

Y.Derfoufi : tresfacile.net

PYTHON EN

10 JOURS

3 ème édition

2026

Collection Très Facile !



www.tresfacile.net

Version Complète = 5 €

132 pages ---



**ORDER
NOW**



Y.Derfoufi : tresfacile.net

PYTHON EN

10 JOURS

3 ème édition

2026

Collection Très Facile !



www.tresfacile.net

Python En 10 Jours : 3^{ème} édition

Younes Derfoufi

Maître de conférences en Mathématiques et Informatique

CMEF – Centre des **M**étiers de l'**É**ducation et de la **F**ormation

Janvier 2026

Site Web : www.tresfacile.net

Table des matières

1	Jour 1 : Installation des outils	17
1.1	Installation de Python : Une solution tout en un	18
1.2	Premier programme Python à l'aide de l'IDE intégré IDLE	19
1.3	Premier programme Python à l'aide de l'IDE Thonny	22
1.4	Usage des compilateurs Python en ligne	24
2	Jour 2 : Les variables et opérateurs	27
2.1	les variables en Python	27
2.2	Résumé des types de variables Python	32
2.3	Les commentaires en Python	32
2.4	Les opérateurs en Python	34
2.5	Les fonctions en Python	38
2.6	Lire une saisie clavier avec la fonction <code>input()</code>	40
3	Structures de contrôles	43
3.1	La structure sélective <code>if . else .</code>	43
3.2	L'instruction <code>elif</code>	44
3.3	La structure répétitive <code>for</code>	45
3.4	La structure répétitive <code>while</code>	45
3.5	Les exceptions en Python (<code>Try Except</code>)	46
4	Jour 4 : Les chaînes de caractères	49
4.1	Définir une chaîne de caractère en Python	49
4.2	Longueur d'une chaîne de caractères	51
4.3	Accéder aux éléments d'une chaîne de caractères	52

4.4	Opération sur les chaînes de caractères	53
4.5	Les méthodes de chaînes de caractères	54
5	Jour 5 : Les listes en Python	63
5.1	Création d'une liste en Python	63
5.2	Accès aux éléments d'une liste.	65
5.3	Changer la valeur d'un élément de la liste	65
5.4	Parcourir les éléments d'une liste Python	66
5.5	Longueur d'une liste Python	66
5.6	Ajouter ou supprimer des éléments à la liste	67
5.7	Les différentes méthodes destinées aux listes	71
5.8	Tableau récapitulatif	75
6	Jour 6 : Les tuples	77
6.1	Définir un tuple en Python	77
6.2	Accéder aux éléments d'un tuple	77
6.3	Boucle à travers un tuple	78
6.4	Vérifier si un élément existe dans un tuple	78
6.5	Longueur d'un tuple	79
6.6	Ajout ou suppression d'éléments impossible à un tuple	79
6.7	Suppression d'un tuple	79
6.8	Création d'un tuple à l'aide du constructeur	80
6.9	Méthodes associées à un tuple	80
7	Jour 7 Dictionnaires et Ensembles	81
7.1	Les dictionnaires en Python	81
7.2	Les ensembles Python (Python sets)	86
7.3	Fonction Lambda En Python	90
7.4	Compréhension des listes en Python	93
8	Jour 8 : Programmation orientée objet	95
8.1	Les avantages de la POO	95
8.2	Le concept de POO en Python	96
8.3	Terminologie de la POO	97
8.4	Les classes en Python	98
8.5	Les méthodes d'instances en Python	99
8.6	Les méthodes de classes en Python	100
8.7	Attributs d'instances et attributs de classes	101

8.8	Les méthodes statiques	101
8.9	Héritage en Python	102
9	Jour 9 : Les modules en Python	105
9.1	Introduction	105
9.2	Créer votre propre module	106
9.3	les modules standards en Python	107
9.4	Exemple de modules standards	108
10	Jour 10 : Les fichiers en Python	115
10.1	Mode d'ouverture d'un fichier	115
10.2	Ouverture et lecture d'un fichier	116
10.3	Lecture et écriture à une position donnée.	122
10.4	Ouverture en mode écriture	123
10.5	Récapitulatif des méthodes associées aux fichiers	125

A propos de l'auteur

Formateur et **enseignant-chercheur** au **Centre des Métiers de l'Éducation** et de la **Formation**, l'auteur dispose de plus de trente années d'expérience consacrées à la formation des **enseignants stagiaires** en mathématiques et en informatique; **agrégé** de mathématiques et **docteur en topologie robotique**, il a su conjuguer recherche scientifique, rigueur académique et pédagogie appliquée tout au long de sa carrière; il a enseigné de nombreuses disciplines en informatique, notamment le **développement web** (HTML, CSS, Bootstrap, JavaScript), la **programmation et le backend** (PHP, programmation orientée objet en PHP, Laravel), les **bases de données** (MySQL), l'**algorithmique** et la **programmation Python** du niveau débutant jusqu'au niveau expert, le framework Django, ainsi que plusieurs systèmes de gestion de contenu tels que Joomla, WordPress, Moodle et PHPBB; parallèlement, il a assuré des enseignements approfondis en mathématiques aux niveaux classes préparatoires et licence, couvrant l'algèbre, la topologie générale, la topologie des espaces métriques et des espaces normés, ainsi que le calcul différentiel, avec une approche pédagogique fondée sur la clarté, la structuration des savoirs et la mise en pratique au service de la formation et de l'excellence académique.

Droits d'auteur

I YOUNES DERFOUFI. Tous droits réservés. Aucune partie de cette oeuvre, ne peut être reproduite, distribuée, ou transmise sous quelque forme que ce soit, sans l'autorisation écrite préalable de l'auteur, sauf dans les cas prévus par la loi sur le droit d'auteur. Cette oeuvre est protégée par les lois internationales sur le droit d'auteur et les droits moraux de l'auteur. Toute violation des droits d'auteur, y compris la reproduction non autorisée, la distribution ou la modification de cette oeuvre, peut entraîner des poursuites judiciaires et des réclamations pour dommages et intérêts. L'auteur détient tous les droits exclusifs sur cette oeuvre, y compris les droits de reproduction, de distribution, de représentation publique et de modification. L'autorisation de l'auteur doit être obtenue avant toute utilisation ou exploitation de cette oeuvre, qu'elle soit commerciale ou non. Pour toute demande d'autorisation ou pour plus d'informations sur l'utilisation de cette oeuvre, veuillez contacter l'auteur à l'adresse suivante : yderf64@gmail.com. L'auteur se réserve le droit d'apporter des modifications à ce texte de copyright à tout moment sans préavis. Veuillez consulter régulièrement cette section pour vous assurer d'être informé des dernières mises à jour. Merci de respecter les droits d'auteur et de faire preuve de diligence dans l'utilisation de cette oeuvre. Votre coopération est grandement appréciée.

À Qui S'Adresse Ce Livre ?



Python En 10 Jours

L'ouvrage "**Python En 10 Jours**", est un guide pratique conçu spécifiquement pour les débutants passionnés par l'apprentissage rapide et efficace de Python. Cet ouvrage s'adresse à un public varié, allant des élèves avides de nouvelles compétences aux enseignants cherchant à intégrer la programmation dans leurs cours, en passant par les étudiants désireux de renforcer leur bagage informatique.

En **dix jours**, nous vous guiderons à travers les fondements de **Python**, des installations initiales aux concepts avancés tels que la programmation orientée objet et la manipulation de fichiers.

Que vous soyez débutant absolu ou que vous ayez déjà une expérience en programmation, ce livre vous accompagnera pas à pas dans votre voyage pour devenir un développeur Python compétent. Chaque jour, vous explorerez un aspect clé du langage, consoliderez vos connaissances par des exemples pratiques, et vous serez prêt à passer au niveau suivant à la fin de chaque chapitre.

Préparez-vous à plonger dans le monde fascinant de Python, et d'ici la fin de ces dix jours, vous serez équipé pour créer des applications, résoudre des problèmes complexes et exploiter tout le potentiel de ce langage de programmation polyvalent.

Jour 1 : Installation des outils

Bienvenue dans votre première journée d'exploration de Python ! Avant de plonger dans le code, commençons par l'essentiel : l'installation des outils nécessaires pour développer en Python. Cette première étape est cruciale pour vous assurer que votre environnement de développement est configuré correctement. Préparez-vous à découvrir les étapes nécessaires pour installer Python et les outils associés sur votre système.

Jour 2 : Les variables et opérateurs

Maintenant que votre environnement est prêt, passons à l'action ! Aujourd'hui, vous allez plonger dans le monde des variables et des opérateurs en Python. Comprenez comment stocker des données dans des variables, manipuler ces données avec des opérateurs, et commencez à construire les fondations de vos programmes Python.

[.]

Continuez à explorer chaque jour les différents aspects de Python, en apprenant progressivement et en appliquant vos connaissances nouvellement acquises. Bon voyage dans le monde de Python !

Apreçu des extraits du code



Caractéristiques du langage Python

Dans cet ouvrage les **extraits de code** utilisés ont été soigneusement élaborés à l'aide d'outils précieux et de **packages LaTeX** spécifiques facilitant ainsi l'apprentissage et rendant la lecture agréable . Cette approche a permis d'offrir aux lecteurs une expérience visuelle exceptionnelle, avec une **colorisation syntaxique** élégante et une mise en page de haute qualité ajoutant ainsi une dimension supplémentaire à l'expérience d'apprentissage.

Exemple. Extrait de code

```
1 # définir des variables string
2 chaine1 , chaine2 = "Hello" , " World!"
3
4 # concaténation des deux chaines
5 concatenation = chaine1 + chaine2
6
7 print(concatenation) # output : 'Hello World!'
```

Introduction

A propos du langage Python

Python est un **langage de programmation polyvalent** et **populaire**, largement utilisé dans divers domaines : **développement web**, **applications desktop**, **analyse des données**, **intelligence artificielle**, et bien plus encore. Il se distingue par sa **syntaxe simple et lisible**, ce qui en fait un **excellent choix** pour les **débutants en programmation**, tout en offrant une puissance et une flexibilité considérables pour les développeurs.

Caractéristiques principales de Python

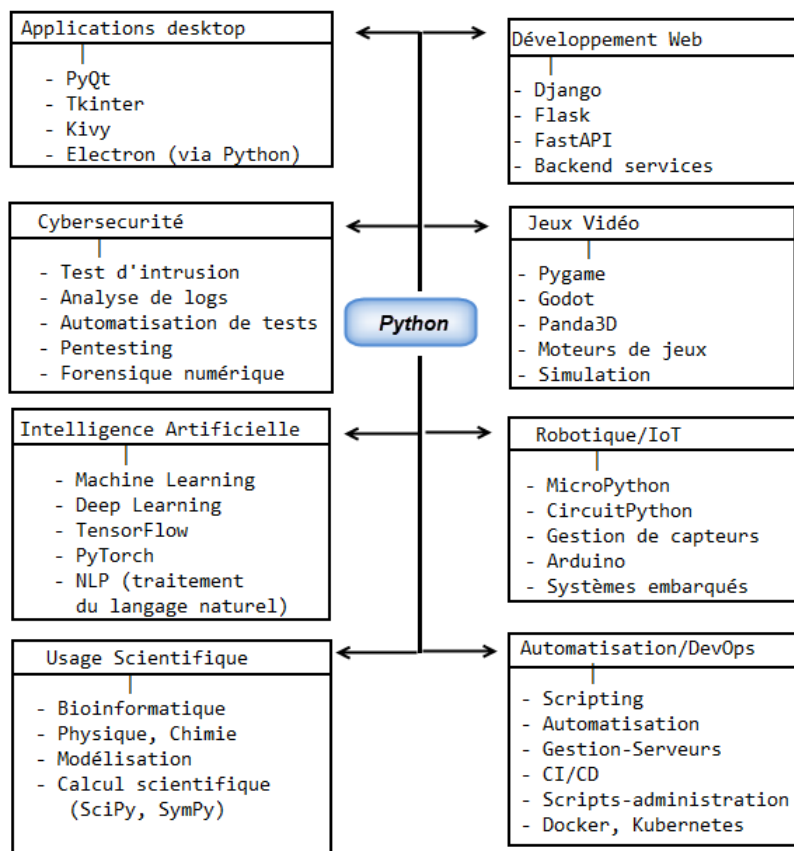
Le **langage Python** est doté de **nombreuses caractéristiques** faisant de lui un **excellent et meilleur langage**



Caractéristiques du langage Python

1. **Syntaxe Lisible**
2. **Un grand nombre de Bibliothèques Standards**
3. **Facilité d'Apprentissage**
4. **Dynamiquement Typé**
5. **Orientation Objet**
6. **Portabilité**
7. **Diversité des Applications**

Les différents usages du langage Python



Chapitre 1

Jour 1 : Installation des outils



Bienvenu dans cette **première journée** de découverte de **Python** ! Avant de nous plonger dans le **code**, démarrons par l'essentiel : **l'installation des outils indispensables au développement en Python**. Cette étape inaugurale revêt une importance cruciale pour garantir une **configuration correcte** de votre **environnement de développement**. Préparez-vous à suivre les étapes nécessaires à **l'installation de Python** et des outils associés sur votre système.

1.1 Installation de Python : Une solution tout en un

Pour configurer les **outils de développement en Python**, plusieurs options s'offrent à vous. Cependant, nous vous recommandons la solution la plus efficace et rapide : l'utilisation de la **distribution PySchool**, un package gratuit et open source. **PySchool** est équipée d'une gamme étendue d'outils visant à simplifier le processus de développement en **Python**, offrant notamment :



Composants de PySchool

1. Une version du langage **Python**
2. **Des Ide intégrés (éditeurs de code)** : IDLE, Thonny Ide, VSCode, Jupyter Notebook.
3. **Qt Designer** : Outil de réalisation d'interfaces graphiques sans code
4. **Bibliothèques** : La distribution PySchool est dotée des principales bibliothèques nécessaire au **machine learning** et **intelligence artificielle**.
5. Ainsi que de nombreux autres outils que vous allez découvrir !

Lien du téléchargement du package **PySchool** :

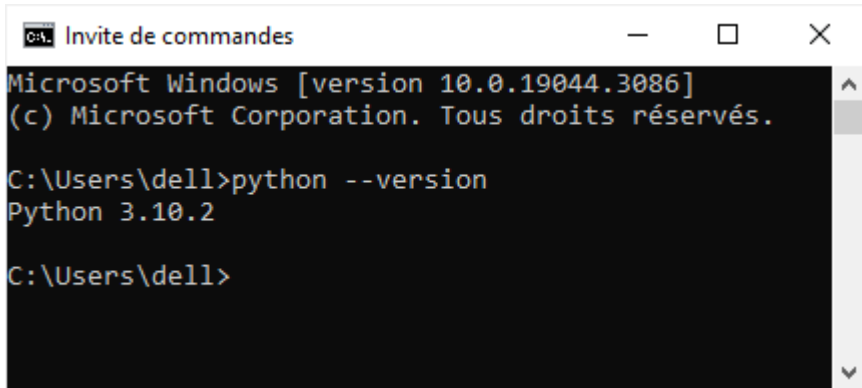
<https://sourceforge.net/projects/pyschool3/>



L'installation se déroule d'une façon très intuitive. Suivez alors les

1.2. PREMIER PROGRAMME PYTHON À L'AIDE DE L'IDE INTÉGRÉ IDLE19

étapes d'installation jusqu'à la fin et vérifiez ensuite cette dernière à l'aide de l'invite de commande en tapant : **python - -version**



```
Microsoft Windows [version 10.0.19044.3086]
(c) Microsoft Corporation. Tous droits réservés.

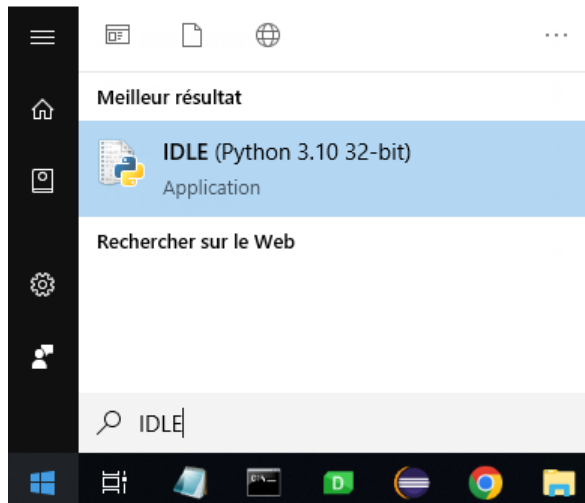
C:\Users\dell>python --version
Python 3.10.2

C:\Users\dell>
```

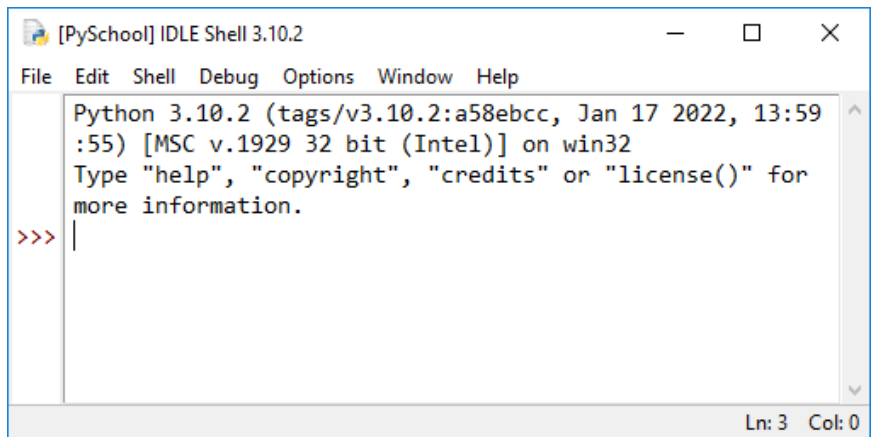
L'invite de commande vous montre ainsi, la version de Python que vous venez d'installer : **Python 3.10.2** dans notre cas.

1.2 Premier programme Python à l'aide de l'IDE intégré IDLE

Python est doté par défaut d'un IDE nommé **IDLE**, pour le lancer, il suffit de taper **IDLE** sur la zone rechercher du menu démarrer :



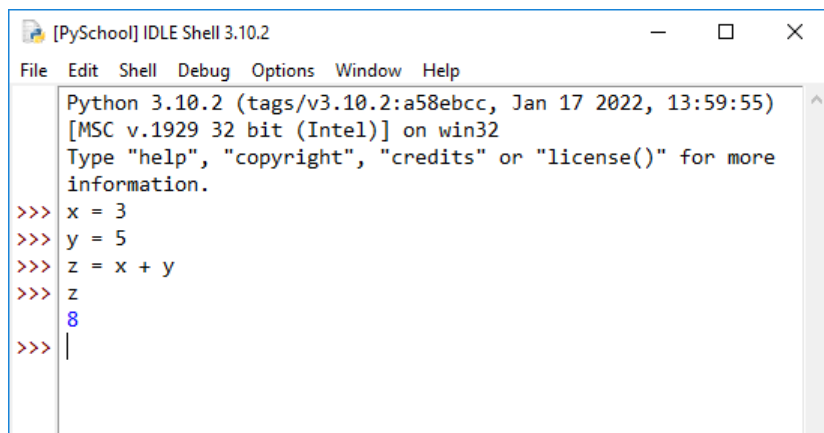
Une fois lancé, vous obtenez la vue suivante :



Comme vous le constatez, il s'agit d'un IDE très minimaliste. Amusons nous maintenant à faire quelques essais :

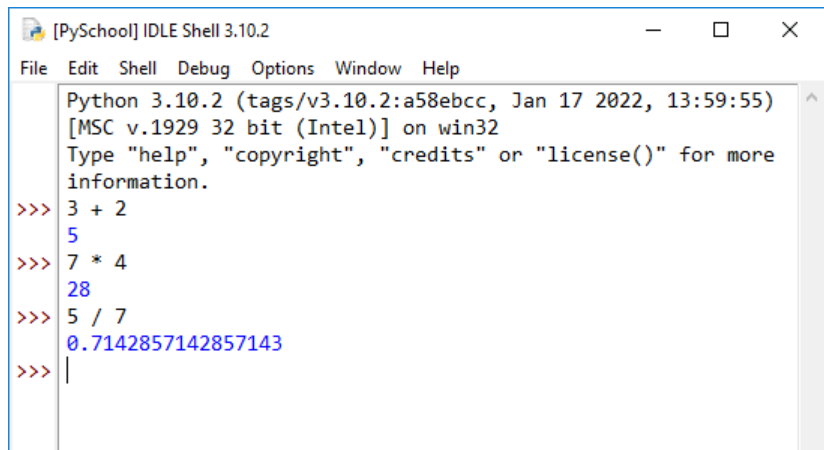
1.2. PREMIER PROGRAMME PYTHON À L'AIDE DE L'IDE INTÉGRÉ IDLE21

- A titre d'exemple on va définir **deux variables** **x** et **y** du type **entier** et on affiche leur somme **z = x + y** :



```
[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55)
[MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> x = 3
>>> y = 5
>>> z = x + y
>>> z
8
>>> |
```

- L'outil IDLE peut aussi jouer le rôle d'une calculatrice :

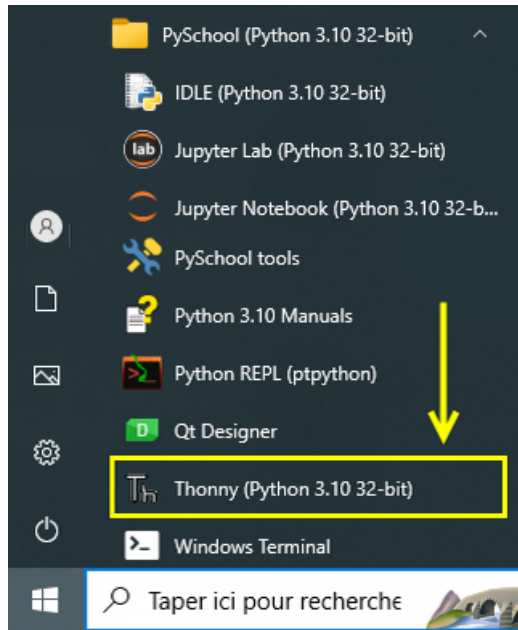


```
[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55)
[MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> 3 + 2
5
>>> 7 * 4
28
>>> 5 / 7
0.7142857142857143
>>> |
```

1.3 Premier programme Python à l'aide de l'IDE Thonny

Comme nous l'avons déjà mentionné ci-dessus, la distribution **PySchool** intègre par défaut un certain nombre d'outils comme : les IDE intégrés : **IDLE**, **Thonny Ide**, **VSCode**, **Jupyter Notebook**. Nous pouvons alors nous servir de l'Ide **Thonny** pour créer notre **premier programme Python**. Nous devons alors suivre les étapes suivantes :

Étape 1 : Lancer votre Ide **Thonny** depuis le menu Démarrer :



Étape 2 : Saisissez ensuite le code suivant et **enregistrer** le fichier :

```
1 s = "Hello World !"
2 print(s)
```

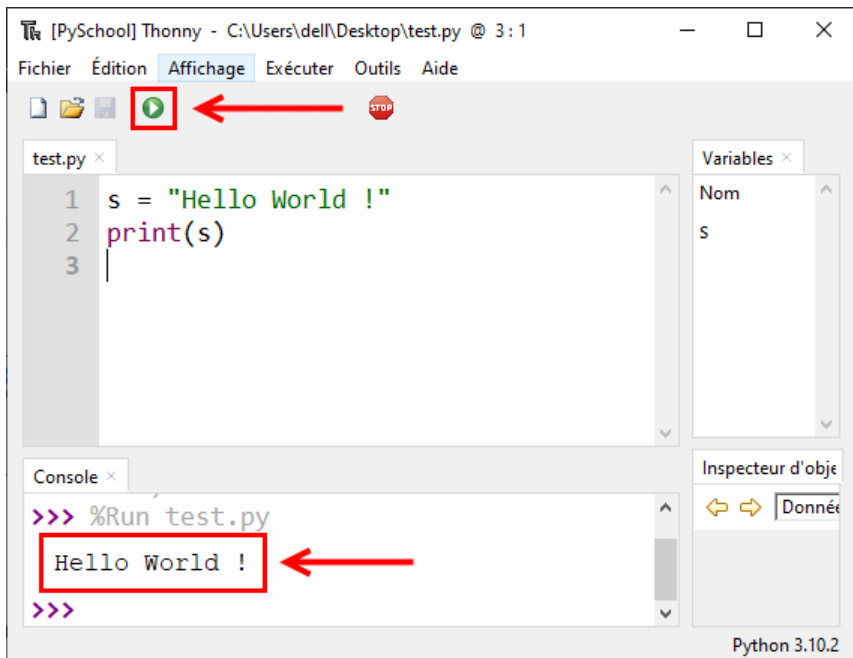
1.3. PREMIER PROGRAMME PYTHON À L'AIDE DE L'IDE THONNY23

Où :

1. Nous avons défini une **variable nommée s** ayant la valeur **'Hello World!'**
2. Nous utilisons l'instruction **print()** pour afficher la **valeur de la variable s**

Étape 3 : Exécution du code

Cliquez maintenant sur l'icône sous forme de petit triangle vert qui se trouve sur la barre d'outil :



Vous allez alors constater l'apparition du message **'Hello World!'** sur la console! Il s'agit de la valeur de la **variable s** affiché à l'aide de l'instruction **print(s)**.

1.4 Usage des compilateurs Python en ligne

Il faut savoir que l'usage des **IDE (environnements de développement intégrés)** comme **Thonny** ou **VS Code** n'est pas la seule solution pour exécuter un script Python. Si vous débutez ou si vous souhaitez tester rapidement un bout de code, vous pouvez utiliser des **compilateurs Python en ligne**. Ces outils offrent une solution pratique, rapide, et accessible **sans installation préalable** !

Avec l'avènement des **outils modernes**, il existe aujourd'hui de nombreux **compilateurs Python en ligne** qui permettent d'**exécuter du code directement depuis un navigateur** ! Ces plateformes sont idéales pour les débutants, les tests rapides ou même pour travailler sur des projets collaboratifs.

Parmi ces outils, je vous suggère [OnlineGDB](#), une plateforme particulièrement adaptée aux débutants et aux utilisateurs souhaitant disposer d'une interface simple mais puissante.

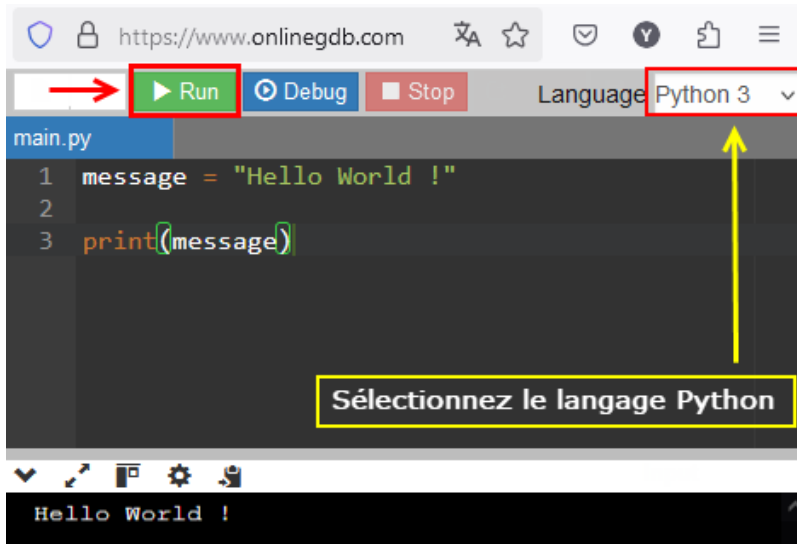
1.4.1 Pourquoi choisir OnlineGDB ?

1. **Interface intuitive** : Son environnement clair facilite la prise en main, même pour les novices.
2. **Débogueur intégré** : Il permet de visualiser pas à pas l'exécution du code, ce qui est très utile pour comprendre les erreurs ou la logique d'un programme.
3. **Exécution rapide** : Les résultats sont affichés instantanément, permettant une expérience fluide.
4. **Support multilingue** : En plus de Python, **OnlineGDB** prend en charge plusieurs autres langages, utile pour les curieux souhaitant explorer davantage.
5. **Accessibilité** : 100% en ligne, il fonctionne sur tous les appareils avec un navigateur, que ce soit un ordinateur, une tablette ou un smartphone.

1.4.2 Comment utiliser OnlineGDB ?

L'usage de la plateforme **OnlineGDB** est aussi simple qu'un clic sur un lien ! Aucune installation nécessaire ! À cet effet, il suffit de :

1. Se rendre sur le site d'**OnlineGDB** : <https://www.onlinegdb.com/>,
2. Choisir **Python** comme langage,
3. Commencer à écrire votre code
4. Exécuter votre code en cliquant sur le bouton **Run**



Chapitre 2

Jour 2 : Les variables et opérateurs

2.1 les variables en Python

Les variables jouent un rôle crucial en matière de stockage des résultats de nos calculs. En substance :



Variable Python

Une **variable** en **Python** est un **identifiant** qui permet de **stocker** une **valeur en mémoire**, de manière à pouvoir y accéder ultérieurement. Elle agit comme une **référence** à cette **donnée**, facilitant son utilisation tout au long du programme.

- Une **variable** peut être vue comme une **étiquette** attribuée à une **valeur**, indiquant comment l'appeler et la manipuler.
- L'**assignation** d'une **valeur** à une **variable** en **Python** se fait à l'aide du **symbole** `' = '`, où la valeur située à droite de l'opérateur `' = '` est stockée dans la variable située à gauche.
- Une variable peut **changer de valeur** pendant l'exécution du programme d'où la nomination **'variable'**

Exemple. On peut définir une variable `age` de la façon suivante :

```

1 age = 25
2
3 # afficher l'age actuel
4 print("L'age actuel est : " , age) # affiche : L'age
   actuel est : 25
5
6 # Après deux ans l'age deviendra
7 age = 27
8 print("L'age maintenant est : " , age) # affiche : L'
   age maintenant est : 27

```

- Vous avez remarqué sans doute que la **valeur d'age** a pris une valeur au début qui est 25
- Mais au cours du programme elle a **changé de valeur** en prenant la nouvelle qui est 27
- '**age**' est ainsi nommée variable car sa valeur peut "**varier**" au cours de l'exécution du programme.

Remarque. Les lignes précédées du symbole '**#**' sont **ignorées** lors de l'exécution du code. On les appelle des **commentaires** (voir paragraphe suivant).

```

1 mavariable = valeur

```

1. **mavariable** : ici on spécifie le nom de la variable qui doit être une séquence de caractères sans espace.
2. **valeur** : ici on spécifie la valeur de la variable qui peut revêtir la forme de n'importe quelle expression, comme évoqué précédemment. Contrairement à d'autres langages de programmation, **Python** n'exige pas de déclarer le **type d'une variable**. Une variable est créée au moment où vous lui affectez une valeur. Le

type sera détecté et assigné à la variable automatiquement (on dit alors que **Python** est un langage à **typage dynamique**).

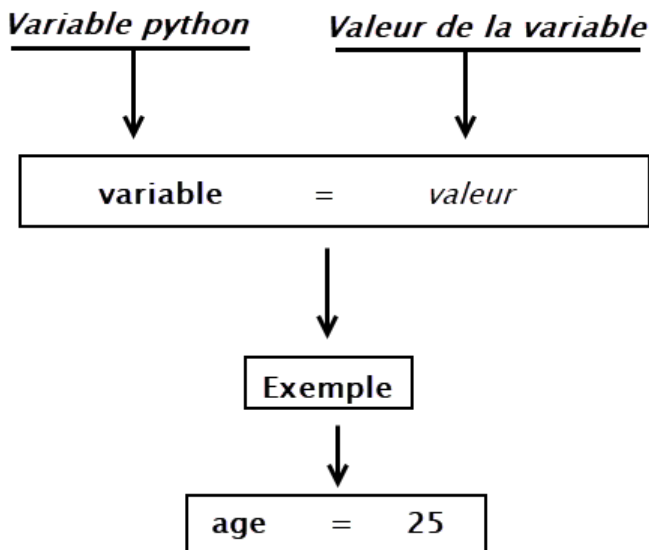
Les principaux types de variables en python sont :



Types de variables Python

1. Type **integer** ou **int** comme : **5** , **7** , **23** .
2. Type **float** ou variable du type **décimal** à virgule flottante : comme **12.5** , **6.75** , **1.25** .
3. Type **string** ou **chaîne de caractères** comme **s1 = 'Python'** **s2 = 'Java'**.
4. Type **booléen** qui prends uniquement deux valeurs **True** ou **False**

Voici une manière simple et explicite de représenter une **variable** :



Remarque. L'instruction `age = 25` définit une **variable** nommée '**age**' et lui affecte en même temps la **valeur** **25**.

Exemple. variables du type entier

```

1 # définir une variable du type entier
2 n = 11
3
4 # afficher la variable
5 print(n)
6 # affiche : 11

```

Après exécution ce programme affiche **11**. On peut aussi afficher un **texte explicatif** conjointement avec la variable **n** :

Exemple. afficher la variable avec un texte explicatif :

```

1 n = 11
2 print("La valeur de n est : " , n)

```

Le programme affiche à l'exécution :

La valeur de n est : 11

Remarque. Pour afficher le type d'une variable, on utilise l'instruction

type(nom_de_la_variable)

Exemple. affichage du type d'une variables

```

1 x = 13
2 y = 12.75
3 s = "Apprendre Python"
4 b = 10 < 3
5 print(type(x)) # affiche : <class 'int'>
6 print(type(y)) # affiche : <class 'float'>
7 print(type(s)) # affiche : <class 'str'>
8 print(type(b)) # affiche : <class 'bool'>

```

Version Complète = 5 €

132 pages ---



**ORDER
NOW**



Y.Derfoufi : tresfacile.net

PYTHON EN

10 JOURS

3 ème édition

2026

Collection Très Facile !



www.tresfacile.net