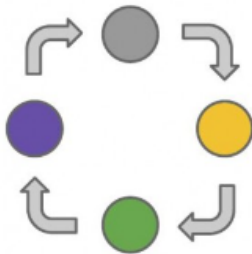




# Runtrack Python

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



En Python, les boucles et les conditions sont des structures de contrôle fondamentales qui facilitent la gestion du flux d'exécution dans un programme. Les boucles « **for** » et « **while** » permettent d'itérer à travers des séquences de données ou d'exécuter un

bloc de code tant qu'une condition spécifique est vraie.

## Job 01

---

Créez un programme qui parcourt les nombres de **0** à **20**. **Affichez** chacun des chiffres dans le terminal.

## Job 02

---

```
0
2
4
6
8
10
12
14
16
18
```

Créez un programme qui parcourt les nombres de **0** à **20**.  
**Afficher 1 nombre sur 2** dans le terminal.



## Job 03

---

Créer un programme qui **parcourt** les chiffres de **1** à **20** et **affiche** le carré de ces nombres.

## Job 04

---

```
Entrez un entier supérieur à zéro (N) :  
Table de multiplication de 1 :  
1 x 1 = 1  
1 x 2 = 2  
1 x 3 = 3  
1 x 4 = 4  
1 x 5 = 5  
1 x 6 = 6  
1 x 7 = 7  
1 x 8 = 8  
1 x 9 = 9  
1 x 10 = 10
```

```
Table de multiplication de 2 :  
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18  
2 x 10 = 20
```

Créez un programme qui affiche dans le terminal **les tables de multiplications** de **1** à **N**. N étant un **entier** supérieur à zéro saisi par l'utilisateur.

## Job 05

---

Transformer cette boucle **for** en boucle **while** afin d'obtenir le même résultat en console.

```
for N in range(1, 13):  
    print(N)
```



## Job 06

---

Écrire un programme qui **affiche** à l'aide d'une boucle de type **while** les premiers résultats de la multiplication de **N** par **7**. **N** étant un entier renseigné par l'utilisateur.

## Job 07

---

Créer une boucle qui effectue **12 tours** et **affiche** le triple du tour moins 2.

**Résultat attendu :**

```
Tour 1: 1
Tour 2: 4
Tour 3: 7
Tour 4: 10
Tour 5: 13
Tour 6: 16
Tour 7: 19
Tour 8: 22
Tour 9: 25
Tour 10: 28
Tour 11: 31
Tour 12: 34

[Execution complete with exit code 0]
```

## Job 08

---

Créer une boucle qui effectue 12 tours qui affichent le numéro du tour précédent + 2.

**Résultat attendu :**

```
Tour 1: 2
Tour 2: 4
Tour 3: 6
Tour 4: 8
Tour 5: 10
Tour 6: 12
Tour 7: 14
Tour 8: 16
Tour 9: 18
Tour 10: 20
Tour 11: 22
Tour 12: 24
```



## Job 09

---

Uniquement à l'aide d'une **boucle**, **afficher** les nombres **pairs** et **impairs** de 1 à 30.

## Rendu

---

Que vous aillez fini vos jobs ou pas, rendre sur le projet sur votre intranet et remplir le [QCM](#).

## Compétences visées

---

→ Maîtriser les bases de Python

## Base de connaissances

---

- [Les bases du développement en Python](#)
- [Les boucles](#)