

vous galérez avec les fonctions ?

Posez vous les bonnes questions

Appel de fonctions.

Si mon programme doit utiliser (appeler) une fonction :

question 1 : quelles sont les informations dont la fonction à besoin ?

Ces informations sont les **paramètres** que l'on passe à la fonction.

exemple : Je dois utiliser une fonction appelée *somme* qui AFFICHE la somme de 2 entiers.

- je passe ces deux entiers à ma fonction.
- si dans mon programme, je stocke ces entiers dans des variables *x* et *y*, je fais l'appel de la fonction comme suit :

```
somme(x,y)
```

exemple : Je dois utiliser une fonction appelée *moyenne* qui AFFICHE la moyenne d'un tableau.

- je passe ce tableau à ma fonction.
- si dans mon programme, je stocke ce tableau dans une variable *tab*, je fais l'appel de la fonction comme suit :

```
moyenne(tab)
```

exemple : Je dois utiliser une fonction appelée *ajoute* qui MODIFIE le contenu d'un tableau pour ajouter un nombre à chaque case du tableau

- je passe ce nombre ET le tableau à ma fonction.
- si dans mon programme, je stocke le tableau dans une variable *tab* et l'entier dans une variable *x*, je fais l'appel de la fonction comme suit :

```
ajoute(tab,x)
```

Remarque : Pour des raisons pédagogiques, les deux premiers exemples sont des fonctions qui affichent des calculs. Dans la vraie vie, on ne fait souvent pas ça. On utilise ce que la fonction répond, ou **renvoie** ou **retourne**. C'est ce qu'on va voir maintenant.

Question 2 : Est ce que la fonction renvoie quelque chose ?

Si une fonction calcule quelque chose, les gens qui l'ont conçue se sont débrouillés pour qu'elle renvoie cette information à ce qui l'appelle. Le programme doit donc stocker sa réponse dans une variable.

exemple : Je dois utiliser une fonction appelée *somme* qui CALCULE la somme de 2 entiers.

- la fonction va renvoyer la somme.
- imaginons que dans mon programme, je stocke ces entiers dans des variables *x* et *y*.
- dans mon programme, je vais récupérer la somme dans une variable que je nomme COMME JE VEUX. Par exemple, *resu*

```
# appel de la fonction.  
resu = somme(x,y)    # la réponse de la fct est stockée dans la variable resu  
  
# utilisation de la réponse de la fonction  
print("la somme vaut", resu)
```

exemple : Je dois utiliser une fonction appelée *moyenne* qui CALCULE la moyenne d'un tableau.

- la fonction va renvoyer la moyenne
- imaginons que dans mon programme, je stocke ce tableau dans une variable *tab*.
- dans mon programme, je vais récupérer la moyenne dans une variable que je nomme COMME JE VEUX. Par exemple, *moy*

```
moy = moyenne(tab)
```

exemple : Je dois utiliser une fonction appelée *ajoute* qui MODIFIE le contenu d'un tableau pour ajouter un nombre à chaque case du tableau

- la fonction ne renvoie rien, elle va modifier le contenu du tableau qu'on lui passe en argument.
- imaginons que dans mon programme, je stocke le tableau dans une variable *tab* et l'entier dans une variable *x*.

```
ajoute(tab,x)
```

```
# après cet appel, le contenu de tab a changé (on a ajouté x à chaque case)
```

Avec ces deux questions, vous SAVEZ appeler une fonction quelconque qu'on vous décrit.

Attention, néanmoins à ce qui suit : **L'ordre des paramètres qu'on passe à la fonction est choisi par le concepteur de la fonction.** Vous devez respecter cet ordre. Si c'est une fonction faite par quelqu'un d'autre, la documentation vous indiquera cet ordre.

exemple : pour la fonction *ajoute* qui MODIFIE le contenu d'un tableau pour ajouter un nombre à chaque case du tableau, la documentation vous dira par exemple que le tableau est le premier argument, et le nombre le second.

auquel cas

```
# Je l'appelle comme suit :  
ajoute(tab,x)
```

```
# ET PAS COMME CA :  
ajoute(x,tab)
```

Création de fonctions.

Si je veux CREER une fonction qui fait quelque chose, les questions sont les mêmes, mais j'en rajoute deux.

1. **Question 1** : quel nom va porter ma fonction ?

Je choisis ce que je veux. J'essaye de prendre quelque chose de parlant... C'est ce nom que le programme qui appelle la fonction utilisera (pour l'appeler)

2. **Question 2** De quelles informations ma fonction a besoin ?

- Ces informations sont les paramètres de ma fonction.
- dans la définition de ma fonction, je vais placer entre parenthèses un nom de variable par paramètre.
- je choisis librement les noms de ces variables.

exemple : Je veux créer une fonction qui calcule la somme de deux entiers

- je choisis un nom pour ma fonction (par exemple, *somme*).
- ma fonction a besoin de deux informations (les deux entiers), je choisis un nom pour chaque entier. Par exemple *a* et *b*

=> je peux écrire le début du code de ma fonction :

```
def somme (a,b):  
    # des trucs que ma fonction fera
```

exemple : Je veux créer une fonction qui calcule la moyenne d'un tableau

- je choisis un nom pour ma fonction (par exemple, *moyenne*).
- ma fonction a besoin d'une information (le tableau), je choisis un nom pour le tableau. Par exemple *mon_tableau*

=> je peux écrire le début du code de ma fonction :

```
def moyenne (mon_tableau):  
    # des trucs que ma fonction fera
```

exemple : Je dois utiliser une fonction qui MODIFIE le contenu d'un tableau pour ajouter un nombre à chaque case du tableau

- je choisis un nom pour ma fonction (par exemple, *ajoute*).
- ma fonction a besoin de deux informations (le tableau et le nombre), je choisis un nom pour le tableau et pour le nombre. Par exemple *mon_tableau* et *nombre*

=> je peux écrire le début du code de ma fonction :

```
def ajoute(mon_tableau,nombre):
```

Dans l'exemple qui précède, l'ordre dans lequel j'ai mis les paramètres de la fonction (tableau puis nombre) imposera à l'utilisateur de les passer dans cet ordre.

3. Question 3 Comment écrire le contenu de la fonction ?

Dès que vous avez écrit l'entête (le **def truc(x,y):**), votre fonction dispose des variables x et y, et elle peut les utiliser.

exemple : Je veux créer une fonction qui calcule la somme de deux entiers

```
# Si j'ai déjà écrit la ligne qui suit  
def somme (a,b):  
  
    # je peux écrire  
    resu = a+b
```

la somme de mes deux nombres est bien $a+b$, vu que c'est sous ce nom que la fonction connaît les nombres.

exemple : Je veux créer une fonction qui calcule la moyenne d'un tableau

```
# Si j'ai déjà écrit la ligne qui suit  
def moyenne (mon_tableau):  
  
    # je peux écrire  
    somme = 0  
    for e in mon_tableau  
        somme = somme + e  
  
    moy = somme / len(tableau)
```

c'est bien *mon_tableau* que j'utilise dans ma fonction, car c'est sous ce nom que la fonction connaît le tableau.

4. Question 4 : Est ce que ma fonction renvoie quelque chose.

- Si oui, elle doit RENVoyer la variable qu'elle calcule.
- le mot clef pour renvoyer est **return**.
- le nom de la variable, vous l'avez choisi en écrivant le code au moment de la **question 3**

exemple : Je veux créer une fonction qui calcule la somme de deux entiers

la fonction va renvoyer le résultat du calcul

```
# Si j'ai déjà écrit ce qui les deux lignes ci dessous,  
def somme (a,b):  
    resu = a+b  
  
    # pour renvoyer le résultat, je fais  
    return resu
```

exemple : Je veux créer une fonction qui calcule la moyenne d'un tableau

la fonction va renvoyer cette moyenne

```
# Si j'ai déjà écrit toutes les lignes suivantes jusqu'à moy =
def moyenne (mon_tableau):

    # je peux écrire
    somme = 0
    for e in mon_tableau
        somme = somme + e

    moy = somme / len(tableau)

    # je dois renvoyer le résultat du calcul, qui est dans la variable moy :
    return moy
```

exemple : Je veux créer une fonction qui MODIFIE un tableau pour y ajouter un nombre

la fonction ne renvoie rien, on ne met pas de return

```
def ajoute (mon_tableau, nombre):
    for i in range(len(tab)):
        mon_tableau[i] = mon_tableau[i] + nombre

    # y a pas de return
```

Gros résumé en exemples

Je vais prendre deux exemples de type d'un examen (même si la première est au delà de ce que vous savez faire, ce n'est pas grave, vous verrez).

Donner le code d'une fonction qui calcule la valeur des mensualités d'un crédit d'un certain montant, sur n mois, à un certain taux d'intérêt

En appliquant les questions qui précèdent :

- il me faut un nom pour ma fonction (disons *calculer_mensualites*)
- il me faut un nom de variable pour chaque information dont a besoin la fonction (ces informations sont : le montant du crédit, la durée en mois du crédit, le taux d'intérêt). Je peux prendre comme noms de variables *montant, duree, taux*.
- ma fonction va renvoyer la valeur d'une mensualité, que la fonction va placer dans une variable dont je choisis le nom. par exemple, *valeur*

Je peux donc écrire:

```
def calculer_mensualites (montant, duree, taux):

    # Ici, je fais un calcul qui utilise les variables
    #   montant, durée et taux.
    # Je vais par exemple faire le calcul FAUX suivant :
    valeur = montant * (1+taux) / duree

    # je fais mon return sur le résultat du calcul
    return valeur
```

Notez que pour le prochain examen, si vous faites ça (alors que le calcul des mensualités est faux), je vous met au moins la moitié des points de la question. Car l'essentiel du passage d'arguments à une fonction est compris.

Par ailleurs, ne cherchez pas à résoudre cette question complètement, le calcul d'une mensualité est assez complexe.

Faire un programme qui utilise cette fonction pour afficher la valeur des mensualités d'un crédit. Les informations sont entrées par l'utilisateur.

- je sais que je dois appeler la fonction `calculer_mensualites`
- je dois placer les informations (montant, durée, taux) dans des variables dont je choisis les noms. Disons *prix*, *nb_mois*, *t*
- l'utilisateur va entrer ces informations au clavier, donc je vais utiliser `input`, en mettant les bons types de variables.

```
print("entrez le montant du crédit")
prix = float(input())

print("entrez le taux du crédit")
t = float(input())

print("entrez la durée en mois du crédit")
nb_mois = int(input())

# J'appelle ma fonction en passant ces informations dans le bon ordre
# La fonction renvoie la valeur des mensualités, que je stocke dans une variable dont je choisis le nom pour
# Disons que je choisis argh comme nom de variable.

argh = calculer_mensualite(prix, nb_mois, t)

# je peux afficher la valeur des mensualités
print("chaque mois, vous paierez", argh)
```

Les petits trucs en plus

Si vous avez bien assimilé ce qui précède, tout va bien. Je rajoute là quelques informations intéressantes :

Qui connaît quelles variables ?

Une fonction ne connaît normalement que les variables qu'elle déclare et ses paramètres. Ces variables sont “*de portée locale à la fonction*”.

Le programme principal ne connaît que les variables qu'il déclare.

exemple :

```
def somme(a,b):
    resu = a+b
    return resu

print("entrez un nombre")
x = float(input())

print("entrez un nombre")
y = float(input())

val = somme(x,y)
print("la somme vaut", val)
```

Dans ce qui précède :

- la fonction ne connaît que les variables *a,b* (ses paramètres) et *resu*, une variable créée par la fonction pour la fonction
- la fonction ne devrait pas utiliser *x* ou *y* ou *val* qui sont des variables du programme principal
- le programme principal ne connaît que *x,y* et *val*.

- il ne connaît ni *a*, ni *b*, ni *resu*

Qui peut appeler une fonction ?

En fait, le programme principal peut appeler une fonction, mais une fonction peut appeler une autre fonction. Chaque fonction ne connaît que ses variables locales.

Ce qui me permettrait de faire ceci

```
# une fonction qui calcule la somme des éléments d'un tableau
def somme(tableau):
    s = 0
    for e in tab:
        s = s + e
    return s

# une fonction qui calcule la moyenne des éléments d'un tableau
def moyenne(tab):
    # on calcule la somme des elements du tableau
    # en appelant la fonction précédente
    sum = somme(tab)

    resu = sum / len(tab)

# le programme principal lit 3 notes au clavier et calcule la moyenne
notes = []
for i in range(3):
    print("entrez une note")
    x = float(input())
    notes.append(x)

m = moyenne(notes)
print("la moyenne des notes est",m)
```

Dans ce qui précède :

- Pour la fonction *somme*, le tableau s'appelle *tableau*. Elle fait ses affaires avec.
- Pour la fonction *moyenne*, le tableau s'appelle *tab*.
- La fonction *moyenne* appelle la fonction *somme* pour calculer la somme et renvoyer la moyenne.
- Pour le programme principal, le tableau s'appelle *notes*.
- Le programme principal appelle la fonction *moyenne* et récupère le résultat dans une variable *m*, qu'il affiche.

Et voilà !