IntelliPaat

Python Programming Basics

Agenda

01 Conditional Statements

02 Python Loops

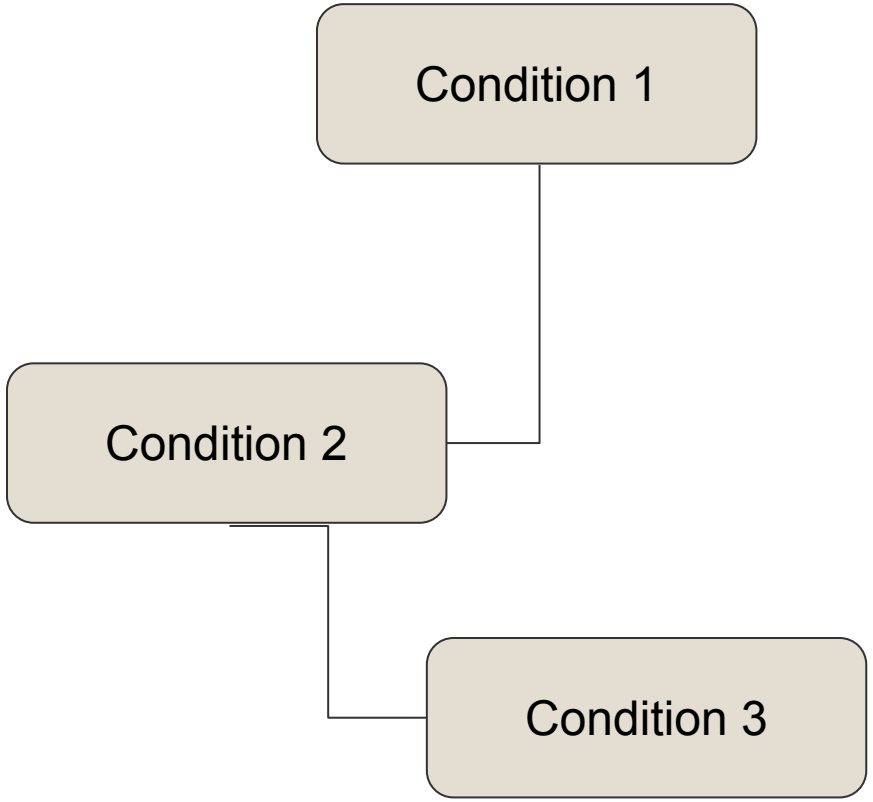
03 Functions

04 Lambda Functions

Conditional Statements

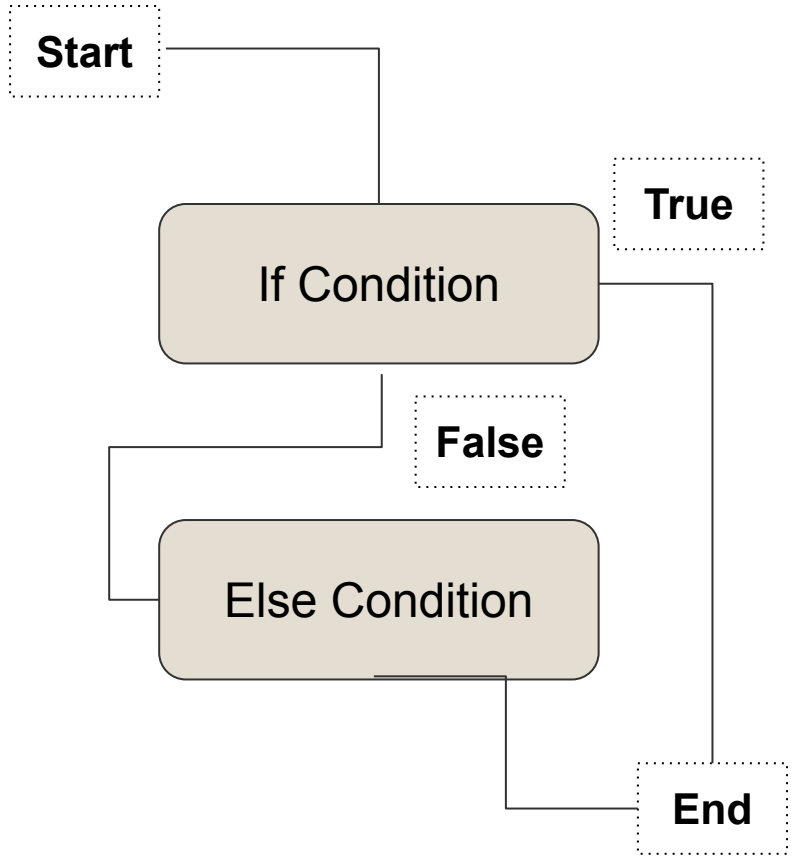
Conditional Statements

A conditional statement in python or any other programming language tests a few conditions, and based on the outcome, the execution of the code is performed.



Python If-else Statement

the if-else statement in python consists of primary condition if, which if turns out to be false, the execution moves to the else statement.



Python If-else Statement

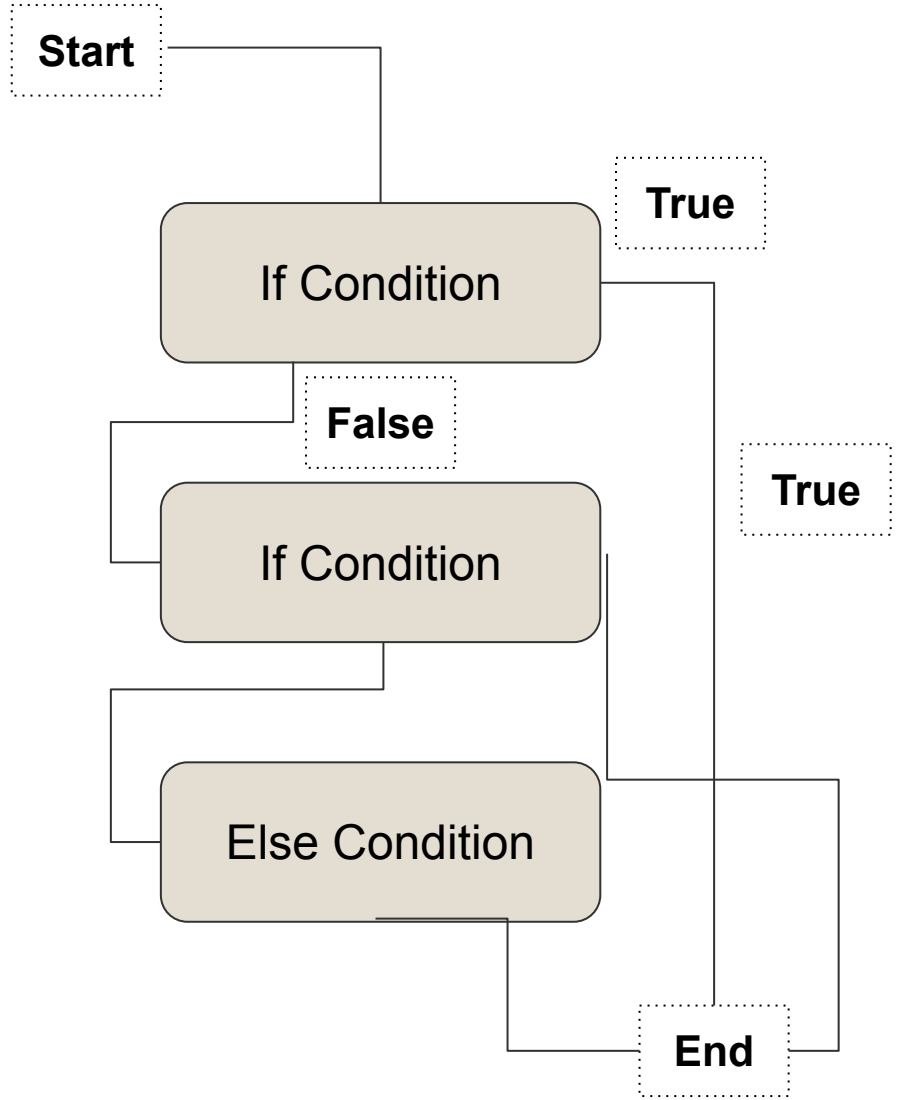
The program shows the implementation of if-else in python to compare two numbers and print if they are equal or not.

```
A = 10

B = 20

if A == B:
  print('equal')
else:
  print('not equal')
```

```
not equal
```

Python Elif Statements

The elif statement is the secondary statement that is checked when the if statement turns out to be false, and then moves to the else statement if the elif statement is also false.



Python Elif Statements

Implementation of ELIF in python, that adds multiple secondary conditions that can be applied to the problem statement.

```
A = 10
B = 20

if A == B:
  print('equal')
elif A > B:
  print('A is Greater')
elif A < B:
  print("B is Greater")
else:
  print('not equal')
```

```
B is Greater
```

Shorthand if-else Statement

Shorthand if-else statement is a quick and concise way to write a conditional statement that consists of if and else statement.

```
A = 10
B = 20

print("equal") if A == B else print('Not Equal')
```

```
Not Equal
```

Shorthand Elif Statement

Shorthand elif statement consists of primary secondary and penultimate condition else in a short concise manner.

```
A = 10
B = 20

print('A is greater') if A > B else print('equal') if A == B else print('A is smaller')
```

```
A is smaller
```

Nested if-else Statement

```
A = 20
B = 10

if A != 0:
  if A > B:
    print('Greater than B')
    if A%2 == 0:
      print("Even Number")
    else:
      pass
  else:
    pass
else:
  print('invalid entry')
```

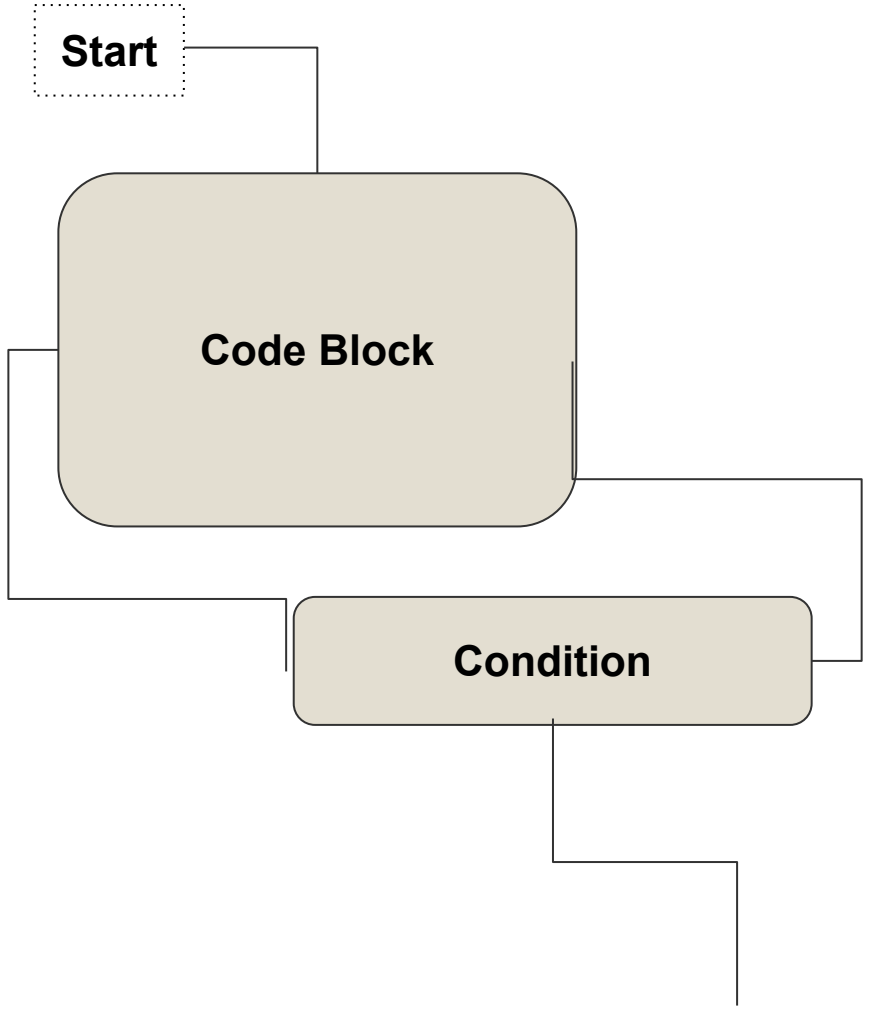
```
Greater than B
Even Number
```

A nested if-else block will have multiple if conditions inside an if condition block.

Python Loops

Python Loops

A loop in python or any other programming language is a way to iteratively execute a piece of code until certain conditions are met.



Python For Loop

A for loop in python will start with a condition and will go on until the last iteration is performed, or if there aren't any interruptions involved.

```
A = [1,2,3,4,5,6,7,8,9,10]
for i in A:
  print(i, end="")
```

```
12345678910
```

Break, Continue and Pass

The break statement will stop the execution when encountered in the program.

The continue statement stops the current iteration and moves to the next iteration

The pass statement does nothing in the current iteration.

Break, Continue and Pass

Break

```
A = [1,2,3,4,5,6,7,8,9,10]
for i in A:
  print(i, end="")
  if i >= 6:
    break
```

```
123456
```

Break, Continue and Pass

Continue

```
A = [1,2,3,4,5,6,7,8,9,10]
for i in A:
  if i == 6:
    continue
  print(i, end="")
```

```
1234578910
```

Break, Continue and Pass

Pass

```
A = [1,2,3,4,5,6,7,8,9,10]
for i in A:
  if i == 6:
    pass
  print(i, end="")
```

```
12345678910
```

Python While Loop

A while loop will continue as long as the predefined condition is true, as soon as the condition turns false, the execution moves out of the loop.

```
A = [1,2,3,4,5,6,7,8,9,10]
n = len(A)- 1
while n >= 0:
  print(A[n], end=" ")
  n = n - 1
```

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

Nested For Loops

Nested loops will have a loop inside of a loop, just like how we have discussed the nested if statements.

```
rows = 5

for i in range(rows):
    for j in range(i+1):
        print("* ", end="")
    print("\n")
```

```
*

* *

* * *

* * * *

* * * * *
```

Nested While Loop

A nested while loop will have a while loop in the same block of code.

```
n = 10
while n > 0:
  for i in range(0,11,2):
    print(i, end=" ")
    n = n - 1
```

```
0 2 4 6 8 10 0 2 4 6 8 10
```

Functions

Functions

A function can be described as a piece of code that executes a certain task and can be reused again and again.

```
def func():
  x = int(input("enter a number"))
  y = int(input("enter a number"))
  z = x + y
  return z

func()
```

```
enter a number10
enter a number10
20
```

In-Built Functions

The in-built functions are those functions that are provided by the python programming language.

```
import math

A = 10
math.pow(10, 2)
```

```
100.0
```

User Defined Functions

A user defined function is a block of code defined by the user to perform certain tasks.

```
def pow():
  x = int(input("enter a number"))
  return x**2

pow()
```

```
enter a number10
100
```

Parameterized Functions

A parametrized function will have parameters in the defined function.

```
def func(x, y):
  x = x + 10
  y = y + 20
  return x + y

func(10,15)
```

```
55
```

Lambda Functions

Lambda Functions

Lambda functions is just another shorter and concise way of defining a function similar to shorthand if-else statements.

```
func = lambda a,b: a+b

func(10,20)
```

```
30
```

Thank You

India: +91-7022374614

US: 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor