

Unit-2

Decision Making Statements

Python If Statements

There are various types of if statements in Python.

1. if.else statement
2. if elif ladder statement
3. nested if statement

1. Python If Statement

Syntax

```
if(condition):  
    True statements  
  
else:  
    false statements
```

Example-

```
a=input("Enter Number :")  
if a%2==0:  
    print( "Even")  
else:  
    print ("Odd")
```

Que- Read two values and print Largest.

If elif ladder

In python, we can use nested If Else to check multiple conditions. Python provides elif keyword to make nested If statement.

This statement is like executing a if statement inside a else statement.

Syntax

If condition-1:

 statement-1

elif condition-2:

 statement-2

elif condition-n:

 statement-n

else:

 default statements

Example-1

Que- Read three values and print the largest .

```
a=input("Enter First Number :")
```

```
b=input("Enter Second Number :")
```

```
c=input("Enter Third Number :")
```

```
if a>b and a>c:
```

```
    print "Largest=",a
```

```
elif b>a and b>c:
```

```
    print "Largest=",b
```

```
else:
```

```
    print "Largest=",c
```

Exercise-1:

Write a python program that displays a message as follows for a given number:

If it is a multiple of three, display "Zip"

If it is a multiple of five, display "Zap".

If it is a multiple of both three and five, display "Zoom".

If it does not satisfy any of the above given conditions, display "Invalid".

Exercise-2:

A teacher in a school wants to find and display the grade of a student based on his/her percentage score. The criterion for grades is as given below:

| Score (both inclusive) | Grade |
|-------------------------------|--------------|
| Between 80 and 100 | A |
| Between 73 and 79 | B |
| Between 65 and 72 | C |
| Between 0 and 64 | D |
| Any other value | Z |

Assume that the percentage score is a whole number.

Python Nested If

If , if statement contains one or more than one statement into its body then this term is known as nested if.

Example-

```
# In this program, we input a number
```

```
# check if the number is positive or
```

```
# negative or zero and display
```

```
# an appropriate message
```

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

```
if num >= 0:
```

```
    if num == 0:
```

```
        print("Zero")
```

```
    else:
```

```
        print("Positive number")
else:
    print("Negative number")
```

Evaluating Python expressions

Python provides several ways to interact with the interpreter from within a program. For example, the `eval` function evaluates a string as if it were a Python expression. You can pass it a literal, simple expressions, or even use built-in functions:

Example: Using the `eval` function

```
def dump(expression):
    result = eval(expression)
    print expression, "=>", result, type(result)
```

```
dump("1")
dump("1.0")
dump("'string'")
dump("1.0 + 2.0")
dump("'*' * 10")
dump("len('world')")
```

Output-

```
1 => 1 <type 'int'>
1.0 => 1.0 <type 'float'>
'string' => string <type 'string'>
1.0 + 2.0 => 3.0 <type 'float'>
```

```
'*' * 10 => ***** <type 'string'>
```

```
len('world') => 5 <type 'int'>
```

Loop

It is used for repeating a part of program during a finite or infinite times.

1. For
2. While

For Loop

Python for loop is used to iterate the elements of a collection in the order that they appear. This collection can be a sequence(list or string).

Syntax

for val in sequence:

 Body of for

Example

```
for i in range(1,11):  
    print i;
```

Example-1: Python Example to Find Sum of 10 Numbers.

```
sum=0  
for n in range(1,11):  
    sum+=n  
print sum
```

Increment of more than one -

```
for i in range(1,10,2):  
    print i
```

Exercises

1. WAP for printing 1 to 10.
2. WAP for printing the table of given number.
3. WAP for printing the list of even values b/w 1 to 50.
4. WAP for printing 10 to 1.
5. WAP for printing the following series-

1

0

1

0

.

.

n times

6. WAP for printing the sum of even numbers between 1 to 50.
7. WAP for printing the factorial of given number.
8. WAP for calculating x to the power n.
9. WAP for printing the sum of even numbers b/w 1 to 100.

While Loop

In Python, while loop is used to execute number of statements or body till the specified condition is true. Once the condition is false, the control will come out of the loop.

Syntax

```
while <expression>:
```

```
    Body
```

Example-1

```
i=1
```

```
while i<=10:
```

```
print i
```

```
i=i+1
```

Example-2

```
a=10
```

```
while a>0:
```

```
    print "Value of a is",a
```

```
    a=a-2
```

```
print "Loop is Completed"
```

Example-3

```
n=153
```

```
sum=0
```

```
while n>0:
```

```
    r=n%10
```

```
    sum+=r
```

```
    n=n/10
```

```
print sum
```

Exercises

1. WAP for reversing the digits of a given number.

```
123
```

```
321
```

2. Check whether the given number is palindrome or not?

```
121
```

3. WAP for printing the sum & average of digits of a given number.
4. WAP for printing the largest digits of a given number.
5. WAP for checking that the given number is armstrong or not?

$$153=1^3+5^3+3^3$$

Python Break

Break statement is a jump statement which is used to transfer execution control. It breaks the current execution and in case of inner loop, inner loop terminates immediately.

When break statement is applied the control points to the line following the body of the loop, hence applying break statement makes the loop to terminate and controls goes to next line pointing after loop body.

Example-1

```
for i in [1,2,3,4,5]:  
    if i==4:  
        print "Element found"  
        break  
    print i,
```

Example-2

```
for letter in 'Python3':  
    if letter == 'o':  
        break  
    print (letter)
```

Exercise-1: Check given number is prime or not?

Python Continue Statement

Python Continue Statement is a jump statement which is used to skip execution of current iteration. After skipping, loop continue with next iteration.

Example-1

```
for i in range(1,11):  
    if i>=5 and i<=8:  
        continue;
```



```
print i,
```

Python Pass

In Python, pass keyword is used to execute nothing; it means, when we don't want to execute code, the pass can be used to execute empty. It is same as the name refers to. It just makes the control to pass by without executing any code. If we want to bypass any code pass statement can be used.

Syntax

```
pass
```

Example

```
for i in range(1,11):
    if i==5:
        pass;
    else:
        print i,
```

Python Nested For Loops

Loops defined within another Loop are called Nested Loops. Nested loops are used to iterate matrix elements or to perform complex computation.

When an outer loop contains an inner loop in its body it is called Nested Looping.

Syntax

```
for <expression>:
    for <expression>:
        Body
```

Example-1:

```
for i in range(1,6):
    for j in range (1,i+1):
```

```
    print i,  
    print
```

Python 3

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

Output:

```
>>>  
1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
>>>
```

Example-2

```
for i in range(1,6):  
    for j in range(1,i+1):  
        print j,  
    print
```

Exercises:

1

01

010

1010

10101

A

AB

ABC

ABCD

ABCDE

```
for i in range(65,70):  
    for j in range(65,i+1):  
        print chr(j),  
    print
```

1

23

456

78910

1

12

123

1234

12345

12345

1234

123

12

1

**

*

1

123

12345

1234567

123456789

1

121

12321

1234321

123454321

12345

1234

123

12

1

Que-1: WAP for printing the table of 2 to 10.

Que-2: WAP for printing the list of prime nos between 1 to 100.

Que-3: WAP for printing the list of Palindrome numbers between 1 to 1000.