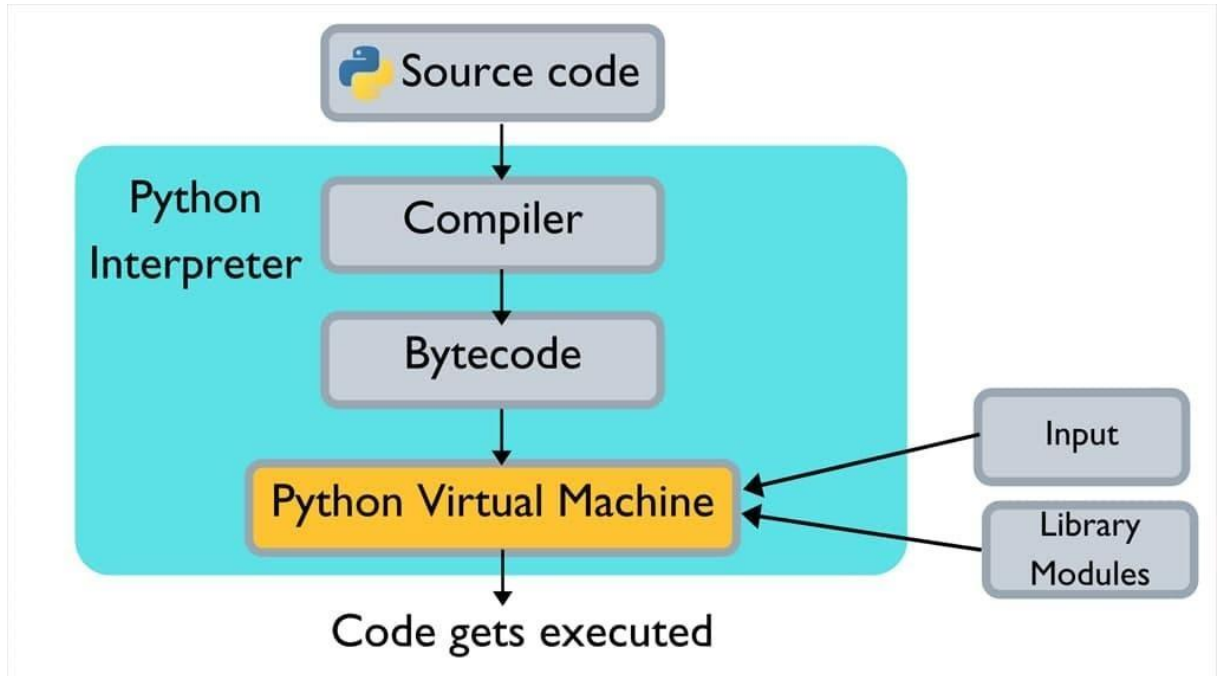


# CUISINES 1 – Python – code - path

## Exercice 1:

1. Schéma de l'architecture interne de Python.



Le code source qui est saisi par un programmeur sera compilé ensuite converti en langage machine puis la machine virtuelle va transformer ce code machine en exécutable tout en traitant les entrées et les librairies s'il y'en a pour pouvoir générer le code exécuté.

2. Une variable globale est une variable créée dans le programme principal et visibles dans tout le programme tandis qu'une variable locale est celle qui est créée dans une fonction du programme et n'est visible que dans cette fonction c'est à dire qu'elle n'existe que lors de l'exécution de la fonction. En Python, on obtient les variables locales en les déclarant dans le programme principal(hors de toute fonction) ou alors la déclarer avec le mot cle 'global'.
3. Un decorateur est une fonction permettant de modifier le comportement d'autres fonctions et/ou d'exécuter du code supplémentaire. Exemple :

```

def mon_decorateur(fonction):
    def ma_fonction_decoree():
        fonction()
        print("La fonction s'est exécutée")
    return ma_fonction_decoree

@mon_decorateur
def ma_fonction():
    print("hello_world")

```

4. La différence entre git et git hub est que git permet de faire des dépôts en local et GitHub permet de faire des dépôts distants (en ligne).

### Exercice 2 :

Comportement de l'interpréteur face aux instructions suivantes :

>>> str(2021) \* int("3") : l'interpréteur affiche '202120212021'.

Car int("3") retourne 3 en entier et donc concatène la chaîne avec elle-même 3 fois.

>>> int("3") + float("3.2") : l'interpréteur affiche 6.2 car les deux retournent un entier et un float

>>> int("3,98") + float("3.2") : retourne une erreur car "3,98" n'est pas convertible en entier à cause du symbole « , » .

>>> int("3.9") + float("3,2") : retourne une erreur car "3,2" n'est pas convertible en entier à cause du symbole « , » .

>>> str(3) \* float("3.2") : retourne une erreur car la chaine "3" ne peut etre concaténée 3.2 fois

>>> str(3/4) \* 2 : l'interpréteur affiche '0.750.75' car  $\frac{3}{4} = 0.75$  est converti en chaine et est concaténé deux fois :

>>> "SEED" + ["innovation", "hub"] : Ici l'interpréteur retournera un message d'erreur car une chaine ne peut être concaténée à une liste.

>>> "SEED" + str(["innovation", "hub"]) : La l'interpreteur pourra retourner une liste car la liste a été convertie en chaine. Il retourne donc 'SEED['innovation', 'hub'] '