

INFORMATIQUE POUR LA ROBOTIQUE 2

Cours n °1 : Introduction à Python

Enseignante : Maëva LECAVELIER - maeva.lecavelier@gmail.com

Sommaire

01

Python

Qu'est-ce que c'est

02

Utiliser Python

Comment faire ?

03

Environnement virtuel

".venv"

04

Un peu de préparation

Avant le TD

05

Recap

Ce qu'il faut retenir

01

Python

Qu'est-ce que c'est

Qu'est-ce que Python ?



Langage polyvalent : haut niveau, facile à lire, utilisé autant pour du Web, du back-end, analyse de données, intelligence artificielle, création d'applications...

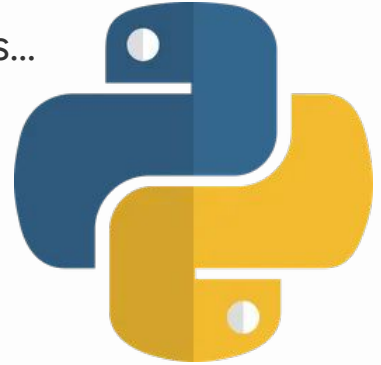


Syntaxe simple et lisible : clair, concis, pas de superflu



Python en quelques chiffres...

Top 3 des langages les plus utilisés, plus de 450 000 bibliothèques sur le dépôt officiel, des millions de développeurs, utilisé dans l'industrie et de grandes entreprises (Google, Netflix, Facebook, Spotify...) que ce soit pour du Web, ou de l'IA



Exemple d'un script Python

```
def main():
    print("Démarrage du robot suiveur de ligne...")
    while True:
        # Lire les données des capteurs (gauche, centre, droite)
        sensor_left, sensor_center, sensor_right = read_sensors()

        # Décision en fonction des capteurs
        if sensor_center:
            move_forward() # Avancer si la ligne est au centre
        elif sensor_left:
            turn_left() # Tourner à gauche si la ligne est à gauche
        elif sensor_right:
            turn_right() # Tourner à droite si la ligne est à droite
        else:
            stop() # Arrêter si la ligne est perdue
```

Un code épuré, simple

Pas de point
virgule

Pas de types
explicites

Pas de gestion de
mémoire

mais avec des règles

Des mots clé (if,
else, while...)

De la tabulation

Des délimiteurs ":"

Un code épuré, simple

1

Pas de point virgule

2

Pas de types explicites

3

Pas de gestion de mémoire

mais avec des règles

1

Des mots clé (if, else, while...)

2

De la tabulation

3

Des délimiteurs ":"

02

Développer en Python

Comment faire ?

Comment coder en Python ?

- Créer un fichier avec l'extension ".py"
- Ecrire son programme : aucune obligation, mais quelques bonnes pratiques...
- L'exécuter avec python3 : `python3 mon_script.py`

Les bonnes pratiques en Python

- Une fonction "main"
- Un appel à cette fonction "main" :

```
def main():  
    print("Hello, World!")  
  
if __name__ == "__main__":  
    main()
```

Explication :

- `main()` : Définit la fonction principale du programme.
- `if __name__ == "__main__":` :: Cette condition vérifie si le script est exécuté directement (et non importé comme un module dans un autre script).

Installer des bibliothèques

On utilise un gestionnaire de bibliothèques. Pour Linux, c'était apt-get, avec Python c'est pip3 !

Dans la pratique :

```
sudo apt-get install python3-pip
```

Puis :

```
pip install <ma_bibliothèque>
```



Aparté : Pourquoi Python **3**

Incompatible avec Python 2

Python 3 a introduit des améliorations significatives, mais cela nécessitait de rompre la compatibilité avec Python 2.

Évolution et modernisation

A ajouté des fonctionnalités modernes par rapport à Python 2 : meilleure gestion de chaînes de caractères, meilleurs gestions des erreurs, des exceptions...

Python 2 est obsolète

Fin du support au 1er janvier 2020

Changer de version Python

Version installée de base : Python3.6

```
python3 -version
```

On peut installer jusqu'à Python3.8 facilement

```
sudo apt-get install 3.8  
sudo apt-get install 3.8-venv  
sudo apt-get install binutils
```

Pour une version plus récente, il faut build!

<https://jetsonhacks.com/2023/06/12/upgrade-python-on-jetson-nano-tutorial/>

03

Environnement virtuel

`".venv"`

Les environnements virtuels Python

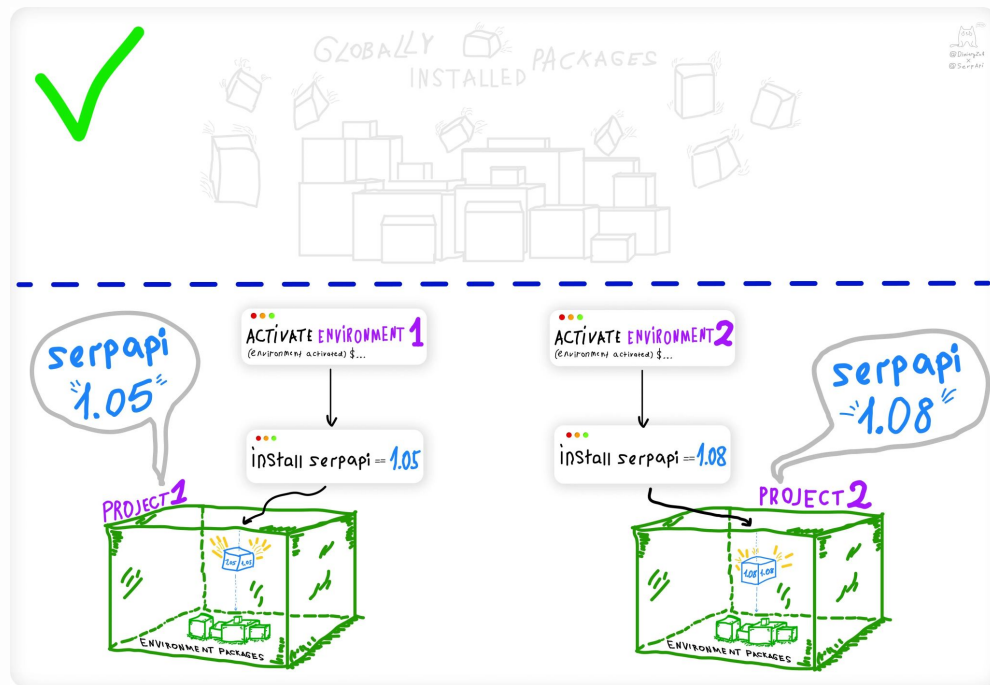
450 000 librairies, plusieurs versions de Python3 (3.6, 3.8, 3.11...) et autant de versions des librairies...

Alors, on sépare et on organise !

Dans la pratique :

```
python3 -m venv .<nom_env>  
source .<nom_env>/bin/activate
```

```
#création d'un environnement virtuel  
#active l'environnement virtuel
```



Pourquoi utiliser des environnements virtuels

Isoler les projets

Les bibliothèques utilisées pour un projet n'affecteront pas les autres

Gestion des versions des bibliothèques

Les versions des bibliothèques évoluent et peuvent introduire des changements incompatibles. En isolant, on gère nous-mêmes les versions installées. Cela protège aussi l'environnement global !

Faciliter la collaboration et le déploiement

Avec des fichiers comme requirements.txt, vous pouvez lister les dépendances exactes d'un projet. Un collaborateur peut recréer le même environnement sur sa machine avec `pip install -r requirements.txt`, garantissant la reproductibilité du projet.

04

Un peu de préparation

Avant le TD

Augmenter l'espace disque disponible

Petit point sur la Jetson Nano :

- Une mémoire eMMC limitée : l'installation de Ubuntu et de JetPack (suite logicielle de Nvidia) ne laisse que 2 Go de disponibles...
- Possibilité de brancher des supports de stockage : USB (clé, disque dur...), disque dur en NVMe, ou

Carte SD

Pour cela :

https://wiki.seeedstudio.com/reComputer_Jetson_Memory_Expansion/#capacity-expansion-via-usb-storage-device

Point d'attention sur ce tuto :

La carte SD est montée en `/dev/mmcblk<0-9>p<0-9>`(ex: `/dev/mmcblk1p1`) et non en `/dev/sda1`

`./copyRootToUSB.sh -p /dev/mmcblk1p1` prend quelques minutes...

Pour exécuter `./partUUID.sh` il faut préciser `-p` et non `-d` comme écrit sur le site

En cas de doute, demandez de l'aide !



**ATTENTION
DANGER**

05

Recap

Ce qu'il faut retenir

Python :

Un langage polyvalent, utilisé partout et par tous

Par sa simplicité et sa légèreté, Python est largement utilisé dans toutes les industries et entreprises. Son exécution est simple et l'installation de nouvelles librairies l'est tout autant

Les environnements virtuels : indispensable

Pour travailler proprement et pérennément, ces environnements sont essentiels car ils assurent la gestion granulaire des bibliothèques et de leurs versions installées

Attention aux versions !

Python2 et Python3 sont incompatibles ! Mais certaines librairies sont aussi incompatibles avec des "vieilles" versions 3 de Python, d'où l'intérêt des environnements virtuels



Merci !

Des questions ?