

L'indentation, les conditions et les boucles en Python



Par:
Maria Zebda, Salim Moussaoui
Encadrant: Ronan Sicre



L'indentation (les espaces au début des lignes)

Qu'est-ce que l'indentation ?

🧠 En Python, les espaces au début des lignes sont très importants ! Ils indiquent ce qui fait partie d'un bloc de code (exemple : dans une boucle ou une condition).

```
Entrée[6]: x = 7
            if x > 5: # si x est plus grand que 5
                print("Oui, x est plus grand que 5")
            print("fin du programme")
```

```
Oui, x est plus grand que 5
fin du programme
```

Si on oublie l'indentation ➡ Erreur ! ❌





Les conditions:

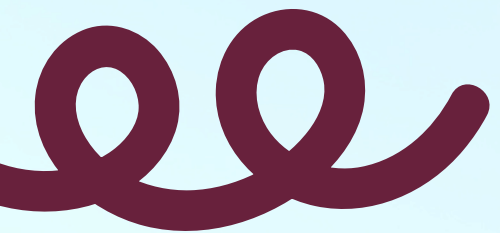
Les conditions permettent de faire des choix.

1- Les conditions if (si) :

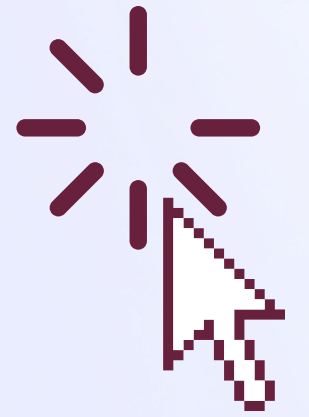
```
Entrée[7]: age = 20  
if age >= 18:  
    print("Tu es adulte")  
Tu es adulte
```

Ce code vérifie si l'âge est 18 ou plus.
Comme age = 20, la condition est vraie, donc ça affiche :
"Tu es adulte".

- ✓ Si la condition est vraie, Python exécute le bloc indenté.
- ✗ Sinon, il passe à autre chose.



2- if else (si sinon) :



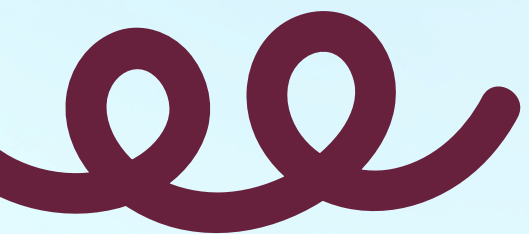
```
Entrée[8]: age = 12
            if age >= 18:
                print("Tu es adulte")
            else:
                print("Tu n'es pas encore un(e) adulte")
            Tu n'es pas encore un(e) adulte
```

Si la condition du if n'est pas vraie, alors Python va dans le else.

C'est comme dire : "Sinon, fais ça à la place."

Donc soit le if s'exécute, soit le else.

⚠ Jamais les deux en même temps.





Les boucles for :

```
Entrée[9]: for i in range(4):  
            print("Bonjour !")
```

```
Bonjour !  
Bonjour !  
Bonjour !  
Bonjour !
```

Python répète 4 fois la phrase "Bonjour".
Même si i change, on n'utilise pas sa valeur : on répète juste le même message.

```
Entrée[10]: for i in range(4):  
             print(i)
```

```
0  
1  
2  
3
```

Ce code répète 4 fois l'instruction print(i).
La variable i prend les valeurs 0, 1, 2 et 3, une à une.
Donc Python affiche les nombres de 0 à 3.

