Python Programming BCA 3643

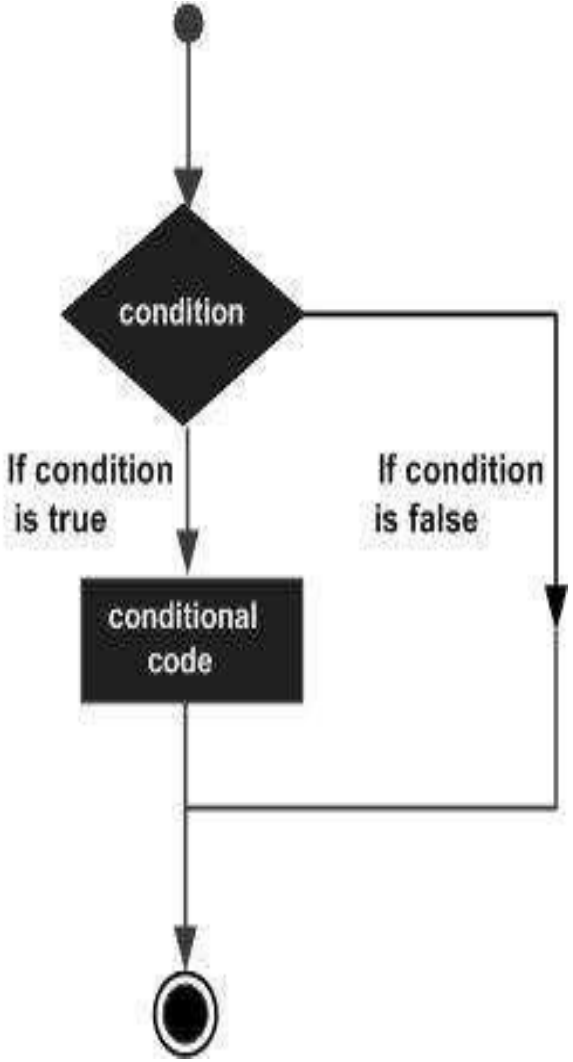
- K.Britto Alex
- Assistant Professor & Head
- BCA Department

Python Decision-making

Python Decision-making

- Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.
- Decision structures evaluate multiple expressions, which produce TRUE or FALSE as the outcome.
- It need to determine which action to take and which statements to execute if the outcome is TRUE or FALSE otherwise.
- Python programming language assumes any non-zero and non-null values as TRUE, and any zero or null values as FALSE value.

Flow chart


condition
If condition is true
If condition is false
conditional code

if Statement

- The if statement is similar to that of other languages.
- The if statement contains a logical expression using which the data is compared and a decision is made based on the result of the comparison.

Syntax

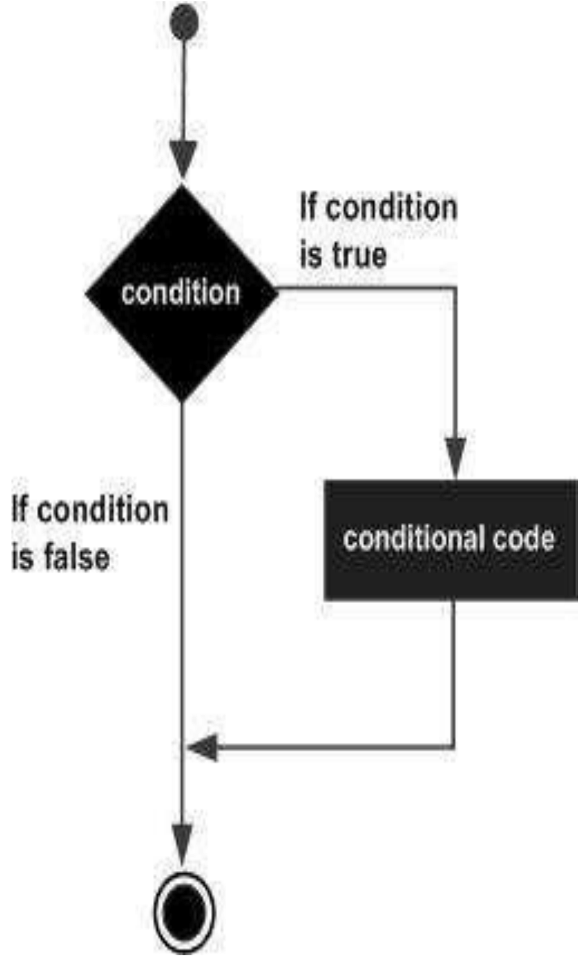
IF TEST EXPRESSION:

TRUE STATEMENTS BLOCK

REST OF PROGRAM STATEMENTS

- If the Boolean expression evaluates to TRUE, then the block of statement(s) inside the if statement is executed.
- In Python, statements in a block are uniformly indented after the: symbol.
- If Boolean expression evaluates to FALSE, then the first set of code after the end of block is executed.

Flow Diagram



Example

```
#!/usr/bin/python3
var1 = 100
if var1:
print ("1 - Got a true expression value")
print (var1)
var2 = 0
if var2:
print ("2 - Got a true expression value")
print (var2)
print ("Good bye!")
```

Output

```
1 - Got a true expression value
100
Good bye!
```

if –else Statement

- An else statement can be combined with an if statement.
- An else statement contains a block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.
- The else statement is an optional statement and there could be at the most only one else statement following if.

Syntax

IF TEST EXPRESSION:

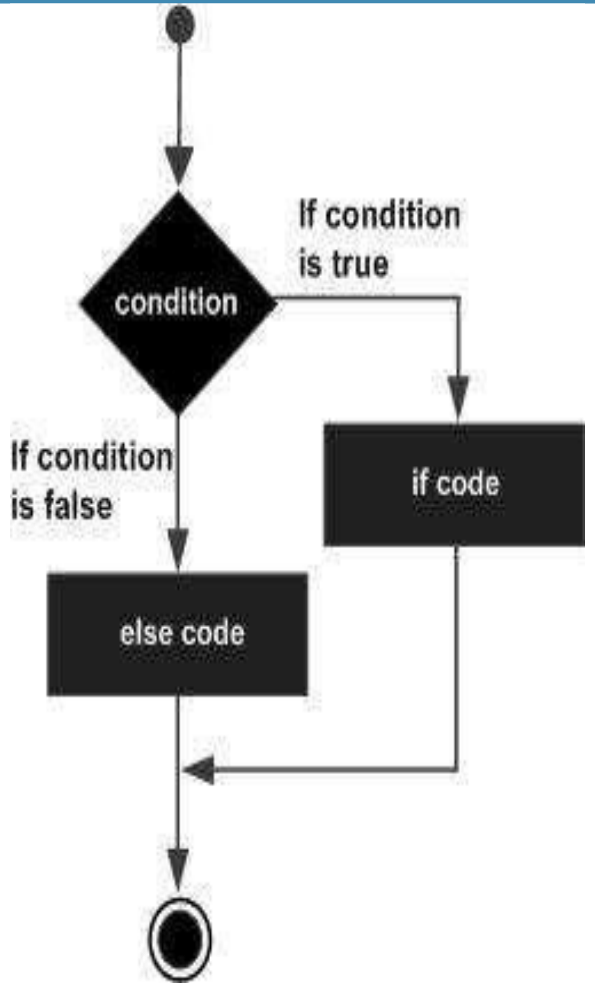
TRUE STATEMENTS BLOCK

ELSE:

FALSE STATEMENTS BLOCK

REST OF PROGRAM STATEMENTS

Flow Diagram



Example

```
#!/usr/bin/python3
amount=int(input("Enter amount: "))
if amount<1000:
discount=amount*0.05
print ("Discount",discount)
else:
discount=amount*0.10
print ("Discount",discount)
print ("Net payable:",amount-discount)
```

Output

```
Enter amount: 600
Discount 30.0
Net payable: 570.0
Enter amount: 1200
Discount 120.0
Net payable: 1080.0
```

if –elif-else Statement

- The elif statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.
- Similar to the else, the elif statement is optional.
- However, unlike else, for which there can be at the most one statement statement, there can be an arbitrary number of elif statements following an if.

Syntax

```
if test expression:
if statements block
elif test expression:
elif statements block
elif test expression:
elif statements block
else:
else statements block
rest of program statements
```

Example

```
#!/usr/bin/python3
amount=int(input("Enter amount: "))
if amount<1000:
discount=amount*0.05
print ("Discount",discount)
elif amount<5000:
discount=amount*0.10
print ("Discount",discount)
else:
discount=amount*0.15
print ("Discount",discount)
print ("Net payable:",amount-discount)
```

Output

```
Enter amount: 600

Discount 30.0

Net payable: 570.0

Enter amount: 3000

Discount 300.0

Net payable: 2700.0

Enter amount: 6000

Discount 900.0

Net payable: 5100.0
```

Nested IF Statements

- There may be a situation when you want to check for another condition after a condition resolves to true.
- In such a situation, it can use the nested if construct. In a nested if construct, it can have an if...elif...else construct inside another if...elif...else construct.

Syntax

```
if
expression1:
statements block
if expression2:
statements block
elif expression3:
statements block
else
statements block
elif expression4:
statements block
else:
statements block
rest of program statements
```

Example

```
# !/usr/bin/python3
num=int(input("enter number"))
if num%2==0:
if num%3==0:
print ("Divisible by 3 and 2")
else:
print ("divisible by 2 not divisible by 3")
else:
if num%3==0:
print ("divisible by 3 not divisible by 2")
else:
print ("not Divisible by 2 not divisible by 3")
```

Output

```
enter number8
divisible by 2 not divisible by 3
enter number15
divisible by 3 not divisible by 2
enter number12
Divisible by 3 and 2
enter number5
not Divisible by 2 not divisible by 3
```

Thank You