

Chapitre 4 : Fonctions (Première partie)

- Une fonction est un algorithme spécifique effectuant une tâche précise (par exemple, un calcul ou un affichage) et est destinée à être utilisée par un autre algorithme.
- L'algorithme appelle une fonction, son exécution est arrêtée et le flux de contrôle est transmis à la fonction jusqu'à ce que cette dernière ait terminé sa tâche et "rende la main".

Intérêts d'écrire des fonctions :

- Réutilisation d'algorithmes spécifiques effectuant une tâche précise ;
- Amélioration de la lisibilité et de la clarté d'un algorithme en le decoupant en fonctions avec des noms explicites.

Définition d'une fonction sans paramètres et sans valeur de retour

Syntaxe :

```
def nom_fonction() :  
    # algorithme de la fonction
```

- def : indique que nous souhaitons définir une fonction ;
- nom_fonction : est le nom de la fonction, il suit les mêmes règles que les noms de variables ;
- les parenthèses ouvrantes et fermantes sont obligatoires pour indiquer que nom_fonction est bien une fonction (on est dans le cas d'une fonction qui ne possède pas de paramètres).

Exemple

```
In [1]: # Définition de la fonction hello()  
def hello() :  
    """Affiche un message de bienvenue Bonjour le monde"""  
    print("Bonjour le monde !")  
    #####  
# Algorithme principal  
#  
i=1  
while i<4 :  
    hello()  
    i=i+1
```

Définition d'une fonction avec paramètres et sans valeur de retour

Syntaxe :

```
def nom_fonction(param_1, param_2, ..., param_n) :  
    # algorithme de la fonction
```

- def : indique que nous souhaitons définir une fonction ;
- nom_fonction : est le nom de la fonction, il suit les mêmes règles que les noms de variables ;
- une parenthèse ouvrante, suivie des paramètres en entrées de la fonction, on parle de *paramètres formels*, suivis d'une parenthèse fermante. L'ordre des paramètres est important.

Exemple d'utilisation de la fonction affiche_age

```
In [3]: # Définition de la fonction affiche_age()  
def affiche_age(est_femme, nom, age) :  
    """ Affiche un message personnalisé suivant le sexe, le nom et l'âge d'une personne  
        est_femme (entrée de type bool)  
        nom (entrée de type str)  
        age (entrée de type int)  
    """  
    prefix=""  
    if est_femme :  
        prefix="Mme"  
    else :  
        prefix="Mr"  
  
    print(prefix, nom, "a", age, "ans.\n");  
#####  
# Algorithme principal  
    affiche_age(False,"Martin",23);  
    affiche_age(True,"Sophie",32);
```

Portées de variables

- La portée/visibilité d'une variable est l'endroit dans le code où cette variable est accessible.
- Une variable déclarée dans le corps d'une fonction n'est visible (ou accessible) qu'à l'intérieur de cette fonction.

Passage des paramètres par copie

- Lors de l'appel d'une fonction, la transmission des paramètres se fait par copie de la valeur des paramètres effectifs (de l'algorithme appelant).
- Toute modification de la valeur des paramètres dans le corps de la fonction n'entraîne pas la modification des paramètres effectifs (de l'algorithme appelant).

Vocabulaire

- fonction, paramètre effectif, paramètre formel, portée de variables, variable locale, transmission des paramètres par copie.