Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)
Python Programming By
Mrs. Geeta
Python Instructor
Zed-King Group of Institutions
Kaithal

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)

Python Programming

Video Tutorial

O-LEVEL (M3-R5.1) Full Video Playlist Available on YouTube Channel **Zed-king Computer Institute Kaithal**

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Free Online Computer Classes On YouTube Channel

https://www.youtube.com/@zedkinginstitute

For Free PDF Notes

Our Telegram Channel

t.me/boost/Zedking_Institute

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)
CHAPTER-4

Operators,Expressions and Python Statement

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Definition Of Operators

Operators are used to perform operations on variables and values.

Expression is the combination of operator and operand.

c=a+b

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Types of operator

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. Identity operators
6. Membership operators
7. Bitwise operator

Arithmetic operator

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operator	Name	About Operator	Example	Result
+	Addition	Add value of its operands	7+7 8+9	14 17
-	Subtraction	Subtract the value of right operand from left operand	6-5 6-7	1 -1
*	Multiplication	Multiplies value of its operands	5*6	30
**	Exponential	Return base (left operand) raised to power (right operand) exponent.	5**3 (it behave like 5^3)	125
/	Division	Divide the value of left operand with right operand and store quotient in float.	5/2	2.5
//	Floor Division	Same as division, but store quotient after removing the fractional part.	5//2	2
%	Modulus (Mod)	Same as division, but store remainder.	5%2	1

Relational /comparison Operators

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operator	Name	About Operator	Example	Result
==	Equal to	returns True if two operands are equal	A=5, B=7 A == B	False
!=	Not equal to	returns True if two operands are not equal	A=5, B=7 A != B	True
>	Greater than	returns True if left operand is greater than right operand	A=5, B=7 A > B	False
<	Less than	returns True if left operand is smaller than right operand	A=5, B=7 A < B	True
>=	Greater than or equal to	returns True if left operand is greater than or equal to right operand	A=7, B=7 A >= B	True
<=	Less than or equal to	returns True if left operand is smaller than or equal to right operand	A=7, B=7 A >= B	True

Logical Operators:

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operator	About Operator
or	if any of the expression is True then the condition becomes True.
and	if both the operands are True, then the condition becomes True.
not	reverse the state of expression

Expression 1	Expression 2	and operator	or operator
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Bitwise Operators:

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operator	Name	About Operator
&	AND	Compare two bits and generate result 1 if both bits are 1, else return 0.
|	OR	Compare two bits and generate result 1 if one of the bit is 1 else return 0.
^	XOR	Compare two bits and generate result 0 if both the bits are identical (same) else return 1.
~	Complement	It generate result opposite of the bits i.e. if bit is 0 then it generates 1 and vice versa.
>>	Right Shift	It shifts the bits of the number to the right and fills 0 on blank left as a result. It gives result very similar as to divide the number with some power of two
<<	Left Shift	It shifts the bits of the number to the left and fills 0 on blank left as a result. It gives result very similar as of multiplying the number with some power of two.

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Membership Operators:

- Membership operators are used to check if a value exists in a sequence or not.
- They return the result as **True** or **False**.
- There are 2 membership operators in python :
  - **in :** returns True if a values is found in a sequence.
  - **not in :** returns True if a value is not found in a sequence.

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Identity Operators:

- Identity operators are used to check whether two values are on the same part of the memory or not.
- They return the result as **True** or **False**.
- There are 2 membership operators in python :
  - **is :** returns True if two operands are identical.
  - **is not:** returns True if two operands are not identical.

Assignment operators

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operator	Example	Equals To
=	a = 10	a = 10
+=	a += 10	a = a+10
-=	a -= 10	a = a-10
*=	a *= 10	a = a*10
/=	a /= 10	a = a / 10
%=	a %= 10	a = a % 10
//=	a //= 10	a = a // 10
**=	a **= 10	a = a ** 10
&=	a &= 10	a = a & 10
|=	a |= 10	a = a |10
^=	a ^= 10	a = a ^10
>>=	a >>= 10	a = a >> 10
<<=	a <<= 10	a = a << 10

Operator Precedence

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Operators	Associativity
() Highest precedence	Left - Right
**	Right - Left
+x , -x, ~x	Left - Right
*, /, //, %	Left - Right
+, -	Left - Right
<<, >>	Left - Right
&	Left - Right
^	Left - Right
|	Left - Right
Is, is not, in, not in, <, <=, >, >=, ==, !=	Left - Right
Not x	Left - Right
And	Left - Right
Or	Left - Right
If else	Left - Right
Lambda	Left - Right
=, +=, -=, *=, /= Lowest Precedence	Right - Left

Introduction-

Control statements define the direction or flow according to which a program should be executed. In control statements, unless the given expression is true, it does not execute its statement.

There are following 2 types of control statements-

1. Conditional branching or statement
2. Looping or iteration

Conditional Statements/Selection Statement:-

Control statement means that it does not execute its statement until any condition given in the program is true.

There are following types of control statements in Python programming language.

1. if statement
2. if_else statement
3. if_elif_else statement
4. nested if

if Statement:-

In an if statement, when the expression is true then the statement is executed and if the expression is false then it is not executed.

Syntax

```
if expression :
    if_statement(s)
```

or

```
if condition:
    statement(s)
```

Example

```
age =  int(input("enter age ———— "))

if age >= 18:

      print ("you are above 18 ")
```

output 1 :-

```
enter age ———— 20          # condition is true

you are above 18
```

if-else statement:-

When any condition in the if-else statement is true then the statement of the if block will be executed. And when any condition is false then the statement of else block will be executed.

Syntax:-

```
if condition :
        statement 1 (s)
else :
        statement 1 (s)
```

Example:-

```
email = "xyz@gmail.com"

passwd = "123450022"

em = input ("enter email——— ")

pa = input ("enter your passwd —- ")

if email==em and passwd ==pa :

        print (" log in. ")

else :

        print ("wrong email id ..")
```

 # output 1 :–

```
enter email——— xyz@gmail.com

enter your passwd —- 12345

log in.
```

if-elif-else statements:-

In the if_elif_else statement, if the condition of the if block is true then the statement of the if block will be printed. If the condition of the block is false then elif block will be checked and if elif is true then the statement of elif block will be printed. If both if and elif are false then the statement of else block will be printed.

Syntax:-

```
if condition :
        statement (s)
elif condition :
        statement (s)
else :
    statement (s)
```

example:-

```
a = 2

b = 2

 if(a < b) :

        print("a is less than b")

elif(a > b) :

         print("a is greater than b")

else :

          print("a is equal to b")
```

Output

```
a is equal to b
```

What is loop:-

In python, a loop is a set of blocks that repeat and execute a given statement repeatedly. A loop executes any statement until the condition given in the loop becomes false.

Loop has two parts – first loop body and second control statement.

Types of Loop:-

There are three types of loops in Python.

1. while loop
2. for loop
3. nested

Note:- do while loop is not there in Python

while loop:-

While Loop simply repeatedly executes the statements until the given condition is False. The loop will execute as long as the condition is true.

While Loop is also called Entry Control Loop because before executing the loop, the given condition is checked, entry into the loop occurs only when the condition is true.

syntax :-

```
while condition / expression :
        statements

or

while (condition / expression) :

        statements
```

Example:-

```
i= 1

while i<=10:

    print ("hello world")

    i=i+1
```

For Loop:-

The syntax of for loop in Python is quite different from other programming languages. The for loop is used to iterate a sequence (such as lists, tuples, dictionaries and sets etc.). To iterate these sequences we use membership operator in.

Syntax:-

```
for < Variable_Name > in < Sequence_Name >:
        statements
```

Important Point:-

1. while is used to conditional repeated a block of code
2. for iterates n times, where n is number of elements in a sequence.

Example:-

```
a = [5, 6, 9, 8, 5, 7]
for num in a :
    print(num)
```

Output

```
5
6
9
8
5
7
```

```
str = "Hello World"
for a in str:
print(a)
```

Output

```
H
e
l
l
o
W
o
r
l
d
```

THE range() FUNCTION:-

The range() function in Python accepts an integer and returns a range object, which is nothing but a sequence of integers.

Syntax: range(start, stop, step)

Rules:- 1. All arguments must be integer, i.e. cannot pass string, float, or any other type into the start, stop, and step arguments of a range() .

2. All three arguments can be positive or negative.
3. step value should not be zero. If a step is zero then gives ValueError exception in python.

Important point:-

1. range is a class
2. range is immutable sequence
3. range can contain only **int** type value
4. range contains sequence of integers with common deference (Arithmetic progression).

Jump Statement:-

Such statements which transfer program control from one place to another within the program without condition are called jump statements which are as follows.

1. Break statement
2. Continue statement
3. Pass Statement

Break statement:-

break is a keyword in Python that is used to bring program control out of a loop. The break statement breaks the loops one by one, i.e. in case of nested loop, it first breaks the inner loop and then moves on to the outer loop.

In Python, break is used to terminate the current for Loop and while Loop. By using means break keyword we can exit the loop on a particular condition.

Example

```
n=1
while n < 10 :
    print(n)
     if(n==5) :
          break
     n=n+1
```

Output

1
2
3
4
5

```
for n in range(1, 10) :
      print(n)
       if(n==5) :
            break
```

Output

1
2
3
4
5

```
for n in 'python' :
   if(n=='o') :
         break
   print(n)
```

Output

p
y
t
h

Continue Statement:-

In Python, continue is used to skip the current loop iteration according to a given condition. And after skipping, the loop starts from next iteration. Simply we can say that we use continue in Python when we have to skip loop execution on some condition.

Example

```
for n in range(1, 8) :
        if(n == 3) :
            continue
        print(n)
```

Output

```
1
2
4
5
6
7
```

```
n = 0
while n < 8 :
        n = n+1
        if(n==3) :
            continue
         print(n)
```

Output

```
1
2
4
5
6
7
7
```

Pass Statement:-

In Python, the Pass keyword is used to do nothing; This means, when we do not want to execute the code, Pass can be used to execute empty. It is exactly what the name refers to. This simply makes the control pass without executing any code.

Example:-

```
name=input("Enter your name: ")
if(name=='atul'):
    print('Welcome',name)
elif(name=='vimal'):
elif(name=='akhilesh'):
print("Access Denied")
```

Note: यह कोड नहीं चलेगा क्योंकि एक खाली ब्लॉक पायथन में illegal है

```
name=input("Enter your name: ")
if(name=='atul'):
    print('Welcome',name)
elif(name=='vimal'):
    pass
elif(name=='akhilesh'):
    print("Access Denied")
```

assert statement:-

'Assert' keyword is used for debugging in Python program. With 'assert' keyword, when the condition is true then nothing happens and when the condition is false then 'AssertionError' occurs.

assert 4==4 #No Exception

assert 4==5 #AssertionError Exception Occurred

```
num=int(input('enter a number '))

assert num>=0

print('You entered: ',num)
```

Nested For Loops:-

Using another For Loop inside a For Loop is called Nested For Loop. Nested For Loop is used like this in Python.

Example:

```
for i in range(1, 4):
    for j in range(i):
        print(i, end='')
    print()
```

Output

```
1
2 2
3 3 3
```