boucle de lecture-évaluation-impression ou **Read-Eval-Print-Loop** (REPL) est un environnement de programmation informatique interactif qui prend les entrées individuelles de l'utilisateur, les exécute et renvoie le résultat à l'utilisateur.

### 3.1 Activation

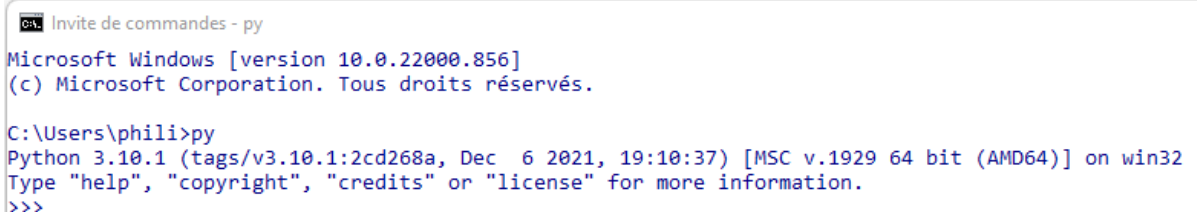
- **Sous Windows**

- **Cas 1** : menu **Démarrer** → **Python 3.x.x**. Une console s'ouvre comme ci-dessous. Les trois chevrons > > > identifient l'**invite de commande** Python.



```
Python 3.10 (64-bit)
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- **Cas 2** : ouvrir une invite de commande, entrer **py**.



```
Invite de commandes - py
Microsoft Windows [version 10.0.22000.856]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\phili>py
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- **Sous Linux**

Par défaut, Python est installé sur GNU/Linux. Sous Ubuntu, saisir “**python3**” dans un terminal pour avoir accès à l'invite de commande Python.

```
mno@PC-BUREAU: /mnt/c/WINDOWS/system32

mno@PC-BUREAU:/mnt/c/WINDOWS/system32$ python3
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

### 3.2 Ecrire dans l'interpréteur

- **Ecrire sur une seule ligne dans l'interpréteur**

Pour tester le fonctionnement de l'interpréteur Python, entrer le code ci-dessous : une commande par ligne.

\*.py

```
>>> 2+3
>>> a=2+3
>>> print("Le résultat de 3+2 est",a)
```

Résultat attendu

```
>>> 2+3
5
>>> a=3+2
>>> print("Le résultat de 3+2 est",a)
Le résultat de 3+2 est 5
>>>
```

- **Ecrire sur plusieurs lignes dans l'interpréteur**

Il existe deux méthodes pour **écrire sur plusieurs lignes** :

- la continuation de ligne **implicite** et
- la continuation de ligne **explicite**.

- **Continuation de ligne implicite**

Toute instruction contenant une **parenthèse ouvrante** ( '(' ), un **crochet** ( '[' ), une **accolade** ( '{' ), ou **deux points** ( ':' ) est considérée par l'interpréteur Python comme incomplète et **peut être poursuivie sur les lignes suivantes** jusqu'à ce que la parenthèse, crochet ou accolade correspondante soit rencontrée.

**Ctrl Maj Entrée** pour sortir de l'édition et exécuter le code.

Exemples

\*.py

```
>>> s = ('a' + 'b'
        + 'c' + 'd'
        + 'e' + 'f'
        )

>>> for i in range(10):
        print(i)
```

### • Continuation de ligne explicite

La continuation de ligne explicite est obtenue en spécifiant une **barre oblique inverse** (\) à la fin de chaque ligne à poursuivre. La barre oblique inverse doit être le dernier caractère d'une ligne continue. Il ne peut y avoir rien qui le suit, pas même des espaces.

Exemple

\*.py

```
>>> s = 'a' + 'b' \
        + 'c' + 'd' \
        + 'e' + 'f'
```

Les méthodes de continuation de ligne ne s'applique pas à plusieurs commandes. Pour exécuter plusieurs commandes dans l'interpréteur sur **une seule ligne** : les séparer par un **point-virgule**.

Exemple

 Invite de commandes - py

```
>>> print("Hello"); nom=input("Nom ? "); print("Hello",nom); humeur=input("Comment vas-tu ? ") ; print(nom,"va",humeur);
Hello
Nom ? Lamarche
Hello Lamarche
Comment vas-tu ? bien
Lamarche va bien
>>>
```

## 3.3 Aide en ligne

1. **help**(sorted) ou **help**(numpy. ...)
2. **s** = set({3, 4}); **help**(s) : donne de l'aide sur le type ou la classe à laquelle l'objet appartient.
3. **help**(argparse) : donne l'aide sur le module.
4. Depuis le shell on peut aussi faire **pydoc** argparse pour avoir l'aide sur un module, etc.

Exemple : liste des méthodes d'une classe.

```
>>> import math
>>> dir(math)
['_doc_', '__loader__', '__name__',
, 'asin', 'asinh', 'atan', 'atan2', 'i
'degrees', 'e', 'erf', 'erfc', 'exp',
od', 'frexp', 'fsum', 'gamma', 'gcd',
```

### 3.4 Sortie

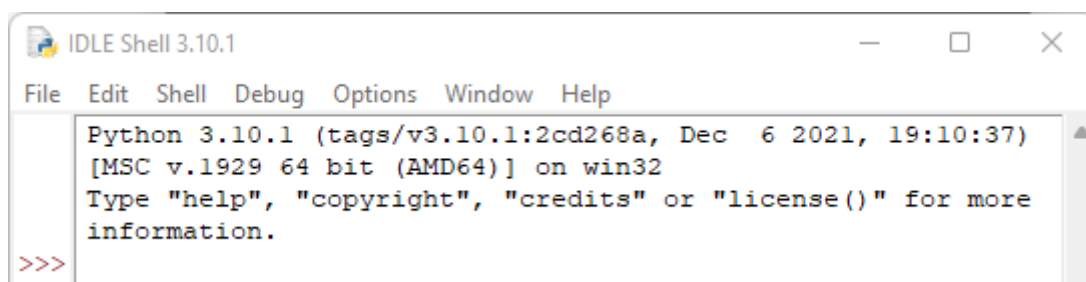
\*.py

```
exit()
```

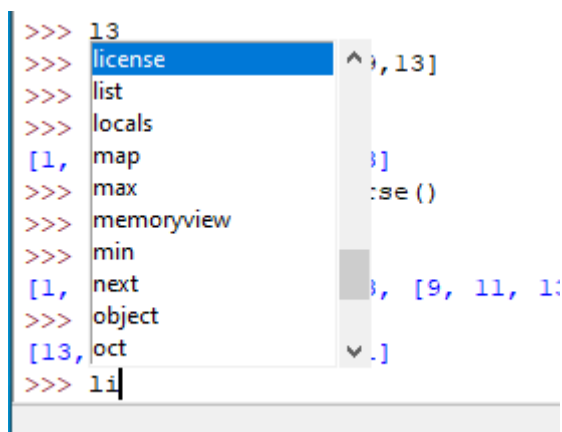
## 4. IDLE : l'éditeur fourni avec Python

### 4.1 Ouvrir IDLE

- Dans le menu démarrer : Python → IDLE



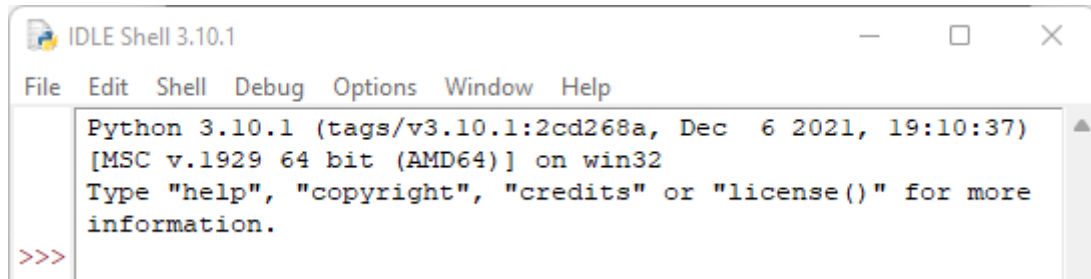
L'interpréteur de commande ci-dessus fonctionne comme celui présenté au paragraphe précédent. Cet interpréteur est cependant un peu plus confortable. Il dispose notamment de la **complétion de code** (touche **TAB**).



En cas d'**erreur lors de l'écriture**, il suffit de remonter dans le code avec les flèches puis "Entrée" pour corriger.

## 4.2 Créer un fichier

- Dans la fenêtre, sélectionner **file → new file**. Une nouvelle fenêtre s'ouvre. Il est alors possible de créer, de sauvegarder et d'exécuter un code Python.



## 4.3 Bonnes pratiques

- **Plusieurs instructions par ligne dans un fichier**

Le caractère utilisé pour séparer les instructions sur la même ligne est le **point-virgule (;)**. Cependant, placer plus d'une déclaration sur une ligne (dans un fichier) est généralement déconseillé par **PEP 8**.

- **Commenter son code**
  - Avec un **dièse** sur une seule ligne
  - Avec des **triples quote** ou des **guillemets** sur plusieurs lignes

Cependant, **PEP 8 décourage cette pratique** car, par convention, ce type de littéral de chaîne indépendant entre guillemets triples est réservé aux **docstrings** de fonction.

\*.py

```
# Ceci est un commentaire sur une seule ligne
""" Des commentaires sur
plusieurs lignes """
''' D'autres commentaires sur
plusieurs lignes '''
```

## 5. Créer un exécutable

Python ne fournit pas de méthode native pour produire un exécutable autonome ou une copie autonome d'un programme Python. La bibliothèque [PyInstaller](#) permet de conditionner des applications utilisant de nombreuses bibliothèques.

---

### Résumé

- L'**interpréteur de commandes Python (REPL)** permet de tester du code au fur et à mesure qu'on l'écrit. Il accepte des nombres et peut réaliser des calculs. Un nombre décimal s'écrit avec un point.
- L'éditeur **IDLE** (Integrated DeveLopment Environment), fourni dans l'installation de Python, permet d'écrire des programmes et dispose de la coloration syntaxique, l'autocomplétion, l'indentation et un débogueur intégré.
- Une **extension Python** proposant les mêmes fonctionnalités peut également être installée dans **VSCode**.
- Un commentaire sera ignoré lors de l'exécution du code. Il existe deux façons de commenter son code :
  - Avec un **dièse** sur une seule ligne
  - Avec des **triples quote** ou **guillemets** sur plusieurs lignes. (réservé aux docstrings)



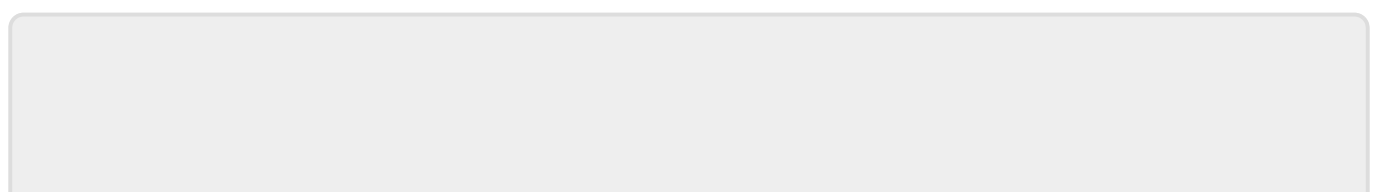
### Quiz

- [Python Program Structure Quiz](#)



### Pour aller plus loin ...

- [Managing Multiple Python Versions With pyenv](#)





From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<http://webge.fr/dokuwiki/doku.php?id=python:installation&rev=1662612845>



Last update: **2022/09/08 06:54**