



Runtrack Python

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

Les fonctions en programmation sont des morceaux de code que l'on peut réutiliser pour effectuer des tâches spécifiques. Elles aident à organiser le code, le rendent plus lisible, et évitent la duplication. Les fonctions permettent aussi de passer des informations (*paramètres*) et de recevoir des résultats (*retours*). En les utilisant, on peut simplifier des tâches complexes, faciliter la maintenance du code, et rendre le programme plus modulaire et compréhensible. Les fonctions sont comme des outils pratiques qui rendent le développement de logiciels plus efficace.





Job 01

Créer une fonction nommée **My_print_hello**. Cette fonction affichera « **Hello world** » dans le terminal à son appel.

Job 02

Créer une fonction nommée « **My_print_name** ». Cette fonction prendra un variable nommé « **name** » en paramètre. Elle affichera la valeur de name. Depuis votre programme, appelez cette fonction plusieurs fois en y passant des paramètres différents pour afficher ces résultats.

Job 03

Créer une fonction nommée « **Add** ». Cette fonction devra prendre **deux nombres entiers** en paramètres et afficher la **somme** de ces deux entiers. Depuis votre programme, appelez cette fonction plusieurs fois en y passant des paramètres différents pour afficher ces résultats.

Job 04

Créez une fonction nommée « **GetHello** ». Cette fonction doit retourner « **Hello la Plateforme** ». Depuis votre programme, appelez cette fonction et affichez le résultat.

Job 05

Créer une fonction nommée « **calcule** » qui prend **3 paramètres** :

- Le premier, « **num1** », est un nombre,
- Le deuxième, « **operator** », est un caractère (*string*) contenant le type d'opération(+, -, *, /, %),



→ Le troisième, « **num2** », est un nombre.

La fonction doit retourner le résultat de l'opération.

Job 06

Créer une fonction qui prend en paramètre un nombre nommé « **nombre** ».

Afficher « **positif** » si le nombre est supérieur à 0.

Afficher « **négatif** » si le nombre est inférieur à 0.

Appeler cette fonction plusieurs fois en y passant des paramètres différents pour afficher ces résultats.

Job 07

Créez une fonction qui prend en paramètre une String nommée « **langage** ».

Si la valeur de « **langage** » est égale à « **JavaScript** » affichez : « **tu es un développeur web** »

Sinon si la valeur de « **langage** » est égale à « **python** » affichez : « **tu es un développeur IA** »

Sinon si la valeur de « **langage** » est égale à « **java** » affichez : « **tu es un développeur logiciel** »

Sinon si la valeur de « **langage** » est égale à « **reactjs** » affichez : « **tu es un développeur mobile** »

Sinon, affichez : « **un jour, je serai le meilleur développeur...** »



Job 08

Créer une fonction qui prend **2 paramètres** :

- Le premier reçoit un String nommé « **type** »
- Le second reçoit un String nommé « **saison** »

Si la valeur de « **type** » est égale à « **fruits** » et que celle de saison est égale à « **hiver** », affichez « **orange, mandarine et kiwi** »

Si la valeur de « **type** » est égale à « **fruits** » et que celle de saison est égale à « **été** », affichez « **Poire, fraise, cassis** »

Si la valeur de « **type** » est égale à « **légume** » et que celle de saison est égale à « **hiver** », affichez « **carotte, topinambour, endive** »

Si la valeur de « **type** » est égale à « **légume** » et que celle de saison est égale à « **été** », affichez « **artichaut, aubergine,navet** ».

Job 09

Créez une fonction nommée « **moyenne** » qui prend en paramètre trois notes et retourne la moyenne de ces notes.

Demandez à l'utilisateur de saisir trois notes, puis enregistrez le résultat de la fonction « **moyenne** » dans une variable appelée « **moyenne_etudiant** ».

Afficher ensuite la phrase suivante en fonction de la moyenne de l'étudiant :

- **Très bon élève** si la moyenne est comprise entre **20 et 15**.
- **Bon élève** si la moyenne est comprise entre **14 et 11**.
- **Élève moyen** si la moyenne est comprise entre **10 et 8**.
- **Élève devant faire des efforts** si la moyenne est comprise entre **0 et 7**.



Job 10

Créer une fonction qui vérifie que le nombre est pair ou impair. **Pensez à vérifier si le nombre est bien un chiffre entier et positif.**

Appeler cette fonction plusieurs fois avec des valeurs différentes.

Job 11

Créer une fonction nommée « **time_to_text** » qui prend en paramètre un « **nombre entier** » de minutes et « **affiche en console** » une chaîne de caractère sous la forme de : « **X heures et Y minutes** ».

```
2 heures et 40 minutes
```

... pour aller plus loin.

Créer une fonction qui prend en paramètre **un string** et qui la retourne inverser. Par exemple, « nikana » deviendra « anakin ».

Compétences visées

- Maîtriser les bases de python
- Implémenter un algorithme
- Maîtriser les fonctions



Rendu

Créer sur github un répertoire nommé "runtrack-python". Créer dans ce répertoire un dossier « **jour03** », et pour chaque étape, un dossier « **jobXX** » où **XX** est le numéro de l'étape.

Base de connaissances

- [Site officiel python, documentation et téléchargement.](#)
- [Les bases du développement en python](#)
- [Les fonctions](#)