

# CONDITIONAL STATEMENTS

**Conditional statements otherwise called control structures check for stated conditions or expressions and change the behavior of the program.**

## If Statement

```
In [2]: x = 6
        y = 4
        if x > y:
            print('x is greater') #Take note here that there is an indentation when wr
            iting the statement.

x is greater
```

## Elif Statement:

Sometimes there are more than two possibilities, in that case we can use the elif statement It stands for "else if," which means that if the original if statement is false and the elif statement is true, execute the block of code following the elif statement.

```
In [3]: x = 6
        y = 4
        if x > y:
            print ('x is greater than y')
        else:
            print ('y is greater than x')

        #In this example of if else statement the first statement will be printed beca
        use x is greater than y

x is greater than y
```

```
In [4]: x = 6
        y = 4
        if x < y:
            print ('x is greater than y')
        else:
            print ('y is greater than x')

        #In this example of if else statement the second statement will be printed bec
        ause x is lesser than y

y is greater than x
```

**You can have several statement under the conditions as shown below**

```
In [5]: x = 60

if x < 50:
    print('(I am below 60)')
    print('I am not old yet')
else:
    print('(I am already 60)')
    print('I am an old folk')
```

```
(I am already 60)
I am an old folk
```

### **Elif Statement**

Sometimes there are more than two possibilities, in that case we can use the elif statement. It stands for "else if," which means that if the original if statement is false and the elif statement is true, execute the block of code following the elif statement.

```
In [7]: level = 'Beginner'
if level == 'Rookie':
    print('I am a Rookie player')
elif level == 'Beginner':
    print('I am a Beginner player')
elif level == 'Middle':
    print('I am a Middle level player')
elif level == 'Expert':
    print('I am an Expert level player')
else:
    print("I don't know your level!")
```

```
I am a Beginner player
```

### **NOTE**

At most, one of the code blocks specified will be executed. If an else clause isn't included, and all the conditions are false, then none of the blocks will be executed.

```
In [9]: names = {
        'Fred': 'Hello Fred',
        'Xander': 'Hello Xander',
        'Joe': 'Hello Joe',
        'Arnold': 'Hello Arnold'
    }

    print(names.get('Joe', "I don't know who you are!"))

    print(names.get('Rick', "I don't know who you are!"))
```

```
Hello Joe
I don't know who you are!
```

## NOTE

An if statement with elif clauses uses short-circuit evaluation, unlike the and and or operators. Once one of the expressions is found to be true and its block is executed, none of the remaining expressions are tested. See the example below:

```
In [10]: if 'a' in 'Learning':
        print('I am learning Python')
    elif 1/0:
        print("This won't happen")
    elif var:
        print("This won't either")
    #The second expression contains a division by zero, and the third references a
    n undefined variable var.
    #Either would raise an error, but neither is evaluated because the first condi
    tion specified is true.
```

```
I am learning Python
```

## One-Line if Statements

While it is customary to write if on one line and indented on the following line, it is permissible to write an entire if statement on one line with each statement separated by a semi colon (;)

```
In [11]: if 'P' in 'Pyhton': print('You are learning Pyhton'); print('Do you love it?')
        ; print('How far have you learnt?')
```

```
You are learning Pyhton
Do you love it?
How far have you learnt?
```

```
In [13]: if 'c' in 'Pyhton': print('You are learning Pyhton'); print('Do you love it?')
        ; print('How far have you learnt?')
    #Nothing gets printed out cos the if statement does not hold.
```

**NOTE**

Multiple statements may be specified on the same line as an elif or else clause as well.

```
In [14]: x = 1
         if x == 1: print('Positive'); print('Needs Drugs'); print('Avoid smoking')
         elif x == 0: print('Negative'); print('Continue a healthy lifestyle')
         else: print('Test result inconclusive'); print('Further examination required')
```

```
Positive
Needs Drugs
Avoid smoking
```

While the above technique works, and the interpreter allows it, it is generally discouraged on the grounds that it leads to poor readability, particularly for complex if statements. The PEP 8 specifically recommends against it.

```
In [15]: x = 1
         if x == 1:
             print('Positive')
             print('Needs Drugs')
             print('Avoid smoking')
         elif x == 2:
             print('Negative')
             print('Continue a healthy lifestyle')
         else:
             print('Test result inconclusive')
             print('Further examination required')
```

```
Positive
Needs Drugs
Avoid smoking
```

**Python's Ternary Operator**

This is different from the if statement forms listed above because it is not a control structure that directs the flow of program execution. It acts more like an operator that defines an expression.

if else In example above, is evaluated first. If it is true, the expression evaluates to . If it is false, the expression evaluates to .

```
In [16]: stocked = False
print("Is it time for shopping?", 'not time for shopping, as fridge is stocke
d' if not stocked else 'Yes, fridge is empty.')
stocked = True
print("Is it time for shopping?", 'not time for shopping, as fridge is stocke
d' if not stocked else 'Yes, fridge is empty.')
```

```
Is it time for shopping? not time for shopping, as fridge is stocked
Is it time for shopping? Yes, fridge is empty.
```

The python ternary operator can help reduce the amount of code you write for some basic operations and also help improve readability. See the 2 code blocks below for an implementation of finding the greatest of two numbers

```
In [17]: #Normal if statement to find the larger of two (2) numbers
if x > y:
    m = x
else:
    m = y
```

```
In [18]: #Using a conditional expression ( ternary operator) to find the larger of two
(2) numbers
m = x if x > y else y
```

## Sample Project 1

Using conditional statement write a python program to compare two (2) strings based on user input

```
In [23]: # Program that compares two strings base on what a user inputs.
# Get a string from the user.
user_input = input('Enter the your string: ')
# Determine whether the given string matches what is stored
if user_input == 'python':
    print('Correct String entered')
else:
    print('Sorry, you have entered the wrong string, try again.')
```

```
Enter the your string: python
Correct String entered
```

## Sample Project 2

Build a medical test kit for diagnosing whether a patient has malaria or not based on the temperature reading of the patient. If the user temperature is equal or below 37.5 the program should print 'Negative'. If the temperature is above 37.5 the program should print 'Positive, go for check up' and finally print 'Your result is not clear when a user enters any figure outside the defined range such as -2, -10

```
In [40]: #Enter patient temperature
userTemp = int(input('Enter your body temperature: ')) #The int here is called
type casting
# Determine whether the given string matches what is stored
if userTemp == 0 or userTemp == 37.5:
    print('Negative')

elif userTemp > 37.5:
    print('Positive, go for check up')

else:
    print('Your result is not clear')
```

Enter your body temperature: 40  
Positive, go for check up