Informatics Practices

Flow of Control
By
Rajesh Verma

Rajesh Verma, MRA DAV Public School, Solan

Types of statements in python

- *Statements are the instructions given to the computer to perform any kind of actions, be it data movements, making decisions or be it repeating action.*
- *Three types of python statement:*

1. *Empty statement*
2. *Simple Statement*
3. *Compound statement*

Types of statements in python

- *Empty statement : The simplest statement is the empty statement i.e., a statement which does nothing. For example: pass*
- *The pass statement of python is do nothing . A pass statement is useful in those instances where the syntax of language require presence of statement but where the logic of the program does not.*

Types of statements in python

- ***Simple Statement:*** *Any single executable statement is a simple statement in python. For example:*

```
name = input ("Enter your name")
```

Types of statements in python

- *Compound Statement: A compound statement represents a group of statements executed as a unit. Compound statement has :*
- *A header line which begin with a keyword and with a colon.*
- *A body consisting of one or more Python statements, each inside the header line. All the statements in the body are at the same indentation.*

Flow of Control

- *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.*

Flow of Control

Two types of statements in control flow :

1. *Conditional Statement*

2. *Repeated or Iterative or Loops statement*

Conditional Statements

The if statement is the conditional statement in Python.

There are 3 forms of if statement:

1. *Simple if statement*
2. *The if..else statement*
3. *The If..elif..else statement*

Simple if statement

- *The if statement tests a condition & in case the condition is True, it carries out some instructions and does nothing in case the condition is False.*

- Syntax

```
if <condition>:
        statement
        [statements]
```

The header of **if** statement, colon (:) at the end

Body of if

- e.g.

```
if  amount>1000:
        disc = amount * .10
```

Remember :

Python uses indentation to define code blocks, instead of brackets. The standard Python indentation is 4 spaces, although tabs and any other space size will work, as long as it is consistent. Notice that code blocks do not need any termination.

Example of *if* statement

- *Progam to find discount (10%) if amount is more than 1000.*

```
Price = float (input("Enter Price ? " ))
Qty = float (input("Enter Qty ? " ))
Amt = Price * Qty
print(" Amount :" , Amt)
if Amt >1000:
        disc = Amt * .10
        print("Discount :", disc)
```

Body of if statement
(will be executed incase condition is true)

The if-else statement

- *The if - else statement tests a condition and in case the condition is True, it carries out statements indented below if and in case the condition is False, it carries out statement below else.*

- *Syntax*

```
if <condition> :
        statement       }
        [statements]    } Block 1
else :
        statement       }
        [statements]    } Block 2
```

- *e.g.*

```
if amount>1000:
        disc = amount * 0.10
else:
        disc = amount * 0.05
```

Example of if-else statement

- *Program to find discount (10%) if amount is more than 1000, otherwise (5%).*

```
Price = float (input("Enter Price ? " ))
Qty = float (input("Enter Qty ? " ))
Amt = Price * Qty
print(" Amount :" , Amt)
if Amt >1000 :
        disc = Amt * .10                    }  block 1
        print("Discount :", disc)           }  (will be executed incase condition is true)
else :
        disc = Amt * .05                    }  block 2
        print("Discount :", disc)           }  (will be executed incase condition is False)
```

The *if..elif* statement

- *The if - elif statement has multiple test conditions and in case the condition1 is True, it executes statements in block1, and in case the condition1 is False, it moves to condition2, and in case the condition2 is True, executes statements in block2, so on. In case none of the given conditions is true, then it executes the statements under else block*

- *Syntax*

```
if <condition1> :
            statement
            [statements]      }  Block 1

elif <condition2> :
            statement
            [statements]      }  Block 2

elif <condition3> :
            statement
            [statements]      }  Block 3
  :
  :
else :
            statement
            [statements]
```

Example of *if-elif* statement

- *Prog to find discount (20%) if amount>3000, disc(10%) if Amount <=3000 and >1000, otherwise (5%).*

```
Price = float (input("Enter Price ? " ))
Qty = float (input("Enter Qty ? " ))
Amt = Price* Qty
print(" Amount :" , Amt)
if Amt >3000 :
            disc = Amt * .20
            print("Discount :", disc)
elif Amt>1000:
            disc = Amt * .10
            print("Discount :", disc)
else :
         disc = Amt * .05
            print("Discount :", disc)
```

block 1
(will be executed incase condition 1 is true)

block 2
(will be executed incase condition2 is True)

block 3
(will be executed incase both the condition1 &conditon2 are False)

Example of Nested if statement

- *Program to find Largest of Three Numbers (X,Y,Z)*

```
X = int (input("Enter Num1 ? " ))
Y = int (input("Enter Num2 ? " ))
Z = int (input("Enter Num3 ? " ))

if   X > Y :
            if  X > Z:
                Largest = X
            else:
                Largest = Z
else:
            if  X > Z:
                Largest = X
            else:
                Largest = Z
print("Largest Number :", Largest)
```

Assignments

- *WAP to input a number and check whether it is Even or Odd.*
- *WAP to input a number print its Square if it is odd, otherwise print its square root.*
- *WAP to input a Year and check whether it is a Leap year.*
- *WAP to input a number check whether it is Positive or Negative or ZERO.*
- *WAP to input Percentage Marks of a students, and find the grade as per following criterion:*

Marks	*Grade*
>=90	*A*
75-90	*B*
60-75	*C*
Below 60	*D*

Loops

There are two types of loops in Python:

1. *for loop*
2. *while loop*

LOOPS

- *It is used when we want to execute a sequence of statements (indented to the right of keyword for) a fixed number of times.*
- *Syntax of for statement is as follows:*
  - *i) with range() function*
  - *ii) with sequence*

The for Loop

- *Syntax 1:*

Starting value

Ending value

Step value

```
for <Var> in range (val1,   val2,   Val3):
          Statements to be repeated
```

- Syntax 2:

List or a String

```
for <Var> in <Sequence> :
          Statements to repeat
```

The range() function:

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

- *range(1 , n):*

  will produce a list having values starting from 1,2,3… upto n-1.

  The default step size is 1
- *range(1 , n, 2):*

  will produce a list having values starting from 1,3,5… upto n-1.

  The step size is 2

1) *range(1 , 7): will produce 1, 2, 3, 4, 5, 6.*
2) *range(1 , 9, 2): will produce 1, 3, 5, 7.*
3) *range(5, 1, -1): will produce 5, 4, 3, 2.*
3) *range(5): will produce 0,1,2,3,4.*

   default start value is 0

for loop implementation

```
Sum=0
for  i  in range(1, 11):
            print(i)
            Sum=Sum + i
print("Sum of Series", Sum)
```

OUTPUT:

```
    1
    2
    :
    10
    Sum of Series: 55
```

```
Sum=0
For   i   in range(10, 1, -2):
            print(i)
            Sum=Sum + i
print("Sum of Series", Sum)
```

OUTPUT:

```
    10
    8
    :
    2
    Sum of Series: 30
```

for loop: Prog check a number for PRIME

METHOD 1

```
num= int(input("Enter Num?"))
flag=1
for  i  in range(2, num//2+1):
                if num%i == 0:
                        flag = 0
                        break
if  flag == 1:
    print("It is Prime No.")
else:
     print("It is Not a Prime No.")
```

METHOD 2: using loop else

```
num= int(input("Enter Num?"))
for  i  in range(2, num):
                if num%i == 0:
                    print("It is not  Prime No.")
                    break
else:
    print("It is a Prime No.")
```

Note: the else clause of a loop will be executed only when the loop terminates normally (not when break statement terminates the loop)

Nested *for* Loop

```
for  i  in range ( 1, 6 ):
        print( )
        for j  in range (1,  i + 1):
                print("@", end=" ")
```

Will produce following output

```
@
@@
@@@
@@@@
@@@@@
```

The while Loop

- It is a conditional loop, which repeats the statements with in itself as long as condition is true.
- The general form of while loop is:

```
 while <condition> :
            Statement
           [Statements]
```

$\Big\}$ loop body (these statements repeated until condition becomes false)

- Example:

```
k = 1, sum=0
while  k <= 4 :
          print (k)
          sum=sum + k
print("Sum of series:", sum)
```

$\Rightarrow$

OUTPUT

```
1
2
3
4
          Sum of series: 10
```

while loop: Implementation

- *Prog. To find digit sum*

```
num = int(input("No.?"))
ds = 0
while num>0 :
        ds = ds +num % 10
        num = num // 10
print("Digit Sum :", ds)
```

- *Prog. To find reverse*

```
num = int(input("No.?"))
rev = 0
while num>0 :
        d =  num % 10
        rev = rev*10 + d
        num = num // 10
print("Reverse :", rev)
```

for loop with string

e.g.1:

```
for  ch  in  "Hello" :
             print(ch)
```

Output:

```
H
e
l
l
o
```

e.g. 2:

```
T = "Hello"
for ch in T :
     print(ch)
```

Output:

```
H
e
l
l
o
```

Any Questions Please