

Python En 10 Jours

Younes. Derfoufi
Enseignant de mathématique
et d'informatique
au (CRMEF) Oujda

Janvier 2024

Younes Derfoufi

PYTHON EN

10 JOURS

Collection **Très Facile!**

2024



Ouvrage destiné à ceux qui
souhaitent apprendre le
langage Python rapidement

Table des matières

1	Jour 1 : Installation des outils	9
1.1	Installation de Python : Une solution tout en un	10
1.2	Premier programme en Python	11
2	Jour 2 : Les variables et opérateurs	16
2.1	les variables en Python	16
2.2	Les commentaires en Python	18
2.3	Les opérateurs en Python	20
2.4	Les fonctions en Python	22
2.5	Lire une saisie clavier avec la fonction input()	23
3	Jour 3 : Structures de contrôles	25
3.1	La structure sélective if ... else	25
3.2	L'instruction elif	26
3.3	La structure répétitive for	26
3.4	La structure répétitive while	27
3.5	Les exceptions en Python (Try Except)	28
4	Jour 4 : Les chaînes de caractères	31
4.1	Définir une chaîne de caractère en Python	31
4.2	Longueur d'une chaîne de caractères	32
4.3	Accéder aux éléments d'une chaîne de caractères	32
4.4	Opération sur les chaînes de caractères	34
4.5	Les fonctions de chaînes de caractères en Python	35

5	Jour 5 : Les listes en Python	40
5.1	Création d'une liste en Python	40
5.2	Accès aux éléments d'une liste.	40
5.3	Changer la valeur d'un élément de la liste	41
5.4	Parcourir les éléments d'une liste Python	41
5.5	Longueur d'une liste Python	41
5.6	Ajouter ou supprimer des éléments à la liste	42
5.7	Les différentes méthodes destinées aux listes Python	46
6	Jour 6 : Les tuples	47
6.1	Définir un tuple en Python	47
6.2	Accéder aux éléments d'un tuple	47
6.3	Boucle à travers un tuple	48
6.4	Vérifier si un élément existe dans un tuple	48
6.5	Longueur d'un tuple	48
6.6	Ajout ou suppression d'éléments impossible à un tuple	49
6.7	Suppression d'un tuple	49
6.8	Création d'un tuple en utilisant le constructeur tuple()	50
6.9	Méthodes associées à un tuple	50
7	Jour 7 : Les dictionnaires et les ensembles	51
7.1	Les dictionnaires en Python	51
7.2	Les ensembles Python (Python sets)	56
7.3	Fonction Lambda En Python	60
7.4	Compréhension des listes en Python	63
8	Jour 8 : Programmation orientée objet POO	64
8.1	Le concept de POO en Python	64
8.2	Terminologie de la POO	65
8.3	Les classes en Python	66
8.4	Les méthodes d'instances en Python	67
8.5	Les méthodes de classes en Python	68
8.6	Attributs d'instances et attributs de classes	69
8.7	Les méthodes statiques	69
8.8	Héritage en Python	70

9 Jour 9 : Les modules en Python	73
9.1 Introduction	73
9.2 Créer votre propre module	74
9.3 les modules standards en Python	75
9.4 Exemple de modules standards	76
10 Jour 10 : Les fichiers en Python	82
10.1 Mode d'ouverture d'un fichier	82
10.2 Ouverture et lecture d'un fichier	83
10.3 Lecture et écriture à une position donnée à l'aide de la méthode seek()	87
10.4 Ouverture en mode écriture des fichiers en Python . . .	88
10.5 Récapitulatif des méthodes Python associées à un objet fichier avec description :	90

A propos de l'auteur

L'auteur, enseignant-chercheur docteur agrégé en topologie robotique, possède près de trente ans d'expérience dans l'éducation. Sa spécialité réside dans la didactique des mathématiques et l'enseignement informatique. Il a développé des méthodes pédagogiques pointues pour faciliter la compréhension des concepts mathématiques et enseigner efficacement l'informatique.

Son expertise a formé de nombreux enseignants, couvrant des sujets variés en mathématiques avancées et maîtrisant divers langages informatiques tels que HTML, CSS, JavaScript, PHP, MySQL, Python, et Java. En tant que défenseur des technologies de l'information dans l'éducation, il utilise des outils comme Geogebra, SPSS, Excel et des bibliothèques Python.

Sa passion pour l'enseignement se traduit par une contribution significative au développement des connaissances et des compétences essentielles chez les apprenants.

Droits d'auteur

I YOUNES DERFOUFI. Tous droits réservés. Aucune partie de cette uvre, 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 uvre 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 uvre, peut entraîner des poursuites judiciaires et des réclamations pour dommages et intérêts. L'auteur détient tous les droits exclusifs sur cette uvre, 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 uvre, qu'elle soit commerciale ou non. Pour toute demande d'autorisation ou pour plus d'informations sur l'utilisation de cette uvre, veuillez contacter l'auteur à l'adresse suivante : [adresse de contact de l'auteur]. 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 uvre. Votre coopération est grandement appréciée.

À Qui S'Adresse Ce Livre ?

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'installa-

tion 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 !

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**

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. La dernière version du langage Python
2. Des Ide intégrés : **IDLE**, **Thonny Ide**, **VSCode**, **Jupyter Notebook**...
3. L'outil graphique visuel **Qt Designer**
4. Les bibliothèques Python nécessaire au **machine learning** et **intelligence artificielle**.
5. Ainsi que de nombreux autres outils que vous allez découvrir !

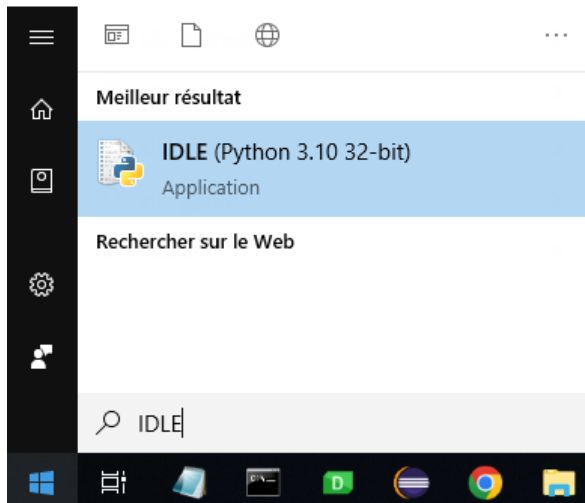
Lien du téléchargement du package PySchool : <https://sourceforge.net/projects/pyschool3/>



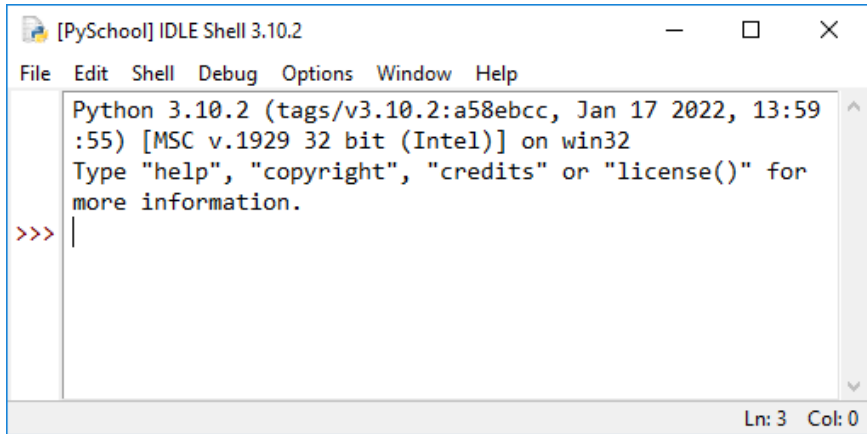
1.2 Premier programme en Python

1.2.1 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 recherche du menu démarrer :



Une fois lancé, vous obtenez la vue suivante :

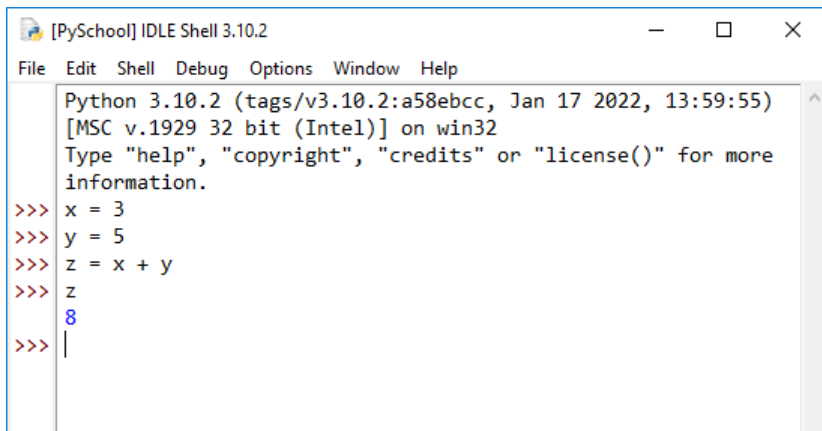


```
[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55) [MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 3 Col: 0

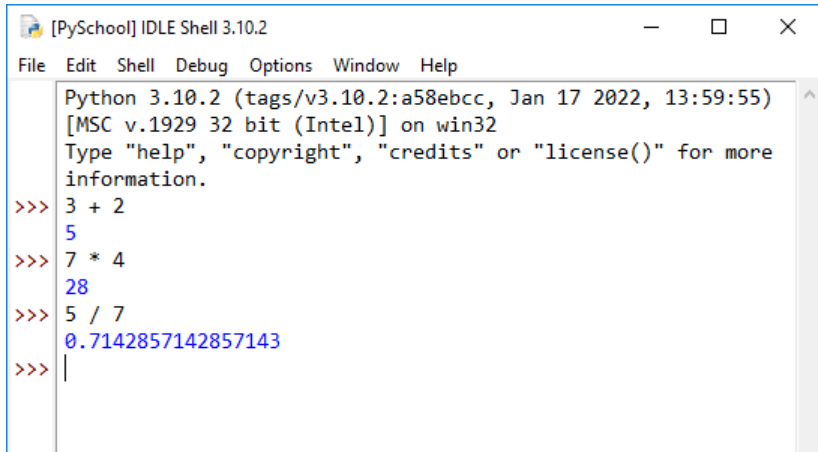
Comme vous le constatez, il s'agit d'un IDE très minimaliste. Amusons nous maintenant à faire quelques essais :

- À 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 :

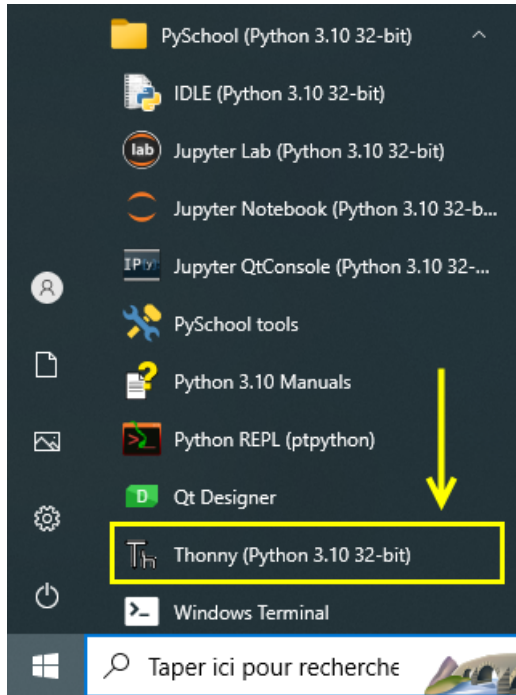


```
[PySchool] IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 13:59:55)
[MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> 3 + 2
5
>>> 7 * 4
28
>>> 5 / 7
0.7142857142857143
>>>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

1.2.2 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 :

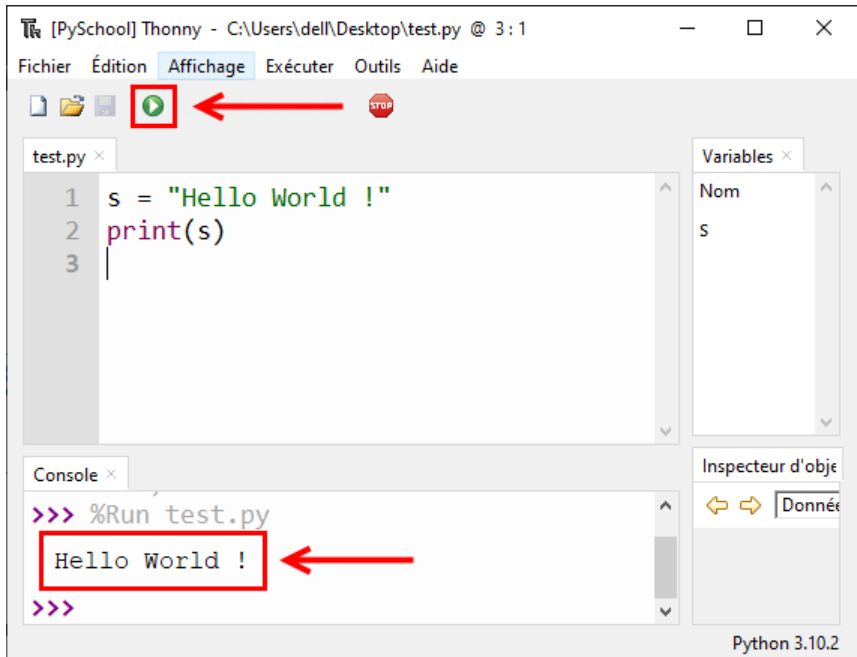
```
1 s = "Hello World !"  
2 print(s)
```

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)**.

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, une variable se résume à un nom attribué à une valeur, permettant ainsi à Python de la mémoriser (de ne pas la supprimer) et de la retrouver ultérieurement grâce à son nom. On peut concevoir la variable comme une sorte de marquage apposé sur notre valeur, indiquant comment elle est appelée. En Python, l'assignation :

```
1 mavariable = valeur
```

d'une valeur à une variable se réalise à l'aide de l'opérateur `=`. Sur la gauche,

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 automatiquement lors de l'assignation.

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**

Exemple. variables du type entier

```
1 n = 11
2 print(n)
3 # 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 :

```
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'>

```

Remarque. Une **variable** python possède toujours un **type**, même s'il est non déclarée. le type se définit au moment de l'introduction de la variable et peut être changé par la suite, ce qui justifie le dynamisme et la puissance du langage Python

Exemple. Type d'une variable.

```

1 x = 3 # x est de type int
2 print(x) # affiche 3
3
4 # changer le type de x en float
5 x = float(x) # x est maintenant du type float
6 print(x) # affiche 3.0

```

2.2 Les commentaires en Python

2.2.1 Qu'est-ce qu'un commentaire en Python ?

Les langages de programmation fournissent une méthode pour l'insertion de commentaires au sein du code afin de fournir des informations supplémentaires. Un commentaire n'est autre qu'un texte qui sera ignoré lors de l'exécution du programme. Les commentaires peuvent

être utilisés pour expliquer une partie compliquée d'un programme, ou pour mettre des indications dans le code, comme le code source, la version du langage ou script

2.2.2 Commentaire sur une seule ligne

En Python, nous insérons un commentaire sur une seule ligne avec le caractère `#` (dièse).

Syntaxe

```
1 # Ceci est un commentaire qui sera ignoré à l'exécution
```

Exemple. de commentaire en python

```
1 # définir une variable de type entier
2 n = 5
3 # Affichage de la variable
4 print ("La valeur de n est :", n)
```

Ce qui affiche à l'exécution :

La valeur de n est : 5

2.2.3 Commentaire sur plusieurs lignes

Si nous voulons insérer un commentaire sur plusieurs lignes en Python, nous utilisons le symbole des guillemets doubles

2.2.3.1 Syntaxe

```
1 """
2 Ceci est un commentaire
3 en plusieurs lignes
4 qui sera ignoré lors de l'exécution
5 """
```