

Blocs logiques - quelles briques utiliser ?

Boucles

- `for`
- `while`

for vs while

for	while
Besoin d'itérer sur une séquence (ou <i>range</i> de valeurs)	Besoin d'attendre qu'une condition soit vraie / fausse
On connaît le nombre d'itérations à l'avance	Le nombre d'itérations dépend d'une condition
	Ne pas oublier les cas extrêmes. Est-ce que la boucle dépasse la taille de la liste ?

for vs for each

Note

En python, il n'y a pas d'instruction `foreach` à part entière, mais une syntaxe proche :

```
# for classique
for index in range(len(ma_liste)):
    print(index)

# équivalent d'un "for each"
for element in ma_liste:
    print(element)
```

for	for each
Permet d'accéder à l'index	Si pas besoin de l'index, plus facile à manier / lisible
Permet de modifier la liste avec la syntaxe <code>liste[index] = ...</code>	Ne permet pas de modifier la liste

Conditions

- `if`
- (Pas encore vu, pour les curieuses/eux) : `match... case`

On peut enchaîner plusieurs conditions (les deux formes sont équivalentes) :

```
if condition_une:
    if condition_deux:
        pass

if condition_une and condition_deux:
    pass
```

1. Choisir la forme qui semble la plus logique / simple à manier.
2. Ne pas enchaîner *trop* de conditions (il faut que le code reste lisible).
3. Si plusieurs conditions s'enchaînent, python (et de nombreux langages, voir [short-circuit evaluation](#)) les testent dans l'ordre :

```
if condition_une and condition_deux:
    # si condition_une est fausse, condition_deux ne sera testée
    pass
```