Python If Statements

The if statement in python is same as c language which is used test a condition. If condition is true, statement of if block is executed otherwise it is skipped.

**Syntax of python if statement:**

**if**(condition):

   statements

**Example of if statement in python**

1. a=10
2. **if** a==10:
3. **print** ( “Hello User”)

**Output:**

Hello User

Python If Else Statements

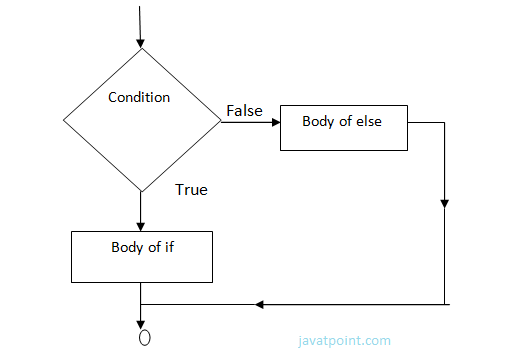
**Syntax:**

**if**(condition):  True

             statements

**else**:   false

              statements



**Example-**

year=2000

**if** year%4==0:

**print**  "Year is Leap"

**else**:

**print** "Year is not Leap"

**Output:**

Year is Leap

Nested If Elif Statement:

When we need to check for multiple conditions to be true then we use elif Statement.

This statement is like executing a if statement inside a else statement.

**Syntax:**

If statement:

    Body

**elif** statement:

    Body

**else**:

    Body

**Example:**

a=10

**if** a>=20:

**print** "Condition is True"

**elif** a>=15:

**print** "Checking second value"

**else**:

**print** "All Conditions are false"

**Output:**

All Conditions are false.

For Loop

for Loop is used to iterate a variable over a sequence(i.e., list or string) in the order that they appear.

**Syntax:**

**for** <variable> **in** <sequence>:

**Program to display table of Number:**

1. num=2
2. **for** a **in** range (1,6,4):
3. **print**  num \* a

**Output:**

1. 2
3. 4
5. 6
7. 8
9. 10

**Program to find sum of Natural numbers from 1 to 10.**

1. sum=0
2. **for** n **in** range(1,11):
3. sum+=n
4. **print** sum

**Output:**

1. 55

Nested Loops

Loops defined within another Loop is called Nested Loop.

When an outer loop contains an inner loop in its body it is called Nested Looping.

**Syntax:**

**for**  <expression>:

**for** <expression>:

            Body

**eg:**

**for** i **in** range(1,6):

**for** j **in** range (1,i+1):

**print** i,

**print**

**Output:**

1. >>>
2. 1
3. 2 2
4. 3 3 3
5. 4 4 4 4
6. 5 5 5 5 5
7. >>>

**Explanation:**

For each value of Outer loop the whole inner loop is executed.

For each value of inner loop the Body is executed each time.

**Program to print Pyramid:**

**for** i **in** range (1,6):

**for** j **in** range (5,i-1,-1):

**print** "\*",

**print**

for i in range (1,6):

for j in range (5,i-1,-1):

print (i,end=' ')

print ("\n")

**Output:**

1. >>>
2. \* \* \* \* \*
3. \* \* \* \*
4. \* \* \*
5. \* \*
6. \*

While Loop

while Loop is used to execute number of statements or body till the condition passed in while is true. Once the condition is false, the control will come out of the loop.

**Syntax:**

1. **while** <expression>:
2. Body

Here, body will execute multiple times till the expression passed is true. The Body may be a single statement or multiple statement.

**Eg:**

1. a=10
2. **while** a>0:
3. **print** "Value of a is",a
4. a=a-2

print "Loop is Completed"

**Output:**

1. >>>
2. Value of a **is** 10
3. Value of a **is** 8
4. Value of a **is** 6
5. Value of a **is** 4
6. Value of a **is** 2
7. Loop **is** Completed
8. >>>

**Explanation:**

* Firstly, the value in the variable is initialized.
* Secondly, the condition/expression in the while is evaluated. Consequently if condition is true, the control enters in the body and executes all the statements . If the condition/expression passed results in false then the control exists the body and straight away control goes to next instruction after body of while.
* Thirdly, in case condition was true having completed all the statements, the variable is incremented or decremented. Having changed the value of variable step second is followed. This process continues till the expression/condition becomes false.
* Finally Rest of code after body is executed.

**Program to add digits of a number:**

1. n=153
2. sum=0
3. **while** n>0:
4. r=n%10
5. sum+=r
6. n=n/10
7. **print** sum

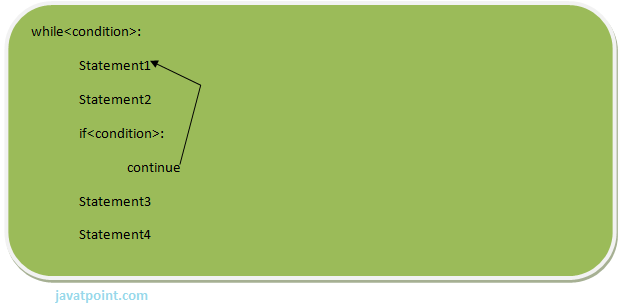
**Output:**

1. >>>
2. 9
3. >>>

Python Break

break statement is a jump statement that is used to pass the control to the end of the loop.

When break statement is applied the control points to the line following the body of the loop , hence applying break statement makes the loop to terminate and controls goes to next line pointing after loop body.



**eg:**

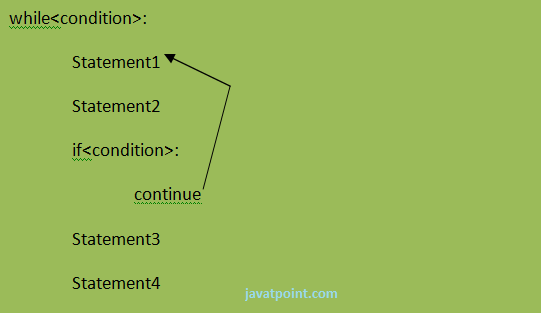
1. **for** i **in** [1,2,3,4,5]:
2. **if** i==4:
3. **print** "Element found"
4. **break**
5. **print** i,

**Output:**

1. >>>
2. 1 2 3 Element found
3. >>>

Continue Statement

continue Statement is a jump statement that is used to skip the present iteration and forces next iteration of loop to take place. It can be used in while as well as for loop statements.



**eg:**

1. a=0
2. **while** a<=5:
3. a=a+1
4. **if** a%2==0:
5. **continue**
6. **print** a
7. **print** "End of Loop"

**Output:**

1. >>>
2. 1
3. 3
4. 5
5. End of Loop
6. >>>

Python Pass

When you do not want any code to execute, pass Statement is used. It is same as the name refers to. It just makes the control to pass by without executing any code. If we want to bypass any code pass statement can be used.

**Syntax:**

1. **pass**

**eg:**

**for** i **in** [1,2,3,4,5]:

**if** i==3:

**pass**

**print** "Pass when value is",i

**print** i,

**Output:**

1. >>>
2. 1 2 Pass when value **is** 3
3. 3 4 5
4. >>>