CONDITIONAL STATEMENTS (Decision Making)

- The basic decision statements in computer is selection structure.
- The decision is described to computer as a conditional statement that can be answered True or False.

Python language provide the following conditional (Decision making) statements.

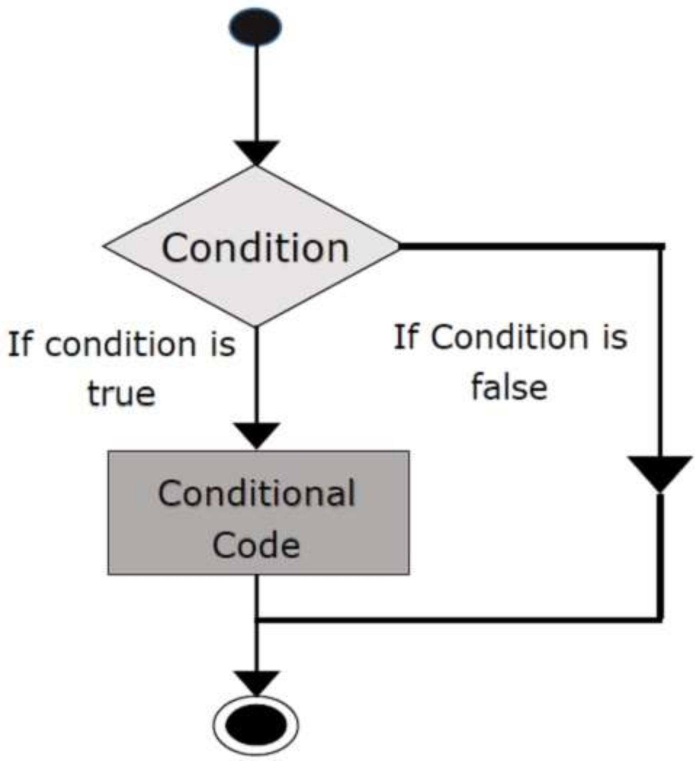
- if statement
- if...else statement
- if...elif...else staement
- Nested if..else statement

The if statement

- The **if** statement is a decision making statement. It is used to control the flow of execution of the statements and also used to test logically whether the condition is true or false.

Syntax

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



Example program

```
i=int(input(“Enter the number:”))
If (i<=10):
    print(“ condition is true”)
```

OUTPUT

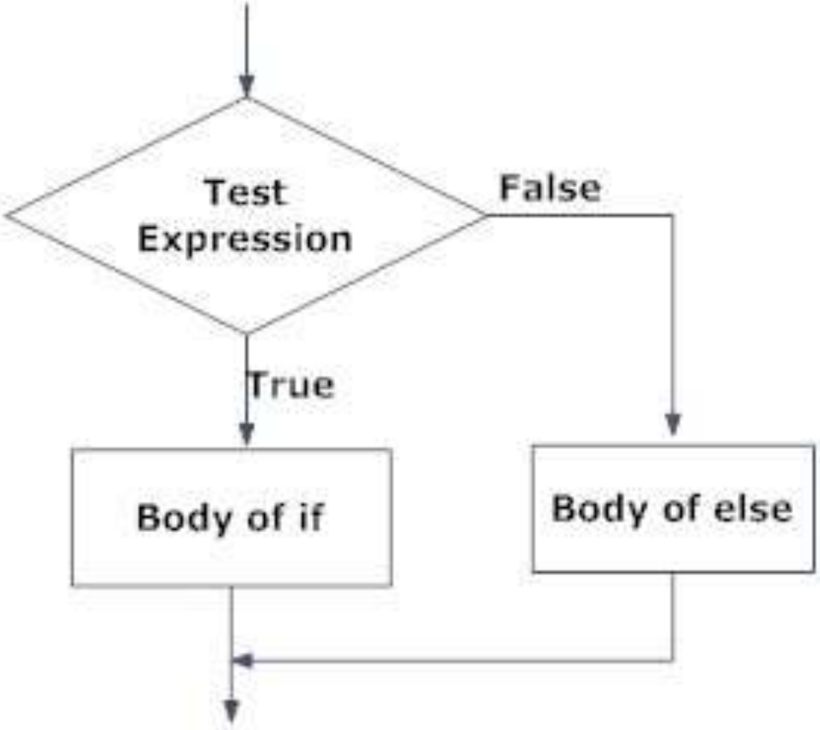
```
Enter the number: 9
Condition is true
```

If … else statement

- The if…else statement is called alternative execution, in which there are two possibilities and the condition determines wich one gets executed.

Syntax

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



Write a program to check if a number is Odd or Even

```
num = int(input(“Enter the number:”))
if (num % 2)== 0:
    print (“Given number is Even”)
else:
    print(“ Given number is Odd”)
```

OUTPUT

```
Enter the number: 9
Given number is Odd
```

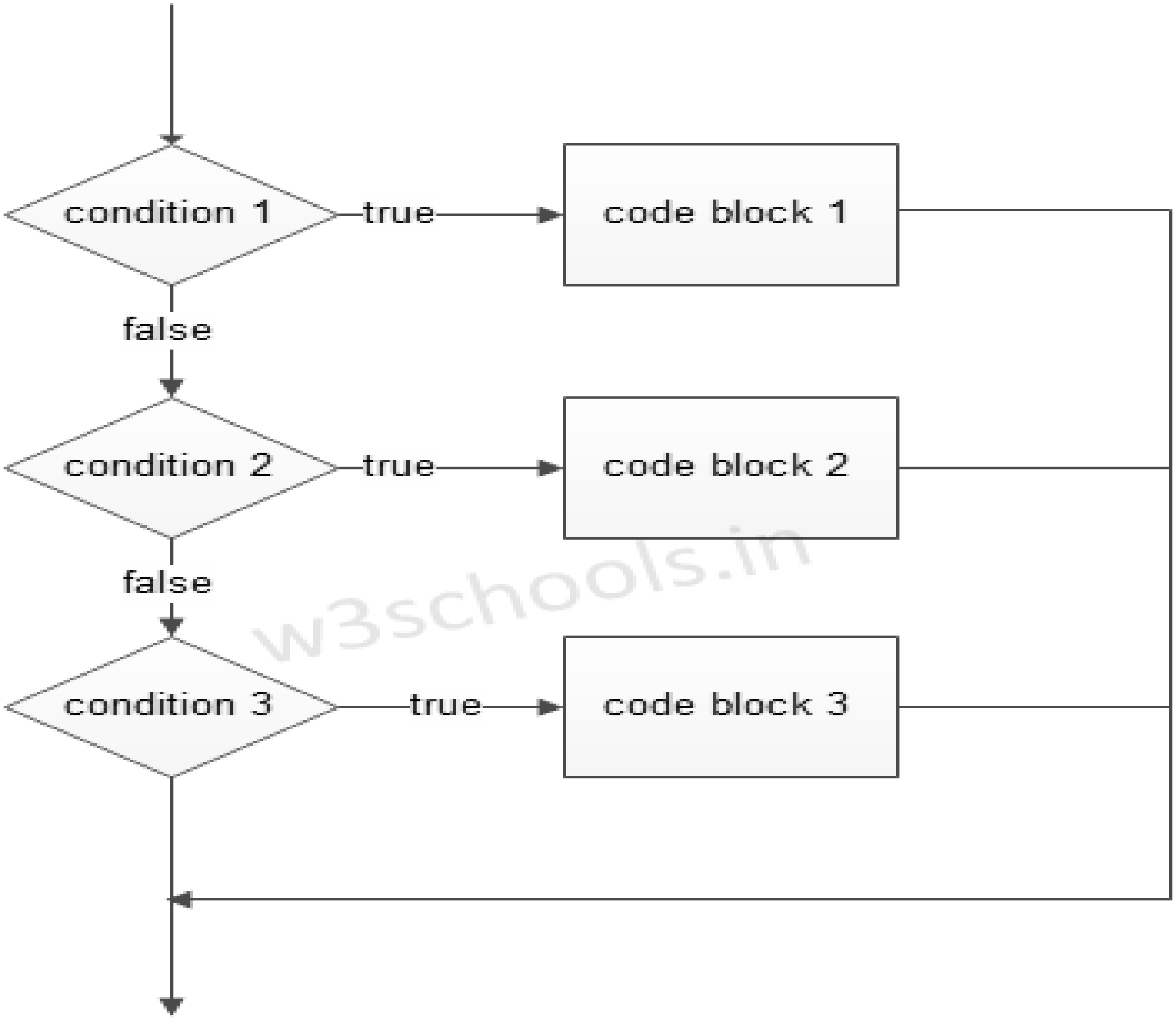
elif Statements

- elif – is a keyword used in Python in replacement of else if to place another condition in the program. This is called chained conditional.
- Chained conditions allows than two possibilities and need more than two branches.

SYNTAX

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

Figure – elif condition Flowchart



Example: largest among three numbers

```
a = int(input(“Enter 1st number:”))
b= int(input(“Enter 2nd number:”))
c= int(input(“Enter 3rd number:”))
if (a > b) and (a > c):
        print("a is greater")
 elif (b < a) and (b < c):
        print(“b is greater")
 else:
        print(“c is greater")
```

OUTPUT

```
Enter 1st number:10
Enter 2nd number:25
Enter 3rd number:15
B is greater
```

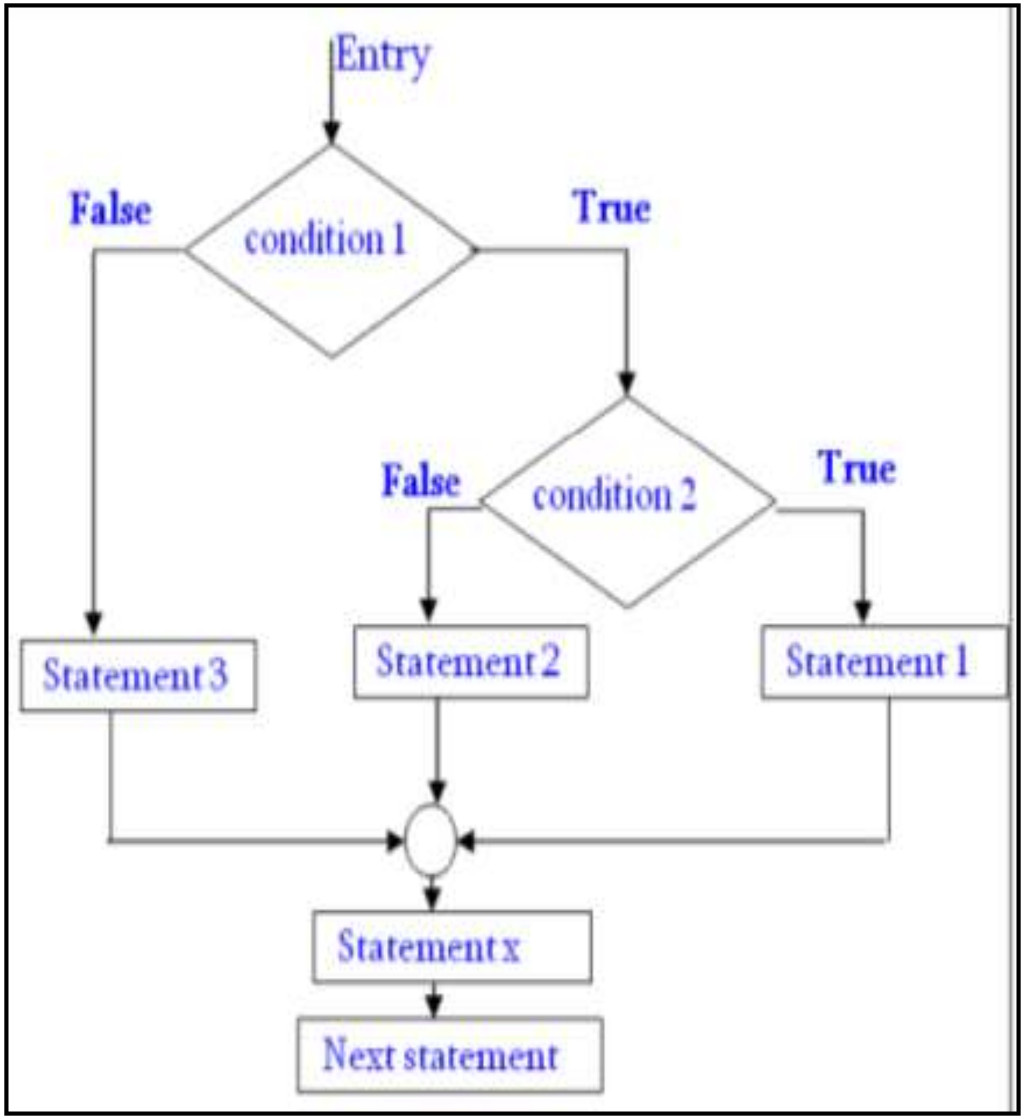
Nested if … else Statements

- We can write an entire if… else statement in another if… else statement called nesting, and the statement is called nested if.

- In a nested **if** construct, you can have an if … elif … else construct inside an if … elif.. Else construct.

Syntax

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



- Example program

```
n = int(input(“Enter  number:”))
If (n<=15):
   if (n == 10):
         print(‘play cricket’)
   else:
         print(‘play kabadi’)
Else:
   print(‘Don’t play game’)
```

```
OUTPUT
Enter number : 10

Play cricket
```

CONTROL STATEMENT (Looping Statement)

- Program statement are executed sequentially one after another. In some situations, a block of code needs of times.
- These are repetitive program codes, the computers have to perform to complete tasks. The following are the loop structures available in python.

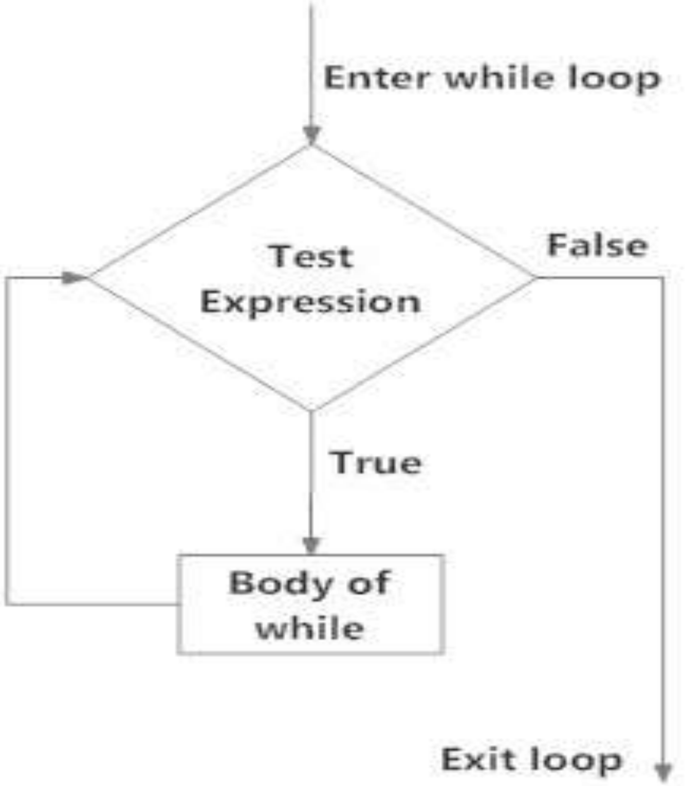
- while statement
- for loop statement
- Nested loop staement

While loop statement

- A **while** loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

Syntax of while loop

```
while expression:
      statement(s)
```



Write a program to find sum of number

```
num = int(input("Enter a number: "))
sum = 0
while(num > 0):
   sum = sum+num
num = num-1
   print("The sum is",sum)
```

```
OUTPUT

Enter a number: 10

The sum is 55
```

Using else statement with while loops

- Python supports t have an else statement associated with a loop statement.
- If the else statement is used with a while loop, the else statement is executed when the condition false.

Program to illustrate the else in while loop

```
counter = 0
while counter < 3:
    print("Inside loop")
    counter = counter + 1
else:
    print(“Outside loop")
```

```
OUTPUT
Inside loop
Inside loop
Inside loop
 Outside loop
```

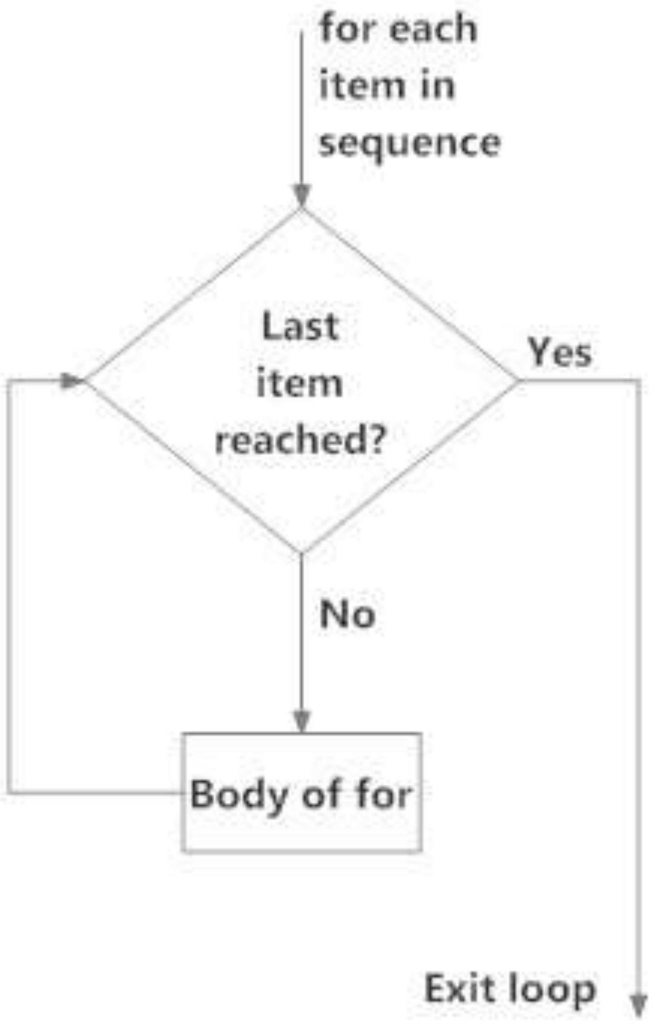
For loop statement

- The for loop is another repetitive control structure, and is used to execute a set of instructions repeatedly, until the condition becomes false.
- The for loop in Python is used to iterate over a sequence (list, tuple, string) or other iterable objects. Iterating over a sequence is called traversal.

Syntax

```
for val in sequence:
        Body of for loop
```

For loop flow chart



Addition of number using for loop

```
numbers = [6, 5, 3, 8, 4, 2, 5, 4]
sum1 = 0
for val in numbers:
sum1 = sum1+val
print("The sum is", sum1)
```

OUTPUT

```
The sum is 37
```

for Loop and for Loop with else

EX-01:

```
genre = ['pop', 'rock', 'jazz']
for i in range(len(genre)):
    print("I like", genre[i])
```

```
OUTPUT
I like pop
I like rock
I like jazz
```

EX-02:

```
genre = ['pop', 'rock', 'jazz']
for i in range(len(genre)):
        print("I like", genre[i])
else:
    print("No items left.")
```

```
OUTPUT
I like pop
I like rock
I like jazz
No items left.
```