BAC Pro CIEL	Python – Module 4 Les boucles avec for	Année 2025/2026

A. Boucle for

La boucle for est utilisée pour itérer sur une séquence (comme une liste, une chaîne de caractères, un tuple, ou une plage de nombres) ou d'autres objets itérables. C'est l'une des boucles les plus utilisées en programmation, notamment en Python lorsque l'on connaît le nombre de répétitions à effectuer.

La syntaxe est la suivante :

```
for element in sequence :
    # Bloc de code à exécuter pour chaque 'element'
    # de la 'sequence'
    # (Ce bloc est indenté)
```

Avec :

- ⇨ element : une variable qui prendra la valeur de chaque élément de la séquence, un par un
- ⇨ sequence : indique la séquence sur laquelle la boucle va itérer

Exemples :

Itérer sur une liste :

```
mes_fruits = ["pomme","banane","cerise"]
for fruit in mes_fruits :
    print(f"J'aime manger des {fruit}s.")
```

Résultat du code :

```
J'aime manger des pommes.
J'aime manger des bananes.
J'aime manger des cerises.
```

Itérer sur une chaîne de caractère :

```
mot = "Python"
for lettre in mot :
      print(lettre)
```

Résultat du code :

```
P
y
t
h
o
n
```

Itérer avec range()

```
for i in range(5) :
      print(f"Compteur : {i}")
```

Résultat du code :

```
Compteur : 0
Compteur : 1
Compteur : 2
Compteur : 3
Compteur : 4
```

```
for j in range(2,6) :
      print(f"Compteur : {j}")
```

Résultat du code :

```
Compteur : 2
Compteur : 3
Compteur : 4
Compteur : 5
```

```
for k in range(0,11,2) :
      print(f"Compteur : {k}")
```

Résultat du code :

```
Compteur : 0
Compteur : 2
Compteur : 4
Compteur : 6
Compteur : 8
Compteur : 10
```

En bref, range() :

⇨ `range(stop)` : génère des nombres de 0 jusqu'à `stop-1`
⇨ `range(start,stop)` : génère des nombres de `start` jusqu'à `stop-1`
⇨ `range(start,stop,step)` : génère des nombres de `start` jusqu'à `stop-1` avec un pas (`step`)

```
for m in reversed(range(5)):
    print(f"Compteur : {m}")
```

Résultat du code :

```
Compteur : 4
Compteur : 3
Compteur : 2
Compteur : 1
Compteur : 0
```

B. Erreurs fréquentes

Oublier les deux-points :

```
for i in range(5)
    print(i)
```

```
SyntaxError: invalid syntax
```

Mauvaise indentation :

```
for i in range(5) :
print(i)
```

```
IndentationError: expected an indented block
```

Inverser la variable et la séquence :

```
mes_insectes = ["fourmis","araignées","scarabées","vers"]
for mes_insectes in jardin :
    print(jardin)
```

```
NameError: name 'jardin' is not defined
```

Penser que range(5) va jusqu'à 5 :

```
for i in range(5):
    print(i)
```

Résultat :

```
0
1
2
3
4
```

C. A retenir

for permet de parcourir une séquence

la variable placée après for change de valeur à chaque tour de boucle

une séquence peut être une liste, une chaîne de caractères ou une plage de nombres

range(5) génère les valeurs de 0 à 4

range(start,stop) s'arrête à stop-1

range(start,stop,step) permet d'utiliser un pas

l'indentation est obligatoire

D. Synthèse