



La Plateforme

Runtrack Python

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



Job 01

Créez un programme qui demande à l'utilisateur de renseigner son prénom via l'invite de commande grâce à la fonction `input()`. Le programme doit alors afficher dans le terminal "Hello xx !" ou xx est le prénom entré par l'utilisateur.

Job 02

Écrire un programme qui affiche dans le terminal un rectangle avec des '-' et des '|' en fonction des paramètres d'entrées, (width, height), par exemple :

```
draw_rectangle(10, 3)
```

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

Job 03

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

Exemple pour une taille de 10 :

```
1 +-----+
2 |#####|
3 |##### #|
4 |##### ##|
5 |##### ###|
6 |##### ####|
7 |##### #####|
8 |##### #####|
9 |##### #####|
10 |##### #####|
11 |##### #####|
12 |##### #####|
13 +-----+
```

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 3 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 parcours 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 07

Créer un programme qui demandera à l'utilisateur de renseigner un mot et un seul, sans espace ni aucun autre caractère que les 26 lettres de l'alphabet (sans accent ni majuscule).

Votre programme devra modifier ce mot, en y changeant de place certains caractères (ou tous) afin de donner un mot plus "loin" dans l'ordre alphabétique que le mot renseigné par l'utilisateur.

Attention: Le nouveau mot doit être le mot le plus proche possible, dans l'ordre alphabétique, du mot original !

Par exemple, "abcde" donnerait "abced". "acedb" est aussi "valide" mais n'est PAS le plus proche du mot original dans l'ordre alphabétique.

Rendu

Créer un dossier "**jour05**", et pour chaque étape, un dossier "jobXX" où XX est le numéro de l'étape.

Comment rendre ce projet ? Seul l'avenir nous le dira !

Compétences visées

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

Base de connaissances

- [Python.org](https://python.org)
Site officiel python, documentation et téléchargement.
- [Tutoriel python](#)
Les bases du développement en python