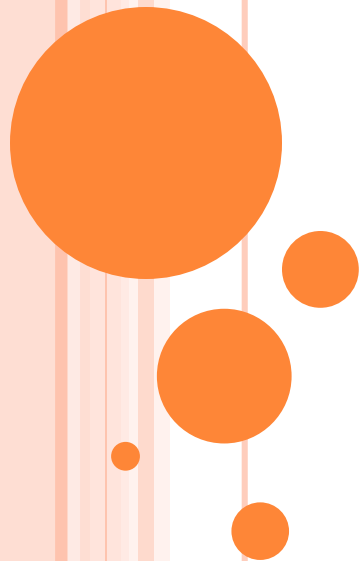


## **CHAPITRE N° 3:**

# **LES FONCTIONS EN PYTHON**

L1 Mathématiques appliquées



# Plan de Cours

1. Introduction aux fonctions en Python
2. Syntaxe de fonction
3. Fonctions avec valeur de retour
4. Conclusion

# 1- Introduction aux fonctions en Python

## ◆ Qu'est-ce qu'une fonction ?

Une fonction est un **bloc de code réutilisable** qui exécute une tâche spécifique.

Elle est définie une seule fois et peut être **appelée plusieurs fois** dans un programme.

# 1-Introduction aux fonctions en Python

## ◆ Qu'est-ce qu'une fonction ?

Une fonction est un **bloc de code réutilisable** qui exécute une tâche spécifique.

Elle est définie une seule fois et peut être **appelée plusieurs fois** dans un programme.

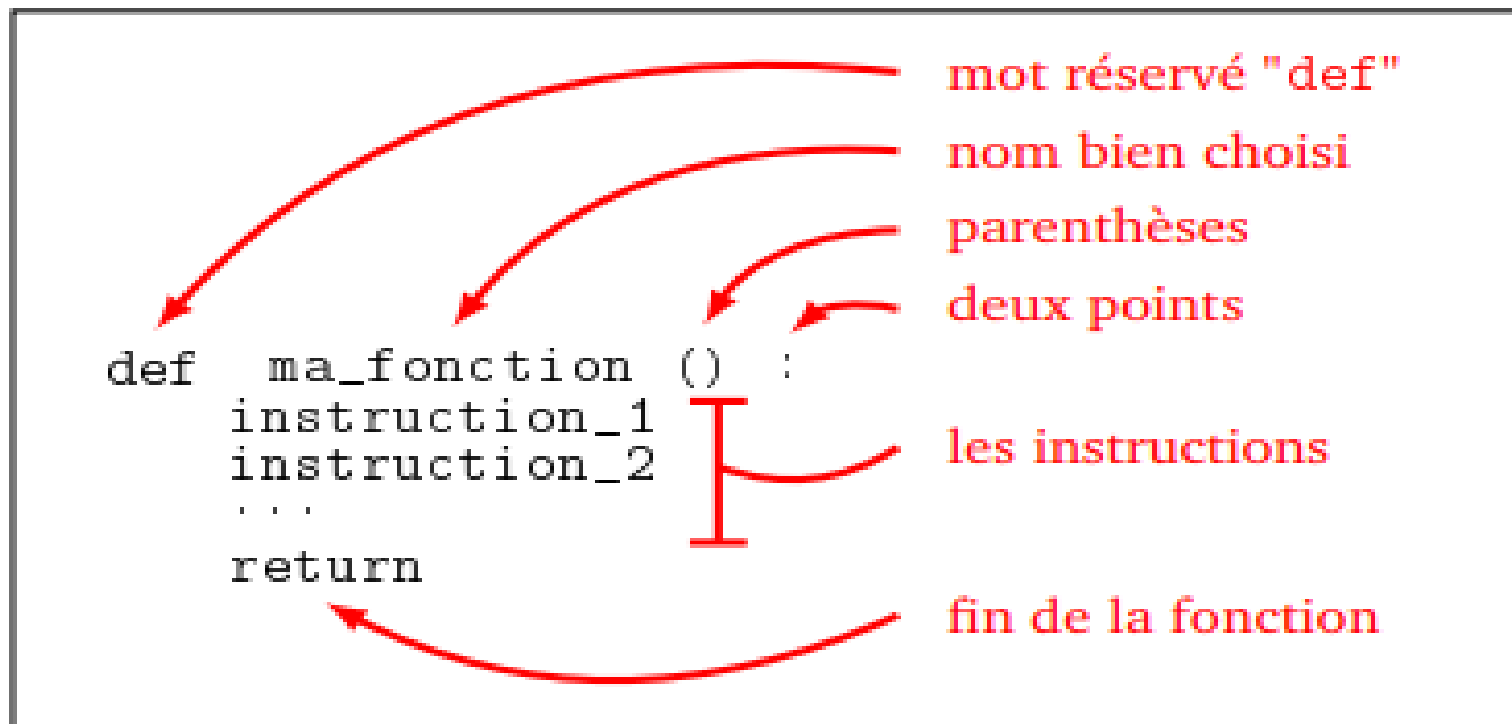
## ◆ Avantages des fonctions :

- **Réutilisation du code** : utiliser la même fonction plusieurs fois.
- **Organisation du programme** : rendre le code plus clair et structuré.
- **Simplification des programmes complexes** : les programmes deviennent plus faciles à comprendre et à maintenir.

## 2- Syntaxe de Fonction:

◆ En Python, une fonction est définie avec le mot clé **def**.

◆ **Structure générale:**



## 2- Syntaxe de Fonction (suite):

### Exemples:

```
def dire_bonjour():  
    print("Bonjour !")  
    return
```

```
def affiche_carres():  
    for i in range(20):  
        print(i**2)  
    return
```

- ❑ Les instructions sont regroupées dans un **bloc indenté**.
- ❑ Le mot `return` (optionnel) indique la fin de la fonction et permet de **renvoyer un résultat**.
- ❑ Les instructions de la fonction **ne s'exécutent que lorsque la fonction est appelée**.

### Entrée et sortie d'une fonction

- **Entrée (paramètres)** : valeurs transmises à la fonction pour effectuer un traitement.
- **Sortie (résultat)** : valeur renvoyée par la fonction grâce au mot clé `return`.

## Exemple :

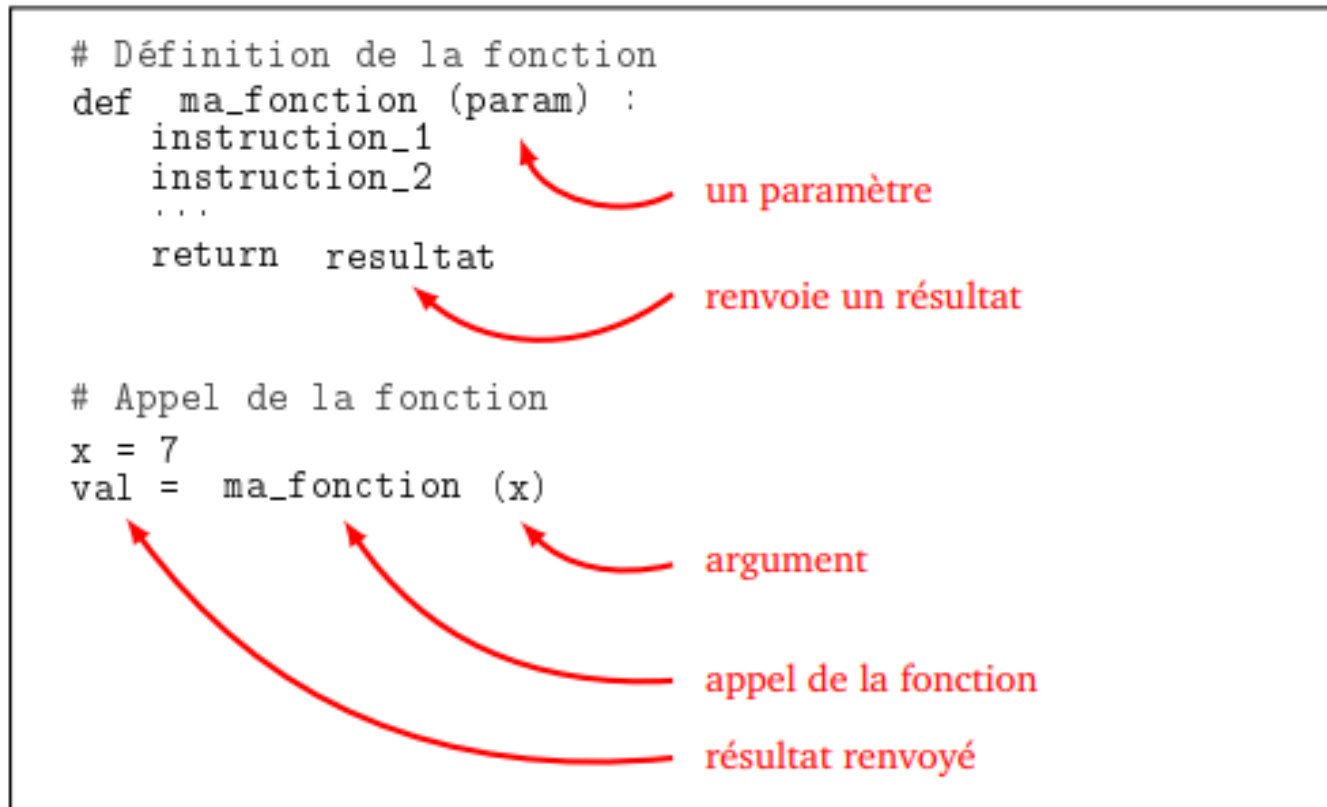
```
def affiche_mois(numero):  
    if numero == 1:  
        print("Nous sommes en janvier.")  
    if numero == 2:  
        print("Nous sommes en février.")  
    if numero == 3:  
        print("Nous sommes en mars.")  
    # etc.  
    return  
  
affiche_mois(3)    #Appel de la fonction
```

## Résultat Obtenu :

Nous sommes en mars.

# 3-Fonctions avec valeur de retour

## ◆ Structure générale:



## Remarque:

- Paramètre = variable dans la fonction.
- Argument = valeur donnée à la fonction.

# 3-Fonctions avec valeur de retour

## Exemple:

```
import math
```

```
def calcule_cube(a):  
    cube = math.pow(a, 3)  
    return cube
```

```
x = 2
```

```
y = 1
```

```
z = calcule_cube(x) + calcule_cube(y)
```

```
print(z)
```

**Résultat Obtenu :**

**9.0**

## 4-Conclusion

- Les fonctions permettent d'organiser et structurer les programmes Python.
- Elles facilitent la réutilisation du code et réduisent la répétition.
- Une fonction peut recevoir des paramètres et retourner un résultat.
- L'utilisation des fonctions est essentielle pour écrire des programmes plus clairs et plus efficaces.