

Class-XI
Computer Science
Conditional Statements
Notes

Types of statements in Python : Statements are the instructions given to the computer to perform any kind of action. Python statements can belong to one of the following three types.

Empty Statement

A statement that does nothing. In Python an empty statement is pass statement. It takes the following form:

Pass

Whenever Python encounters a pass statement, Python does nothing and moves to next statement in the flow of control.

Simple Statement

Any executable statement is a simple statement in Python. For example:

Name=input("enter your name")

Compound Statement

A compound statement represents a group of statements executed as a unit. The compound statement in Python are written in a specific pattern .

It has :

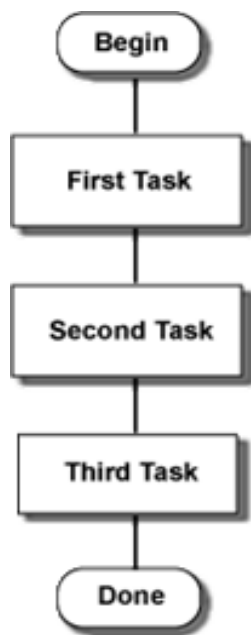
-> A header line which begins with a keyword and ends with a colon.

-> A body consisting of one or more Python statements, each indented inside the header line.

Statement Flow Control : In a program statements may be executed sequentially, selectively or iteratively. Every programming language provides constructs to support sequence, selection or iteration.

Sequential Statements:

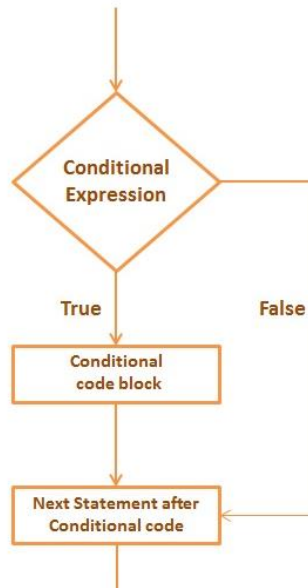
- Sequential statements are set of statements where the execution process will happen in sequence manner. So, these kind of statements are called as sequential statements.
- The problem in sequential statements is, if the logic has broken in any one of the line, then complete source code execution will get broken. So, we are going to decision making, iterative and jump statements to avoid this kind of problems.



Decision Making Statements:

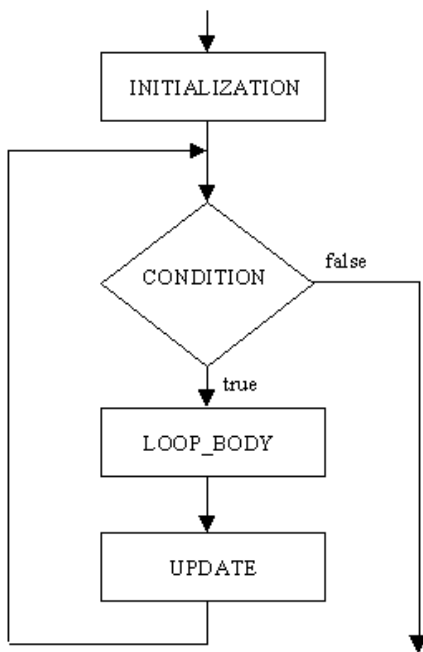
- Decision Making Statements allow the program to take the decision as which statement should be executed next.
- Decision Making statements are used when we want a set of instructions should be executed in one situation and different instructions should be executed in another situation .
- Decision making can be implemented in python using,
 1. **if statements**
 2. **if-else statements**
 3. **if-elif-else statements**

4. **nested if statements**

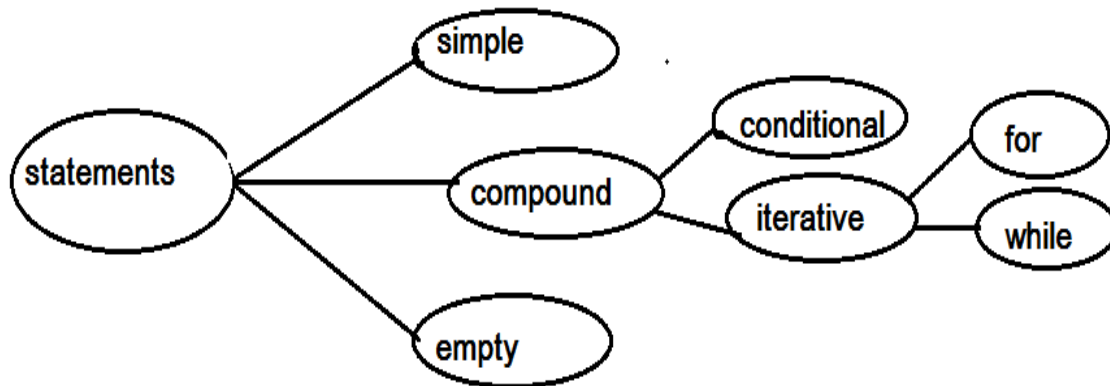


Iteration

The iteration constructs mean repetition of a set of statements depending upon a condition-test. Till the time a condition is true, a set of statements are repeated again and again. As soon as the condition becomes False, the repetition stops. The iteration construct is also called looping construct.



TYPE OF STATEMENTS-MIND MAP



Program Logic Development Tools:

Before developing the solution of a problem in terms of a program, you should read and analyze the given problem and decide about basic sub tasks needed to solve a problem and the order of these subtasks.

The various logic development tools are:

Algorithm and Flowchart






Algorithm : An Algorithm is a step by step procedure to solve a given problem.

For example, the algorithm to find the remainder of given two numbers is:

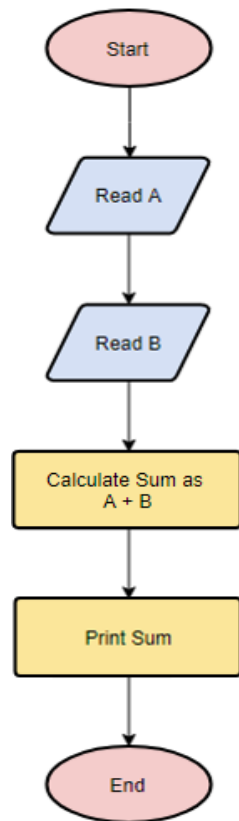
1. Input first the number.
2. Input second number.
3. Divide first number with second number and store the remainder as third number.
4. Display the result.

It is a set of ordered and finite steps to solve a given problem.

Flowchart: A flowchart is a graphical representation of an algorithm. A flowchart shows different subtasks with different symbols. Some commonly used flowchart symbols are :

Flow Chart Symbol	Meaning	Explanation
	Start and end	The symbol denoting the beginning and end of the flow chart.
	Step	This symbol shows that the user performs a task. (Note: In many flow charts steps and actions are interchangeable.)
	Decision	This symbol represents a point where a decision is made.
	Action	This symbol means that the user performs an action. (Note: In many flow charts steps and actions are interchangeable.)
	Flow line	A line that connects the various symbols in an ordered way.

For Example the flowchart to find sum of two numbers is as follows:



Conditional statement :

In programming, very often we want to check the conditions and change the behavior of the program.

How to use Conditional Statements

We can write programs that has more than one choice of actions depending on a variable's value.

Perhaps the most well-known statement type is the if statement.

You use the if statement to perform one action if one thing is true, or any number of other actions, if something else is true.

We must use indentation to define that code that is executed, based on whether a condition is met.

The if statements of Python : The if statements are the conditional statements in Python and these implement selection constructs.

An if statement tests a particular condition; if the condition evaluates to true, a course-of-action is followed i.e. a statement or set-of-statements is executed.

Syntax :

```
if <condition>:  
    statement
```

Example 1:

```
if grade=='A':  
    print("well done")
```

Example 2:

```
if a>b :  
    print("A has more than B has")
```

Example 3:

```
age=18  
age=17  
if age>= 18:
```

```
print "Person eligible for vote"
```

```
print "Person not eligible for vote"
```

```
print "*****"
```

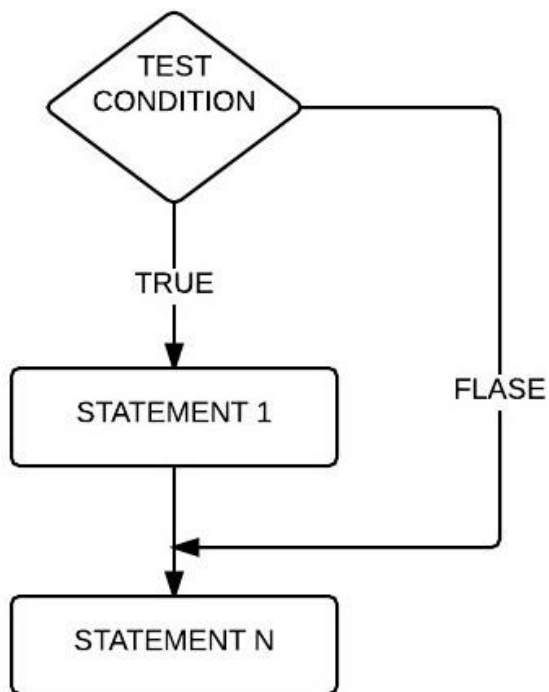
OUTPUT

Person not eligible for vote

Person eligible for vote

Person not eligible for vote

FLOW CHART



If-else statement :

An else statement can be combined with an if statement.

An else statement contains the 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 most only one else statement following if.

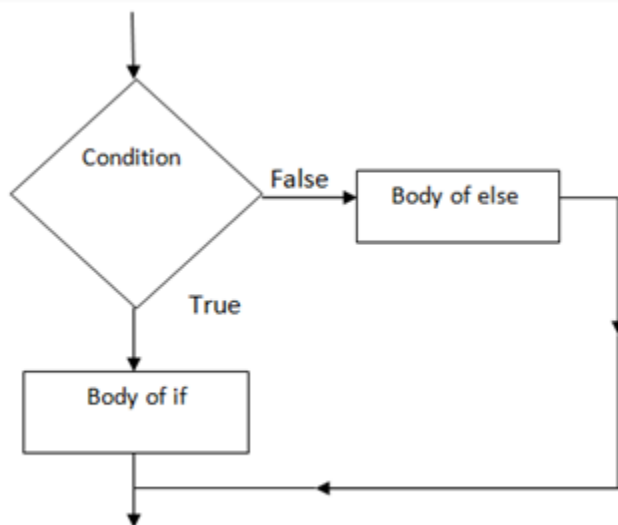
The syntax of if..else is:

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

Example :

```
if grade == 'A':
    print ("well done")
else:
    print ("try again")
```

FLOW CHART



Example1:

age=25


```
if age>18:

    print "Person eligible for vote"

elif age==18:

    print "Person just eligible for vote"

else:

    print "Person not eligible for vote"
```

Example 2:

This script will compare two strings based on the input from the use

```
# This program compares two strings.

# Get a password from the user.
password = raw_input('Enter the password: ')

# Determine whether the correct password
# was entered.

if password == 'hello':
    print'Password Accepted'

else:
    print'Sorry, that is the wrong password.'
```

Example 3:

Program to check whether the given number is odd or even

```
num= int("Enter an integer")

if (num%2==0) :

    print(num, "is even number")

else:
```

```
print(num, "is odd number ")
```

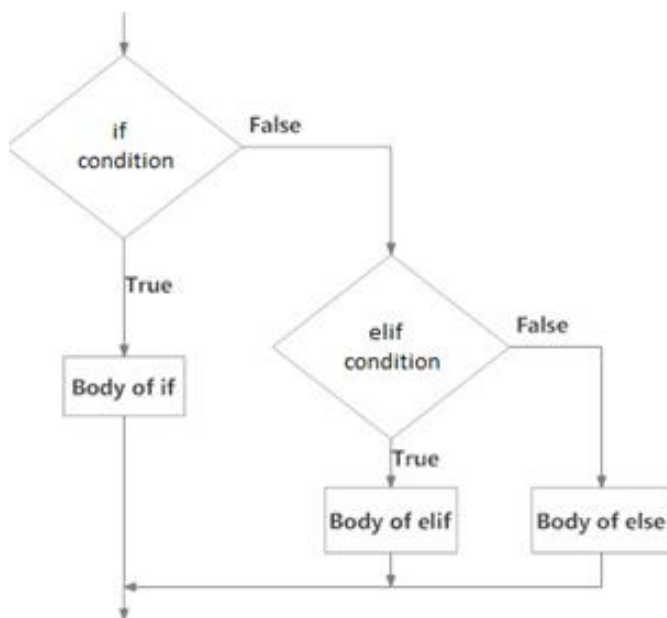
The if-elif Statement

Sometimes there are more than two possibilities; in that case we can use the elif statement

It stands for “else if,” which means that if the original if statement is false and the elif statement is true, execute the block of code following the elif statement.

The **Syntax** of the if...elif statement is:

```
if expression1:  
    statement(s)  
elif expression2:  
    statement(s)  
elif expression3:  
    statement(s)  
else:  
    statement(s)
```



Example 1:

```
num=int(input("enter the number"))  
if num<0:  
    print("Invalid entry. Valid range is 0 to 999.")  
elif num<10:  
    print("single digit number is entered")  
elif num<100:  
    print("double digit number is entered")  
elif num<=999:  
    print("three digit number is entered")  
else:  
    print("invalid entry.valid range is 0 to 999.")
```

Sample runs of this program are given below;

Enter a number (0..999) : -3

Invalid entry. Valid range is 0 to 999

Example 2:

Let's show one more examples, in which will also make use of the elif statement.

```
number = 20  
  
guess = int(input('Enter an integer : '))  
  
if guess == number:  
    print('Congratulations, you guessed it.')  
  
elif guess < number:  
    print('No, it is a little higher than that')
```

```
else:  
    print('No, it is a little lower than that')
```

Example 3 :

Program to read two numbers and an arithmetic operator and displays the computed result.

```
num1= int (input("enter first number"))  
num2= int(input("enter second number "))  
op =input("enter operator [+ - * / % ] :")  
result=0  
if op == '+' :  
    result= num1+num2  
elif op== '-' :  
    result=num1-num2  
elif op== '+' :  
    Result=num1+num2  
elif op== '/' :  
    result=num1+num2  
elif op== '%' :  
    result=num1%num2  
print(num1, op , num2,'=' , result)
```

The Nested-if statement :

One conditional can also be **nested** within another. i.e., 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.

How *nested if* works:

Initially, the flow of control enters the **outer if** and test expression is evaluated.

When resolving as **true**, it executes inside statements of *if part* and goes to *inner if*.

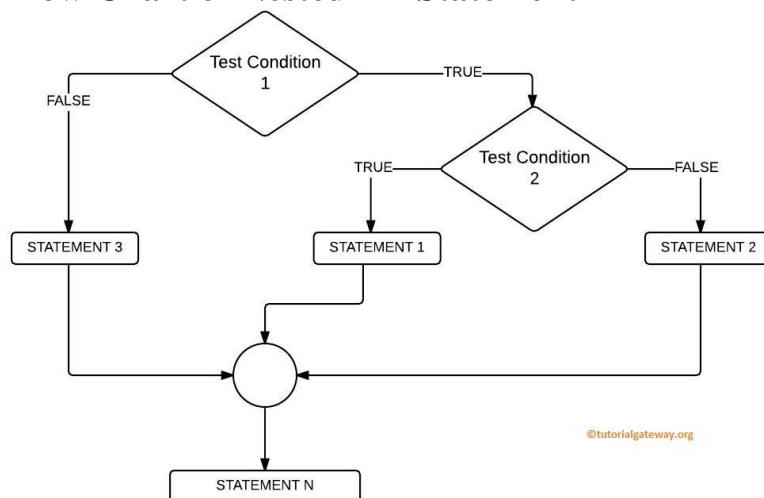
Then, test expression of *inner if* is evaluated. If it is **true**, the body part of *inner if* statements are executed. Otherwise, if it is **false**, the flow of control jumps out to the *outer if* body part to evaluate **test expression**.

When resolve as false(outr if), the flow of control jumps to *its else* part.

Syntax of Nested if..else statement is

```
if TEST EXPRESSION1:  
    if TEST EXPRESSION2:  
        STATEMENTS_B  
    else:  
        STATEMENTS_C  
else:  
    if TEST EXPRESSION3:  
        STATEMENTS_D  
    else:  
        STATEMENTS_E
```

Flow Chart of Nested – if Statement



Example 1:

```
a=10
b=20
c=5
if a>b:
    if a>c:
        print("Greatest number is ",a)
    else:
        print("Greatest number is",c)
else:
    if b>c:
        print("Greatest number is ",b)
    else:
        print("Greatest number",c)
```

Example 2:

```
age=20
gpa=1.1
if age>18:
    if gpa>0.8:
        print("Your age and GPA are both enough");
        print("You have been selected for university")
    else:
        print("your gpa is not enough")
else:
    print("your age is not enough")
```

Example 3:

To check if the number is a positive or negative number or zero

```
#a number get from user
# tis program used to check number
#is it negative or positive or Zero
number=input("Enter the number for check\n")
if number>=0: #outer if
    if number==0: #inner if
        print("number is Zero")
```

```
    else:
        print("Number is positive")
else:
    print("Number is Negative")
```

When we execute the above program, it will be produced the following result.

Output 1

Enter a number: 5
Number is positive

Output 2

Enter a number: -5
Number is Negative

Output 3

Enter a number :0
number is zero

Example 4: Program to check whether a given year is leap year or not.

```
year = int(input("Enter a year: "))
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".year)
        else:
            print("{0} is not a leap year".year)
    else:
        print("{0} is a leap year".year)
else:
    print("{0} is not a leap year".year)
```

SUMMARY:

1. Statements are the instructions given to the computer to perform any kind of tasks.
2. Python statements can be on one of these types: empty statement, single statement and compound statement.
3. An empty statement is the statement that does nothing. Python offers pass statement as empty statement.
4. Single executable statement forms a simple statement.
5. A compound statement represents a group of statement executed as a unit.
6. Every compound statement of a python has a header and an indented body below the header.
7. The flow of control in a program can be in three ways : sequentially, selectively and iteratively.
8. The sequence constructs means execution of statements one after the another.
9. The selection constructs means execution of statements depending upon a condition.
10. The iteration constructs means execution of a set of statements depending upon a condition test.
11. Python provides one selection statement in different forms-if...else and if.....elif...else.
12. The if.....else statement can be nested also.