

Nom :	Prénom :	Classe :
-------	----------	----------

## NSI 1re — Boucle bornée (boucle for) #2 : utilisations de range()

Objectifs :

- Comprendre l'utilité de range()
- Découvrir les différentes façons d'utiliser range()

### Introduction

`range()` sert à fabriquer une **suite de nombres entiers**, souvent utilisée pour compter le nombre de tours dans une boucle bornée (boucle for), c'est à dire une boucle pour laquelle on connaît d'avance le nombre d'itérations.

Voici un premier exemple d'utilisation de `range()` :

```
1 for _ in range(3):
2     print('Hello')
```

Dans cet exemple, nous avons créé une boucle bornée qui va itérer 3 fois sur le bloc d'instructions qu'elle contient. Ici, elle permet d'afficher trois fois `'Hello'`.

### Variable auto-incrémentée

Une possibilité offerte par l'utilisation de `range()` dans le cadre d'une boucle bornée est de pouvoir **incrémenter automatiquement une variable**, dont il suffit de donner le nom pour :

1. La définir.
2. La faire incrémenter.

Prenons un nouvel exemple :

```
1 for a in range(3):
2     print(a)
```

Dans cet exemple, nous avons défini une variable `a`, dès la création de la boucle (ligne 1). Cette boucle va itérer 3 fois sur le bloc d'instructions qu'elle contient.

Cet exemple affichera dans la console :

```
0
1
2
```

### ► Exercice 1 : prédire

Essayez de deviner pourquoi ce dernier exemple affiche `0`, `1` puis `2` dans la console Python :

`Range renvoie successivement les valeurs allant de 0 (inclus) à 3 (exclus), donc 0, 1 puis 2`

## ► Exercice 2 : prédir

Pour chaque programme ci-dessous, indiquez à droite ce qui sera affiché dans la console Python.

```
1 for age in range(2):  
2     print("J'ai", age, "ans !")
```

J'ai 0 ans !  
J'ai 1 ans !

```
1 for h in range(4):  
2     print(h)  
3     if h > 2:  
4         print('Partez !')
```

0  
1  
2  
3  
Partez !

```
1 nbre_1 = 5  
2 for nbre_2 in range(11):  
3     res = nbre_1 * nbre_2  
4     print(nbre_1, "*", nbre_2, "=", res)
```

5 \* 0 = 0  
5 \* 1 = 5  
5 \* 2 = 10  
5 \* 3 = 15  
5 \* 4 = 20  
5 \* 5 = 25  
5 \* 6 = 30  
5 \* 7 = 35  
5 \* 8 = 40  
5 \* 9 = 45  
5 \* 10 = 50

## Personnalisation de la valeur de départ et de fin de range()

La personnalisation de la valeur de départ et de fin de `range()` désigne le fait de choisir soi-même à partir de quel entier commence la séquence et à quel entier (non inclus) elle s'arrête.

Voici quelques exemples d'utilisations de `range()` pour illustrer cela :

Instruction Python	Liste des entiers retournés par range()
<code>range(3)</code>	0, 1, 2
<code>range(0, 3)</code>	0, 1, 2
<code>range(1, 3)</code>	1, 2
<code>range(-1, 3)</code>	-1, 0, 1, 2
<code>range(-2, 1)</code>	-2, -1, 0

## ► Exercice 3 : prédir

Pour chaque programme ci-dessous, indiquez à droite ce qui sera affiché dans la console Python.

```
1   for d in range(0, 3):  
2       print(d)
```

0  
1  
2

```
1   for a in range(3):  
2       print(a)
```

0  
1  
2

```
1   for g in range(-3, 0):  
2       print(g)
```

-3  
-2  
-1

```
1   for oy in range(-1, 3):  
2       print(oy)
```

-1  
0  
1  
2

```
1   for hum in range(-2, 2):  
2       print(hum)
```

-2  
-1  
0  
1

## ► Exercice 4 : créer

(a) Créez une boucle permettant l'affichage de trois "bonjour" (str) successifs :

```
for _ in range( 3 ):  
    print( 'bonjour' )
```

(b) Créez une boucle itérant sur les entiers allant de -1 à 3 inclus, et qui affiche chacune de ces valeurs :

```
for i in range( -1, 4 ):  
    print( i )
```

(c) Créez quatre "bonsoir" (str) successifs, en utilisant deux boucles, **imbriquées** l'une dans l'autre :

```
for a in range( 2 ):  
    for b in range( 2 ):  
        print( 'bonsoir' )
```

## ► Exercice 5 : modifier

Aaron souhaite calculer la somme des 5 premiers nombres entiers, en commençant par 1 :

**1 + 2 + 3 + 4 + 5      soit un total de 15**

Sur la papier, aucun soucis, mais en Python ça ne fonctionne pas. Pouvez-vous corriger son erreur ?

```
1 somme = 0
2 for v in range(5):
3     somme = somme + v
4 print(somme)
```

À vous de jouer :

```
somme = 0
for v in range( 1, 6 ):
    somme = somme + v
print(somme)
```

## ► Exercice 6 : modifier

En récupérant votre code pour l'exercice précédent, ajoutez la possibilité de saisir un nombre entier, afin de calculer la somme des nombres entiers allant jusqu'à ce nombre entier inclus.

Illustrations de fonctionnement :

```
Quelle est la valeur max ? 3
La somme allant de 1 à 3 inclus est : 6
```

```
Quelle est la valeur max ? 6
La somme allant de 1 à 6 inclus est : 21
```

À vous de jouer :

```
nbre_max_inclus = int( input( "Jusqu'à quelle valeur souhaitez-vous aller ? " ) )
nbre_max_inclus = nbre_max_inclus + 1    # ou nbre_max_inclus += 1
somme = 0
for v in range( 1, nbre_max_inclus ):
    somme = somme + v
print(somme)
```

## Personnaliser le saut

Hop, autre possibilité avec `range()`, celle de pouvoir personnaliser le saut (hop !).



```
range(valeur_de_départ_incluse, valeur_de_fin_exclue, valeur_du_saut)
```

Voici quelques exemples d'utilisations de `range()` pour illustrer cela :

N°	Instruction Python	Liste des entiers retournés par range()	Commentaire
1	<code>range(5)</code>	0, 1, 2, 3, 4	Seule la valeur de fin exclue est définie.
2	<code>range(0, 5)</code>	0, 1, 2, 3, 4	Début inclus, fin exclue.
3	<code>range(0, 5, 1)</code>	0, 1, 2, 3, 4	Début inclus, fin exclue, saut.
4	<code>range(0, 5, 2)</code>	0, 2, 4	Début inclus, fin exclue, saut.
5	<code>range(0, 5, 3)</code>	0, 3	Début inclus, fin exclue, saut.
6	<code>range(4, -1, -1)</code>	4, 3, 2, 1, 0	Début inclus, fin exclue, saut négatif.
7	<code>range(4, -1, -2)</code>	4, 2, 0	Début inclus, fin exclue, saut négatif.

### ► Exercice 7 : prédire

Pour chaque programme ci-dessous, indiquez à droite ce qui sera affiché dans la console Python.

1 2	<code>for paris in range(2):     print(paris)</code>	0 1
1 2	<code>for berlin in range(-1, 3):     print(berlin)</code>	-1 0 1 2
1 2	<code>for istanbul in range(5, 0, -1):     print(istanbul)</code>	5 4 3 2 1  # PS : c'est Ankara la capitale, pas Istanbul !
1 2	<code>for hanoi in range(1, 7, 2):     print(hanoi)</code>	1 3 5
1 2	<code>for luanda in range(4, -1, -2):     print(luanda)</code>	4 2 0

## ► Exercice 8 : créer

Créez le programme contenant une boucle itérant sur les valeurs allant de **3** à **1** incluses, et qui affiche chacune de ces valeurs, puis qui affiche à la fin "Partez!" :

Illustrations de fonctionnement :

```
3  
2  
1  
'Partez!'
```

À vous de jouer :

```
for i in range ( 3, 0, -1 ):  
    print( i )  
print('Partez!')
```

## ► Exercice 9 : créer

Créez le programme demandant un maximum et une valeur de saut, puis qui affiche toutes les valeurs comprises entre **0** et ce maximum, en prenant en compte le saut.

Illustration de fonctionnement :

```
Valeur max (inclus) ? 6  
Valeur du saut ? 2  
0  
2  
4  
6
```

À vous de jouer :

```
max_inclus = int( input( 'Valeur max (inclus) ?' ) )  
max_inclus = max_inclus + 1 # ou max_inclus += 1  
saut = int( input( 'Valeur du saut ?' ) )  
for i in range( 0, max_inclus, saut ):  
    print( i )
```