

Comment fonctionnent les boucles en python

Une boucle sert à répéter plusieurs fois la même suite d'instructions. C'est donc particulièrement pratique quand on veut répéter un morceau de code de nombreuses fois sans avoir à le recopier.

Pour faire ça, plusieurs instructions sont disponibles en python, par exemple l'instruction `for` qui permet notamment de répéter du code un nombre déterminé de fois. Pour ça on doit passer à l'instruction une fonction qui va décrire ce que l'on veut faire et une variable pour récupérer le résultat de cette fonction.

Petit exemple:

```
for i in range(10):
    print(i)
```

Ce programme affiche les chiffres de 0 à 9 grâce à la fonction `range`. Celle-ci permet de boucler entre deux valeurs : dans `range(10)`, les valeurs vont de 0 à 9 (10 non inclus).

On peut aussi préciser la borne de départ, par exemple : `range(6, 21)` boucle de 6 à 20.

Les chaînes de caractères (strings) en python

Une string est simplement le nom que l'on donne à du texte en Python : "bonjour", "ABC", "1234", etc. Ce type est très utilisé, notamment pour manipuler des codes, des mots de passe ou des messages.

Voici quelques notions utiles pour cet exercice :

1. Transformer un nombre en string

Et oui ici 42 (un entier) et "42" (une string) ne sont pas la même chose, et pour passer de l'un à l'autre on peut utiliser la fonction `str` et lui passer le nombre que l'on veut pour récupérer les chiffres dans une string.

```
str(42) → "42"
```

2. Répéter une string un certain nombre de fois

Certaines opérations mathématiques peuvent être faites sur les string. Par exemple l'opérateur '+' permet de concaténer (coller deux textes l'un à la suite de l'autre) deux string. Mais l'opérateur qui va nous intéresser ici est '*'. Il permet normalement de multiplier mais ici il fait que l'on peut répéter une string un certain nombre de fois.

```
"a" * 5 → "aaaaa"  
"test" * 3 → "testtesttest"
```

3. Compléter une string avec des zéros à gauche

Pratique pour afficher des chiffres avec une longueur particulière. Pour ça les strings offrent une fonction dédiée, `zfill` à laquelle on donne la longueur finale souhaitée et qui retourne la string avec les 0 en plus.

```
"42".zfill(5) → "00042"
```

4. Tester si une string commence par un texte précis

Essentiel pour le traitement automatique du texte, la fonction permet en fonction d'une string de base, de regarder si oui ou non, cette string commence par une autre string passé en paramètre.

```
"text 1".startswith("text") → True  
"test 2".startswith("text") → False
```