



## ▼ Introduction

La programmation nécessite souvent l'examen d'un ensemble de conditions et de prendre des décisions par rapport aux actions à exécuter. L'instruction Python **if** permet d'examiner l'état actuel du programme et répondre convenablement.

```
cars = ['Audi', 'BMW', 'Toyota', 'Honda', 'Hyundai', 'Renault', 'Peugeot']
```

La boucle vérifie d'abord si la valeur de la variable **car** est 'Audi'. Si c'est le cas, la valeur est affichée en majuscules, sinon elle est affichée comme titre.

```
for car in cars:
    if car == 'Audi':
        print(car.upper())
    else:
        print(car.title())
```

```
AUDI
Bmw
Toyota
Honda
Hyundai
Renault
Peugeot
```

## ▼ Expressions booléennes

Les expressions conditionnelles ou expressions booléennes sont évaluées comme étant **True** ou **False** et sont appelées des tests conditionnels. Python utilise les valeurs **True** et **False** pour

décider si le code à l'intérieur du bloc **if** doit être exécuté. Si le test est évalué à **True**, Python

## Vérification des égalités

La plupart des tests conditionnels comparent la valeur actuelle d'une variable à une valeur spécifique. Le test conditionnel le plus simple vérifie si la valeur d'une variable est égale à une valeur déterminée.

```
car = 'bmw'  
car == 'bmw'
```

True

La première ligne étant une affectation de valeur à une variable en utilisant l'opérateur d'affectation **=**. La deuxième ligne vérifie si la valeur de la variable **car** est égale à 'bmw' en utilisant l'opérateur d'égalité **==**. Ce dernier retourne **True** si la valeur à gauche et la valeur à droite correspondent et **False** le cas contraire.

```
car = 'audi'  
car == 'bmw'
```

False

## Ignorer la casse lors de la comparaison

Le test d'une égalité en Python respecte la casse. Par exemple deux valeurs avec une capitalisation différente ne sont pas considérées égales.

```
car = 'Audi'  
car == 'audi'
```

False

Si la casse n'est pas importante, on peut convertir la valeur de la variable en minuscules avant la comparaison.

```
car = 'Audi'  
car.lower() == 'audi'
```

True

## Vérification d'inégalité

Quand on veut déterminer si deux valeurs ne sont pas égales, on peut combiner un point d'exclamation et le signe égal **!=**. Le point d'exclamation représente la contradiction **not**.

```
car = 'bmw'
```

```
if car != 'Audi':  
    print('Ce n''est pas une voiture Audi')
```

```
Ce nest pas une voiture Audi
```

La clause if compare la valeur de **car** avec 'Audi', si les deux valeurs ne correspondent pas, la valeur **True** est retournée sinon **False** est retournée. La plupart des tests conditionnels utilisent des expressions d'égalité, mais des fois, il est plus efficace d'utiliser des inégalités.

## Comparisons numériques

Tester des valeurs numériques est simple. Le code suivant vérifie si l'âge de la personne est 18 ans.

```
age = 18  
age == 18
```

```
True
```

On peut également tester si deux nombres sont non égaux.

```
age = 17  
  
if age != 43:  
    print("L'âge ne correspond pas à l'âge exigé")
```

```
L'âge ne correspond pas à l'âge exigé
```

On peut inclure plusieurs comparaisons mathématiques dans nos expressions conditionnelles.

```
age = 19
```

```
age < 21
```

```
True
```

```
age <= 21
```

```
True
```

```
age > 21
```

```
False
```

```
age >= 21
```

False

## Vérification de plusieurs expressions

Parfois, on a besoin de vérifier plusieurs conditions mathématiques en même temps. Les mots clés **and** et **or** peuvent être utilisés dans ces cas.

```
min_quantity = 10
max_quantity = 30
quantity = 20

if quantity > min_quantity and quantity < max_quantity:
    print('La quantité est convenable')
```

La quantité est convenable

```
car = 'Peugeot'

if car == 'Peugeot' or car == 'Citroën':
    print("Le constructeur de la voiture est PSA.")
```

Le constructeur de la voiture est PSA.

## Vérification d'appartenance à une liste

Parfois, il est important de vérifier si une liste contient une valeur afin d'entreprendre toute action. Par exemple, on peut tester si un nouvel utilisateur appartient à la liste des utilisateurs avant de compléter son enregistrement. L'opérateur **in** peut être utilisé dans ces cas.

```
cars = ['Audi', 'Mercedes', 'Honda']

'Mercedes' in cars
```

True

## Vérification de non appartenance à une liste

On peut également tester si une liste ne contient pas une valeur en utilisant **not in**.

```
blacklisted_users = ['david', 'alex', 'jean']
user = 'jack'

if user not in blacklisted_users:
    print(f"L'utilisateur {user.title()} n'est pas blacklisté.")

L'utilisateur Jack n'est pas blacklisté.
```

## ▼ Structures conditionnelles

### Les structures conditionnelles simples

Les clauses **if** les plus simples se composent d'un seul **if** et d'une seule action.

```
age = 19

if age >= 18:
    print("Vous avez la possibilité d'accéder à la plateforme.")

    Vous avez la possibilité d'accéder à la plateforme.
```

L'indentation joue un rôle primordial pour les structures conditionnelles. Toutes les lignes indentées après la clause **if** seront exécutées si la conditions est remplie et ignorées le cas contraire.

```
age = 19
if age >= 18:
    print("Vous avez plus que 18 ans.")
    print("Vous êtes éligibles pour la promotion.")

    Vous avez plus que 18 ans.
    Vous êtes éligibles pour la promotion.
```

### Les structures conditionnelles if-else

Souvent, on a besoin d'exécuter une action quand une conditions est vérifiée et une autre action le cas contraire. Le bloc **if-else** peut être utilisé dans ces cas.

```
age = 17
if age >= 18:
    print("Vous avez plus que 18 ans.")
    print("Vous êtes éligibles pour la promotion.")
else:
    print("Malheureusement vous n'êtes pas éligibles pour la promotion.")
    print("Vous pouvez bénéficier de la promotion une fois l'âge exigé est atteint.")

    Malheureusement vous n'êtes pas éligibles pour la promotion.
    Vous pouvez bénéficier de la promotion une fois l'âge exigé est atteint.
```

### Les structures conditionnelles if-elif-else

Souvent, on a besoin de tester plusieurs situations où on peut utiliser la syntaxe if-elif-else.

```
salary = 2000

if salary > 1000:
    print("Votre commission est de: ", salary * 1.2)
elif salary < 2000:
    print("Votre commission est de: ", salary * 1.3)
else:
    print("Votre commission est de: ", salary * 1.4)

    Votre commission est de:  2400.0
```

## Utilisation de plusieurs blocs elif

On peut utiliser n'importe quel nombre de blocs elif.

```
salary = 2500

if salary > 1000 and salary < 2000:
    print("Votre commission est de: ", salary * 1.2)
elif salary < 2000:
    print("Votre commission est de: ", salary * 1.3)
elif salary > 2000 and salary < 3000:
    print("Votre commission est de: ", salary * 1.4)
else:
    print("Votre commission est de: ", salary * 1.5)

    Votre commission est de:  3500.0
```