AMRITA VISHWA VIDYAPEETHAM

Python Control Flow

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

Python Control Flow

- A program's control flow is the order in which the program's code executes.
- The control flow of a Python program is regulated by conditional statements, loops, and function calls.
- **Decision Making**
  - if
  - if ... else
  - if ...elif … else
  - nested if
- **Loops**
  - while
  - for
- **Control Statements**
  - break
  - Continue
- **pass Statement**

if Statement

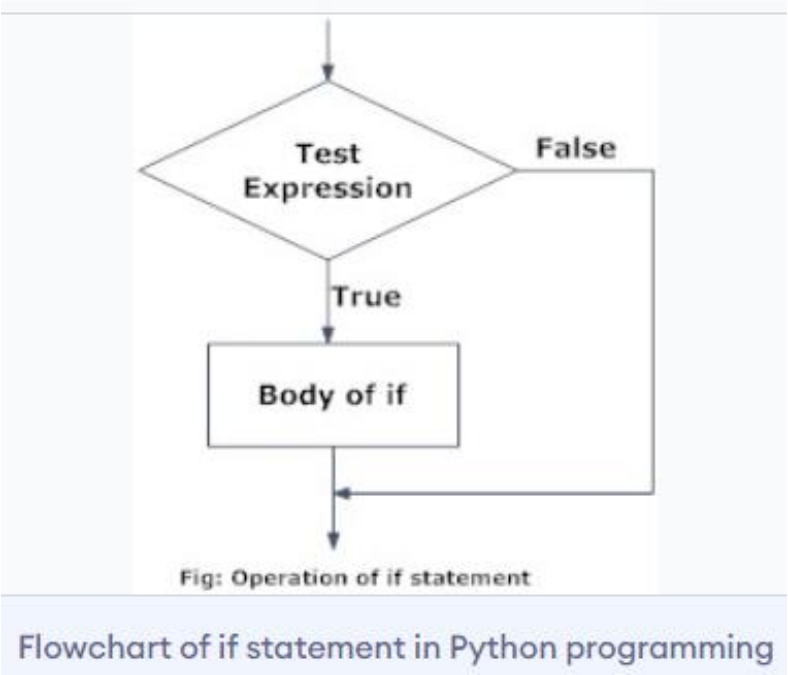
- Decision making is required when we want to **execute a code only if a certain condition is satisfied.**

Syntax

```
if test expression:
    statement(s)
```

- The program **evaluates** the **test expression** and will execute statement(s) **only if the test expression is True**.
- If the test expression is **False**, the **statement(s) is not executed.**
- Python interprets **non-zero values as True**. **None** and **0** are interpreted as **False**.

if Statement Flowchart



Flowchart of if statement in Python programming

Example: if Statement

```
firstNumber = 3
if firstNumber > 0:
  print(firstNumber, "is a positive number.")
print("This is always printed.")

secondNumber = -1
if secondNumber > 0:
  print(secondNumber, "is a positive number.")
print("This is also always printed.")
```

Output

```
3 is a positive number.
This is always printed.
This is also always printed.
```

if...else Statement

Syntax

```
if test expression:
    Body of if
else:
    Body of else
```

- The if..else statement evaluates test expression and will **execute the body of if only when the test condition is True.**
- If the **condition is False**, the **body of else is executed**.

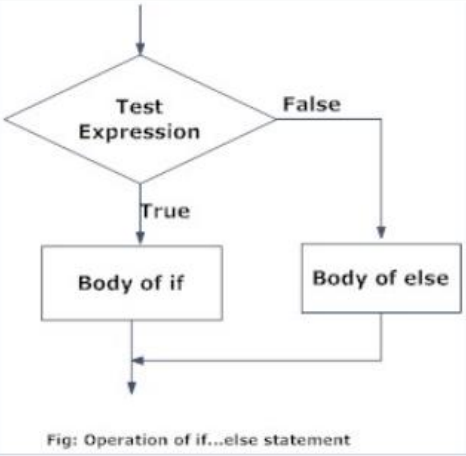


Fig: Operation of if...else statement

Flowchart of if...else statement in Python

Example: if...else Statement

```
number = -5

if number >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
```

Output

```
Negative number
```

if...elif...else Statement

Syntax

```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```

- The **elif** is short for else if. It allows us to **check for multiple expressions**.
- If the **condition for if is False**, it **checks the condition** of the **next elif block** and so on.
- If **all the conditions are False**, the **body of else is executed**.
- **Only one block** among the several if...elif...else blocks is executed according to the condition.
- The if block can have only one else block. But it can have multiple elif blocks.

Flowchart of if...elif...else

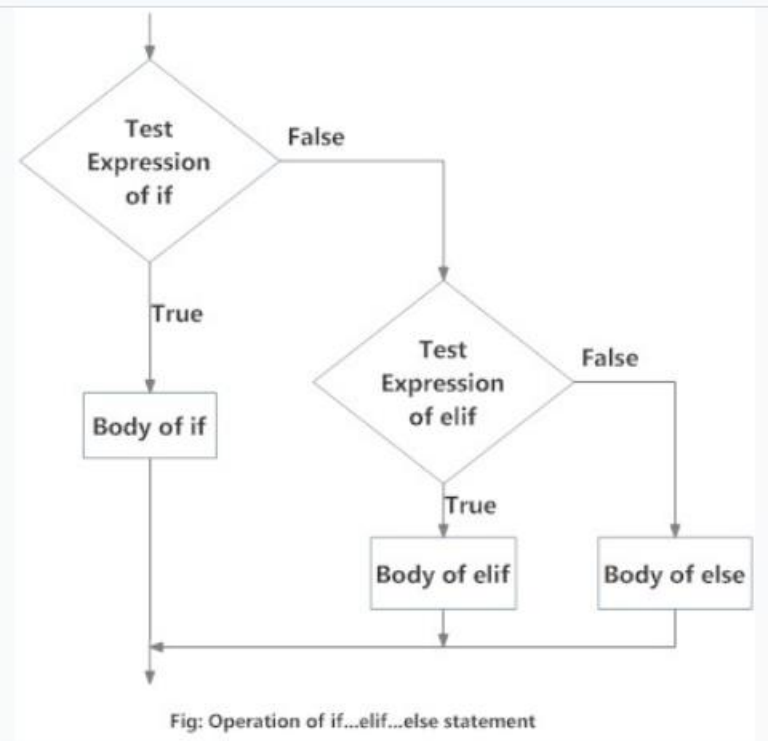


Fig: Operation of if...elif...else statement

Flowchart of if...elif....else statement in Python

Example: if...elif...else Statement

```
number = 0

if number > 0:
  print("Positive number")
elif number == 0:
  print ("Zero")
else:
    print("Negative number")
```

Output

```
Zero
```

Nested if statements

- We can have a if...elif...else statement inside another if...elif...else statement. This is called **nesting** in computer programming.
- **Any number** of these statements can be **nested inside one another.**
- **Indentation** is the only way to figure out the level of nesting.

Example: Nested if

```
number = float (input("Enter a number: "))

if number >= 0:
  if number == 0:
    print("Zero")
  else:
    print("Positive number")
else:
    print("Negative number")
```

Output 1

```
Enter a number: 5
Positive number
```

Output 2

```
Enter a number: -1
Negative number
```

Output 3

```
Enter a number: 0
Zero
```