



## UE121 - Introduction à la programmation

HAUTE ÉCOLE DE NAMUR-LIÈGE-LUXEMBOURG

Technologie de l'informatique - bloc 1

Sécurité des systèmes - bloc 1

# Exercices 3 – Fonctions avec paramètres et avec retour

### Objectifs

- manipuler les notions élémentaires : variables, affectations, entrées/sorties, expressions, instructions de contrôle de flux...
- définir et utiliser des fonctions
- veiller au Clean Code,
- veiller à la portabilité du programme

### A. Introduction

Les séries d'exercices visent à mettre en pratique les notions vues lors des ateliers et des séances de mise en commun. Il est important de faire un maximum d'exercices pour vous familiariser avec l'IDLE et avec le langage Python.

Au cas où vous ne termineriez pas les exercices durant la séance, nous vous conseillons de les achever chez vous au plus tôt et surtout avant l'atelier suivant.

Cette deuxième série concerne les structures de contrôle, c'est-à-dire les alternatives, les répétitives et les appels de fonction. La notion de fonction est approfondie dans le module suivant.

#### Note.

Dans les exemples qui suivent, les passages en texte normal sont à sortir tels quels.

Les parties en *italique souligné* correspondent aux entrées de l'utilisateur.

Les portions en **gras** varient en fonction des entrées.

Sur votre partition de travail (U:), créez un répertoire appelé Python. Ensuite, pour chaque exercice ci-après, ajoutez-y un fichier intitulé Ex3-XX.py avec XX à remplacer par le numéro de l'exercice en question.

## B. Exercices à réaliser dans l'IDLE

### Exercice 1

Écrivez une fonction permettant d'afficher le produit des deux nombres passés en paramètres.

### Exercice 2

Écrivez une fonction permettant de renvoyer le plus grand des deux nombres passés en paramètres.

### Exercice 3

Écrivez une fonction permettant d'afficher alternativement "Tic" ou "Tac" chaque seconde pendant un nombre entier de secondes passé en paramètre. Utilisez la fonction `sleep()` du module `time`.

### Exercice 4

Écrivez une fonction permettant d'afficher l'adresse IPv4 formée des quatre nombres passés en paramètres. Si ce n'est pas une adresse valide, afficher un message d'erreur à la place. Servez-vous du paramètre `sep` de la fonction `print()`. (Consultez la documentation)

### Exercice 5

Écrivez une fonction permettant d'afficher un message d'accueil composé d'une formule de politesse et du nom d'une personne passés en paramètres (ex: "Bonjour Alice !"). Utilisez la valeur par défaut "Bienvenue" pour le paramètre de la formule de politesse.

### Exercice 6

Écrivez une fonction permettant d'afficher la table de multiplication d'un nombre passé en paramètre. La fonction prend deux autres paramètres, début et fin, qui définissent l'intervalle de la table à afficher. Par défaut, de 1 à 10 inclus. Testez la fonction en vous servant des paramètres nommés.

### Exercice 7

Réécrivez l'exercice 4 en ajoutant une fonction déterminant si un nombre est représentable sur un byte ou non. Utilisez cette fonction pour vérifier la validité de l'adresse.

### Exercice 8

Écrivez une fonction permettant de renvoyer le nombre de jours dans un mois, dont le nom est passé en paramètre. Un autre paramètre définit si l'année est bissextile (défaut : non).

---

### Exercice 9

---

Écrivez une fonction permettant d'afficher l'équivalent majuscule (ou minuscule) d'une lettre passée en paramètre. Si la lettre en paramètre est minuscule, il faut afficher la lettre majuscule et inversement. Utilisez les fonctions natives de Python `ord` et `chr`.

Commentez votre code là où vous pensez que cela est nécessaire.

---

### Exercice 10

---

Écrivez une fonction permettant de déterminer si une année passée en paramètre est bissextile (elle l'est si elle est divisible par 4 ET non-divisible par 100 OU divisible par 400). Créez ensuite une version modifiée de l'exercice 8 qui utilise cette fonction.

Commentez votre code là où vous pensez que cela est nécessaire.

---

### Exercice 11

---

Écrivez une fonction permettant de déterminer si un nombre entier positif passée en paramètre est un nombre premier. Ne vous en faites pas pour le moment si votre algorithme n'est pas efficace.

Commentez votre code là où vous pensez que cela est nécessaire.

---

### Exercice 12

---

Écrivez une fonction permettant de renvoyer la division des deux entiers passés en paramètres.

Écrivez une *docstring*, contenant notamment les conditions sur les paramètres, et leurs *type hints*.

---

### Exercice 13

---

Écrivez une fonction permettant de renvoyer la racine carrée du nombre réel passé en paramètre.

Écrivez une *docstring*, contenant notamment les conditions sur les paramètres, et leurs *type hints*.

---

### Exercice 14

---

Écrivez une fonction permettant de renvoyer l'aire d'un disque dont le rayon est passé en paramètre. Utilisez le module *math* pour obtenir la valeur de  $\pi$ .

Écrivez une *docstring*, contenant notamment les conditions sur les paramètres, et leurs *type hints*.

---

## Exercice 15

---

Écrivez un script comprenant un ensemble de fonctions implémentant les fonctionnalités demandées. Écrivez les *docstring* de ces fonctions, les *type hints* et commentez le code si nécessaire.

Votre script est un petit bot avec lequel vous pourrez interagir. Il doit être capable de :

- Demander à l'utilisateur de choisir l'action qu'il veut faire grâce au clavier (prévoyez une action pour quitter le programme).
- Exécuter l'action correspondante à son choix tant qu'il n'a pas choisi de quitter le programme

Les actions sont :

- Afficher une blague à l'écran au hasard (prévoyez en au moins 3).
- Jouer à la devinette de nombre (adaptez votre code de l'exercice 2.12, pour jouer avec un nombre au hasard).
- Afficher un décompte des secondes depuis un plafond défini par l'utilisateur, jusqu'à 0.

Toutes ces fonctionnalités doivent prendre la forme d'une fonction. Servez-vous des modules *time* et *random*.