Runtrack Python

Python is powerful... and fast; and open; and ... many other things.

Job 01

Écrire un programme qui affiche dans la console un rectangle avec des « **-** » et des « **|** » en fonction des paramètres d'entrées, (**width, height**), par exemple :

draw_rectangle(10, 3) :

```
|--------|
|        |
|--------|
```

Job 02

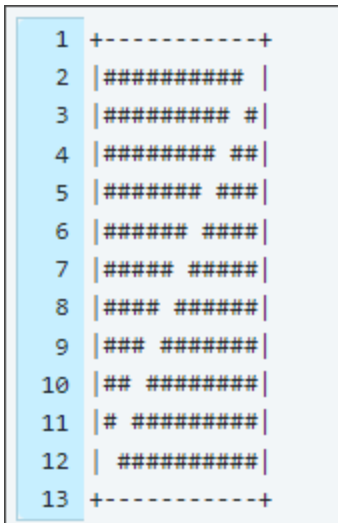
Écrire un programme qui affiche dans la console un triangle avec des « **_** », des « **** » et des « **/** » en fonction de l'input entré, qui représentera la hauteur.

Exemple si l'input entré est 5 :

Job 03

Écrire une fonction qui, recevant une taille **n** en paramètre, affiche un tapis de n+1 ligne/n+1 colonne traversé par une diagonale.

Exemple pour une taille de 10 :

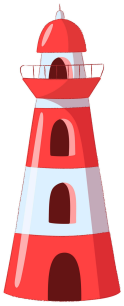


Job 04

Jules César, général et stratège romain, a été le premier militaire officiel à chiffrer ses messages. Sa méthode était assez simple : il décalait les lettres de **trois rangs** dans l'alphabet.

Créer une fonction à laquelle on donne un **message** et un **décalage**, et la fonction renvoie alors le message **décalé dans l'alphabet**. Il faudra gérer le dépassement (« z » décalé vers la droite revient sur « a », et « a » décalé vers la gauche revient sur « z »).

Job 05



Un gardien de phare va aux toilettes **cinq fois par jour**. Or les WC sont au rez-de-chaussée…

Écrire une fonction qui reçoit en paramètres, le nombre de marches du phare et la hauteur de chaque marche (en cm), cette fonction doit calculer combien de mètre le gardien effectué par semaine pour aller aux toilettes. La sortie du code doit être :

Pour **x** marches de **y** cm, le gardien parcourt **z.zz m** par semaine.

On n'oubliera pas :

- ➔ Qu'une semaine comporte 7 jours ;
- ➔ Qu'une fois en bas, le gardien doit remonter ;
- ➔ Que le résultat est à exprimer en m.

Job 06

Luke Skywalker, un professeur de Math, fait passer un test et décide de noter ses élèves sur une échelle allant de **0 à 100 inclus.**

Si un étudiant obtient moins de 40 sur 100, **il échoue au test.**

S'il a plus de 40, **il réussit le test**. Luke est un professeur fort sympathique et décide donc d'arrondir à la hausse les notes des étudiants ayant réussi le test. Mais Luke n'est quand même pas trop gentil. Cet arrondi à la hausse ne bénéficiera qu'aux étudiants remplissant certains critères, car tout de même, il ne faut pas exagérer, sans blague.

Le critère est simple : Si un étudiant a eu une note de moins de strictement 3 points de son prochain multiple de 5, alors sa note est arrondie à ce multiple de 5. Par exemple, un 83 sera arrondi à 85 alors qu'un 82 restera un 82.
Pour simplifier le travail de Luke, écrivez une fonction qui prend en paramètre une liste de notes et qui renvoie une liste de notes, arrondies comme il se doit, quand cela est nécessaire.

... Pour aller plus loin

Job 08

Créer une fonction nommée « **my_sort** » qui compare les éléments d'une liste et échange de place uniquement avec l'élément de la liste adjacente jusqu'à que la liste soit entièrement triée dans l'ordre croissant. La fonction doit retourner la liste triée et afficher le nombre total de coups nécessaire pour trier cette liste.

Pensez à commenter chaque étape de votre logique.

```
Nombre total de coups nécessaires pour trier la liste : 14
Liste triée : [11, 12, 22, 25, 34, 64, 90]
```

Compétences visées

- ➔ Installer un environnement de développement python
- ➔ Maîtriser les bases de python
- ➔ Implémenter un algorithme

Rendu

Pour valider votre **jour 6** remplir le QCM.

Base de connaissances

➔ Les bases du développement en python