

Fiche de révision 2

Boucles for

Introduction

En Python, les boucles permettent de répéter un ensemble d'instructions. La boucle `for` est l'outil principal : elle repose sur la notion d'**itérable**, c'est-à-dire un objet qui contient plusieurs éléments que l'on peut parcourir les uns après les autres.

0.1 Itérables

Itérable

Un **itérable** est un objet dont on peut parcourir les éléments un par un. Exemples :

- une chaîne : "NSI"
- une liste : [10, 20, 30]
- un dictionnaire : {"a": 1, "b": 2}
- un ensemble : {1, 2, 3}
- un objet spécial comme `range(5)`

Note

Un itérable peut être utilisé dans une boucle `for` ou transformé directement (`list()`, `sum()`, `len()`...).

0.2 La boucle for

Structure générale

```
1 for variable in iterable:  
2     # bloc d'instructions (indenté)  
3     ...
```

Boucle for

La boucle `for` prend les éléments d'un **itérable** un par un et exécute le bloc d'instructions pour chacun. À chaque tour, la **variable** change automatiquement de valeur.

Exemple 1 — Boucles sur différents itérables

```

1  # Chaîne de caractères
2  for c in "NSI":
3      print(c)      # N S I
4
5  # Liste de nombres
6  for x in [10, 20, 30]:
7      print(x)      # 10 20 30
8
9  # Dictionnaire (par défaut : clés)
10 d = {"a": 1, "b": 2}
11 for cle in d:
12     print(cle, d[cle]) # a 1 ; b 2

```

⚠️ Attention

Indentation obligatoire : tout le bloc de la boucle doit être aligné, sinon Python renvoie une erreur.

0.3 L'outil range

range

range est une fonction qui crée un **itérable d'entiers**. Il ne calcule pas tout d'un coup : c'est un objet spécial qui peut être parcouru ou transformé.

Trois formes possibles

Forme	Exemple	Valeurs générées
range(stop)	range(5)	0, 1, 2, 3, 4
range(start, stop)	range(2, 6)	2, 3, 4, 5
range(start, stop, step)	range(1, 10, 3)	1, 4, 7

Exemple 2 — Utilisation hors for

```

1  r = range(5)
2  print(r)          # range(0, 5)
3  print(len(r))    # 5
4  print(list(r))   # [0, 1, 2, 3, 4]
5  somme = sum(range(100)) # 0+1+...+99

```

⚠️ Attention

La borne stop est toujours **exclue**. Exemple : `range(5)` → 0,1,2,3,4 (et pas 5).

0.4 Boucles `for` avec `range`

💡 Intérêt

`range` est très souvent utilisé avec une boucle `for` pour répéter une action un certain nombre de fois ou parcourir une suite d'entiers.

Exemple 3 — Exemples classiques

```
1  # 5 répétitions
2  for i in range(5):
3      print("Salut")
4
5  # Compter de 2 à 8 par pas de 2
6  for i in range(2, 10, 2):
7      print(i)          # 2 4 6 8
8
9  # Compter à rebours
10 for i in range(5, 0, -1):
11     print(i)          # 5 4 3 2 1
```

💡 Astuce

On utilise `range` même si la valeur de la variable ne sert pas, par exemple pour répéter une instruction un nombre fixe de fois.