# Cuisine1

# Python_code_path

Exercice1

1. Donnons et expliquons le schéma de l'architecture interne de Python :

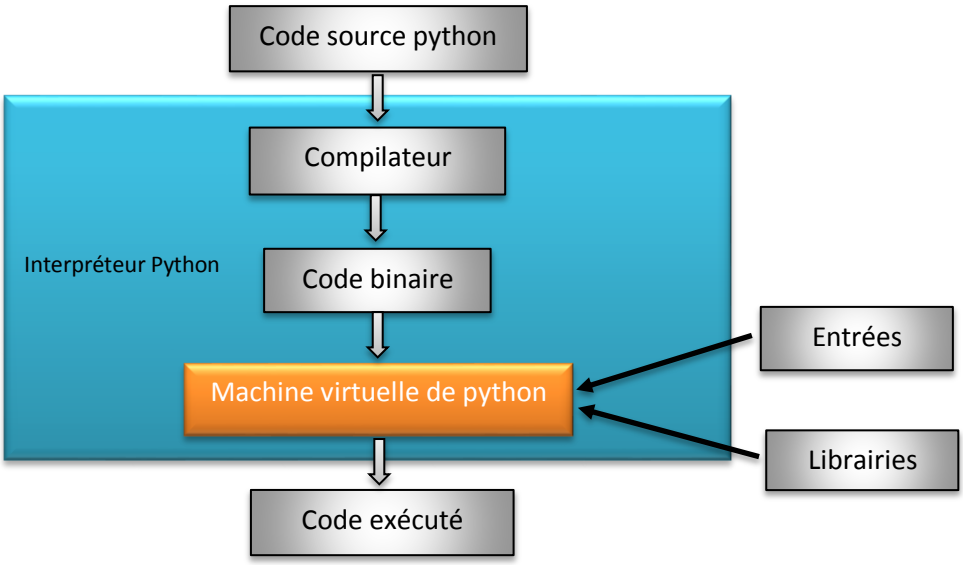


Schéma de la structure interne de Python

Le programmeur saisit un programme ou code qui représente le code source. Par la suite, il va compiler son code grâce au compilateur et obtenir un fichier comportant un code binaire, directement interprété par la machine virtuelle de python cette dernière va produire un exécutable. Une fois interprété, le code va être exécuté.

2.

- Une **Variable** est dite **globale** lorsqu'elle est accessible dans tout le code. Elle est dite **locale** lorsqu'elle n'est accessible que dans une portion de code (comme une fonction) à l'intérieur de laquelle elle a été déclarée.
- Les deux cas possibles de déclaration d'une variable globale en Python sont :
  - Définir la variable dans n'importe quelle partie du programme principal :
  - Déclarer la variable précédé du mot-clé **global** suivi du nom de la variable :

Ex : global a=0.

3. Les décorateurs en Python sont des fonctions dont le rôle est de modifier le comportement d'autres fonctions.
4. La différence entre git et GitHub et que **git** est un logiciel qui permet aux développeurs de sauvegarder des instantanées de leurs projets au fil du temps, tandis que **GitHub** est une plateforme web qui intègre les fonctionnalités de contrôle de versions de git afin de pouvoir les utiliser en collaboration.

Exercice2

Comportement de l'interpréteur Python à la lecture des instructions données

- >>> str(2021)*int ("3") : En effet, l'interpréteur va castrer l'entier 2021 en caractère ainsi que le caractère "3" en entier. Par la suite, effectuer la multiplication entre le nouveau caractère obtenu (2021) par. Ainsi, le résultat obtenu sera **'202120212021'**.
- >>> int("3")+float("3.2") : Ici, il est question de castrer le caractère 3 en entier avant d'effectuer une addition au caractère 3.2 préalablement castré en réel. En sortie, on va obtenir **6.2.**
- >>> int("3.98")+float("3.2") : Ici, l'interpréteur l'interpréteur va castrer le caractère 3.98 en entier ainsi que le caractère 3.2 en float avant d'effectuer la somme des deux résultats obtenus après castres.
- >>> int("3.9")+float("3.2") : Il sera question ici pour l'interpréteur de castrer le caractère 3.9 en entier, castrer le caractère 3.2 en réel ensuite effectuer la somme des résultats obtenus après castre. l'interpréteur va générer une erreur.
- >>> str(3)*float("3.2") : l'interpréteur va effectuer une opération d'addition après avoir castrer l'entier 3 en caractère et le caractère 3.2 en réel.
  Ici, l'interpréteur va générer une erreur.
- >>> str(3/4)*2 : il sera question pour l'interpréteur de castrer le réel 3/4 en caractères avant de le multiplier par l'entier 2. le résultat obtenu sera **'0.750.75'.**
- >>> "SEED"+["innovation","hub"] : il est question d'ajouter la liste de caractères ["innovation","hub"] à la chaine SEED. Il s'agit d'une concaténation entre un caractère ( ici SEED) et une liste de caractères( qui est ["innovation","hub"]).
- >>> "SEED"+str(["innovation","hub"]) : Ici, l'interpréteur va faire une concaténation. En effet, à la suite du caractère SEED, va être ajouté le résultat du castre de la liste de caractère en caractère pour obtention d'une chaine de caractères. Ainsi, le résultat sera **"SEED['innovation', 'hub']".**

Exercice3