

Windows

Documentations concernant Windows

- [Python](#)
 - [01 - Bases de Python](#)
- [VStream](#)
 - [01 - Installation de VStream sur Kodi](#)
- [Hyper-V](#)
 - [01 - Création d'un VM](#)
- [Réparation Windows Dador](#)

Python

01 - Bases de Python

A - Python

1 Qu'est ce que Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

2 Documentation

L'interpréteur Python et sa vaste bibliothèque standard sont disponibles librement, sous forme de sources ou de binaires, pour toutes les plateformes majeures depuis le site Internet

<https://www.python.org/> et peuvent être librement redistribués. Ce même site distribue et pointe vers des modules, des programmes et des outils tiers. Enfin, il constitue une source de documentation.

B - Terme de Base

1 Notion de Variable

Définition du mot ordinateur d'après "Le Petit Larousse" :

"Machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques."

Qui dit "traitement de l'information", dit donc données à manipuler. Un programme "passe" donc son temps à traiter des données. Pour pouvoir traiter ces données, l'ordinateur doit les ranger dans sa mémoire (RAM - Random Access Memory). La RAM se compose de cases dans lesquelles nous allons ranger ces données (une donnée dans une case). Chaque case a une adresse (ce qui permet au processeur de savoir où sont rangées les données).

Alors, qu'est-ce qu'une variable ?

Eh bien, c'est une petite information (une donnée) temporaire que l'on stocke dans une case de la RAM. On dit qu'elle est "variable", car c'est une valeur qui peut changer pendant le déroulement du programme.

Une variable est constituée de 2 choses :

- une valeur présente en mémoire (par exemple le nombre entier 5)
- un nom

On dira donc qu'une variable est l'association d'un nom et d'une valeur

```
i = 12
```

Grâce à cette ligne, nous avons défini une variable qui porte le nom i. Ce nom i est associé à la valeur 12.

Dans la partie "éditeur" de Visual Studio Code, saisissez le code suivant :

```
point_de_vie = 15  
print (point_de_vie)
```

```
1 point_de_vie = 15  
2 |
```

Après avoir exécuté le programme en cliquant sur le triangle vert, il est possible de connaître la valeur associée à un nom en utilisant la partie "console" de Visual Studio Code.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
PS C:\Documents\Site\Python\Graven> & C:/Users/Mattheo/AppData/Local/Programs/Python/Python312/python.exe "c:/Documents/Site/Python/Graven/point_de_vie = 15.py"  
15  
PS C:\Documents\Site\Python\Graven> |
```

Dans la suite la procédure sera toujours la même :

- Vous utiliserez la partie "éditeur" pour saisir votre programme
- vous utiliserez la partie "console" pour afficher la valeur associée à un nom

Les variables sont des espaces de stockages pour tout type d'élément comme des caractères ou des nombres, néanmoins on ne peut pas tout stocker. Le langage python fait la différence entre les types d'éléments.

Ainsi on obtient des variables de type « int() » qui stocke des **nombres entiers** donc « SANS virgule ».

```
1 variable_int = 1
2
```

Nous avons les « float() » qui stocke des **nombres réels** donc, « avec ou sans virgules ».

```
1 variable_float = 1.111
2
```

Et enfin les « strings » qui servent à **stocker les caractères**.

```
1 variable_string = 'Salut !'
2
```

C - Commande de Base

1 Print()

La commande « print() » sert à afficher sur l'écran aussi bien des valeurs que des phrases. Il est obligatoirement précédé de parenthèses pour lui indiquer quoi écrire.

```
test.py
1 print ('Salut, ça vas ?')
2
3 print (3*4)
4
5 print ('Mon calcul :', (3+4))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** [Open file in editor \(ctrl + click\)](#)

```
PS C:\Documents\Site\Python\Graven> & C:/Users/Mattheo/AppData/Local/Programs/Python/Python39-64/Python.exe test.py
Salut, ça vas ?
12
Mon calcul : 7
PS C:\Documents\Site\Python\Graven>
```

2 Input ()

La commande `input()` en Python est utilisée pour demander à l'utilisateur d'entrer des données à partir du clavier pendant l'exécution du programme. Elle affiche généralement un message à l'utilisateur, attend que celui-ci entre une valeur depuis le clavier, puis retourne cette valeur sous forme de chaîne de caractères.

Précédemment nous avons vu comment déclarer des variables, maintenant, nous allons voir comment récupérer une variable demandée à l'utilisateur.

```
1 variable_float=float(input("rentre une variable de type float"))
2
3 variable_string=str(input("rentre une variable de type float"))
4
5 variable_int=int(input("rentre une variable de type int"))
6 |
```

La commande « = » permet d'affecter une variable à une valeur.

Exemple :

```
test.py > ...
1 print(' ')
2 print(' ----- Bonjour ----- ')
3 print(' ')
4 a=float(input(" Veuillez entrer la valeur de a "))
5 d=2*a - 5
6 print(" La valeur de d vaut : ",d)
7 |
```

PS C:\Documents\Site\Python\Graven> & C:/Users/Mattheo/Applications/Python/Python38-64/python test.py

```
----- Bonjour -----

Veuillez entrer la valeur de a 3
La valeur de d vaut : 1.0
PS C:\Documents\Site\Python\Graven> |
```

Le programme demande à l'utilisateur la valeur de a, ici $a = 3$, calcule $d = 2a - 5$ et affiche la valeur de d.

3 La Fonction import ()

Il est possible d'utiliser des fonctions déjà existantes dans Python, pour cela, il faut les importer !

Au début du programme, il faut rentrer l'instruction : « `from math import *` ». Elle permet dans le programme d'utiliser l'instruction « `sqrt( )` » pour calculer la racine carrée mais aussi l'instruction « `pi` » pour π , « `floor` » pour partie entière, « `cos` » pour le cosinus d'un angle. On peut aussi utiliser « `from random import *` » qui permet d'utiliser l'instruction « `random()` » qui permet de générer un nombre aléatoire.

Exemple :

```
1 import random
```

4 Les Fonctions if, then, else, elif

En langage Python, une condition IF ("si" en français) permet d'exécuter une ou plusieurs instructions spécifiques seulement si une condition est vérifiée.

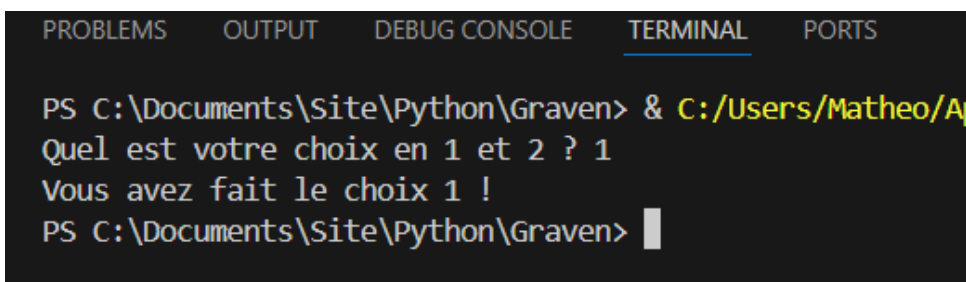
A la suite de l'instruction « if », il faut écrire le test voulu et faire suivre cela par « : » Une fois validée la ligne, le programme va faire un décalage et celui marque les lignes d'instruction qui seront effectuées si la condition est vérifiée. « == » permet de vérifier si deux valeurs sont égales et « != » permet de vérifier si elles sont différentes.

On peut rajouter une instruction si la condition n'est pas vérifiée avec « else : »

Enfin, l'instruction « elif » permet de rajouter une condition si la première n'est pas vérifiée.

Exemple :

```
1 choix1 = 1
2 choix2 = 2
3
4 choix = int(input('Quel est votre choix en 1 et 2 ? '))
5 if choix == choix1:
6     print ('Vous avez fait le choix 1 !')
7 elif choix == choix2:
8     print ('Vous avez fait le choix 2 !')
9 else:
10    print ('Votre choix est incorrect !')
11
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Documents\Site\Python\Graven> & C:/Users/Mattheo/Ap
Quel est votre choix en 1 et 2 ? 1
Vous avez fait le choix 1 !
PS C:\Documents\Site\Python\Graven> |
```

Ici, on peut voir que si l'utilisateur sélectionne 1, le print du choix 1 sera exécuté et si il sélectionne 2, le print du choix 2 sera exécuté. Si le résultat entré ne correspond à aucune variable, le print "incorrect" s'exécutera.

5 La Fonctions while

La fonction « while » permet de faire les instructions qui suivent tant que l'instruction précédée de « while » n'est pas vérifiée (boucle infini), par exemple, « while a!=b : » qui signifie tant que a est différent de b faire ...
Il ne faut pas oublier les « : » après l'instruction.

Exemple :

```
1  import random
2
3  def jeu1():
4      petit = int(input('Plus petit chiffre:'))
5      grand = int(input('Plus grand chiffre'))
6      chiffre = random.randint (petit, grand)
7      print (chiffre)
8
9      while True:
10         line = int(input('Chiffre aléatoire: '))
11         if line > chiffre:
12             print ('Trop grand')
13         elif line < chiffre:
14             print ('Trop petit')
15         elif line == chiffre:
16             break
17         print('Mauvais resultat')
18     print ('Bon résultat !')
19
```

On peut voir une boucle infinie dans la partie rouge, au début de ce script un chiffre aléatoire est défini et tant que l'utilisateur n'a pas trouvé ce chiffre la boucle continue.

VStream

01 - Installation de VStream sur Kodi

A - Kodi

VStream est une extension de Kodi, pour installer cette extension il faut d'abord installer et configurer Kodi.

1 Installation de Kodi

Pour installer Kodi, cliquer sur le lien ci-dessous.

<https://kodi.tv/download/>

Une fois sur le site, sélectionner la plateforme désiré, ici Windows, puis installer la version 64 bits.

Select Your Platform

Kodi has many flavours, our developers have worked hard to make it support a large range of devices and operating system. We provide final as well as developer builds for future final release versions. To get started, simply select the platform that you want to install it on.



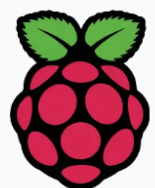
Windows



Linux



Android



Raspberry Pi



macOS



iOS



tvOS



If you need extra help, checkout our ["How To" Guide](#) for Windows.

Recommended

Prerelease

Development Builds

Kodi v20.1 (Nexus)

The installation of Kodi for Windows is straightforward and like most other Windows applications. Simply download the installer, run it, and follow the prompts.

Downloads

Installer (64BIT)

Installer (32BIT)

Windows Store

2 Configuration de Kodi

Langue

Après avoir installé Kodi, rendez-vous dans les paramètres.

Hyper-V

01 - Création d'un VM

A - Création VM

Hyper-V : Debian

Distribution : Debian 12.5

1 Paramètre de création

Pour commencer l'installation d'un VM sur Hyper-V, rendez-vous sur nouveau.

Nom

Commencé par remplir le nom de votre machine virtuel ainsi que l'emplacement dans lequel vous souhaité le situé.

Assistant Nouvel ordinateur virtuel

Spécifier le nom et l'emplacement

Avant de commencer

Spécifier le nom et l'emplacement

Spécifier la génération

Affecter la mémoire

Configurer la mise en réseau

Connecter un disque dur virtuel

Options d'installation

Résumé

Choisissez un nom et un emplacement pour cet ordinateur virtuel.

Le nom est affiché dans le Gestionnaire Hyper-V. Nous vous recommandons d'utiliser un nom qui vous permettra d'identifier facilement cet ordinateur virtuel, tel que le nom de la charge de travail ou du système d'exploitation invité.

Nom :

Vous pouvez créer un dossier ou utiliser un dossier existant pour stocker l'ordinateur virtuel. Si vous ne sélectionnez pas de dossier, l'ordinateur virtuel est stocké dans le dossier par défaut configuré pour ce serveur.

☐ Stocker l'ordinateur virtuel à un autre emplacement

Emplacement :

Parcourir...

 Si vous envisagez de créer des points de contrôle de cet ordinateur virtuel, choisissez un emplacement avec un espace libre suffisant. Les points de contrôle induisent les données des ordinateurs virtuels et peuvent nécessiter un espace considérable.

< Précédent

Suivant >

Terminer

Annuler

Génération

Vous devrez ensuite compléter la génération, la génération est un choisir en fonction du system que l'on veut mettre. Elles présentent chacune leur caractéristique. (la génération 1 est utilisé pour des system plus obsolète)

Voici un tableau qui pourrait vous aider à choisir votre génération.

Caractéristiques	Génération 1	Génération 2
Architecture du matériel	Architecture de PC traditionnelle	Architecture basée sur UEFI
Type de firmware	BIOS	UEFI avec support du Secure Boot
Architecture	32 bits	32 bits et 64 bits
Format de disque virtuel	VHD et VHDX	VHDX
Taille maximale du disque virtuel	Jusqu'à 2040 Go (VHD) et jusqu'à 2 To (VHDX)	Jusqu'à 64 To
Quantité processeurs virtuels	Jusqu'à 64 vCPU	Jusqu'à 240 vCPU
Quantité RAM	Jusqu'à 1 To	Jusqu'à 12 To
Démarrage réseau (PXE)	Nécessite une carte réseau virtuelle héritée	Support intégré dans la carte réseau virtuelle

Avertissement :

Une fois choisie, vous ne pourrez plus changer de génération.

Mémoire

Après avoir choisie votre génération, vous devrez compléter la quantité de RAM que vous voulez alloué à votre machine virtuel. Il existe une option qui permet d'allouer dynamiquement la mémoire en fonction des besoin de la machine.



Affecter la mémoire

Avant de commencer

Spécifier le nom et
l'emplacement

Spécifier la génération

Affecter la mémoire

Configurer la mise en réseau

Connecter un disque dur
virtuel

Options d'installation

Résumé

Spécifiez la quantité de mémoire à allouer à cet ordinateur virtuel. Vous pouvez spécifier une quantité comprise entre 32 Mo et 12582912 Mo. Pour améliorer les performances, spécifiez davantage que la quantité minimale recommandée pour le système d'exploitation.

Mémoire de démarrage : Mo

☐ Utiliser la mémoire dynamique pour cet ordinateur virtuel.

i Pour déterminer la quantité de mémoire à attribuer à un ordinateur virtuel, tenez compte de la façon dont vous envisagez d'utiliser l'ordinateur virtuel et du système d'exploitation qu'il exécutera.

< Précédent

Suivant >

Terminer

Annuler

Switch Virtuel

Il faut ensuite sélectionné un Switch Virtuel qui devras être créé séparément.



Configurer la mise en réseau

Avant de commencer

Spécifier le nom et
l'emplacement

Spécifier la génération

Affecter la mémoire

Configurer la mise en réseau

Connecter un disque dur
virtuel

Options d'installation

Résumé

Chaque nouvel ordinateur virtuel inclut une carte réseau. Vous pouvez configurer celle-ci de façon à utiliser un commutateur virtuel ou la laisser déconnectée.

Connexion : HPE Ethernet port 2 #Virtual



< Précédent

Suivant >

Terminer

Annuler

Espace Disque

Vous devrez maintenant définir l'espace que vous souhaitez allouer à votre VM.



Connecter un disque dur virtuel

Avant de commencer

Spécifier le nom et l'emplacement

Spécifier la génération

Affecter la mémoire

Configurer la mise en réseau

Connecter un disque dur virtuel

Options d'installation

Résumé

Un ordinateur virtuel requiert un espace de stockage pour l'installation d'un système d'exploitation. Vous pouvez spécifier le stockage dès maintenant ou le configurer ultérieurement en modifiant les propriétés de l'ordinateur virtuel.

☒ Créer un disque dur virtuel

Utilisez cette option pour créer un disque dur virtuel de taille dynamique (VHDX).

Nom :

Emplacement :

Taille : Go (Maximum : 64 To)

☐ Utiliser un disque dur virtuel existant

Utilisez cette option pour attacher un disque dur virtuel VHDX existant.

Emplacement :

☐ Attacher un disque dur virtuel ultérieurement

Utilisez cette option pour ignorer cette étape et attacher un disque dur virtuel existant ultérieurement.

< Précédent

Suivant >

Terminer

Annuler

ISO

Pour finir, vous devrez placer un fichier ISO afin de pouvoir démarrer sur le système d'exploitation que vous souhaitez.



Options d'installation

Avant de commencer

Spécifier le nom et
l'emplacement

Spécifier la génération

Affecter la mémoire

Configurer la mise en réseau

Connecter un disque dur
virtuel

Options d'installation

Résumé

Vous pouvez installer un système d'exploitation maintenant si vous avez accès au média d'installation, ou vous pouvez l'installer ultérieurement.

☐ Installer un système d'exploitation ultérieurement

☒ Installer un système d'exploitation à partir d'un fichier image de démarrage

Média

Fichier image (.iso) :

☐ Installer un système d'exploitation à partir d'un serveur d'installation réseau

Réparation Windows Dador

A - Installation à partir du référentiel APT

1 Réparation avec un support d'installation Windows 11

Tu auras besoin d'une **clé USB bootable avec Windows 11**. Si tu ne l'as pas :

1. Télécharge l'outil de création de média sur un autre PC depuis le site officiel de Microsoft :
<https://www.microsoft.com/software-download/windows11>
2. Crée une clé USB bootable (8 Go minimum) avec l'outil.

2 Étape de réparation

1. **Démarre ton PC depuis la clé USB Windows 11 :**
 - Accède au **BIOS/UEFI** (en général `F2`, `DEL`, `ESC`, ou `F12` au démarrage) et sélectionne ta clé USB comme périphérique de démarrage.
2. Quand tu arrives à l'écran d'installation, clique en bas sur **Réparer l'ordinateur** (ne clique pas sur "Installer maintenant").
3. Choisis **Dépannage > Invite de commandes**.

3 Commande de réparation

Tape ces commandes une par une dans l'invite de commandes (appuie sur `Entrée` après chaque ligne) :

```
bootrec /scanos  
bootrec /fixmbr  
bootrec /fixboot  
bootrec /rebuildbcd
```

Si la commande `/fixboot` renvoie une erreur d'accès refusé, tu peux l'ignorer pour l'instant.

Ensuite, tape :

```
bcdboot C:\Windows
```

Cela va tenter de recréer les fichiers de démarrage EFI.

4 Redémarrage

Une fois terminé, redémarre ton PC (retire la clé USB avant que ça ne redémarre) et vois si Windows se lance correctement.

Si ça ne marche toujours pas :

- Tu peux essayer l'outil de réparation automatique :
 - Depuis le menu de la clé USB : **Dépannage** >

Réparation du démarrage.