

# Python

## Notions de base

**Exercice 1:** Ecrire un programme Python qui permet d'afficher le message suivant : *Bonjour*.

**Exercice 2:** Ecrire un programme Python qui collecte la réponse de l'utilisateur à la question : *Est-ce que vous aimez le Python ?* Puis l'affiche sous la forme : *Vous avez répondu :* suivie de la réponse de l'utilisateur.

**Exercice 3:** Ecrire un programme Python qui demande à l'utilisateur les deux côtés d'un rectangle, puis calcule le périmètre et l'aire de ce rectangle et les affiche.

**Exercice 4:** Ecrire un programme Python qui lit trois entiers, calcule puis affiche leur moyenne. La moyenne sera affichée de deux façons :

- directement comme elle est ;
- directement comme elle est en utilisant la méthode *format* ;
- avec une précision de 3 chiffres après la virgule en utilisant la méthode *format*.

**Exercice 5:** Ecrire un programme Python qui lit un nombre réel  $x$ , calcule puis affiche son image, avec une précision de 2 chiffres après la virgule, par la fonction  $f$  définie comme suit :

$f(x) = x^2 - 2x + 7$ . Ceci en respectant la forme d'affichage donnée sur l'exemple suivant :

$f(7.36) = 46.45$

## Conditions

**Exercice 1:** Ecrire un programme Python qui lit les notes de deux contrôles d'une matière donnée, puis calcule et affiche la moyenne obtenue. Dans le cas où la moyenne est supérieure ou égale à 10, il affichera davantage le message de félicitation suivant : *Félicitations ! Vous avez validé cette matière*.

**Exercice 2:** Ecrire un programme Python, qui lit un nombre réels  $x$ , puis affiche son inverse  $1/x$ . Dans le cas où  $x$  est nul, on affichera plutôt le message : *L'inverse n'existe pas pour 0*.

- $a < b$  ;
- $a > b$ .

**Exercice 4:** Ecrire un programme Python qui lit une note entre 0 et 20 puis affiche :

- *Non validé*, si :  $note < 10$  ;
- *Passable*, si :  $10 \leq note < 12$  ;
- *Assez bien*, si :  $12 \leq note < 14$  ;
- *Bien*, si :  $14 \leq note < 16$  ;
- *Très bien*, si :  $16 \leq note < 18$  ;
- *Excellent*, si :  $18 \leq note \leq 20$  ;

**Exercice 5:** Ecrire un programme Python qui choisi aléatoirement un nombre secret entre 1 et 10, puis demande à l'utilisateur de le prédire et donne l'un des verdicts suivants :

- *Bravo !* si le nombre a été bien prédit ;
- Donner une deuxième et dernière chance à l'utilisateur si le nombre entré est juste en dessus ou en dessous du nombre secret et d'afficher :
  - *Bon travail !* si cette deuxième tentative est correcte ;
  - *Pas correct !* si elle est fausse.
- *Pas correct !* si la première tentative est loin du nombre secret.

## Boucles

**Exercice 1:** Ecrire un programme Python qui lit un entier  $n$  puis calcule et affiche la somme des entiers de 1 jusqu'à  $n$  en utilisant la boucle *for*.

**Exercice 2:** Ecrire un programme Python qui lit un entier  $n$  et affiche tous les nombres pairs compris entre 0 et  $n$  (les deux inclus) sur une seule ligne et séparés par des espaces.

**Exercice 3:** Ecrire un programme Python qui lit un entier  $n$ , puis affiche tous les entiers de  $n$  jusqu'à 1 sur une seule ligne et séparés par des espaces. Par exemple si l'utilisateur entre 5, on affichera : 5 4 3 2 1.

**Exercice 4:** Ecrire un programme Python qui lit une somme d'argent puis une liste de dépenses, une par une, jusqu'à ce que la somme s'épuise où devient insuffisante. Afficher dans ce dernier cas le supplément d'argent nécessaire pour couvrir toutes les dépenses y compris la dernière.

actuel. Et arrête le jeu, une fois que l'utilisateur donne une mauvaise réponse et puis affiche le score final.

## Chaînes de caractères

**Exercice 1:** Ecrire un programme Python qui lit une chaîne de caractères, puis l'affiche en séparant ces caractères par des espaces sauf pour l'espace lui-même.

**Exercice 2:** Ecrire un programme Python qui lit séparément le nom et le prénom d'une personne. Puis lit son sexe, à travers un menu qui demande d'entrer 1 pour une personne de sexe masculin, et 2 pour une personne de sexe féminin. Composer par ces données une chaîne de caractères contenant le nom complet constitué de : *Titre Prénom Nom*. Où le titre est soit *M.* ou *Mme*. Afficher ensuite un message de remerciement.

**Exercice 3:** Ecrire un programme Python qui lit une chaîne de caractères puis calcule et affiche sa taille en utilisant les deux méthodes suivante :

- la fonction *len*;
- la boucle *for*.

**Exercice 4:** Ecrire un programme Python qui lit une chaîne de caractères et un caractère, et retrouve la position de la première occurrence de ce caractère dans la chaîne, par les deux méthodes suivantes :

- la boucle *for*;
- la méthode

Si le caractère ne se trouve pas dans la chaîne, on affiche : *Ce caractère ne figure pas sur la chaîne*.

**Exercice 5:** Ecrire un programme Python qui lit une chaîne de caractères et indique si elle est palindrome ou non. On rappelle qu'une chaîne de caractères est palindrome lorsqu'elle se lit dans les deux sens de la même manière. Exemple : kayak, non,...

## Listes

**Exercice 1:** Ecrire un programme Python qui lit une liste d'entiers, puis l'affiche, et affiche ensuite seulement ses nombres pairs.

**Exercice 3:** Ecrire un programme Python qui lit une liste de mots, les réordonne dans l'ordre alphabétique, en utilisant la méthode *sort*, puis les affiche.

**Exercice 4:** Ecrire un programme Python qui lit une liste d'entiers, et supprime toutes les occurrences d'un entier entré par l'utilisateur, en utilisant la méthode *pop*.

**Exercice 5:** Ecrire un programme Python qui lit une liste d'entiers, la trie, puis lit un entier et utilise la méthode *insert* pour l'insérer dans la liste de telle sorte que la liste reste triée.

## Fonctions

**Exercice 1:** Ecrire un programme Python qui définit et appelle une fonction *Bonjour* qui affiche le message : *Bonjour!*.

**Exercice 2:** Ecrire un programme Python qui définit une fonction *Conversion* qui prend comme argument un nombre entier et l'affiche dans les trois bases ; binaire, octale et hexadécimale, en exploitant les fonctions de conversion *bin*, *oct* et *hex*. Utiliser cette fonction pour afficher les équivalents binaire, octal et hexadécimal d'un entier entré par l'utilisateur.

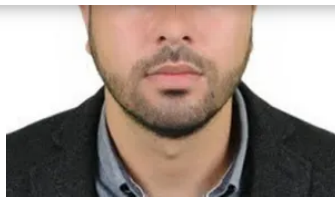
**Exercice 3:** Ecrire un programme Python qui définit une fonction *Somme* qui prend deux nombres en arguments et renvoie leur somme. Utiliser ensuite, cette fonction pour additionner deux entiers, puis deux nombres réels entrés par l'utilisateur.

**Exercice 4:** Ecrire un programme Python qui définit une fonction *Lire* qui permet de lire une liste d'entiers. Puis utiliser cette fonction pour lire deux listes. Assembler ensuite les deux listes dans une troisième et afficher les trois.

**Exercice 5:** Ecrire un programme Python qui définit une fonction *Positif* qui prend un entier comme argument, et renvoie *True* s'il est positif et *False* sinon. Définir ensuite la fonction *AfficherPositifs* qui prend comme argument une liste et affiche ses nombres positifs en exploitant la fonction *Positif*. Lire ensuite une liste d'entiers et afficher ses nombres positifs en faisant appel à la fonction *AfficherPositifs*.

**Cours particuliers à distance**

The C Coder



Pour préparer vos examens et surmonter vos difficultés en programmation ..

Algorithmique, Langage C, Python et Java

..profitez des séances de soutien à distance en me contactant sur mon email.

Cours effectués avec partage d'écran via Microsoft teams, Google Meet, Skype ...

Email: mohamed.tirichen@gmail.com

Mohamed TIRICHEN, Editeur du site

Tarif : **25 €/heure**

[ACCUEIL](#)

[EBOOKS](#)

[EXERCICES C](#)

[PYTHON](#)

[YOUTUBE](#)

Hestia | Développé par Themelsle