

Les structures conditionnelles permettent d'ajouter de la logique dans l'exécution du code.

Grâce à des conditions, vous pouvez contrôler votre code et exécuter une action A plutôt qu'une action B par exemple.

Syntaxe

En Python, on écrit une structure conditionnelle grâce au mot clé `if` :

```
1. if condition:
2.     code
3.     ...
4.
5. suite du code
6. ...
```

Python exécute le code qui se trouve à l'intérieur de votre structure conditionnelle uniquement si la `condition` est vraie (`True`).

Si c'est `False`, le code n'est pas exécuté.

En Python, `None` et `0` sont des valeurs interprétées comme `False`

Autre chose, vous devez **ajouter deux points** `:` après la condition lors de la déclaration du `if`, sans quoi votre code ne fonctionnera pas.

De même, vous devez **indenter votre code à l'intérieur de votre structure conditionnelle** pour qu'il soit correctement interprété par Python.

La première ligne indentée marque le début du bloc de code et cela s'arrête à la première ligne qui ne l'est plus.

```
1. age = 24
2. MAJORITE = 18
3.
4. if age >= MAJORITE:
5.     print("Je suis majeur !")
6.
7. print("J'ai", age, "ans !")
```

Dans cet exemple, je souhaite vérifier qu'un utilisateur est majeur.

Je définis une variable `age` qui correspond à l'âge de l'utilisateur fictif puis une constante `MAJORITE` qui est assignée à la valeur `18`, l'âge légal de la majorité en France.

Je démarre ma structure conditionnelle avec le mot-clé `if` suivi de la condition `age >= MAJORITE` et de deux-points `:`

Si la variable `age` est égale à 18 ou plus, j'indente mon code et j'affiche le message `Je suis majeur !`

Dans tous les cas, le programme se termine par l'affichage de l'âge de l'utilisateur.

if / else

Dans le premier exemple, notre code faisait quelque chose de particulier uniquement si l'utilisateur était majeur.

Qu'en est-il s'il est mineur ?

Le mot clé `else` permet d'exécuter une condition si les conditions précédentes ne sont pas vérifiées.

On doit l'écrire au même niveau d'indentation que le `if`, finir par deux-points `:` puis indenter notre code.

```
1. age = 15
2. MAJORITE = 18
3.
4. if age >= MAJORITE:
5.     print("Je suis majeur !")
6. else:
7.     print("Je suis mineur..")
8.
9. print("J'ai", age, "ans !")
```

if / elif / else

Le mot-clé `elif` (pour « else if », en français « sinon si ») permet d'ajouter des conditions supplémentaires à une structure conditionnelle.

On peut ajouter autant de `elif` que nécessaire.

Si la condition testée dans le `if` est `False` alors on testera celles du/des `elif`.

Si toutes ces conditions retournent `False` alors on exécutera le code se trouvant dans le `else`.

```
1. age = 18
2. MAJORITE = 18
3.
4. if age > MAJORITE:
5.     print("Je suis majeur !")
6. elif age == MAJORITE:
7.     print("Tout juste majeur depuis aujourd'hui")
8. else:
9.     print("Je suis mineur..")
10.
11. print("J'ai", age, "ans !")
```

Imbriquer des structures conditionnelles

Vous pouvez imbriquer une structure conditionnelle dans une autre structure, puis dans une autre, et encore une autre, etc...

On appelle ça de l'imbrication en français (« nesting » en anglais).

Il suffit de déclarer votre nouvelle structure à l'intérieur de la précédente avec une indentation.

Vous imbriquez ainsi, grâce à l'indentation, plusieurs structures les unes dans les autres, comme des poupées russes :

```
1. age = 18
2. MAJORITE = 18
3.
4. if age >= MAJORITE:
5.     print("Je suis majeur !")
6.     if age == MAJORITE:
7.         print("Tout juste majeur depuis aujourd'hui")
8. else:
9.     print("Je suis mineur..")
10.
11. print("J'ai", age, "ans !")
```

Il est conseillé de ne pas trop abuser de cette imbrication pour que votre code reste lisible et facile à suivre.