

Programmer avec Python
Exercices

Séance n° 1	Révisions - bases de la programmation
-------------	---------------------------------------

Objectif : Réalisez chacun des exercices en utilisant le logiciel Spyder comme il a été expliqué en cours. Enregistrez bien votre travail au fur et à mesure !

1 Poids sur la Lune

Écrivez un programme Python qui demande à l'utilisateur de saisir une masse m puis lui affiche le poids P correspondant sur la Lune. La formule de calcul du poids est :

$$P = g \times m$$

Sur la Lune, $g = 1.6 \text{ N.kg}^{-1}$.

2 Indice de Masse Corporelle (IMC)

Écrivez un programme Python qui demande à l'utilisateur de saisir son poids m et sa taille t puis lui affiche son IMC. La formule de calcul de l'IMC est :

$$IMC = \frac{m}{t^2}$$

Prenez garde à $t...$

3 E ?

Écrivez un programme Python qui demande à l'utilisateur de saisir un mot au clavier puis lui affiche le nombre de fois où apparaît la lettre e dans ce mot.

4 Calcul de la factorielle

Écrivez un programme Python qui demande à l'utilisateur de saisir un entier au clavier puis lui affiche la factorielle de ce nombre. Pour rappel, la factorielle de 5 par exemple est :

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

5 Formule de Riegel

La formule de Riegel permet de prédire son futur résultat à une course à pied grâce à un temps de référence. Autrement dit, à partir d'un résultat sur une course récente, un 5km par exemple, qu'on appellera le temps t_1 , on peut calculer le chrono qu'on pourrait obtenir sur une autre course de 10K, un semi-marathon ou un marathon qu'on appellera t_2 . La formule de Riegel est la suivante :

$$t_2 = t_1 \left(\frac{d_2}{d_1} \right) \times 1,06$$

Avec :

- t_2 : le temps qu'on cherche à prédire
- t_1 : le temps de référence
- d_2 : la distance sur la course à prédire
- d_1 : la distance sur la course de référence

1. Écrivez une fonction Python qui prend en entrée le temps t_1 en secondes, les distances d_1 et d_2 en mètres et qui renvoie en sortie le résultat t_2 selon la formule de Riegel. Si d_1 est égal à 0, la fonction renverra 0.
2. Écrivez un programme principal demandant à l'utilisateur de saisir son temps de référence sur une course, la distance de la course de référence et la distance de la course qu'il veut prédire puis qui lui affiche le temps prédit par la formule de Riegel.

6 Calcul de la fréquence cardiaque maximale

La fréquence cardiaque maximale (FCM) correspond au rythme cardiaque maximal qu'un individu peut atteindre lors d'un effort. Il correspond au nombre maximal de battements du cœur en une minute. La FCM correspond à l'intensité maximale lors d'un effort physique ; elle diminue progressivement avec l'âge. La meilleure façon de connaître sa FCM est de la déterminer avec un test en laboratoire. Elle peut aussi être estimée grâce à cette formule :

$$FCM = 192 - 0,007 \times age^2$$

1. Écrivez une fonction Python qui prend en entrée un âge et qui renvoie en sortie la valeur de FCM estimée par cette formule.
2. Écrivez un programme principal demandant à l'utilisateur de saisir son âge et qui affiche ensuite sa FCM estimée. L'âge saisi par l'utilisateur doit être strictement positif. Si l'utilisateur saisit une valeur invalide, le programme lui demandera de recommencer jusqu'à ce que la valeur soit correcte.

7 Ticket de caisse (DS 2021)

Écrivez un programme Python permettant de calculer et d'afficher un ticket de caisse.

Au lancement, le programme demandera à l'utilisateur combien d'articles comporte le ticket.

Ensuite, pour chaque article, le programme demandera à l'utilisateur de saisir la quantité achetée, le prix hors taxe de l'article et si l'article bénéficie d'un taux de TVA réduit à 5.5%. Si l'article ne bénéficie pas du taux de TVA réduit, le taux normal de TVA à 20% sera appliqué.

Le programme affichera pour chaque article le sous-total hors taxe et toutes taxes comprises. Une fois tous les articles saisis, le programme affichera le total du ticket, hors taxe et toutes taxes comprises.

8 Structures conditionnelles (DS Rattrapage 2021)

On veut simuler le passage d'un étudiant en année supérieure en fonction de ses moyennes à 3 unités d'enseignement (UE), simulées par 3 notes n_1 , n_2 et n_3 . Les règles sont les suivantes :

- Si les 3 notes sont supérieures à 10, l'élève passe.
- Si au moins une des notes est inférieure à 8, l'élève redouble.
- Si aucune note n'est inférieure à 8 et que leur moyenne est supérieure à 10, l'élève passe. Sinon, il redouble.

Écrivez un programme qui permet de déterminer si l'étudiant valide son année. Le programme devra préciser la raison pour laquelle l'étudiant passe/redouble. Dans les cas où l'étudiant redouble, il faudra afficher la somme des points qui lui manque dans chaque UE où il n'a pas eu la moyenne. Ex : S'il a eu 8.5, 9.5 et 11, on devra afficher « 2 » ($1.5 + 0.5$).

9 Fonctions (DS Rattrapage 2021)

Un nombre est dit parfait si la somme de ses diviseurs (autres que lui-même) est égale à lui-même.

Exemple :

- 6 est divisible par 1, 2 et 3 (et lui-même que l'on ne compte pas).
- $1 + 2 + 3 = 6$
- 6 est donc un nombre parfait

NB : Les questions 2 et 3 peuvent être réalisées en supposant que les fonctions des questions précédentes sont déjà écrites et fonctionnent.

1. Écrire une fonction qui prend en paramètre un entier n , et qui retourne vrai si n est un nombre parfait, faux sinon.
2. Écrire une fonction qui prend en paramètre 2 entiers a et b et affiche tous les nombres parfaits compris entre a et b . Votre fonction doit bien sûr utiliser la fonction écrite à la question 1.
3. Écrire un script de test qui demande 2 entiers a et b à l'utilisateur, et affiche tous les nombres parfaits compris entre a et b (utilisez la fonction écrite lors de la question 2).