**Cours Python**

Table des matières

[I) Paramétrer votre environnement 2](#_Toc211954378)

[I. 1) Installer VS Code 2](#_Toc211954379)

[I. 2) Paramétrez votre environnement pour python 2](#_Toc211954380)

[I. 3) Installer python sur votre ordinateur 2](#_Toc211954381)

[I. 4) Exécutez votre premier programme Python 3](#_Toc211954382)

[I. 5) Ecrire le programme dans un fichier et l’exécuter 3](#_Toc211954383)

[II) Les bases de python 4](#_Toc211954384)

[II. 1) Les types 4](#_Toc211954385)

[II. 2) Les entrées sorties 5](#_Toc211954386)

[II. 3) Les variables 6](#_Toc211954387)

[II. 4) Les affectations et comparaisons 6](#_Toc211954388)

[II. 5) Les listes 7](#_Toc211954389)

[II. 5. 1) Accédez aux éléments d’une liste 7](#_Toc211954390)

[II. 5. 2) Accédez aux caractères d’une chaîne comme un élément d’une liste 8](#_Toc211954391)

[II. 5. 3) Modifiez les éléments d’une liste 8](#_Toc211954392)

[II. 5. 4) Ajoutez, retirez et triez les listes 9](#_Toc211954393)

[II. 6) Les tuples 10](#_Toc211954394)

[II. 7) Les dictionnaires 10](#_Toc211954395)

[II. 7. 1) Déclaration et initialisation 11](#_Toc211954396)

[II. 7. 2) Manipulation à une valeur 11](#_Toc211954397)

[II. 8) Les raccourcis 12](#_Toc211954398)

[II. 9) Les commentaires 12](#_Toc211954399)

[III) La logique du programme 13](#_Toc211954400)

[III. 1) Affichage 13](#_Toc211954401)

[III. 2) Saisie de valeurs par l’utilisateur 13](#_Toc211954402)

[III. 3) Les conditions 13](#_Toc211954403)

[III. 4) Les boucles 15](#_Toc211954404)

[III. 4. 1) Le FOR 15](#_Toc211954405)

[III. 4. 2) Le WHILE 16](#_Toc211954406)

[III. 5) Les fonctions 17](#_Toc211954407)

[III. 5. 1) Les fonctions sans paramètres 17](#_Toc211954408)

[III. 5. 2) Les fonctions avec paramètres 17](#_Toc211954409)

[III. 5. 3) Les fonctions avec retour 18](#_Toc211954410)

[III. 6) La gestion des exceptions 19](#_Toc211954411)

[III. 7) Les modules et packages 20](#_Toc211954412)

[III. 7. 1) Les modules 20](#_Toc211954413)

[III. 7. 2) Les packages 20](#_Toc211954414)

# Paramétrer votre environnement

## Installer VS Code

Télécharger le fichier d’installation sur <https://code.visualstudio.com/>

Lancer l’installation

Cocher les cases suivantes, pour plus de facilité dans les manipulations régulières.



## Paramétrez votre environnement pour python

Pour faciliter le développement en Python dans Visual Studio Code, nous vous recommandons d'installer quelques **extensions** utiles. Par exemple, l'extension 'Python' développée par Microsoft offre un large éventail de fonctionnalités, notamment la coloration syntaxique, l'autocomplétion intelligente, le débogage interactif, et bien plus encore. L'installation de ces extensions peut grandement améliorer votre productivité et votre confort lors de l'écriture de code Python dans Visual Studio Code. Vous pouvez installer ces extensions directement depuis la vue des extensions de Visual Studio Code.

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## Installer python sur votre ordinateur

Télécharger le fichier d’installation sur [python.org](https://www.python.org/downloads/) et installer la version correspondante à votre système

Si vous utilisez Windows, pensez à bien cocher la case "Add to path" pour ajouter Python aux variables d'environnement.

## Exécutez votre premier programme Python

Maintenant que vous avez installé Python, préparez-vous à exécuter votre premier code Python !

* Ouvrez l’invite de commande
* Écrivez python , et vous devriez voir des informations à propos de Python apparaître avec >>> indiquant où écrire le code.
* Ensuite, écrivez print("hello, world!") .
* Appuyez sur Entrée pour voir le résultat.

Une image contenant texte, capture d’écran, Police

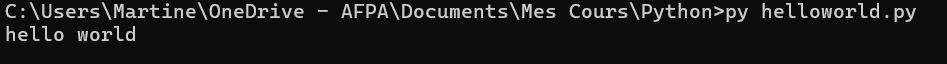
Le contenu généré par l’IA peut être incorrect.

Comme vous l’avez vu, vous pouvez exécuter chaque ligne de code Python directement dans le terminal. Vous pouvez aussi écrire du code dans un **éditeur de texte** séparé, puis l’exécuter en utilisant les commandes du terminal.

## Ecrire le programme dans un fichier et l’exécuter

* Dans l'éditeur, écrivez   print("Hello, world")  .
* Sauvegardez le fichier (par exemple, sous le nom helloworld.py).
* Ouvrez l’invite de commande dans le dossier où est stocké le fichier.
* Ecrivez "py" et le nom du fichier que vous avez créé :  py helloworld.py  .
* Appuyez sur Entrée pour voir le résultat:   Hello, world  .

Et voilà : code exécuté !



Exécution depuis VS Code

Cliquer droit dans l’édition du fichier et choisir Run Code

Voir le résultat dans la fenêtre Sortie

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

# Les bases de python

**Remarques générales sur Python**

* **Python est sensible à l’indentation** (les blocs sont définis par des indentations, pas des accolades ou des mots clés comme "début"/"fin").
* **Les instructions de contrôle (if, for, while, etc.) se terminent par :**.
* **Les commentaires commencent par #**.
* Pas besoin de déclarer les types de variables (int, string, etc.) : Python devine automatiquement.

## Les types

Les types de données les plus simples, ou *primitifs*, utilisés dans Python, sont :

* les entiers (*Integers*, en anglais) ;
* les virgules flottantes (*Floats*, en anglais) ;
* les chaînes de caractères (*Strings*, en anglais) ;
* les booléens.

N’oubliez pas qu’en anglais, les virgules deviennent des points ! Par exemple, si l’on écrit 12,4 en français, dans Python on écrit 12.4.

'912'  et   "912"  sont des chaînes de caractères

912  est un nombre entier

Les variables sont typées en fonction de leur contenu.

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

However the language includes a syntax for type annotations, which allow you to specify the expected type of a variable. This can be useful for improving code readability, documentation, and type checking.

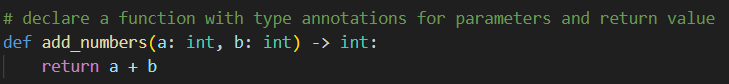
Here's an example of how you can use type annotations in Python:

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

Note that type annotations are optional, and Python will still work even if you do not use them. However, using type annotations can help catch certain types of errors at compile time, rather than at runtime.

Il est possible de caster une variable dans un type différent avec str(), int(), float()

## Les entrées sorties

Pour afficher à l’écran Print()

Pour acquérir une saisie utilisateur Input()

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Les caractères spéciaux

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

## Les variables

Une variable dans Python est constituée de trois éléments :

1. Son nom.
2. Son type.
3. Sa valeur.

Il n’existe pas de réelle différence entre les variables et les constantes en python, on gardera la convention de nommage commune à tous les langages, les constantes sont nommées avec des majuscules uniquement

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

## Les affectations et comparaisons

= sert à affecter une valeur

== sert à comparer 2 valeurs

Une image contenant Police, capture d’écran, texte, Graphique

Le contenu généré par l’IA peut être incorrect.

On peut comparer directement 3 valeurs en les imbriquant

 équivaut à 

On peut affecter avec des coefficients

 donne 

Affectation avec conditions intégrées

Une image contenant Police, texte, capture d’écran, typographie

Le contenu généré par l’IA peut être incorrect.

## Les listes

Une liste permet d’y enregistrer une collection d’objets auxquels vous voulez accéder plus tard.

Dans Python, on utilise des crochets  [ ]  pour indiquer une liste. Le code suivant crée une liste de différentes plateformes de réseaux sociaux, et la sauvegarde dans une variable appelée  plateformes\_sociales  .



### Accédez aux éléments d’une liste

Pour accéder aux éléments d’une liste, on utilise un *indice*. Chaque élément a un indice qui lui correspond, selon sa position dans la liste. Vous obtenez la valeur de cet indice avec la syntaxe suivante :  liste[indice]  . Elle vous renverra la valeur de la liste qui est à la position de l’indice.

L’important est de noter que dans la plupart des langages de programmation, y compris Python, l’indice commence à 0, pas 1.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

### **Accédez aux caractères d’une chaîne comme un élément d’une liste**

Les indices fonctionnent aussi avec les chaînes de caractères ! En fait, les chaînes de caractères sont juste des listes de caractères. Chaque caractère correspond à un indice qui va de zéro à la longueur de la chaîne.

Par exemple, dans la chaîne  langage  = "PYTHON"  ,  langage[2]  vous renverra  "T" . Tout simplement parce que l’indice 2 dans le mot "PYTHON" est le "T". Ou bien, avec l’indice de la position inverse, vous devez utiliser  langage[-4]  pour accéder à  "T"  .

Une image contenant texte, Police, diagramme, capture d’écran

Le contenu généré par l’IA peut être incorrect.

### **Modifiez les éléments d’une liste**

Pour modifier une case d'une liste, il suffit d'utiliser l'indice de la case que l'on souhaite modifier, et d'y affecter la nouvelle valeur à l'aide de l'opérateur d'affectation (=), tout comme pour une variable.

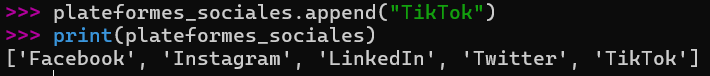
Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

### **Ajoutez, retirez et triez les listes**

Dans Python, c’est très simple de réaliser toutes sortes d’opérations avec les listes. Au lieu de créer une toute nouvelle liste à chaque fois que vous voulez ajouter, retirer ou trier des éléments, vous pouvez faire appel à une **méthode de liste**. Nous allons voir les méthodes plus tard, mais pour l’instant, vous devez juste savoir qu’une méthode est une façon de réaliser une opération spécifique sur un élément.

Par exemple, pour ajouter une plateforme de réseau social à la fin de la liste existante, vous pouvez utiliser la méthode  append()  :



Pour retirer un élément spécifique d’une liste, vous pouvez utiliser la méthode  remove()

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

remove()  retire uniquement la première instance du terme que vous saisissez.

Pour connaître la longueur de la liste, utilisez la fonction len()  .

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

La dernière méthode que nous allons voir est  sort()  . Elle trie les éléments de la liste. Le tri se fait alphabétiquement pour les listes de chaînes, et numériquement pour les listes de nombres.

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

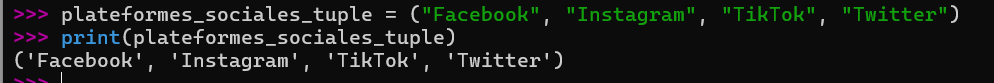
Il y a beaucoup d’autres méthodes que vous pouvez utiliser avec les listes. Vous les trouverez dans la [**documentation Python**](https://docs.python.org/fr/3/tutorial/datastructures.html).

Voici un avant-goût de ces méthodes :

|  |  |
| --- | --- |
| extend() | Ajoute plusieurs éléments à la fin de la liste. |
| insert() | Insère un élément à une position donnée dans la liste. |
| pop() | Supprime et renvoie l'élément à une position donnée de la liste, ou le dernier élément si aucun indice n'est spécifié. |
| index() | Renvoie la première occurrence de l'élément spécifié dans la liste. |
| count() | Renvoie le nombre d'occurrences de l'élément spécifié dans la liste. |
| reverse() | Inverse l'ordre des éléments de la liste. |

## Les tuples

Les tuples sont une autre structure de Python qui sert à enregistrer des données. Au lieu des crochets   [ ]  , ils se caractérisent par les parenthèses   ()  .



Beaucoup des propriétés des tuples sont similaires à celles des listes normales. Par exemple, les listes et les tuples utilisent tous deux les indices. La principale différence est que les tuples sont **immuables**, alors que les listes sont **modifiables**.

Trouvez un élément

Pour savoir si un élément est présent dans une liste ou un tuple, on peut utiliser l'opérateur in  . Cet opérateur retourne True  si l'élément est présent dans la séquence, et False  sinon.

Une image contenant texte, Police, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.

## Les dictionnaires

Un dictionnaire est une structure de données qui enregistre des données dans des **paires clés-valeurs**. Voici un exemple d’une clé et d’une valeur :

"responsable\_de\_campagne": "Jeanne d'Arc"

La **clé** est   "responsable\_de\_campagne"  et la **valeur** est   "Jeanne d'Arc"  .

Les dictionnaires sont indiqués par des accolades {}  au début et à la fin. Chaque paire clé-valeur comprend un deux-points  :  placé entre la clé et la valeur, et une virgule  ,   à la fin. Chaque dictionnaire doit être composé de clés uniques.

Dans le diagramme ci-dessous, le dictionnaire défini a trois éléments, et chaque élément est une paire clé-valeur.

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

### Déclaration et initialisation

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

### Manipulation à une valeur

* Pour accéder aux différentes valeurs, vous pouvez utiliser la clé pour chacune des paires clés-valeurs.

Une image contenant Police, Graphique, texte, graphisme

Le contenu généré par l’IA peut être incorrect.

* Pour ajouter une paire clé-valeur à un dictionnaire, ajoutez juste une nouvelle clé dans le dictionnaire existant. Si la clé existe déjà, vous l’écraserez en définissant une valeur.



* Pour supprimer une paire clé-valeur, vous pouvez utiliser le mot-clé  del  et la clé que vous voulez supprimer, ou encore la méthode   pop

Une image contenant texte, Police, capture d’écran, nombre

Le contenu généré par l’IA peut être incorrect. Une image contenant texte, Police, typographie, conception

Le contenu généré par l’IA peut être incorrect.

* Pour vérifier si une clé spécifique existe dans un dictionnaire, spécifiez la **clé** que vous voulez rechercher, écrivez le mot-clé in  et le nom de la variable du dictionnaire que vous examinez. Le résultat renvoie un booléen qui indique si la clé est dans ce dictionnaire.

Une image contenant Police, texte, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect. Une image contenant texte, Police, typographie, conception

Le contenu généré par l’IA peut être incorrect.

## Les raccourcis

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

## Les commentaires

# pour un commentaire sur une ligne

 pour les commentaires sur plusieurs lignes ou pour la doc autogénérée

Voici un exemple de docstring pour une fonction de calcul de la somme de deux nombres :

Une image contenant texte, capture d’écran, affichage, logiciel

Le contenu généré par l’IA peut être incorrect.

Lorsque vous appelez cette fonction dans l'interpréteur Python, vous pouvez utiliser la fonction  help()  pour afficher la docstring, voici le résultat :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

# La logique du programme

## Affichage

Simple : print

Avec variables

## Saisie de valeurs par l’utilisateur

## Les conditions

**Spécificités :**

* Le if est suivi de la **condition**, puis de :.
* Le **sinon si** se code elif (en un seul mot).
* Le **égal** se code avec ==, pas =.
* Chaque bloc est indenté (généralement 4 espaces).

Une image contenant texte, Police, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.

* Les opérateurs arithmétiques

Une image contenant texte, capture d’écran, Police, diagramme

Le contenu généré par l’IA peut être incorrect.

Attention l’opérateur de comparaison est ==

si vous utilisez = la valeur de la variable sera changée

* Les opérateurs booleén sont And , Or, ^(Xor binaire), Not

L'opérateur **and** n'évalue pas systématiquement les deux conditions. Il utilise ce qu'on appelle une **évaluation paresseuse** (**short-circuit evaluation** en anglais). Idem pour or

Si condensé ou si ternaire s’écrit 

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect. est équivalent à

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

* L’opérateur match case peut remplacer plusieurs si sinon si…

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect. Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Attention aux indentations

## Les boucles

### Le FOR

**Spécificités :**

* range(1, 6) signifie de 1 à 5 **inclus** (le 6 est exclu).
* Le for est suivi de in et d’un range(), puis de :.

Quelques exemples

🔢 range(stop) – Simple de 0 à stop - 1

**Une image contenant Police, texte, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect. Sortie :** 0 1 2 3 4

🔁 range(start, stop) – Départ personnalisé

** Sortie :** 3 4 5 6 7

🔂 range(start, stop, step) – Incrément personnalisé

**Une image contenant Police, texte, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect. Sortie :** 2 4 6 8

🔁 range avec un **pas négatif** (décroissant)

** Sortie :** 10 8 6 4 2

🔁 range(len(some\_list)) – Parcourir les **index** d'une liste

**Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect. Sortie :**

0 pomme

1 banane

2 kiwi

🔁 range utilisé pour répéter une action

Une image contenant Police, texte, capture d’écran, typographie

Le contenu généré par l’IA peut être incorrect. Utilisation du \_ pour indiquer qu'on **n'utilise pas la variable de boucle**.

### Le WHILE

La boucle FOR vous permet d’exécuter du code un nombre spécifique de fois, alors que la boucle WHILE continue de s’exécuter jusqu’à ce qu’une certaine condition soit remplie.

**Spécificités :**

* while suivi de la condition et :.
* Le test est fait avant chaque boucle.
* Ne pas oublier d’incrémenter pour éviter les boucles infinies

Une image contenant Police, capture d’écran, texte, Graphique

Le contenu généré par l’IA peut être incorrect.

* **Utilisez les mot-clés**break**et**continue

Il est courant d'utiliser des boucles pour répéter une série d'instructions plusieurs fois. Parfois, il peut être utile d'interrompre ou de sauter une itération dans la boucle. C'est là que les instructions break  et continue  entrent en jeu.

L'instruction break  permet de **sortir** d'une boucle prématurément. Elle est souvent utilisée lorsqu'une condition est rencontrée, et que l'on souhaite **arrêter** la boucle avant qu'elle ne se termine normalement.

Attention dans l’utilisation du break  
Il est à proscrire dans beaucoup de langages

## Les fonctions

La création d'une fonction se fait à l'aide du mot-clé  def  , suivi du nom de la fonction et des éventuels paramètres entre parenthèses.

### Les fonctions sans paramètres

Elle est assimilable à un **bloc de code** à lancer quand cela est nécessaire. Elle est utile pour **encapsuler** un bloc de code répétitif, et faciliter sa réutilisation dans différents endroits du programme.

Attention au : à la fin de la fonction

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

Cette fonction s'appelle  afficher\_message  et ne prend aucun paramètre en entrée, car elle est définie avec des parenthèses vides. Elle n'a pas de valeur de retour car elle se contente d'afficher un message à l'écran en utilisant la fonction  print()  .

Pour appeler cette fonction, il suffit d'utiliser **son nom** et les **parenthèses** vides, car il n’y a pas de paramètres



### Les fonctions avec paramètres

Les fonctions avec des **paramètres** permettent de transmettre des valeurs à la fonction. Les paramètres sont simplement listés entre parenthèses, séparés par des virgules.

Voici un exemple d'une fonction qui prend deux paramètres, un nom et un prénom, et qui les affiche ensuite.

Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

Pour appeler cette fonction, il faut préciser les valeurs à transmettre aux paramètres. Soit des valeurs directes soit des variables



Une image contenant Police, texte, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

Pour rendre un paramètre facultatif, on peut lui mettre une valeur par défaut

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

### Les fonctions avec retour

Une fonction avec une valeur de **retour** est une fonction qui peut prendre des **paramètres** et effectuer des opérations, mais qui renvoie également une valeur à la fin.

Une image contenant texte, Police, capture d’écran, nombre

Le contenu généré par l’IA peut être incorrect.



Utiliser le mot return pou retourner la valeur

Il est possible de retourner **plusieurs** valeurs en les séparant par des virgules dans l'instruction de retour de la fonction. Les valeurs retournées seront automatiquement regroupées dans un **tuple**.

Le passage des paramètres par valeur ou par référence est transparent, il dépend du caractère **mutable** (ex : liste, dictionnaire, objet custom) ou **immuable** (ex : int, float, str, tuple) du paramètre

## La gestion des exceptions

Lorsqu'un programme rencontre une erreur, cela peut le faire planter complètement, ce qui n'est pas souhaitable. Pour éviter cela, il est possible d'utiliser les **exceptions** pour gérer les erreurs. Les exceptions sont des erreurs qui peuvent être détectées et gérées par le programme. Si une exception est **levée**, le programme peut afficher un message d'erreur, puis continuer à s'exécuter normalement.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Dans ce code, le bloc  try  contient le code qui peut causer une erreur.

Si une erreur se produit, le bloc  except  est exécuté et affiche un message d'erreur personnalisé.

Avec finally et else

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

## Les modules et packages

### Les modules

Un module en Python est simplement **un fichier** contenant du code Python. Ce code peut contenir des fonctions, des classes, des variables, etc. Les modules permettent **d'organiser** votre code de manière logique, et de le réutiliser facilement.

Pour utiliser un module, vous pouvez l'importer dans votre code à l'aide du mot-clé import  . Par exemple, si vous avez un fichier nommé monModule.py  qui contient une fonction nommée maFonction  , vous pouvez l'importer ainsi :

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

Une fois le module importé dans votre nouveau fichier grâce à la commande import  , vous pouvez désormais utiliser la fonction maFonction()  . Cependant, n'oubliez pas de spécifier le nom du module avant d'appeler la fonction.

Vous pouvez également importer des éléments **spécifiques** d'un module, en utilisant la syntaxe  from monModule import maFonction   . Cela vous permet d'utiliser directement la fonction maFonction  sans avoir à spécifier le nom du module.

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

Il est recommandé de placer toutes les déclarations d'importation en **début** de fichier Python, avant toute autre instruction

### Les packages

Un package en Python est simplement un **dossier** contenant un **ensemble de modules** Python. Les packages permettent **d'organiser** votre code en sous-dossiers, et de créer des hiérarchies de modules.

Pour créer un package, vous devez simplement créer un dossier contenant un fichier nommé \_\_init\_\_.py  . Ce fichier est utilisé pour initialiser le package, et peut contenir du code d'initialisation si nécessaire.

Pour utiliser un module d'un package, vous devez spécifier le **nom** du package dans l'import. Par exemple, si vous avez un package nommé monPackage  qui contient un module nommé monModule  , vous pouvez l'importer ainsi :

Une image contenant texte, Police, capture d’écran, Graphique

Le contenu généré par l’IA peut être incorrect.

De nombreux packages populaires sont disponibles sur des dépôts en ligne tels que **[PyPI](https://pypi.org/)**, et peuvent être **facilement** installés à l'aide d'un gestionnaire de packages tel que pip  .

Voici quelques exemples de packages populaires, et leurs fonctions :

* [**Requests**](https://fr.python-requests.org/en/latest/) : un package HTTP élégant et simple pour Python. Fréquemment utilisé pour les appels d’interface REST ;
* [**Beautiful Soup**](https://www.crummy.com/software/BeautifulSoup/bs4/doc/) (ressource en anglais) : un package pour récupérer des données de fichiers HTML et XML ;
* [**Pandas**](https://pandas.pydata.org/) (ressource en anglais) : un outil open source rapide, puissant et accessible pour l’analyse et la manipulation de données.

Des milliers de packages Python sont disponibles pour votre code !