

Boucle bornée for

La boucle **for** est communément utilisée lorsque l'on souhaite répéter plusieurs fois une même instruction. Il faut connaître le nombre de fois que vous souhaitez répéter l'instruction.

Vous aurez besoin d'une variable permettant de compter le nombre de fois que vous répétez l'instruction.

En Python, on indique les valeurs que doit prendre cette variable "compteur" dans une liste.

```
deg PYTHON
for i in [1,2,3,4,5]:
    print("Hello!")
```

Dans cet exemple, la variable qui permet de compter est **i**. À la première iteration, **i** prend la valeur du premier élément de la liste, c'est-à-dire 1, et l'instruction est exécutée : on affiche "Hello!". Une fois l'instruction exécutée, **i** prend la valeur du deuxième élément de la liste et l'instruction est à nouveau exécutée : on affiche "Hello!" une deuxième fois. Une fois la boucle terminée, on aura donc affiché cinq fois "Hello!".

for i in range(n)

La plupart du temps, on n'écrira pas la liste des valeurs que prend **i**. On préférera utiliser **range(n)** qui génère la liste des **n** premiers entiers. Et pour afficher cinq fois "Hello!", on écrira :

```
deg PYTHON
for i in range(5):
    print("Hello!")
```

Attention!

`range(n)` génère la liste des n premiers entiers en commençant par 0 (et en finissant donc en $n - 1$). Ainsi :

```
>>> range(5)
[0, 1, 2, 3, 4]
```

Vous pouvez aussi utiliser `range(n, m)` qui liste les entiers entre n et $m - 1$:

```
>>> range(1, 5)
[1, 2, 3, 4]
```

Enfin, la commande `range(n, m, p)` liste les entiers de n à $m - 1$ par pas de p :

```
>>> range(1, 10, 2)
[1, 3, 5, 7, 9]
```

Exercice

Écrire une fonction `somme(n)` qui prend un nombre entier en argument et renvoie la somme des n premiers entiers $1 + 2 + 3 + \dots + n$ et qui contient une boucle `for`.

Autre exercice

Écrire une fonction `puissance(x, n)` qui prend un réel x et un entier naturel n en argument et renvoie la puissance n -ième de x : x^n .