

Programming Fundamentals Lab



Lab # 06

Conditional Statements

Instructor: Engr. Muhammad Usman

Email: usman.rafiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2021

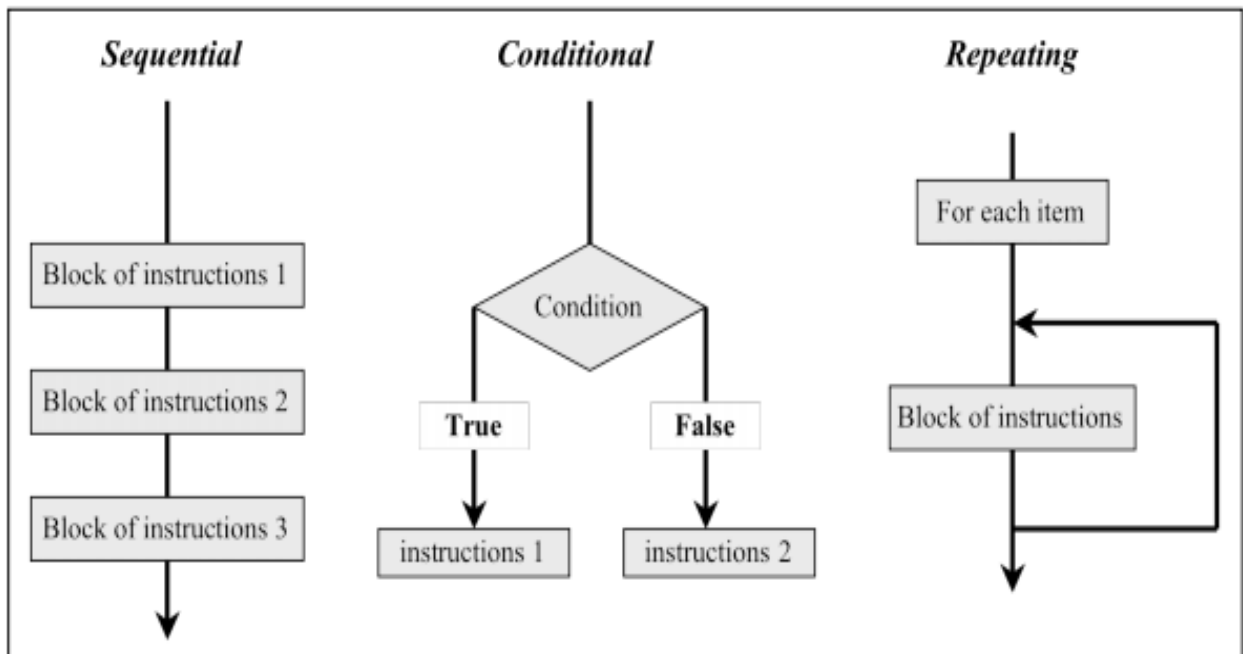
Department of Computer Science,
National University of Computer and Emerging Sciences FAST
Peshawar Campus

Contents

Control Structures	2
Selection Statements/control Statement	2
1. If Statement (One-Way Decision)	3
2. If-else Statement (Two-Way Decisions).....	5
3. Python if...elif...else Statement (Multi-Way Decisions)	7
4. Nested if-else:	9
References	11

Control Structures

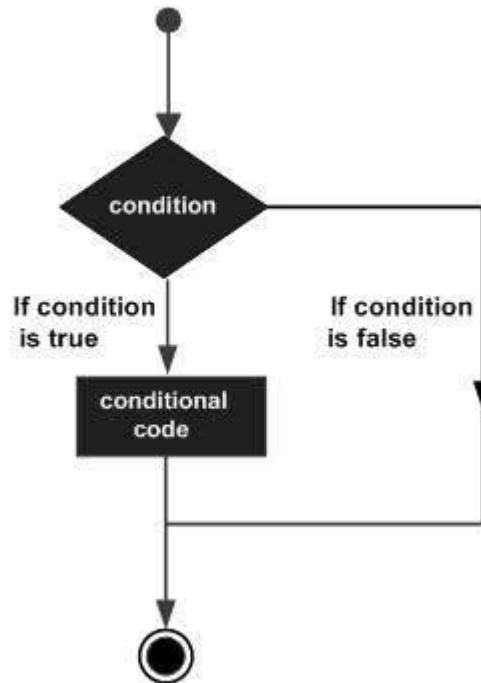
Algorithms require two important control structures: iteration(repeating) and selection(conditional). Both are supported by Python in various forms. The programmer can choose the statement that is most useful for the given circumstance.



Selection Statements/control Statement

Decision making is the most important aspect of almost all the programming languages. As the name implies, decision making allows us to run a particular block of code for a particular decision. Here, the decisions are made on the validity of the particular conditions. Condition checking is the backbone of decision making.

Following is the general form of a typical decision making structure found in most of the programming languages.



1. If Statement (One-Way Decision)

In Python there is only one kind of selection structure: the if statement. There are no switch, cases or whatever other selection structure you know in another language.

The most fundamental control structure is the if structure. It is used to protect a block of code that only needs to be executed if a prior condition is met (i.e., is TRUE). The general format of an if statement is:

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

The condition is one of the logical expressions we have seen in the previous section. The code block is a set of instructions grouped together. The code block is only executed if the condition is TRUE.

Python if Statement Flowchart

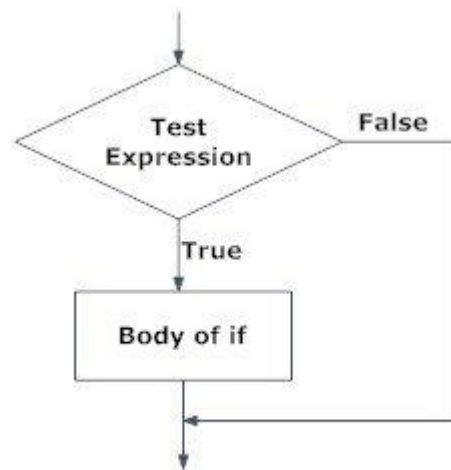


Fig: Operation of if statement

Example 1:

```
1 # If statement
2
3 x = 100
4 y = 300
5
6 if x < y:
7     print("x is less than y")
```

x is less than y

Example 2:

```
In [4]: 1 # If the number is positive, we print an appropriate message
2
3 num = 3
4 if num > 0:
5     print(num, "is a positive number.")
6 print("This is always printed.")
7
8 num = -1
9 if num > 0:
10    print(num, "is a positive number.")
11 print("This is also always printed.")
12
```

When you run the program, the output will be:

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

In the above example, `num > 0` is the test expression.

The body of if is executed only if this evaluates to True.

When the variable `num` is equal to 3, test expression is true and statements inside the body of if are executed.

If the variable `num` is equal to -1, test expression is false and statements inside the body of if are skipped.

The `print()` statement falls outside of the if block (unindented). Hence, it is executed regardless of the test expression.

2. If-else Statement (Two-Way Decisions)

Syntax of if...else

```
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. Indentation is used to separate the blocks.

Python if..else Flowchart

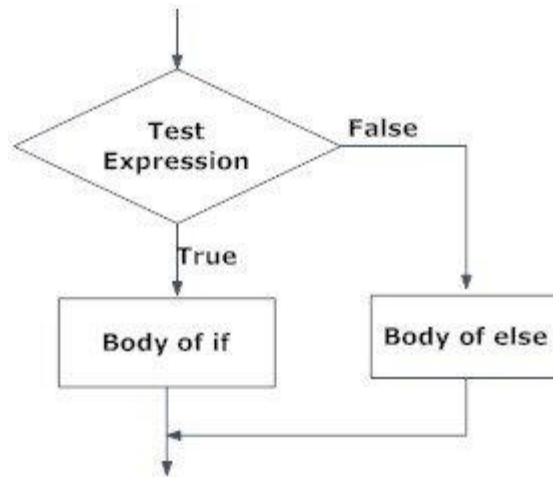


Fig: Operation of if...else statement

Example 3:

```
In [7]: 1 # Program checks if the number is positive or negative
        2 # And displays an appropriate message
        3
        4 num = 3
        5
        6 # Try these two variations as well.
        7 # num = -5
        8 # num = 0
        9
       10 if num >= 0:
       11     print("Positive or Zero")
       12 else:
       13     print("Negative number")
```

Positive or Zero

In the above example, when num is equal to 3, the test expression is true and the body of if is executed and the body of else is skipped.

If num is equal to -5, the test expression is false and the body of else is executed and the body of if is skipped.

If num is equal to 0, the test expression is true and body of if is executed and body of else is skipped.

3. Python if...elif...else Statement (Multi-Way Decisions)

Syntax of if...elif...else

```
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

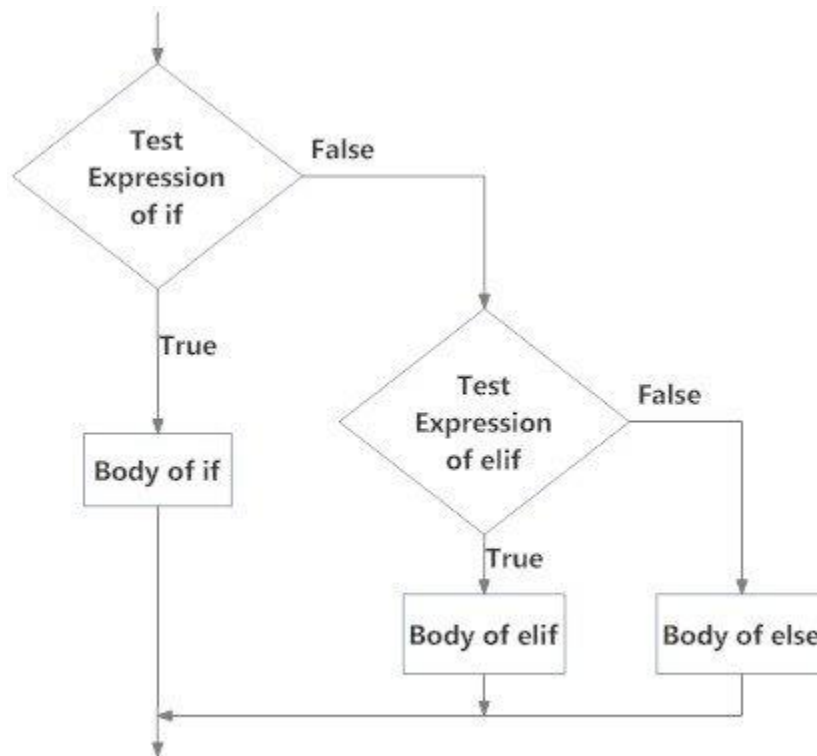


Fig: Operation of if...elif...else statement

Example 4:

```
In [10]: 1 #In this program, we check if the number is positive
2 #or negative or zero and display an appropriate message
3
4
5 num = 5
6 # Try these two variations as well:
7 # num = 0
8 # num = -4.5
9
10 if num > 0:
11     print("Positive number")
12 elif num == 0:
13     print("Zero")
14 else:
15     print("Negative number")
```

Positive number

When variable num is positive, Positive number is printed.

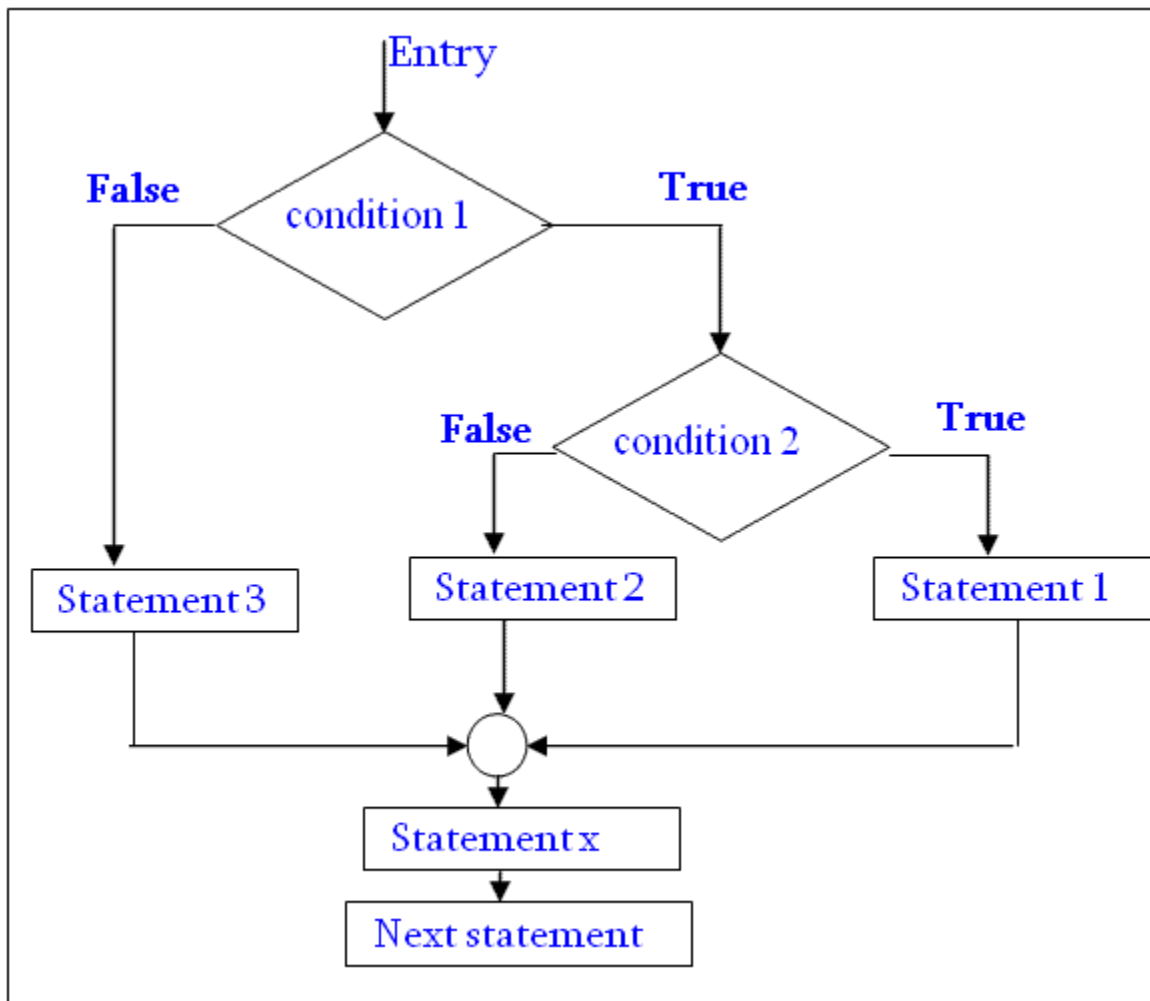
If num is equal to 0, Zero is printed.

If num is negative, Negative number is printed.

4. Nested if-else:

There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if construct.

In a nested if construct, you can have an if...elif...else construct inside another if...elif...else construct.



Example 5:

```
In [14]: 1 '''In this program, we input a number
2 check if the number is positive or
3 negative or zero and display
4 an appropriate message
5 This time we use nested if statement'''
6
7 num = float(input("Enter a number: "))
8 if num >= 0:
9     if num == 0:
10         print("Zero")
11     else:
12         print("Positive number")
13 else:
14     print("Negative number")
```

```
Enter a number: -6
Negative number
```

Example 6:

```
In [16]: 1 # a program which tell us about the grade according to the marks entered
2 # by the user
3 marks = int(input("Enter the marks? "))
4 if marks > 85 and marks <= 100:
5     print("Congrats ! you scored grade A ...")
6 elif marks > 60 and marks <= 85:
7     print("You scored grade B + ...")
8 elif marks > 40 and marks <= 60:
9     print("You scored grade B ...")
10 elif (marks > 30 and marks <= 40):
11     print("You scored grade C ...")
12 else:
13     print("Sorry you are fail ?")
```

```
Enter the marks? 45
You scored grade B ...
```

Example 7:

```
In [17]: 1 # program that grants access only to kids aged between 8-12
          2 age = 18
          3
          4 if ((age>= 8) and (age<= 12)):
          5     print("YOU ARE ALLOWED. WELCOME !")
          6 else:
          7     print("SORRY ! YOU ARE NOT ALLOWED. BYE !")

SORRY ! YOU ARE NOT ALLOWED. BYE !
```

References

<https://www.javatpoint.com/python-if-else>

<https://www.programiz.com/python-programming/if-elif-else>

https://www.tutorialspoint.com/python/python_if_statement.htm