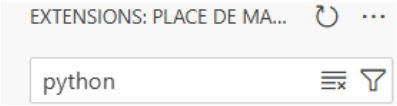
Coanim 1. CORRECTION Tp1 sur les stats

On voit dans ce chapitre comment créer des premiers scripts Python. L'objectif étant de créer une application qui permette de saisir des valeurs, de les stocker dans une liste python et d'en calculer la moyenne et l'écart-type.

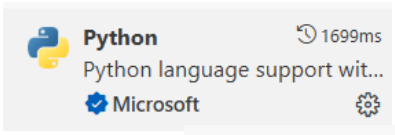
1- MISE EN PLACE DE L'ENVIRONNEMENT PYTHON SUR VSC :

⇨ Ouvrir Visual Studio Code et aller dans le menu Extensions dans la barre d'outils latérale gauche : 

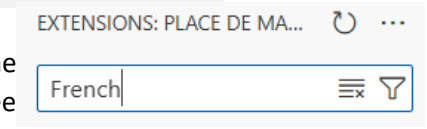
⇨ Dans la barre de recherche 

, rechercher les extensions liées à python.

Vérifier que les 2 extensions suivantes sont installées. Si elles ne le sont pas, les installer.

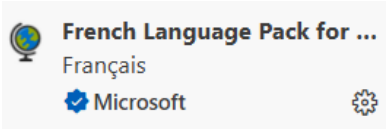


et


⇨ Dans la barre de recherche 

, rechercher les extensions liée

Vérifier que l'extension suivantes est installée. Si elle ne l'est sont pas, l'installer.



2- ANALYSE DU CORRIGE DU TP PRECEDENT STATS.PY:

⇨ Télécharger le corrigé du fichier *stats.py* (https://www.mathsapp.fr/public/documents/btsCiel/tag4/stats.py) et le copier dans un nouveau répertoire sur votre espace sur D :.

⇨ Dans l'onglet *Fichier*, choisir *Ouvrir le dossier* et pointer sur le répertoire *coAnim* créé sur D:.

⇨ Ouvrir le fichier *stats.py* qui apparait dans la fenêtre Dossiers de gauche  . Pour l'exécuter, il suffit de cliquer sur  . Le terminal s'ouvre et permet de suivre l'exécution.

⇨ Il est aussi possible sur VSC, **d'exécuter partiellement** le code. Par exemple, si on souhaite exécuter uniquement la fonction `calculMoyenne(notes: list)` afin de la tester individuellement, on procède ainsi :

- On exécute `$ python` dans le terminal afin de lancer l'interpréteur python. Lle préfixe $ du terminal est alors remplacé par les 3 chevrons de python >>>,
- Dans l'éditeur, **sélectionner** avec la souris, le code de la fonction *calculMoyenne()*
- En tapant sur les touches *Maj* et *Entrée* , python lit et exécute uniquement les lignes sélectionnées.

```python
def calculMoyenne(notes: list) -> float:
    """
    Calcule la moyenne des notes contenues dans la liste en paramètre.
    Args:
        list : liste de nombres
    Returns:
        float: La moyenne des nombres de cette liste
    """
    if len(notes) == 0 :
        return None
    s = 0
    for i in range(len(notes)) :
        s = s + notes[i]
    return s / len(notes)
```

- Comme on n'a que sélectionné le code d'une fonction, celui-ci est uniquement lu et mémorisé.

- On peut ensuite exécuter la fonction *calculMoyenne()* qui est à présent mémorisé :

```
>>> calculMoyenne([5,10,15])
10.0
```

- Pour **sortir de python**, il faut exécuter la fonction *exit()*, cela permet de retrouver le préfixe $ du terminal.

3- EXECUTION DE STATS.PY EN MODE DEBUGGAGE :

On se propose ici de faire fonctionner pas à pas le script de la fonction *calculMoyenne()* en utilisant le *Debogueur* Python.

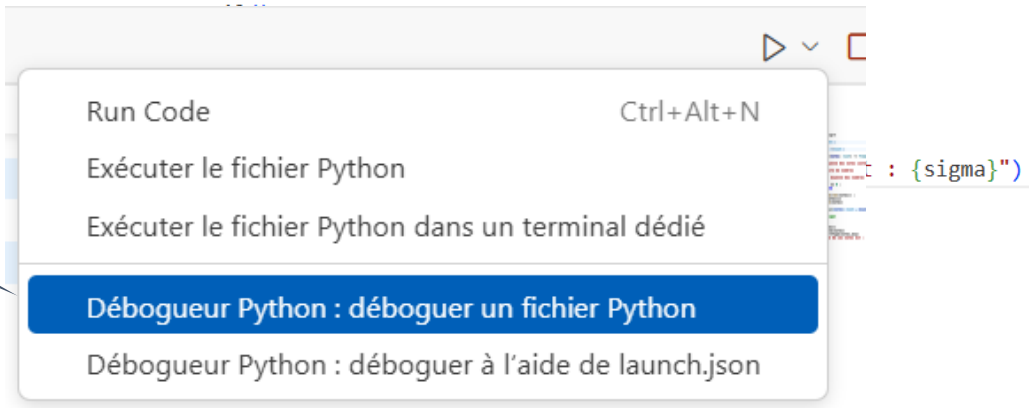
⇨ Par un click, créer *un point d'arrêt* sur la ligne 45, qui se situe au milieu de la boucle for :

Point d'arrêt à créer par un click souris

```
3  > def accueil() ->str :···
14
15 > def saisieNotes() ->list :···
32
33 def calculMoyenne(notes: list) -> float:
34     """
35     Calcule la moyenne des notes contenues dans la liste en paramètre.
36     Args:
37         list : liste de nombres
38     Returns:
39         float: La moyenne des nombres de cette liste
40     """
41     if len(notes) == 0 :
42         return None
43     s = 0
44     for i in range(len(notes)) :
45         s = s + notes[i]
46     return s / len(notes)
47
48 > def calculEcartType(notes: list , moyenne : float) ->float :···
63
64 # programme principal
```

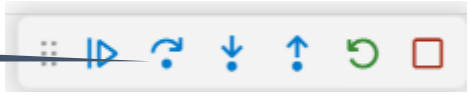
⇨ Lancer une exécution avec deboguage

Exécution avec déboguage

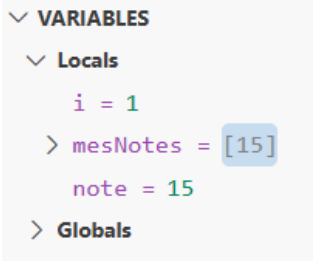


⇨ Utiliser l'avancement Pas à Pas

Avancement Pas à Pas



pour voir les valeurs instantanées des différentes variables

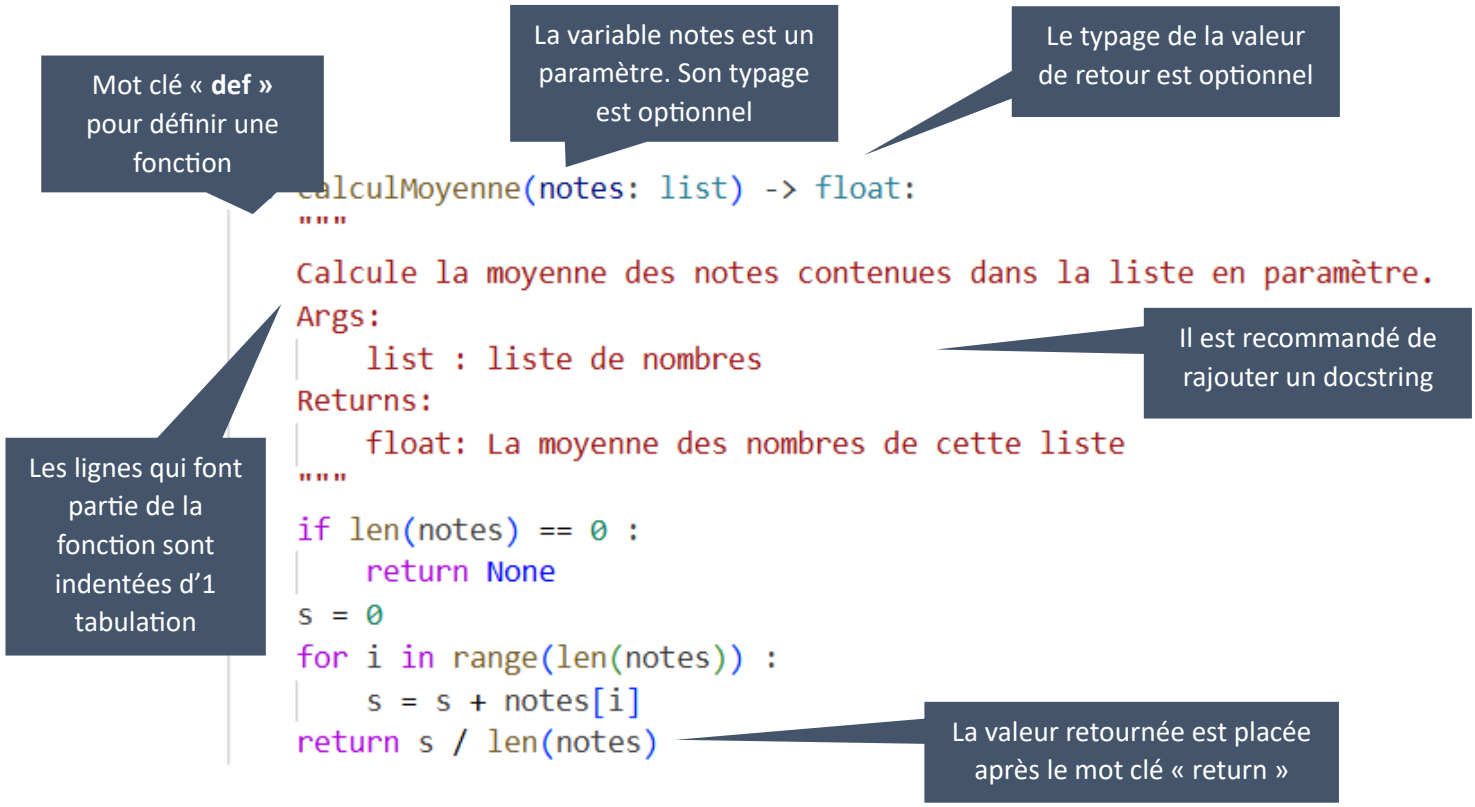



```
23     mesNotes = []
24     i = 1
25     note = input(f"Entre la note {i} (ou rien si fin) : ")
26     while note != "" :
27         note = int(note)
28         mesNotes.append(note)
29         i = i + 1
30         note = input(f"Entre la note {i} (ou rien si fin) : ")
31     return mesNotes
```

Ce déboguage est très utile pour vérifier le contenu des variables lors d'une exécution.

4- Ce qu'il faut retenir de ce TP initiation :

a. Fonctions :




```
calculMoyenne(notes: list) -> float:
"""
Calcule la moyenne des notes contenues dans la liste en paramètre.
Args:
    list : liste de nombres
Returns:
    float: La moyenne des nombres de cette liste
"""
if len(notes) == 0 :
    return None
s = 0
for i in range(len(notes)) :
    s = s + notes[i]
return s / len(notes)
```

Point Cours : Les variables déclarées dans une fonction ont une portée LOCALE.

b. Listes :

Point Cours :

- Créer une liste vide : `nomListe = []`
- Créer une liste non vide : `nomListe = [5,"n",True,3.14]`
- Ajouter un élément en fin de liste : `nomListe.append("s")`
- Taille d'une liste : `len(nomListe)`
- Accéder à la valeur à l'index 2 pour lire ou modifier : `nomListe[2]`
- Supprimer l'élément à l'index 2 : `del(nomListe[2])`
- Savoir si un élément est dans la liste : `test = 5 in nomListe`
- Parcourir la liste par éléments :

```
for elt in nomListe :
    print(elt)
```

- Parcourir la liste par index :

```
for i in range(len(nomListe)) :
    print(nomliste[i])
```

c. BOUCLES WHILE :

```
def saisieNotes() ->list :
    mesNotes = []
    i = 1
    note = input(f"Entre la note {i} (ou rien si fin) : ")
    while note != "" :
        note = int(note)
        mesNotes.append(note)
        i = i + 1
        note = input(f"Entre la note {i} (ou rien si fin) : ")
    return mesNotes
```

Les lignes qui se répètent sont repérées par une INDENTATION

note != ‘’ ‘’ est un test qui prend la valeur True ou False. Tant que cette valeur est à True, la boucle continue

Point Cours : La structure WHILE est utilisée lorsque l’on veut réaliser une boucle pour laquelle le nombre d’itérations n’est pas connu au départ.

Point Cours :

Dans une structure WHILE, si la condition ne prend jamais la valeur *True*, le bouclage tourne à l’infini.
Lors de l’écriture de la condition, il faudra toujours s’assurer qu’elle prendra la valeur False à un moment donné.