

# PYTHON : LES FONCTIONS

Les fonctions vous permettent de mieux structurer votre code, d'éviter de faire des « copier/coller » d'implémentations et donc d'obtenir un code plus maintenable et plus robuste.

## Quand créer une fonction ?

Le fait de devoir créer une fonction ou non ne dépend pas du nombre de lignes de code.

La création de fonction est plutôt liée à la « factorisation du code », c'est à dire éviter de recopier/coller une partie du code pour la réutiliser à un autre endroit.

**Créez une fonction quand vous souhaitez réutiliser une partie du code à un autre endroit, ou quand le code que vous écrivez devient trop compliqué et que vous souhaitez « déléguer » une partie de votre implémentation.**

## Appel de fonction

```
print()                # On utilise toujours les parenthèses pour appeler une fonction
print(« toto »)        # Passage d'un paramètre à la fonction print
print(« je suis » + « toto ») # Concaténation et passage d'un paramètre à la fonction print
print(« je suis », « toto ») # Passage de 2 paramètres (séparation avec la virgule)
```

## Définir une fonction

```
# Cette fonction possède un paramètre obligatoire : nom
def afficher_infos_personne(nom):
    print("La personne est", nom)
    print("le nom possède", len(nom), "caractères")
```

```
afficher_infos_personne(« toto ») # Appel de la fonction
```

# Les paramètres

Les paramètres servent à « envoyer des informations » à la fonction, et ainsi la rendre plus générique (polyvalente).

```
def ma_fonction1(a, b, c):    # Fonction à 3 paramètres obligatoires
    ....

def ma_fonction2(a, b, c=0): # Le paramètre c est optionnel (on lui donne une valeur par défaut)
    ....

def ma_fonction3(a, b=0, c=0):    # Il est possible d'avoir plusieurs paramètres optionnels...
    ....

ma_fonction3(5, c=2) # ... dans ce cas il faudra spécifier le nom du paramètre optionnel à l'appel
```

## Le return

Le return a 2 rôles:

- Pouvoir sortir d'une fonction à tout moment (sans retourner de valeur)
- Pouvoir sortir d'une fonction à tout moment en retournant une valeur.

Il est inutile et déconseillé de mettre « return » à la fin d'une fonction si on ne retourne aucune valeur.

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

def afficher_infos(nom, age):
    if nom == '':
        print(« Erreur, le nom ne doit pas être vide »)
        return
    print(« La personne est », nom, « son age est », age)
```

# Il est possible de retourner plusieurs valeurs

```
def get_data()
    return 4560, 8748
```

```
a, b = get_data()
```

# Envie d'aller plus loin avec Python ?



👉 Accédez à la formation complète pour obtenir enfin **des bases solides en programmation** et savoir créer **tous types de projets** : Web, Jeux, Applications bureau et mobile.

<https://codeavecjonathan.com/formations>

---

A bientôt dans la formation !



Jonathan Roux  
[jonathan@codeavecjonathan.com](mailto:jonathan@codeavecjonathan.com)