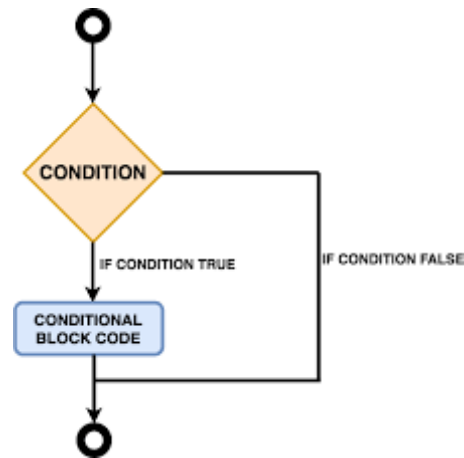


Conditional statements in python



Hey guys, welcome to the 30 days of programming. In this lecture, we'll learn the conditional statements in python. If else statements: As you all know about the conditional operators in python which was discussed in the previous videos. Let us have a quick recap of the conditional statements.

- Equals: `a == b`
 - Not Equals: `a != b`
 - Less than: `a < b`
 - Less than or equal to: `a <= b`
 - Greater than: `a > b`
 - Greater than or equal to: `a >= b`
- An "if statement" is written by using the if keyword.
Syntax:

```
If <condition> :  
<indent>Statements  
:
```

Example :

```
In [ ]: a = 33  
b = 200  
if b > a:  
    print("b is greater than a")
```

1. We end the if condition(s) with a colon (:)
2. To define the scope of the if block we use indentation.(Whitespace before the line)

We can use multiple conditions in **if** using appropriate logical operators for example:

```
In [ ]: if False:  
        print('Inside if')  
        print('Outside if')
```

```
In [ ]: a = int(input("Enter the value for 'a'"))  
if a > 5 and a%2 == 0 :  
    print("a is even no greater than 5")
```

Let's have one more example for a combination of conditions

```
In [ ]: a = True  
b = False  
c = True
```

```
In [ ]: if (a or b) and c:  
        print("30dop")
```

If you don't give indentation then it will raise an error

```
In [ ]: if 10%2 == 0:
        pass
        print("True")
```

elif

elif means 'else if'

if the previous condition fails then check this condition

Example:

```
In [ ]: a = int(input('Enter a number '))
        if a > 0:
            print("Positive no")
        elif a < 0:
            print('Negative no')
```

Note:

elif is always used after a *if* or *elif* statement

```
In [ ]: a = 5
        elif a%2 == 1 :
            print("Odd no")
```

else

if all the above conditions fails then the **else** block will be executed

```
In [ ]: a = int(input('Enter a number '))
        if a > 0:
            print("Positive no")
```

```
elif a < 0:  
    print('Negative no')  
else:  
    print("Number is zero")
```

Let's have one more example for even odd:

```
In [ ]: a = int(input())  
if a % 2 == 0:  
    print("NO is even")  
else:  
    print("Odd no")
```

Short hand if

```
In [ ]: a, b = 5, 4  
if a > b: print("a is greater than b")
```

Short Hand If ... Else

This technique is known as Ternary Operators

```
In [ ]: a, b = 5, 4  
max_number = a if a > b else b  
max_number
```

Nested if

```
In [ ]: a = int(input('Enter no'))  
if a > 0:  
    if a%2 == 0:  
        print('Even no')
```

```
else:
    print('Odd no')
else:
    print('Non positive no')
```

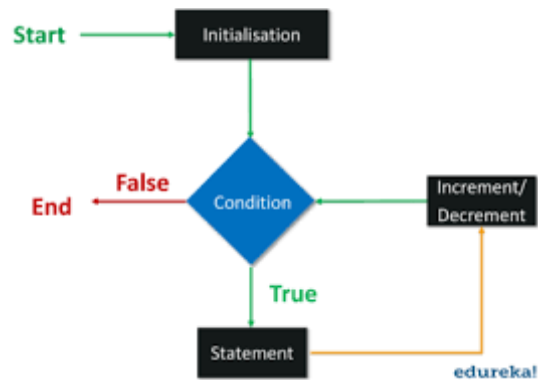
The pass statement

```
In [ ]: a, b = 5, 3
if a > b:
    pass
else:
    print('Else is printed')
```

```
In [ ]:
```

```
In [ ]:
```

Loop in python



Need of loop

When we need to execute some statement repeatedly in any order (or not). We use loop statements
python has two loop commands
1. while loop
2. for loop

While loop

With the while loop we can execute a set of statements as long as a condition is true.

```
In [ ]: # To print numbers from 1 to 5
i = 1
while i < 6:
    print(i, end= ' ')
    i+=1    # i = i+1
```

Note:

1. Make sure your code does not fall in an infinite loop (if not required).

```
i = 0
while i < 5:
```

```
    print(i)
```

</code>

** This code will run infinite times since there is no increment of i

2. Never initialize the looping variable with a fixed value inside the loop.

If you, then the loop will fall in infinite loop

In []:

```
# i = 0
# while i<5:
#     print(i)
```

In []:

break the loop

We can stop the loop even if the while condition is true

In [1]:

```
i = 0
while i < 10:
    print(i, end = " ")
    if i > 5:
        break
    i += 1
```

0 1 2 3 4 5 6

Continue

With the continue statement we can stop the current iteration, and continue with the next:

In [2]:

```
i = 0
while i < 6:
    i += 1
    if i % 3 == 0:
        continue
    print(i, end= ' ')
```

1 2 4 5

loop - else statements

With the else statement we can run a block of code once when the **while** condition no longer is true:

In [3]:

```
i = 0
while i<6 :
    print(i, end= " ")
    i+=1
else:
    print("\nelse part executed")
```

```
0 1 2 3 4 5
else part executed
```

But if you ended the loop using the break statement then the else block will not be executed.

In [4]:

```
i = 0
while True:
    print(i,end=" ")
    if i == 5:
        break
    i+=1
else:
    print("Else executed")
```

0 1 2 3 4 5

Before starting for loops we will take a look at range function in python.

Syntax:

```
range(start, end, step_size)
```

start(inclusive) - lower limit, By default, it starts with 0 if not specified.

end (exclusive) - upper limitgenerate numbers up to this number.

step_size - Difference between any two numbers, by default it is 1

For loop

In [5]:

```
for i in range(5):  
    print(i, end = " ")
```

0 1 2 3 4

In [17]:

```
for i in range(0,10,3):  
    print(i)
```

0
3
6
9

In [6]:

```
for num in range(11,25,4):  
    print(num,end = " ")
```

11 15 19 23

Numbers selected with step 4 as:- **11,12,13,14,15,16,17,18,19,20,21,22,23,24**

for loop on a list

```
In [7]: fruits = ['apple', 'banana', 'orange']  
        for i in fruits:  
            print(i,end = " ")
```

apple banana orange

for loop on a dictionary

```
In [ ]:
```

```
In [8]:
```

```
dict_marks = {'Maths': 78, 'English': 91, 'Science': 89}
for i in dict_marks:
    print('subject {} : Marks: {}'.format(i,dict_marks[i]))

keys()
```

```
subject Maths : Marks: 78
subject English : Marks: 91
subject Science : Marks: 89
```

Loop on a string

In [9]:

```
for i in 'siddhesh':
    print(i,end= " ")
```

```
s i d d h e s h
```

In [18]:

```
string = 'Siddhesh'
for i in range(len(string)):
    print(string[i])
```

```
S  
i  
d  
d  
h  
e  
s  
h
```

Creating a list using for loop

In [19]:

```
numbers = [i+3 for i in range(5)]  
print(numbers)
```

```
[3, 4, 5, 6, 7]
```

Creating a list using for loop with if condition

In [20]:

```
cube_numbers = [n**3 for n in range(1,10) if n%2 == 1]  
for name in cube_numbers:
```

```
print('{} is a member'.format(name))
```

```
1 is a member  
27 is a member  
125 is a member  
343 is a member  
729 is a member
```

Break , continue and else with for loop

In [12]:

```
for i in range(0,10):  
    if i % 3==0: continue  
    if i > 12: break  
    print(i, end= " ")  
else:  
    print("Condition in for is false now")
```

```
1 2 4 5 7 8 Condition in for is false now
```

Nested for loops

In [13]:

```
for i in range(5):
    for j in range(i,5):
        print("# ",end= "")
    print()
```

```
# # # # #
# # # #
# # #
# #
#
```

In [14]:

```
for i in range(5):
    for j in range(i+1):
        print("# ",end= "")
    print()
```

```
#
# #
# # #
# # # #
# # # # #
```

In []:

