

Résumé : structure d'un programme Python type

```
# -*- coding:Utf8 -*-  
  
#####  
# Programme Python type #  
# auteur : G.Swinnen, Liège, 2009 #  
# licence : GPL #  
#####  
  
#####  
# Importation de fonctions externes :  
  
from math import sqrt  
  
#####  
# Définition locale de fonctions :  
  
def occurrences(car, ch):  
    "Cette fonction renvoie le \  
    nombre de caractères <car> \  
    contenus dans la chaîne <ch>"  
  
    nc = 0  
  
    i = 0  
    while i < len(ch):  
        if ch[i] == car:  
            nc = nc + 1  
  
            i = i + 1  
  
    return nc  
  
#####  
# Corps principal du programme :  
  
print("Veuillez entrer un nombre :")  
nbr = eval(input())  
  
print("Veuillez entrer une phrase :")  
phr = input()  
print("Entrez le caractère à compter :")  
cch = input()  
  
no = occurrences(cch, phr)  
rc = sqrt(nbr**3)  
  
print("La racine carrée du cube", end=' ')  
print("du nombre fourni vaut", end=' ')  
print(rc)  
  
print("La phrase contient", end=' ')  
print(no, "caractères", cch)
```

Un programme Python contient en général les blocs suivants, dans l'ordre :

- Quelques instructions d'initialisation (importation de fonctions et/ou de classes, définition éventuelle de variables globales).
- Les définitions locales de fonctions et/ou de classes.
- Le corps principal du programme.

Le programme peut utiliser un nombre quelconque de fonctions, lesquelles sont définies localement ou importées depuis des modules externes.

Vous pouvez vous-même définir de tels modules.

La définition d'une fonction comporte souvent une liste de PARAMÈTRES.

Ce sont toujours des VARIABLES, qui recevront leur valeur lorsque la fonction sera appelée.

Une boucle de répétition de type 'while' doit toujours inclure au moins quatre éléments :

- l'initialisation d'une variable 'compteur' ;
- l'instruction while proprement dite, dans laquelle on exprime la condition de répétition des instructions qui suivent ;
- le bloc d'instructions à répéter ;
- une instruction d'incrémentement du compteur.

La fonction "renvoie" toujours une valeur bien déterminée au programme appelant.

Si l'instruction 'return' n'est pas utilisée, ou si elle est utilisée sans argument, la fonction renvoie un objet vide : 'None'.

Le programme qui fait appel à une fonction lui transmet d'habitude une série d'ARGUMENTS, lesquels peuvent être des valeurs, des variables, ou même des expressions.