

15_python_if_else

November 27, 2023

1 If, else statement in Python

En programmation, il est souvent nécessaire de vérifier si une condition est vraie ou fausse. En fonction du résultat de cette condition, nous pouvons exécuter différentes actions. Par exemple, si un utilisateur a le droit d'accéder à certaines données, nous voulons lui montrer ces données. Sinon, nous voulons lui montrer un message d'erreur.

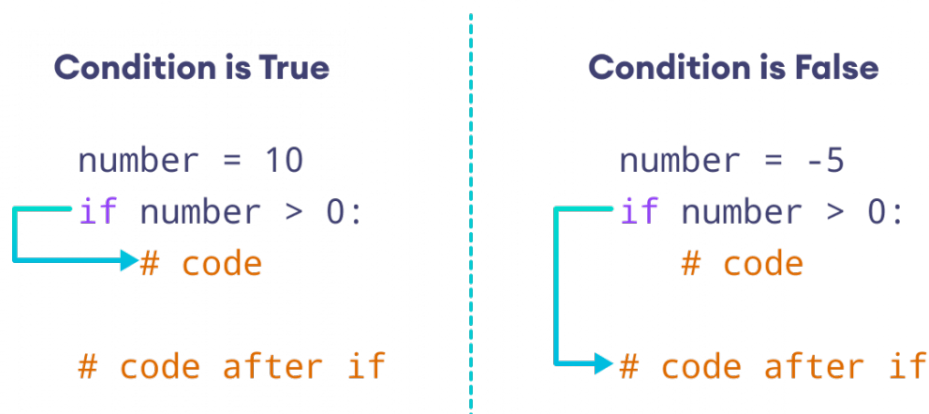
En python, il existe trois formes de conditions: - if statement (si) : exécute un bloc de code si une condition est vraie - if...else statement (si...sinon) : exécute un bloc de code si une condition est vraie et un autre bloc de code si elle est fausse - if...elif...else statement (si...sinon si...sinon) : exécute un bloc de code si une condition est vraie et un autre bloc de code si elle est fausse. Si la première condition est fausse, il vérifie la condition suivante et exécute le bloc de code correspondant si elle est vraie. Si toutes les conditions sont fausses, il exécute le bloc de code du else statement.

1.1 1. Python if statement

La syntaxe de l'instruction if est la suivante:

```
if condition:  
    # code à exécuter si la condition est vraie
```

La condition peut être n'importe quelle expression qui renvoie un booléen. Si la condition est évaluée à **True** alors le bloc de code est exécuté. Si la condition est évaluée à **False** alors le bloc de code n'est pas exécuté.



Source

```
[ ]: number = -1

# check if number is greater than 0
if number > 0:
    print('Number is positive.')

print('The if statement is easy')
```

1.2 2. Python if...else statement

Un if peut avoir une clause `else` optionnelle. La syntaxe de l'instruction `if...else` est la suivante:

```
if condition:
    # code à exécuter si la condition est vraie
else:
    # code à exécuter si la condition est fausse
```

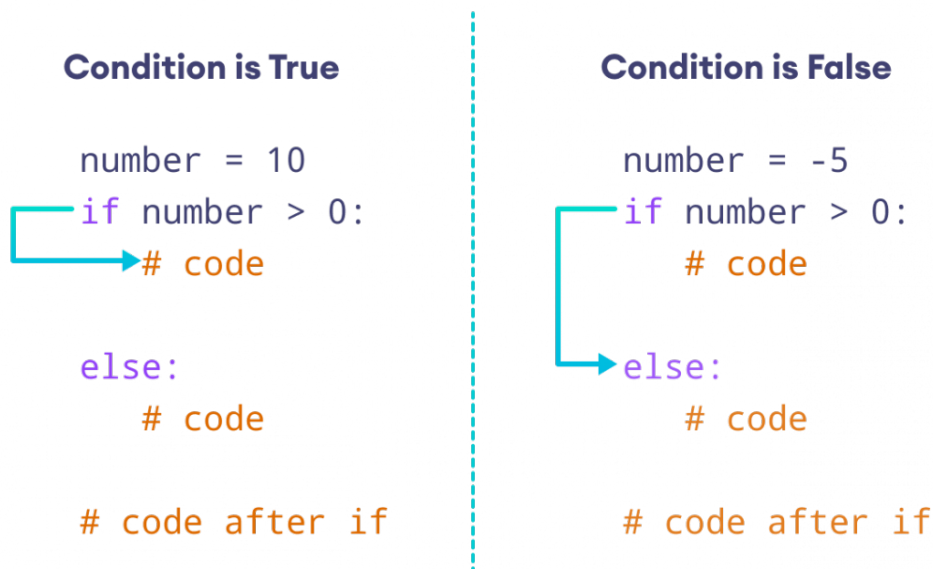
le `if...else` vérifie la condition.

1.2.1 Si la condition est vraie (True),

- le bloc de code à l'intérieur du `if` est exécuté
- le bloc de code à l'intérieur du `else` n'est pas exécuté

1.2.2 Si la condition est fausse (False),

- le bloc de code à l'intérieur du `if` n'est pas exécuté
- le bloc de code à l'intérieur du `else` est exécuté



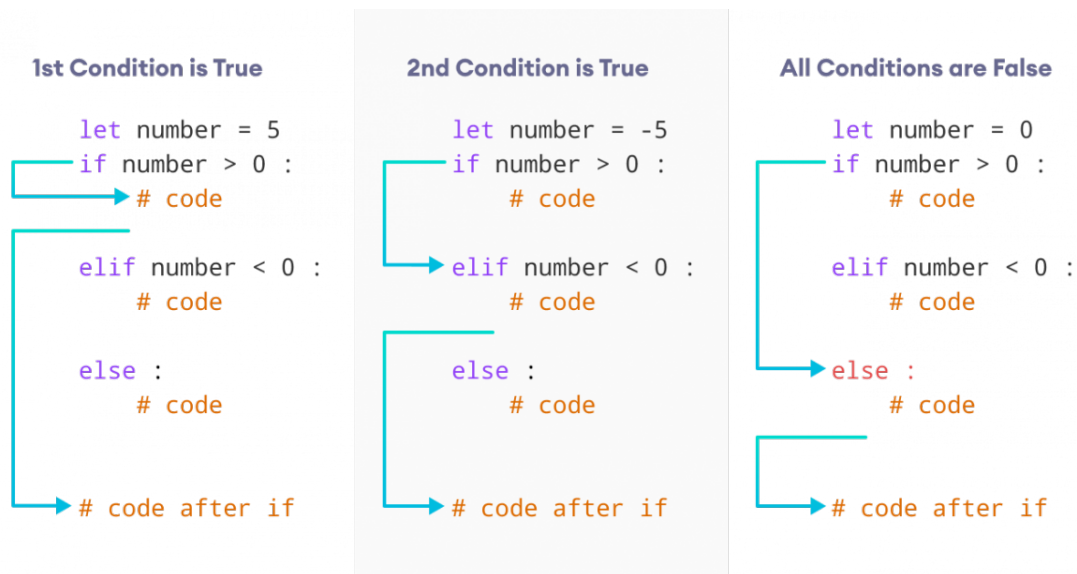
Source

1.3 3. Python if...elif...else statement

Un if peut avoir une ou plusieurs clauses elif optionnelles. La syntaxe de l'instruction if...elif...else est la suivante:

```
if condition1:
    # code à exécuter si la condition1 est vraie
elif condition2:
    # code à exécuter si la condition2 est vraie
else:
    # code à exécuter si toutes les conditions sont fausses
```

1. si condition1 est vraie, le bloc de code à l'intérieur du if est exécuté. Le bloc de code à l'intérieur du elif et du else n'est pas exécuté.
2. si condition1 est fausse:
 - si condition2 est vraie, le bloc de code à l'intérieur du elif est exécuté. Le bloc de code à l'intérieur du else n'est pas exécuté.
 - si condition2 est fausse, le bloc de code à l'intérieur du else est exécuté.



[Source](#)

```
[ ]: number = 1

if number == 10:
    print("10")
elif number == 0:
    print('Zero')
elif number > 0:
    print("Positive number")
else:
    print('Negative number')

print('This statement is always executed')
```

2 Python nested if statements

Nous pouvons avoir un `if...elif...else` à l'intérieur d'un autre `if...elif...else`. C'est ce qu'on appelle un `nested if` en python.

la syntaxe est la suivante:

```
if condition1:
    # code à exécuter si la condition1 est vraie
    if condition2:
        # code à exécuter si la condition2 est vraie
    else:
        # code à exécuter si la condition2 est fausse
else:
    # code à exécuter si la condition1 est fausse
```

```
[ ]: age = 20
majority = "major" if age >= 18 else "minor"
# majority = age >= 18 ? "major" : "minor" -> JS Like

print(majority)
```

```
[ ]:
```