

# PYTHON PAR LA PRATIQUE

---

Présenté par :  
**Xavier TABUTEAU**

## 04. Blocs d'instructions



### Limites des blocs d'instructions

Avec Python, elles sont définies par la mise en page. Vous devez utiliser les sauts à la ligne et l'indentation, mais en contrepartie vous n'avez pas à vous préoccuper d'autres symboles délimiteurs de blocs.

En définitive, Python vous force donc à écrire du code lisible, et à prendre de bonnes habitudes que vous conserverez lorsque vous utiliserez d'autres langages.

Python ne peut exécuter un programme que si sa syntaxe est parfaitement correcte. Dans le cas contraire, le processus s'arrête et vous obtenez un message d'erreur.

```
def power(num, x=1):  
    result = 1  
    for i in range(x):  
        result = result * num  
    return result
```

## 04. Boucles



### Répétitions en boucle – L'instruction while

```
# boucle while
x = 0
while (x < 5):
    print(x)
    x += 1
```

### Répétitions en boucle – L'instruction For

```
# boucle for
for x in range(5, 10):
    print(x)
```

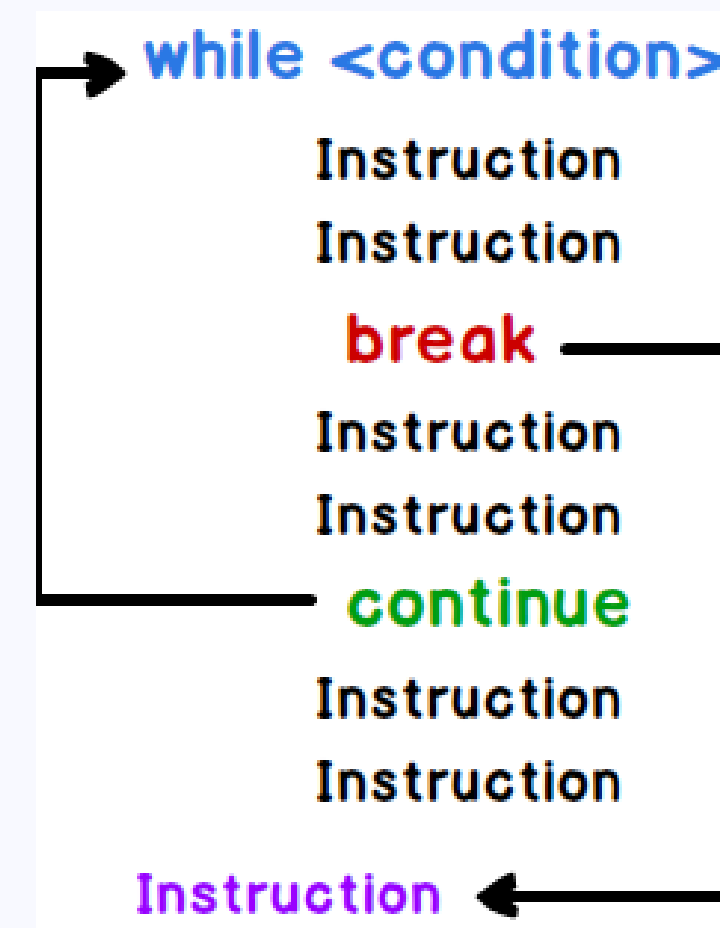
## 04. Boucles



### Instruction continue et break

Break est utilisé pour quitter une boucle for/while en cours d'exécution.

Continue est utilisé pour ignorer la suite du bloc actuel et revenir à l'instruction for/while.



## 04. Conditions



### Instruction conditionnelle If

```
# if, elif, else
if x < y:
    st = "x est plus petit que y"
elif x == y:
    st = "x est égale à y"
else:
    st = "x est plus grand que y"
print(st)
```

Remarques :

Les boucles et conditions sont imbriquables.

Les parenthèses ne sont pas obligatoire mais donne de la visibilité.

Lorsqu'une seule instruction est exécutée, on peut la mettre sur la même ligne que le if, elif ou else.

### Opérateurs de comparaison et booléen

X == Y	# égale
X != Y	# différent
X < Y	# inférieur
X > Y	# supérieur
X >= Y	# supérieur ou égale
X <= Y	# inférieur ou égale

X or Y	# ou
X and Y	# et
not X	# négation de X

Les booléens ont la valeur True ou False avec une majuscule

## 04. Conditions



*condition\_boucles.py*  
Exercices 2 et 3  
TP 1 et 2

### Condition ternaire

Permet d'affecter une valeur selon une condition de type if else en une seule ligne de code.

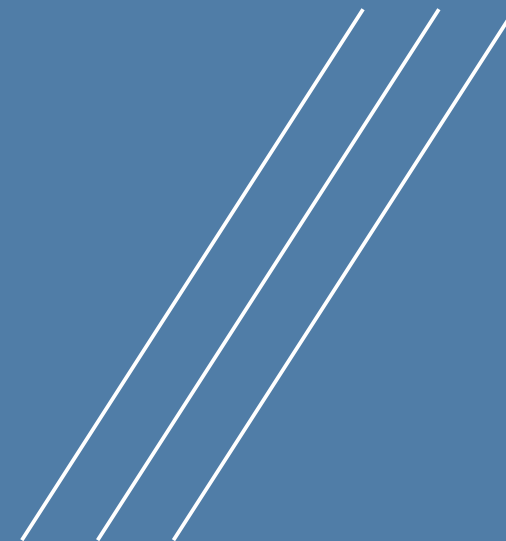
### Compréhension de liste

La compréhension de liste permet d'écrire sur une seule ligne une combinaison d'instruction de condition if (ou if else) avec une boucle for.

### Nouvelle structure conditionnelle depuis python v3.10

```
# match case
match (x):
    case 1:
        y = 0
    case 2:
        y -= 1
    case _:
        y += x + 3
```

# PYTHON INITIATION



Présenté par  
**Xavier TABUTEAU**