Advanced Programming-Python (Module-1)

Presented by

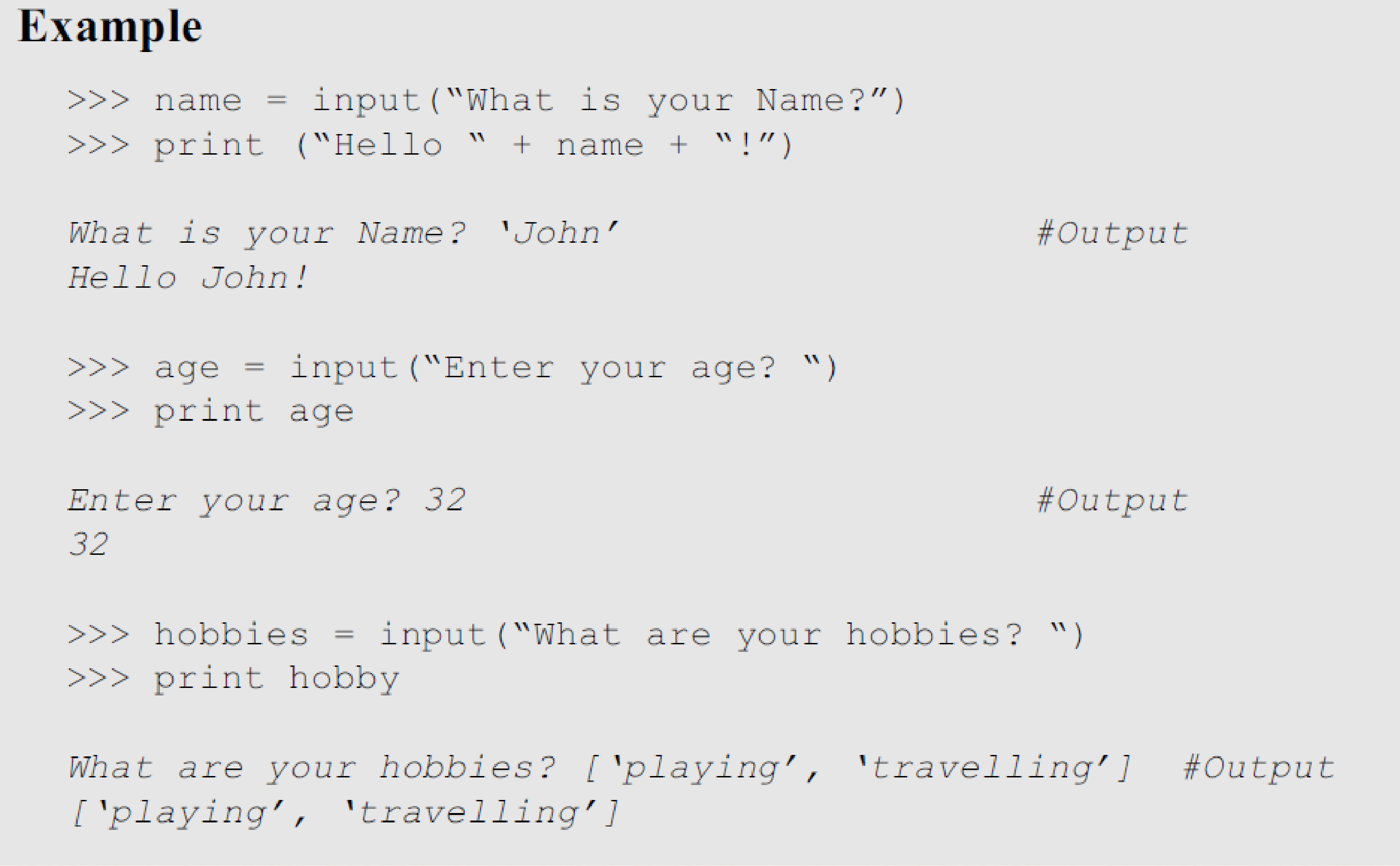
Siddhanta Borah

Contents of Todays (Day-3 & 4) Discussion

- Input from keyboard- input() function
- Different operator in python programming
  - Arithmetic operator
  - Comparison operator
  - Assignment operator
  - Logical operator
  - Bitwise operator
  - Membership operator
  - Identity operator
  - Unary Operator
  - Binary Operator
- Precedence of operator
- Associativity

INPUT( ) FUNCTION

- *input()* function has an optional parameter, which is the prompt string.
- When the *input()* function is called, in order to take input from the user then the execution of program halts and waits for the user to provide an input. The input is given by the user through keyboard and it is ended by the return key.

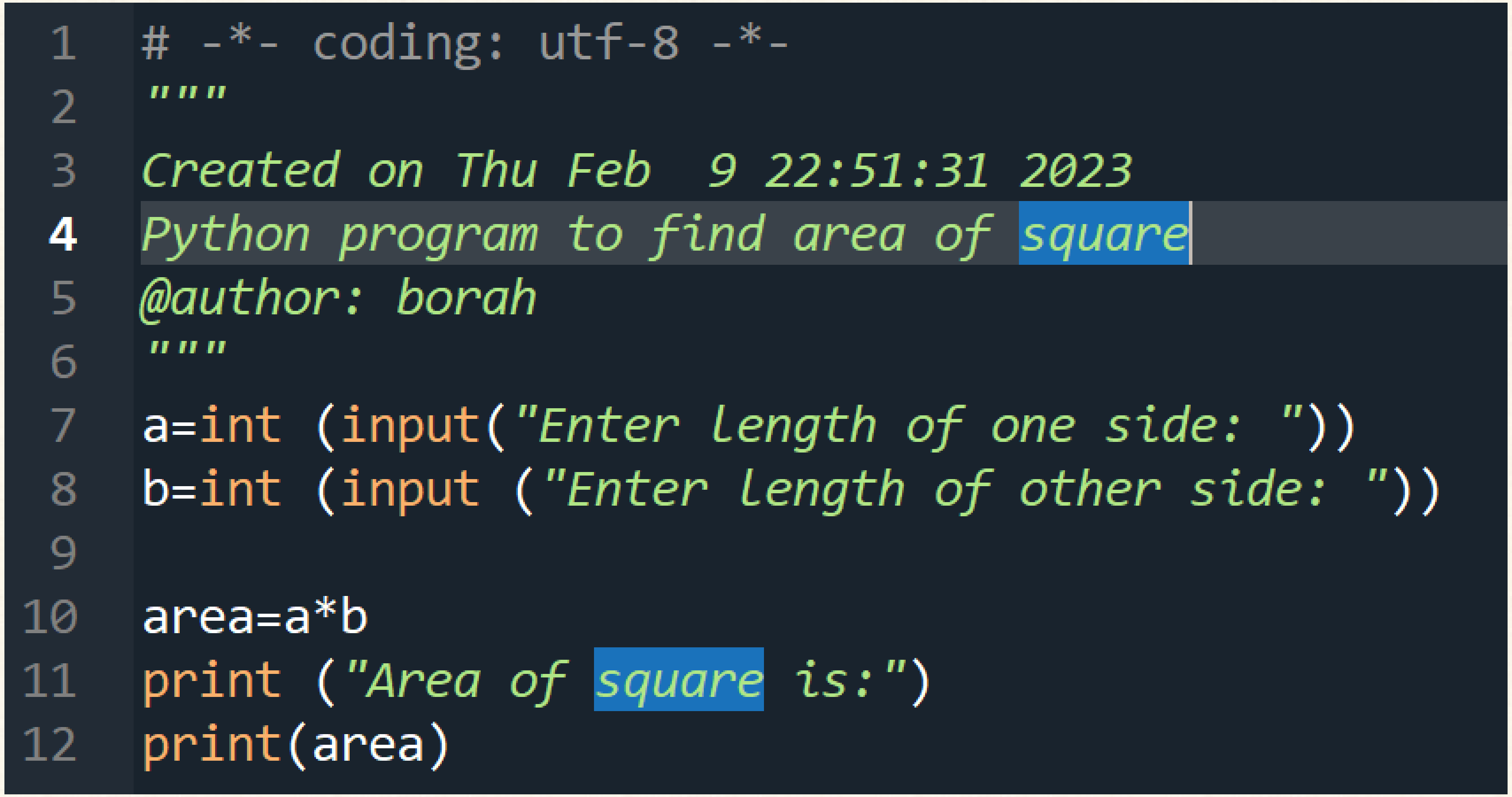


Example

```
>>> name = input(“What is your Name?”)
>>> print (“Hello “ + name + “!”)

What is your Name? ‘John’                 #Output
Hello John!

>>> age = input(“Enter your age? “)
>>> print age

Enter your age? 32                        #Output
32

>>> hobbies = input(“What are your hobbies? “)
>>> print hobby

What are your hobbies? [‘playing’, ‘travelling’]  #Output
[‘playing’, ‘travelling’]
```

INPUT() FUNCTION (CONTD..)

- *Write a program in python to find the area of a rectangle.*
- *Write a program in python to find the area of a right angle triangle.*

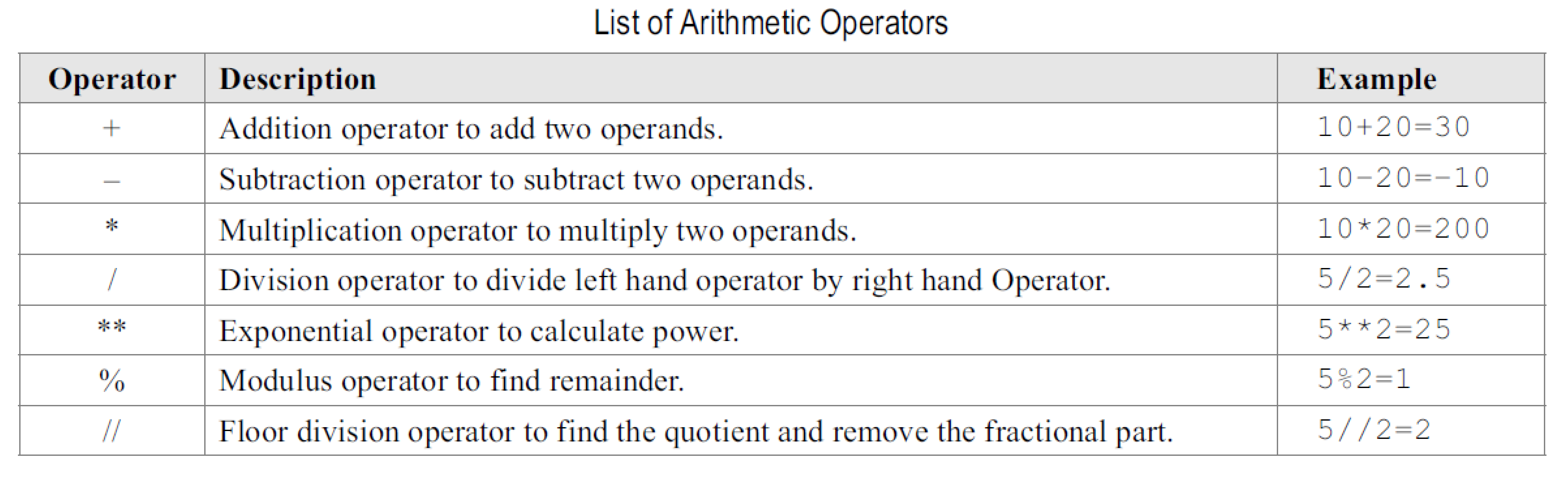


```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb  9 22:51:31 2023
Python program to find area of square
@author: borah
"""
a=int (input("Enter length of one side: "))
b=int (input ("Enter length of other side: "))

area=a*b
print ("Area of square is:")
print(area)
```

OPERATOR IN PYTHON PROGRAMMING

1. Arithmetic operator:

These operators are used to perform arithmetic operations such as addition, subtraction, multiplication and division

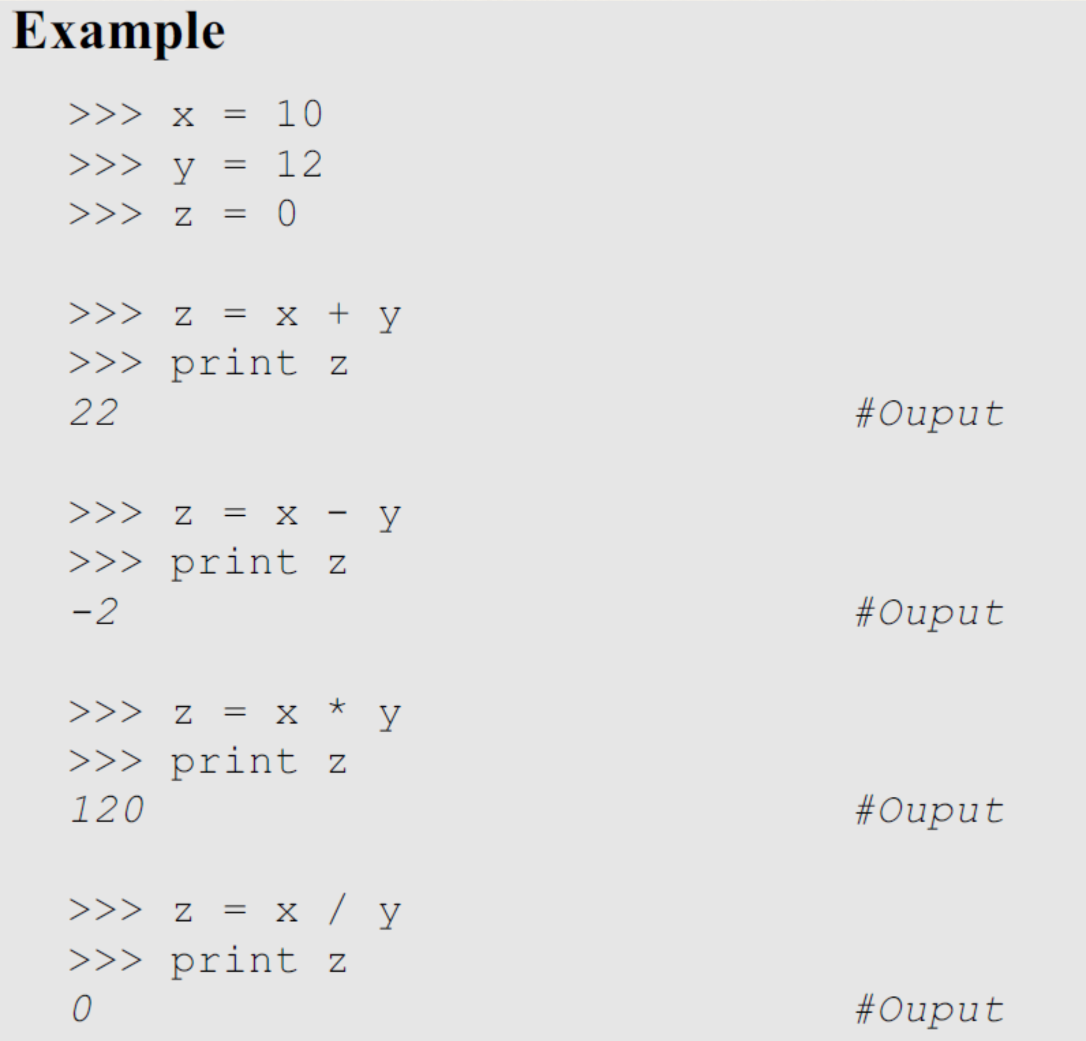


List of Arithmetic Operators

Operator	Description	Example
+	Addition operator to add two operands.	10+20=30
–	Subtraction operator to subtract two operands.	10-20=-10
*	Multiplication operator to multiply two operands.	10*20=200
/	Division operator to divide left hand operator by right hand Operator.	5/2=2.5
**	Exponential operator to calculate power.	5**2=25
%	Modulus operator to find remainder.	5%2=1
//	Floor division operator to find the quotient and remove the fractional part.	5//2=2

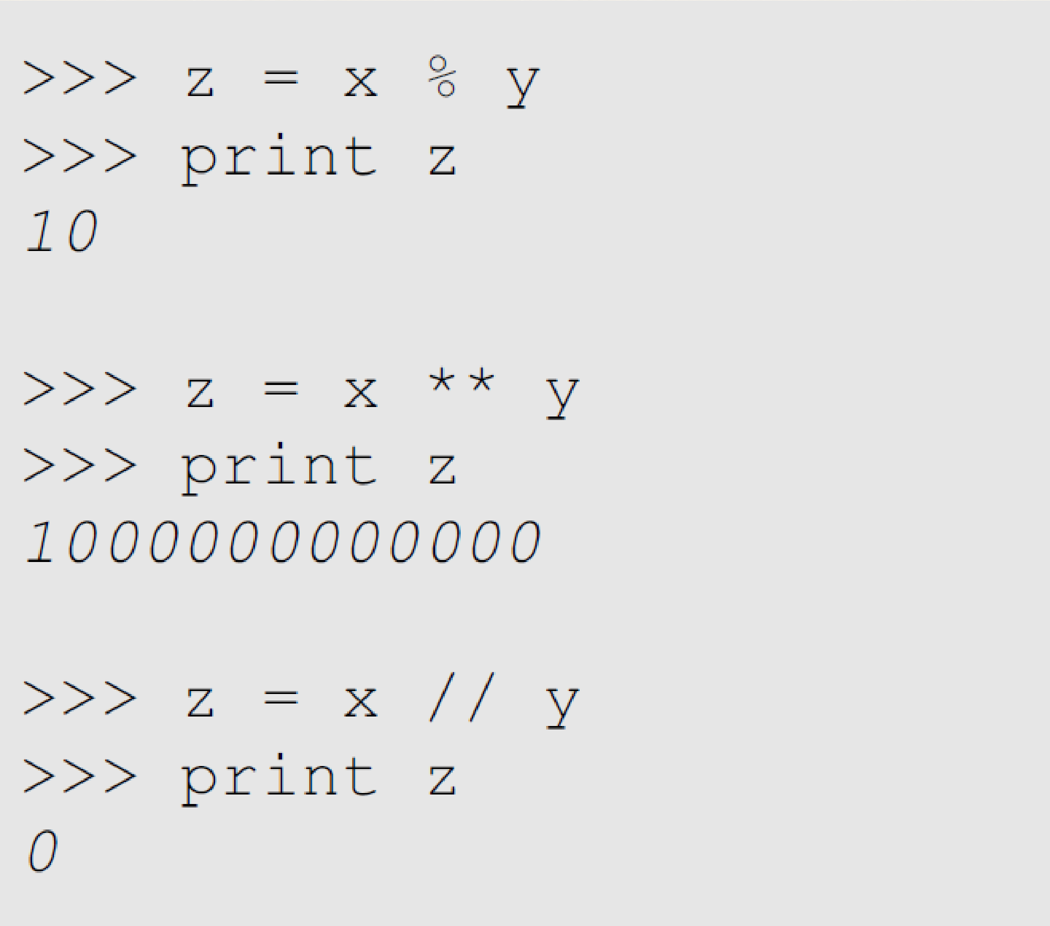
OPERATOR IN PYTHON PROGRAMMING

Arithmetic operator:



Example

```
>>> x = 10
>>> y = 12
>>> z = 0

>>> z = x + y
>>> print z
22                                    #Ouput

>>> z = x - y
>>> print z
-2                                    #Ouput

>>> z = x * y
>>> print z
120                                   #Ouput

>>> z = x / y
>>> print z
0                                     #Ouput
```

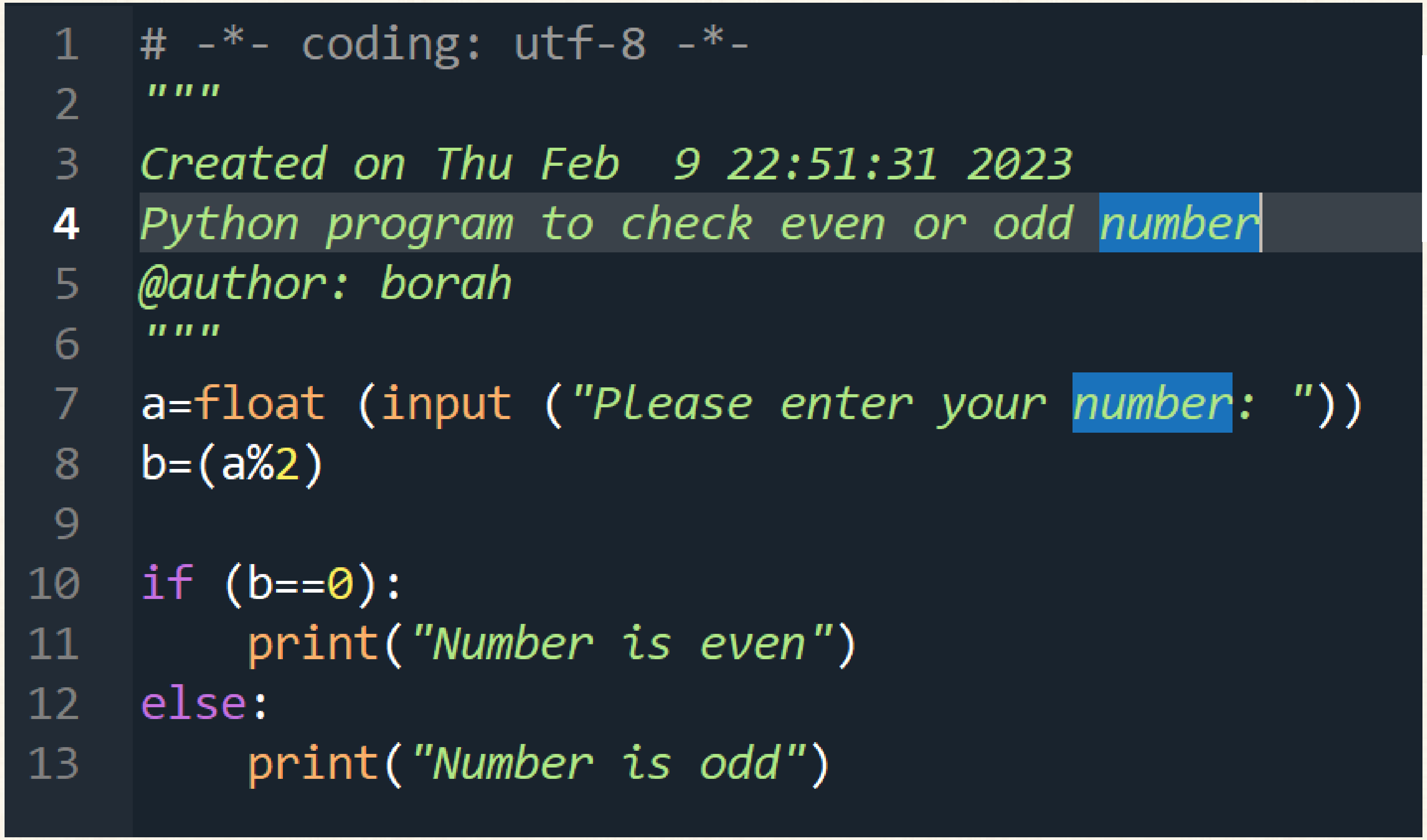


```
>>> z = x % y
>>> print z
10

>>> z = x ** y
>>> print z
1000000000000

>>> z = x // y
>>> print z
0
```

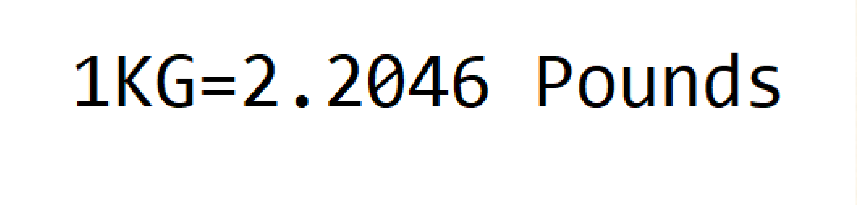
OPERATOR IN PYTHON PROGRAMMING

Arithmetic operator:

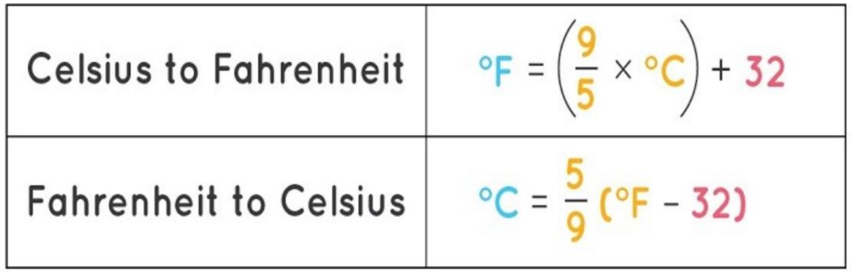
- *Write a program in python to convert kilogram into pound.*
- *Write a program in python to convert degree Celsius into Fahrenheit.*
- ***Write a program in python to check whether a number is even or odd.***



```python
# -*- coding: utf-8 -*-
"""

Created on Thu Feb  9 22:51:31 2023
Python program to check even or odd number
@author: borah
"""
a=float (input ("Please enter your number: "))
b=(a%2)

if (b==0):
    print("Number is even")
else:
    print("Number is odd")
```



1KG=2.2046 Pounds

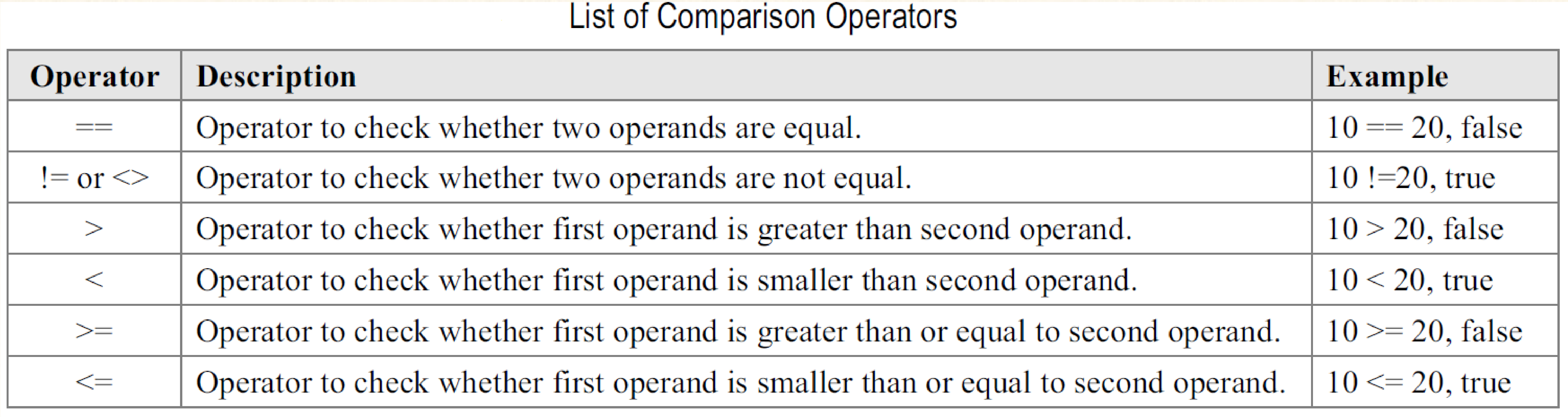


Celsius to Fahrenheit	$°F = \left(\frac{9}{5} \times °C\right) + 32$
Fahrenheit to Celsius	$°C = \frac{5}{9}(°F - 32)$

OPERATOR IN PYTHON PROGRAMMING

2. Comparison operator:

Comparison operators are also called relational operators. The result of these operators is always a Boolean value, that is, either true or false. Following table provides a list of comparison operators.

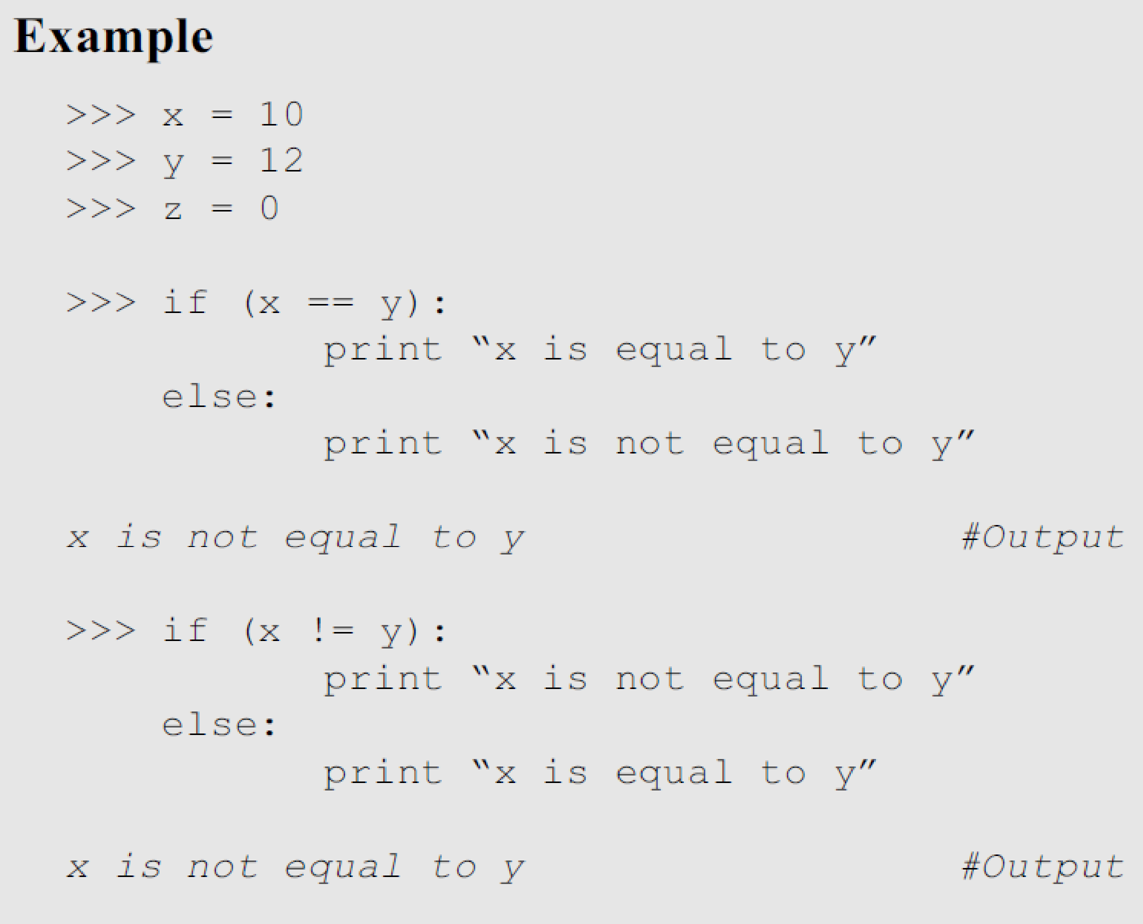


List of Comparison Operators

Operator	Description	Example
==	Operator to check whether two operands are equal.	10 == 20, false
!= or <>	Operator to check whether two operands are not equal.	10 !=20, true
>	Operator to check whether first operand is greater than second operand.	10 > 20, false
<	Operator to check whether first operand is smaller than second operand.	10 < 20, true
>=	Operator to check whether first operand is greater than or equal to second operand.	10 >= 20, false
<=	Operator to check whether first operand is smaller than or equal to second operand.	10 <= 20, true

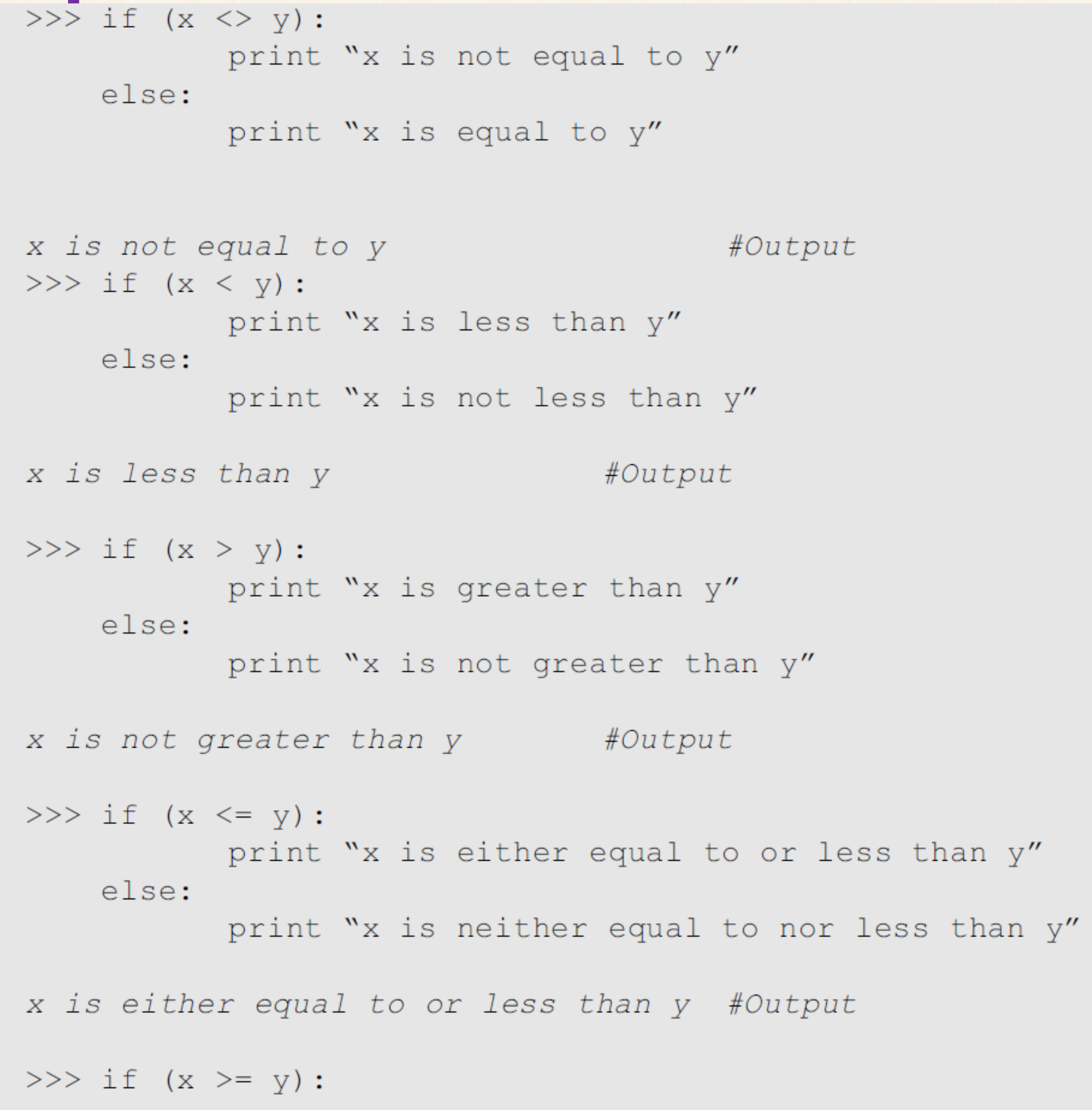
OPERATOR IN PYTHON PROGRAMMING

Comparison operator:



Example

```
>>> x = 10
>>> y = 12
>>> z = 0

>>> if (x == y):
            print “x is equal to y”
    else:
            print “x is not equal to y”

x is not equal to y                    #Output

>>> if (x != y):
            print “x is not equal to y”
    else:
            print “x is equal to y”

x is not equal to y                    #Output
```

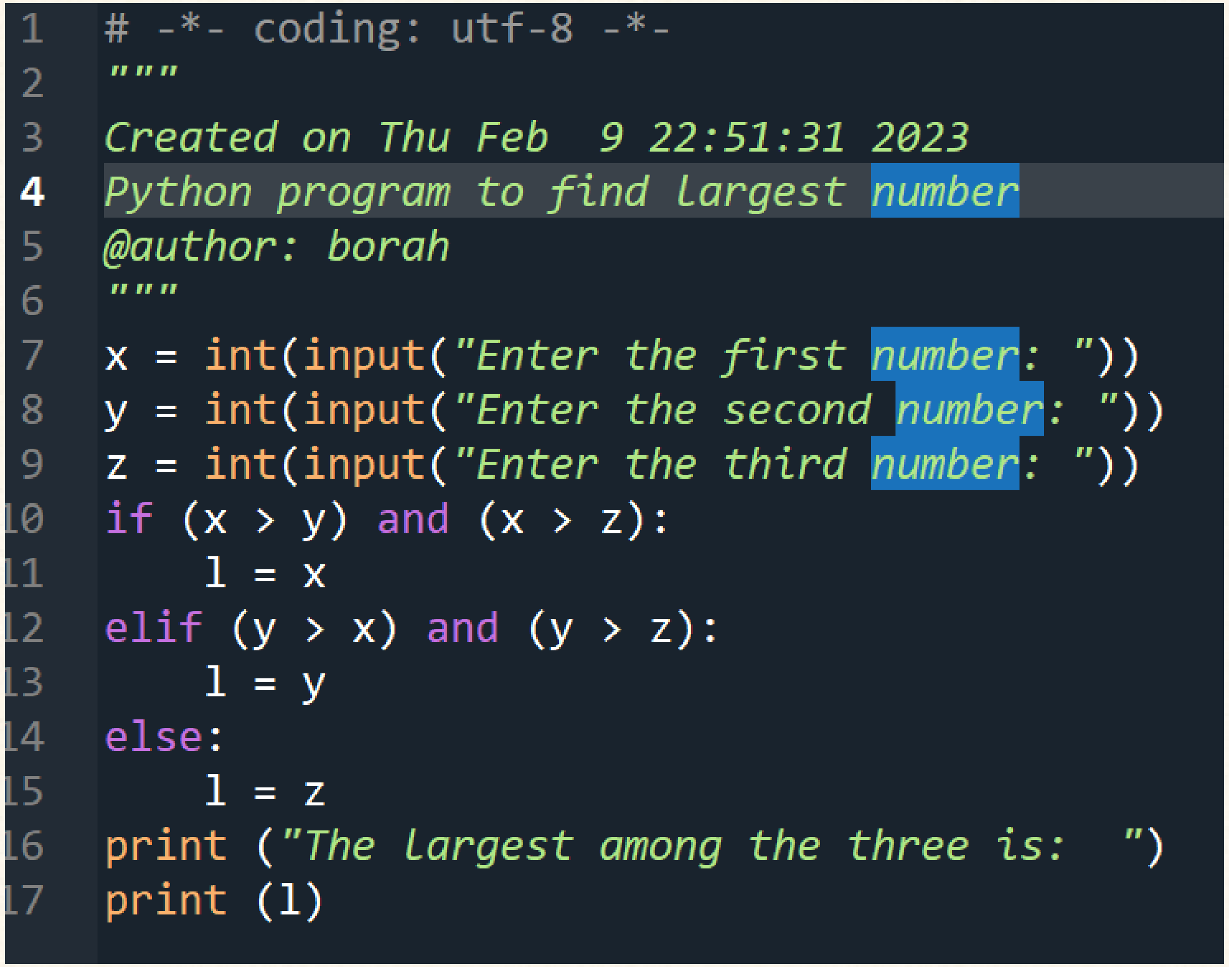


```
>>> if (x <> y):
            print “x is not equal to y”
    else:
            print “x is equal to y”


x is not equal to y                    #Output
>>> if (x < y):
            print “x is less than y”
    else:
            print “x is not less than y”

x is less than y                  #Output

>>> if (x > y):
            print “x is greater than y”
    else:
            print “x is not greater than y”

x is not greater than y         #Output

>>> if (x <= y):
            print “x is either equal to or less than y”
    else:
            print “x is neither equal to nor less than y”

x is either equal to or less than y  #Output

>>> if (x >= y):
```

OPERATOR IN PYTHON PROGRAMMING

Comparison operator:

- *Write a program in python to check if the input year is leap year or not.*
- ***Write a program in python to check largest among the three given numbers.***
- *Write a program in python to find the smallest number in a List.*

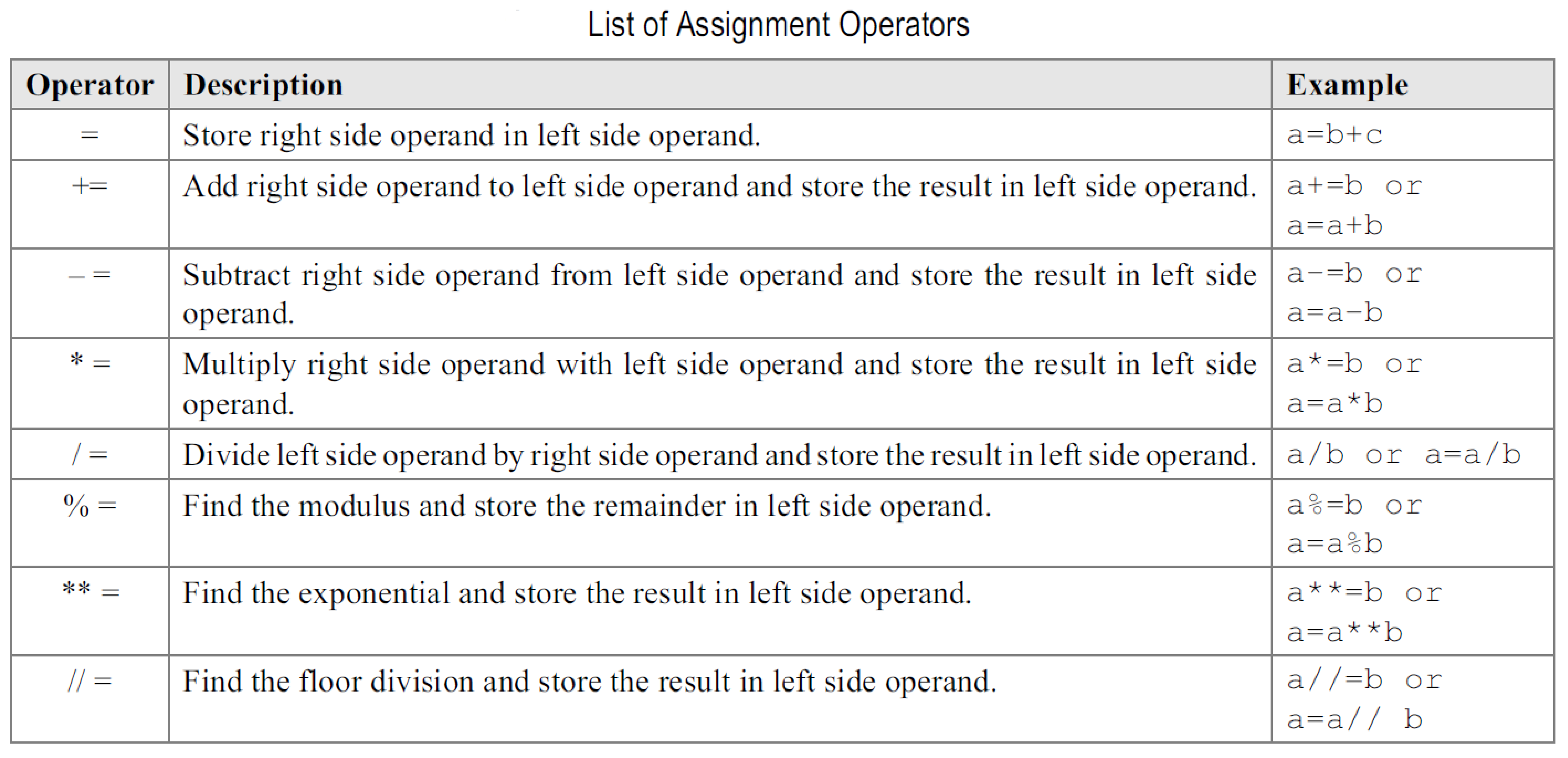


```python
# -*- coding: utf-8 -*-
"""
Created on Thu Feb  9 22:51:31 2023
Python program to find largest number
@author: borah
"""
x = int(input("Enter the first number: "))
y = int(input("Enter the second number: "))
z = int(input("Enter the third number: "))
if (x > y) and (x > z):
    l = x
elif (y > x) and (y > z):
    l = y
else:
    l = z
print ("The largest among the three is:  ")
print (l)
```

OPERATOR IN PYTHON PROGRAMMING

3. Assignment operator:

Following table provides a list of Assignment operators.



List of Assignment Operators

Operator	Description	Example
=	Store right side operand in left side operand.	`a=b+c`

+=	Add right side operand to left side operand and store the result in left side operand.	`a+=b or`
		`a=a+b`

– =	Subtract right side operand from left side operand and store the result in left side operand.	`a-=b or`
		`a=a-b`

* =	Multiply right side operand with left side operand and store the result in left side operand.	`a*=b or`
		`a=a*b`

/ =	Divide left side operand by right side operand and store the result in left side operand.	`a/b or a=a/b`

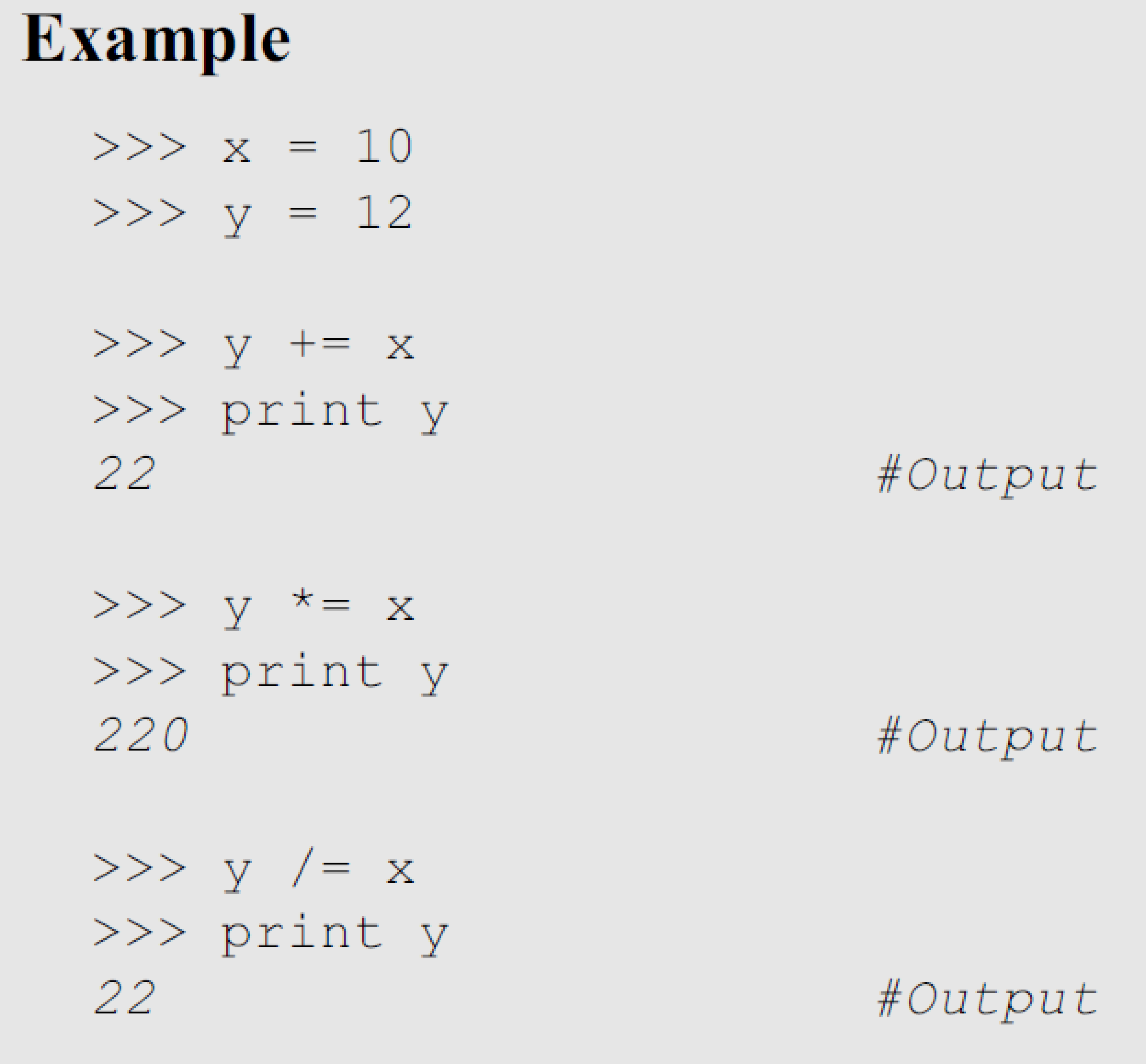
% =	Find the modulus and store the remainder in left side operand.	`a%=b or`
		`a=a%b`

** =	Find the exponential and store the result in left side operand.	`a**=b or`
		`a=a**b`

// =	Find the floor division and store the result in left side operand.	`a//=b or`
		`a=a// b`

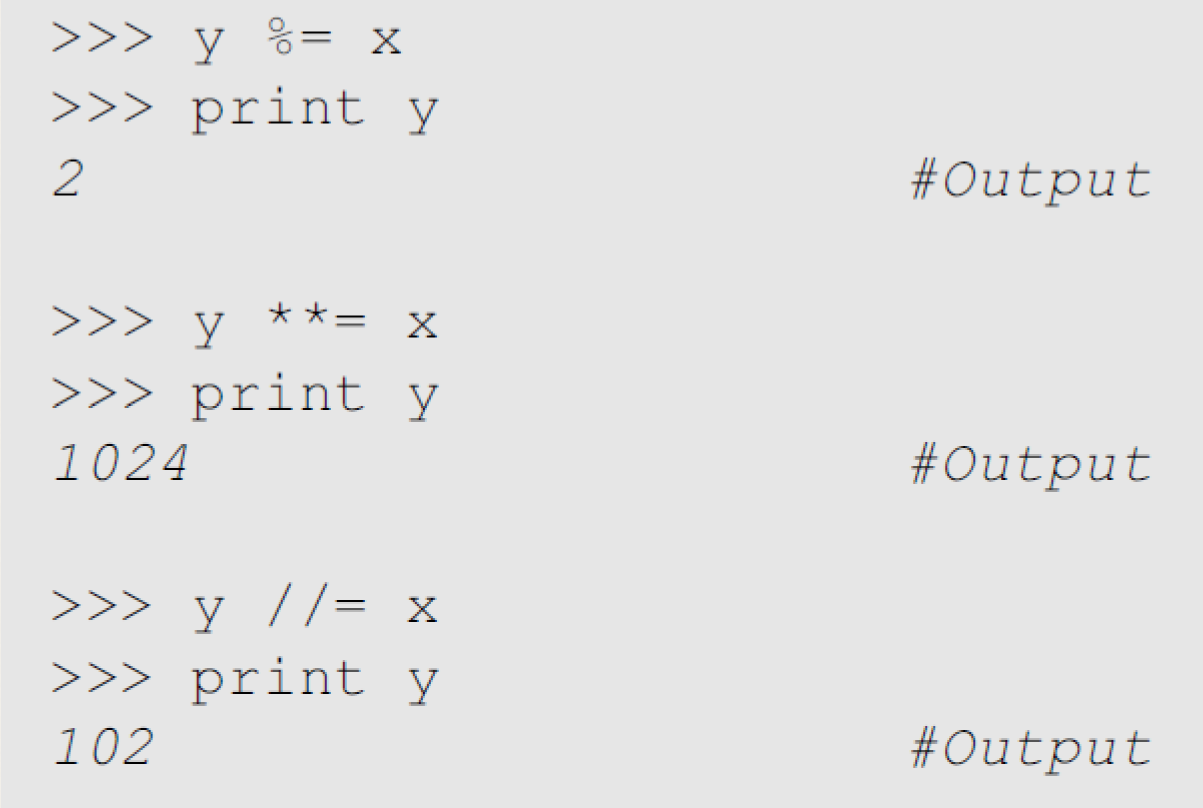
OPERATOR IN PYTHON PROGRAMMING

Assignment operator:



Example

```
>>> x = 10
>>> y = 12

>>> y += x
>>> print y
22                         #Output

>>> y *= x
>>> print y
220                        #Output

>>> y /= x
>>> print y
22                         #Output
```



```
>>> y %= x
>>> print y
2                          #Output

>>> y **= x
>>> print y
1024                       #Output

>>> y //= x
>>> print y
102                        #Output
```

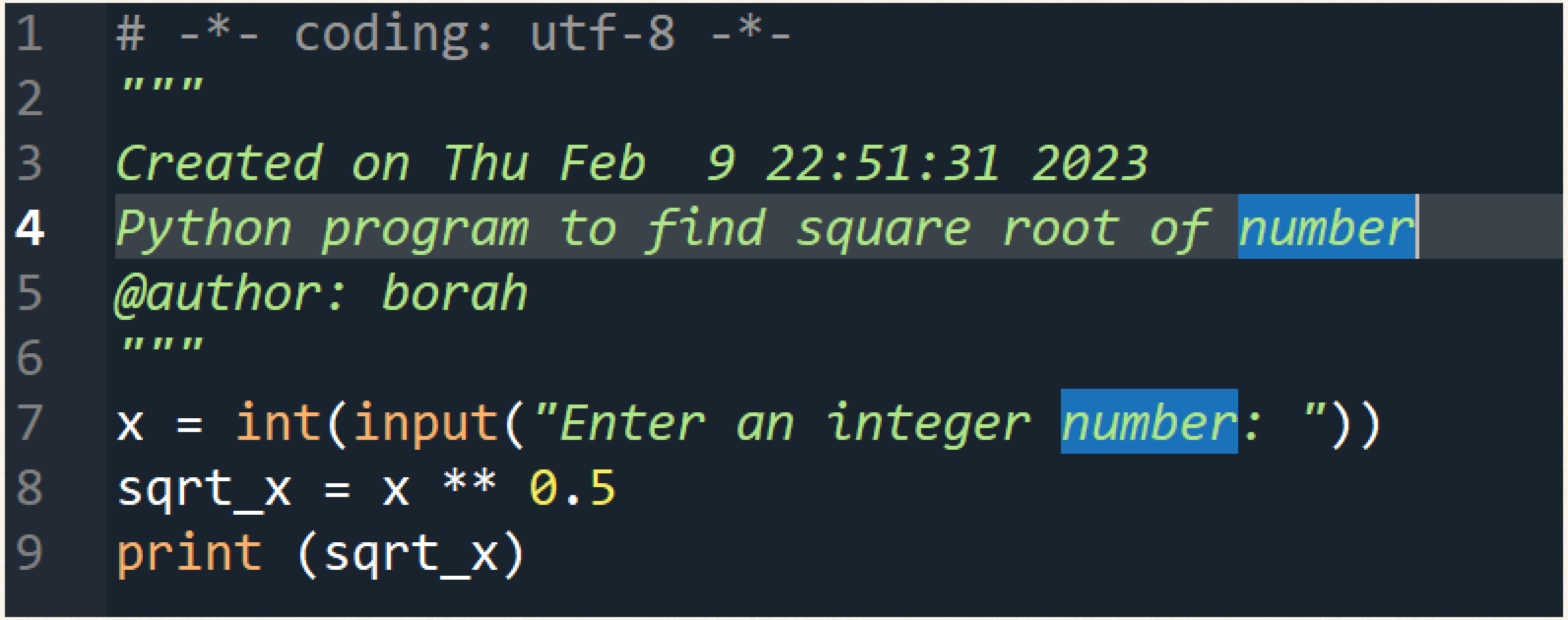
OPERATOR IN PYTHON PROGRAMMING

Assignment operator:

- *Write a program in python to find the value of Y, where x is any integer.*

$$Y = 3^{X+2} + 3^{X}$$

- ***Write a program in python to find the square root of a number.***

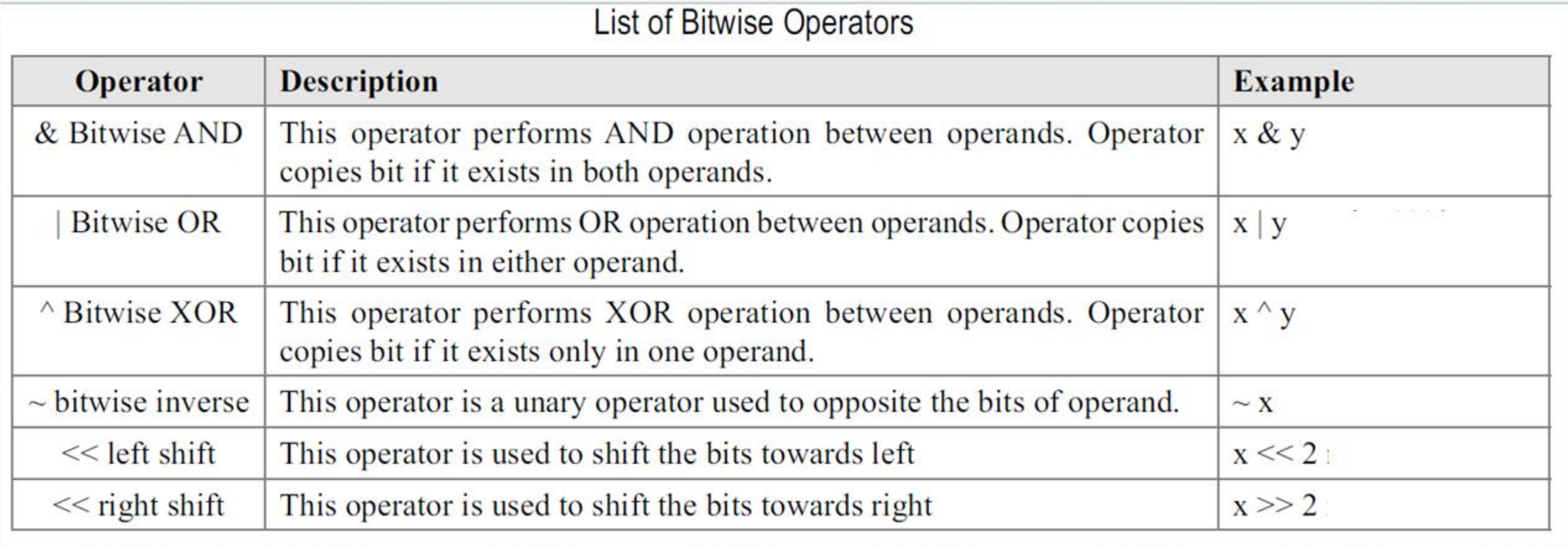


```python
# -*- coding: utf-8 -*-
"""
Created on Thu Feb  9 22:51:31 2023
Python program to find square root of number
@author: borah
"""
x = int(input("Enter an integer number: "))
sqrt_x = x ** 0.5
print (sqrt_x)
```

OPERATOR IN PYTHON PROGRAMMING

4. Bitwise operator:

Following table provides a list of Bitwise operators. Results are obtained for X=1010 and Y= 1000

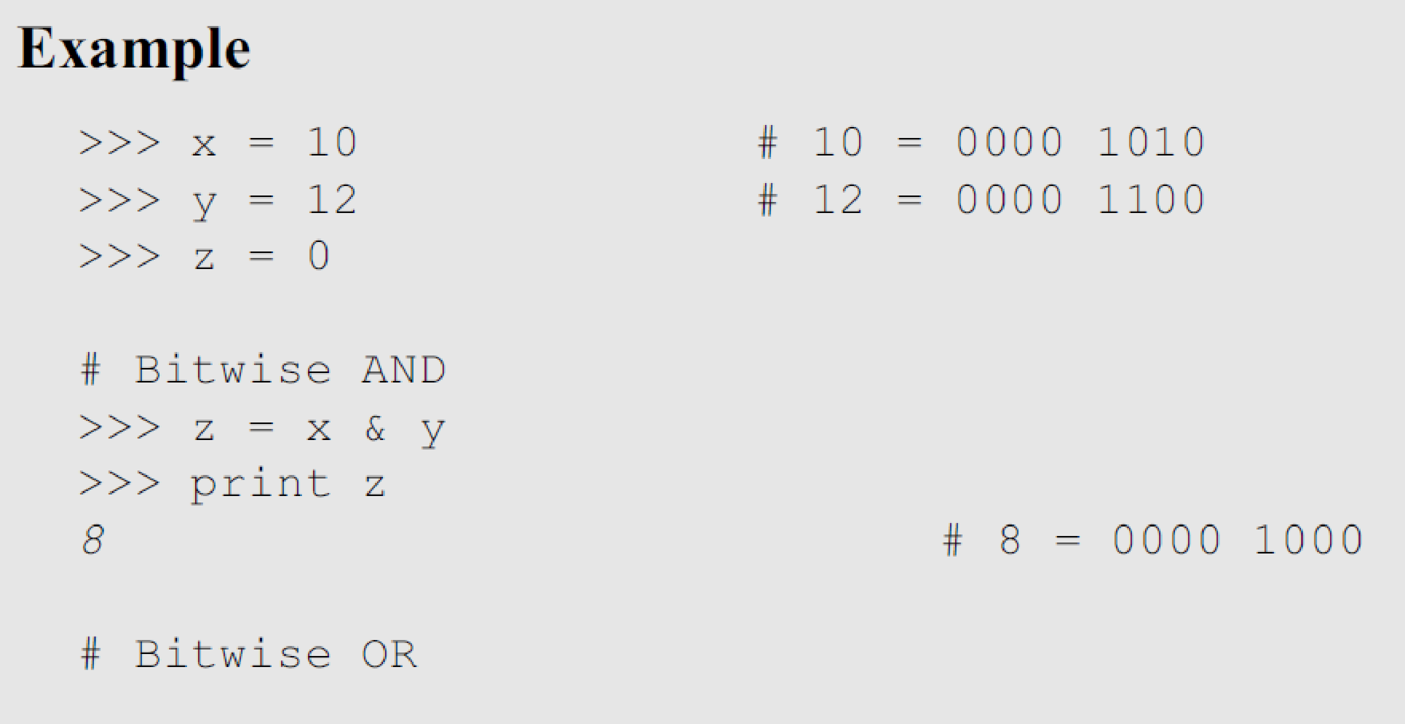


List of Bitwise Operators

Operator	Description	Example
& Bitwise AND	This operator performs AND operation between operands. Operator copies bit if it exists in both operands.	x & y
| Bitwise OR	This operator performs OR operation between operands. Operator copies bit if it exists in either operand.	x | y
^ Bitwise XOR	This operator performs XOR operation between operands. Operator copies bit if it exists only in one operand.	x ^ y
~ bitwise inverse	This operator is a unary operator used to opposite the bits of operand.	~ x
<< left shift	This operator is used to shift the bits towards left	x << 2
<< right shift	This operator is used to shift the bits towards right	x >> 2

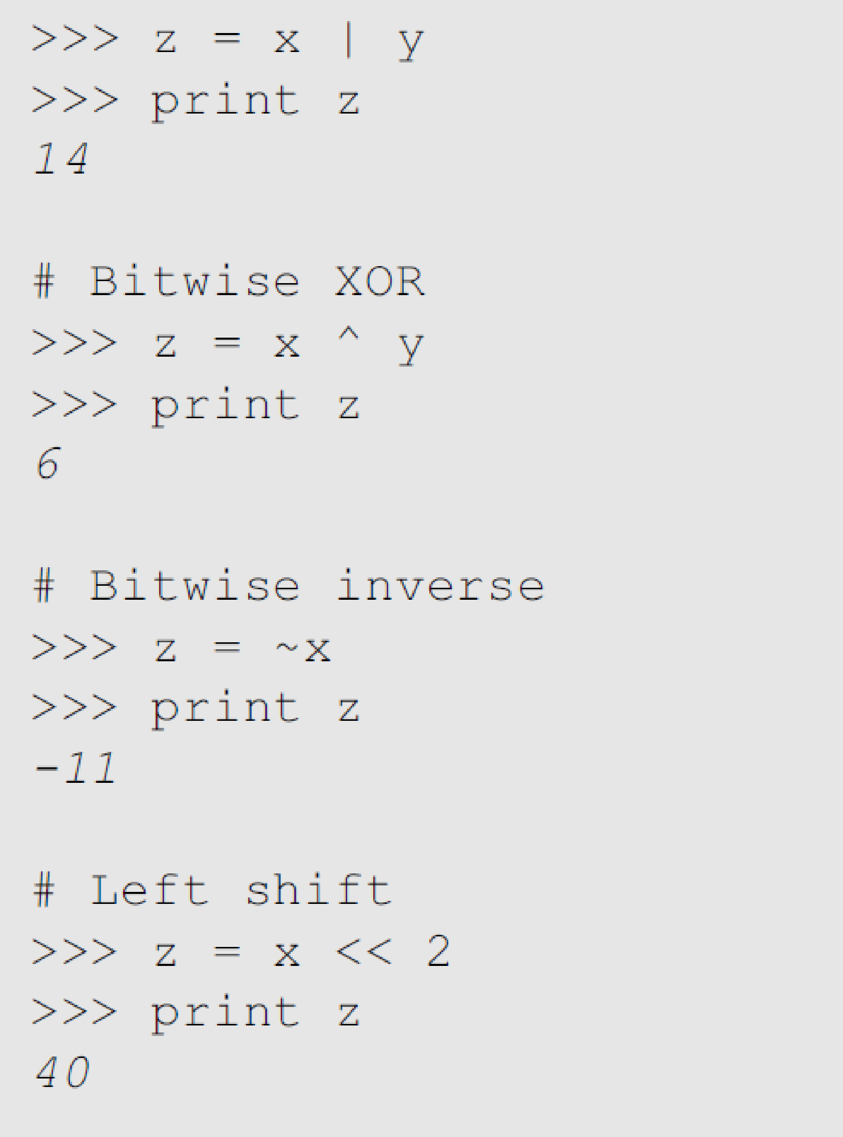
OPERATOR IN PYTHON PROGRAMMING

Bitwise operator:



Example

```
>>> x = 10                    # 10 = 0000 1010
>>> y = 12                    # 12 = 0000 1100
>>> z = 0

# Bitwise AND
>>> z = x & y
>>> print z
8                                    # 8 = 0000 1000

# Bitwise OR
```



```
>>> z = x | y
>>> print z
14

# Bitwise XOR
>>> z = x ^ y
>>> print z
6

# Bitwise inverse
>>> z = ~x
>>> print z
-11

# Left shift
>>> z = x << 2
>>> print z
40
```

OPERATOR IN PYTHON PROGRAMMING

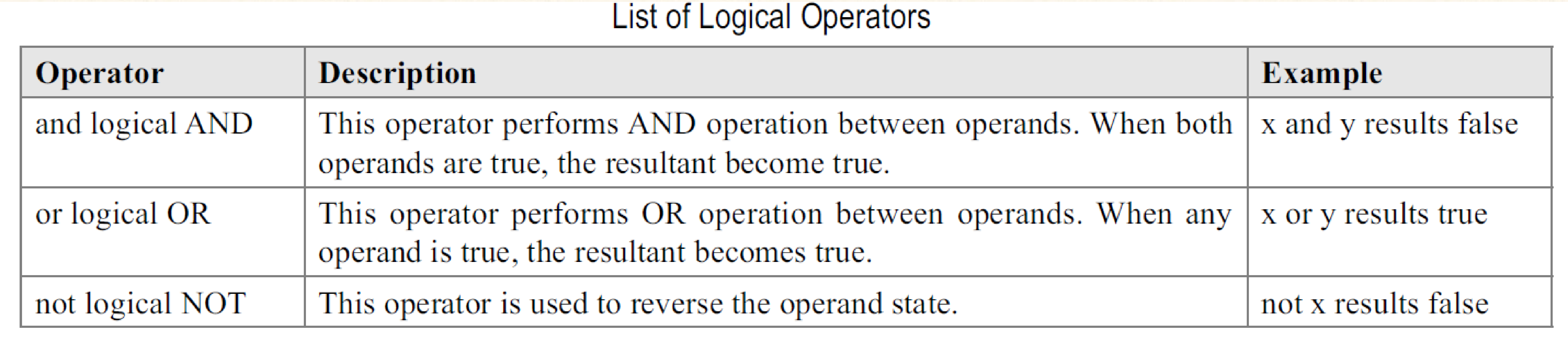
Bitwise operator:

- *Write a program in python to interchange the position of two digit number .*

OPERATOR IN PYTHON PROGRAMMING

5. Logical operator:

Following table provides a list of Logical operators.

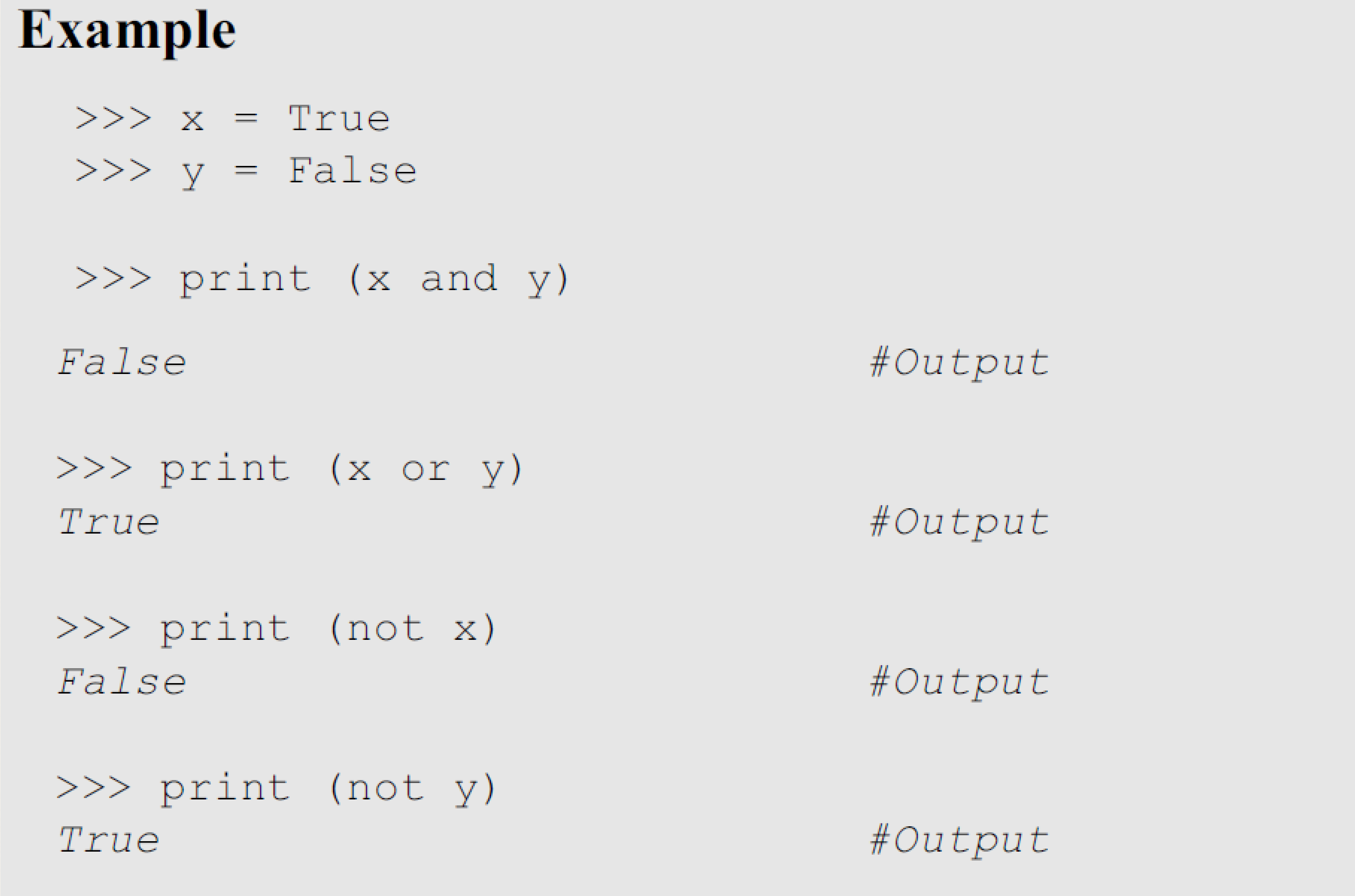


List of Logical Operators

Operator	Description	Example
and logical AND	This operator performs AND operation between operands. When both operands are true, the resultant become true.	x and y results false
or logical OR	This operator performs OR operation between operands. When any operand is true, the resultant becomes true.	x or y results true
not logical NOT	This operator is used to reverse the operand state.	not x results false

OPERATOR IN PYTHON PROGRAMMING

Logical operator:



Example

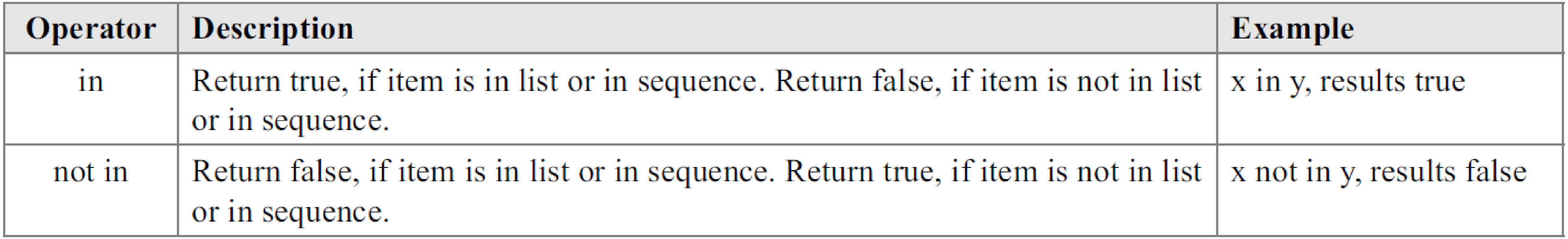
```
 >>> x = True
 >>> y = False

 >>> print (x and y)

False                              #Output

>>> print (x or y)
True                               #Output

>>> print (not x)
False                              #Output

>>> print (not y)
True                               #Output
```

OPERATOR IN PYTHON PROGRAMMING

6. Membership operator:

These operators are used to check an item or an element that is part of a string, a list or a tuple. A membership operator reduces the effort of searching an element in the list. Suppose, x stores a value 20 and y is the list containing items 10, 20, 30, and 40. Then, x is a part of the list y because the value 20 is in the list y. Following Table gives a list of membership operators.

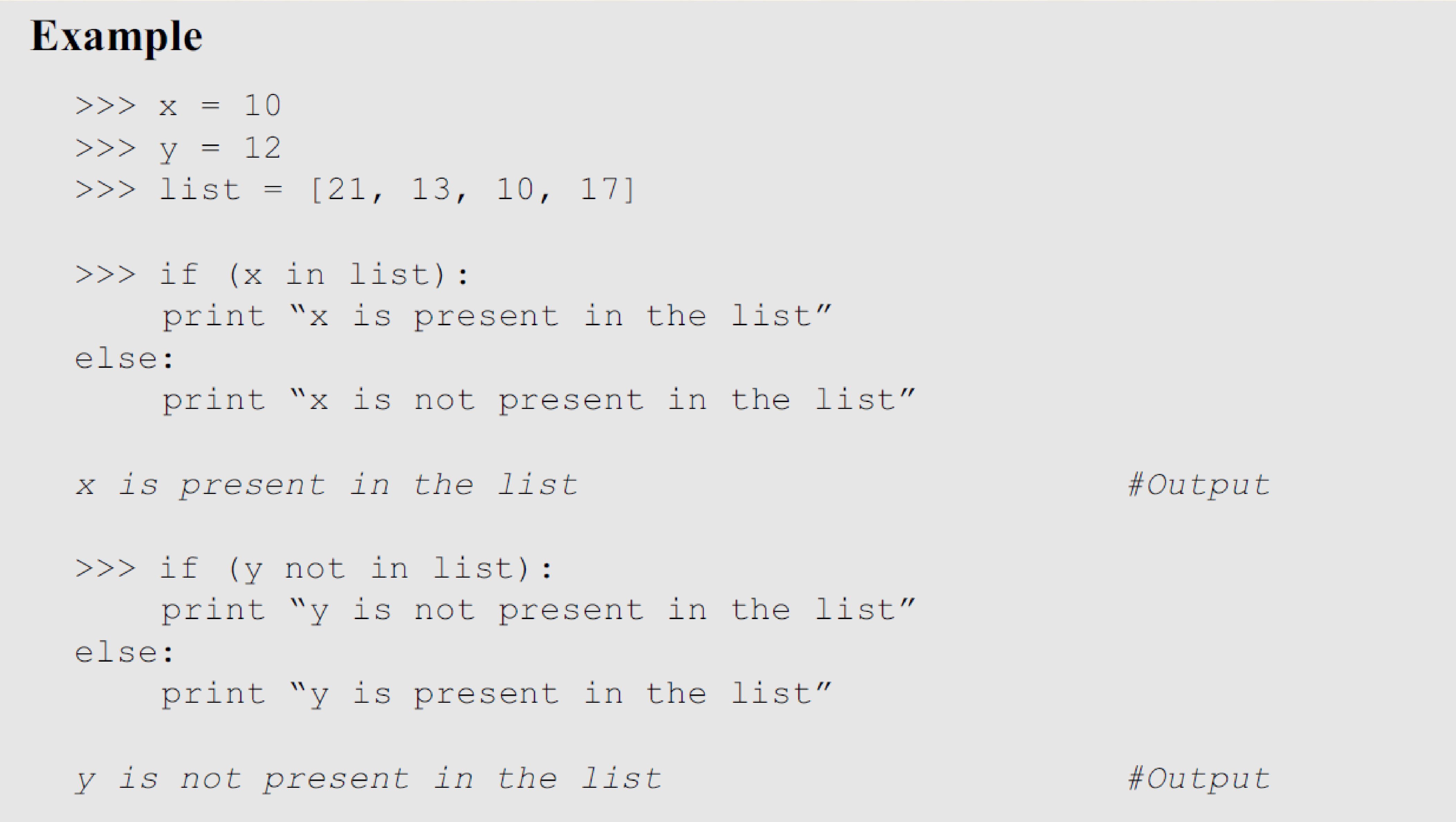
List of Membership Operators



Operator	Description	Example
in	Return true, if item is in list or in sequence. Return false, if item is not in list or in sequence.	x in y, results true
not in	Return false, if item is in list or in sequence. Return true, if item is not in list or in sequence.	x not in y, results false

OPERATOR IN PYTHON PROGRAMMING

Membership operator:



Example

```
>>> x = 10
>>> y = 12
>>> list = [21, 13, 10, 17]

>>> if (x in list):
        print “x is present in the list”
else:
        print “x is not present in the list”

x is present in the list                                    #Output

>>> if (y not in list):
        print “y is not present in the list”
else:
        print “y is present in the list”

y is not present in the list                                #Output
```

OPERATOR IN PYTHON PROGRAMMING

Membership operator:

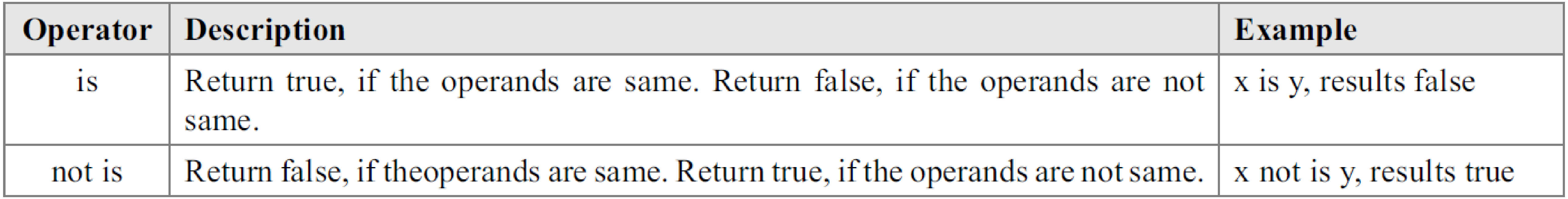
- *Write a program in python to check whether a number present in a list or not.*

OPERATOR IN PYTHON PROGRAMMING

7. Identity operator:

These operators are used to check whether both operands are same or not. Suppose, x stores a value 20 and y stores a value 40. Then *x is y* **returns** *false* **and** *x not is y* **returns** *true***. Following Table provides a list of identity operators**

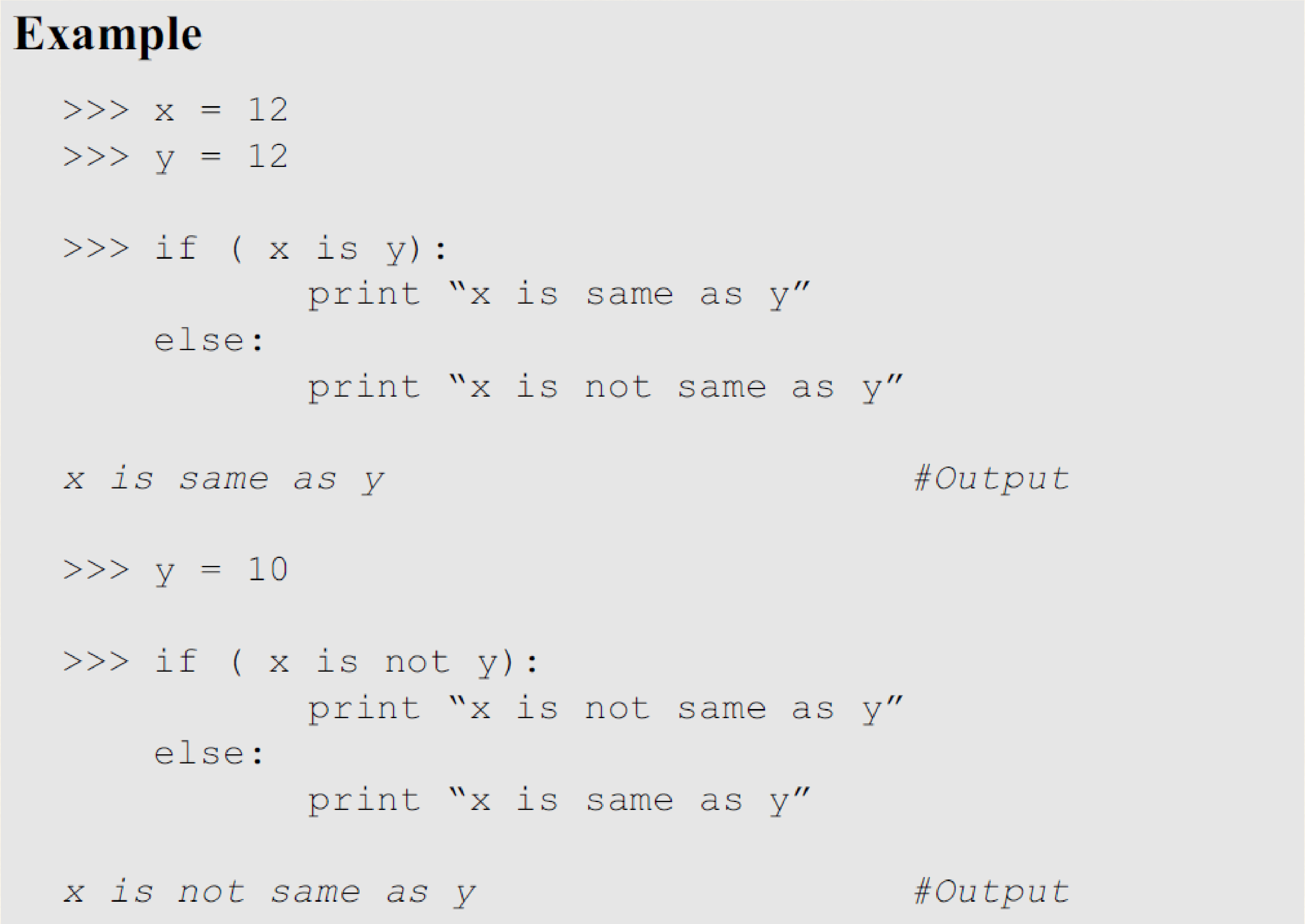
List of Identity Operators



Operator	Description	Example
is	Return true, if the operands are same. Return false, if the operands are not same.	x is y, results false
not is	Return false, if theoperands are same. Return true, if the operands are not same.	x not is y, results true

OPERATOR IN PYTHON PROGRAMMING

Identity operator:



Example

```
>>> x = 12
>>> y = 12

>>> if ( x is y):
           print “x is same as y”
    else:
           print “x is not same as y”

x is same as y                       #Output

>>> y = 10

>>> if ( x is not y):
           print “x is not same as y”
    else:
           print “x is same as y”

x is not same as y                   #Output
```

OPERATOR IN PYTHON PROGRAMMING

8. Unary operator:

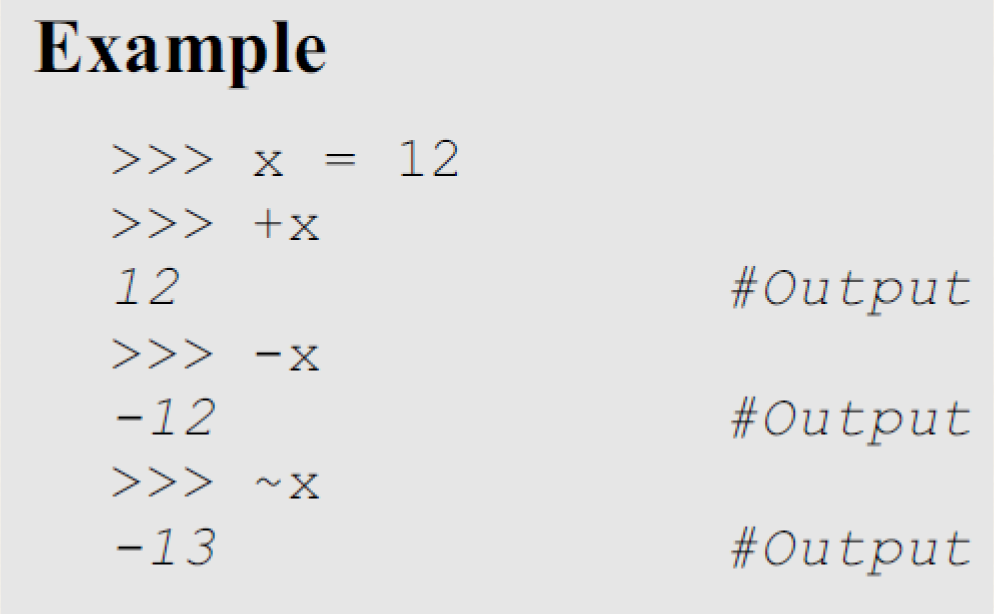
Unary operators are operators with only one operand. These operators are basically used to provide sign to the operand.

The format of the unary operator is:

Operator operand

Some unary operators are as follows:

- **+**
- **-**
- **~**



Example

```
>>> x = 12
>>> +x
12                    #Output
>>> -x
-12                   #Output
>>> ~x
-13                   #Output
```

OPERATOR IN PYTHON PROGRAMMING

9. Binary operator:

- **Binary operators are operators with two operands that are manipulated to get the result. They are also used to compare numeric values and string values.**
- **The format of binary operator is:**

Operand1 **Operator** Operand2

- **Some binary operators are as follows:**

**, *, /, %, +, -, <<,>>, &, | ,^,<,>,<= ,>=, == ,!= ,<>

PRECEDENCE OF OPERATOR

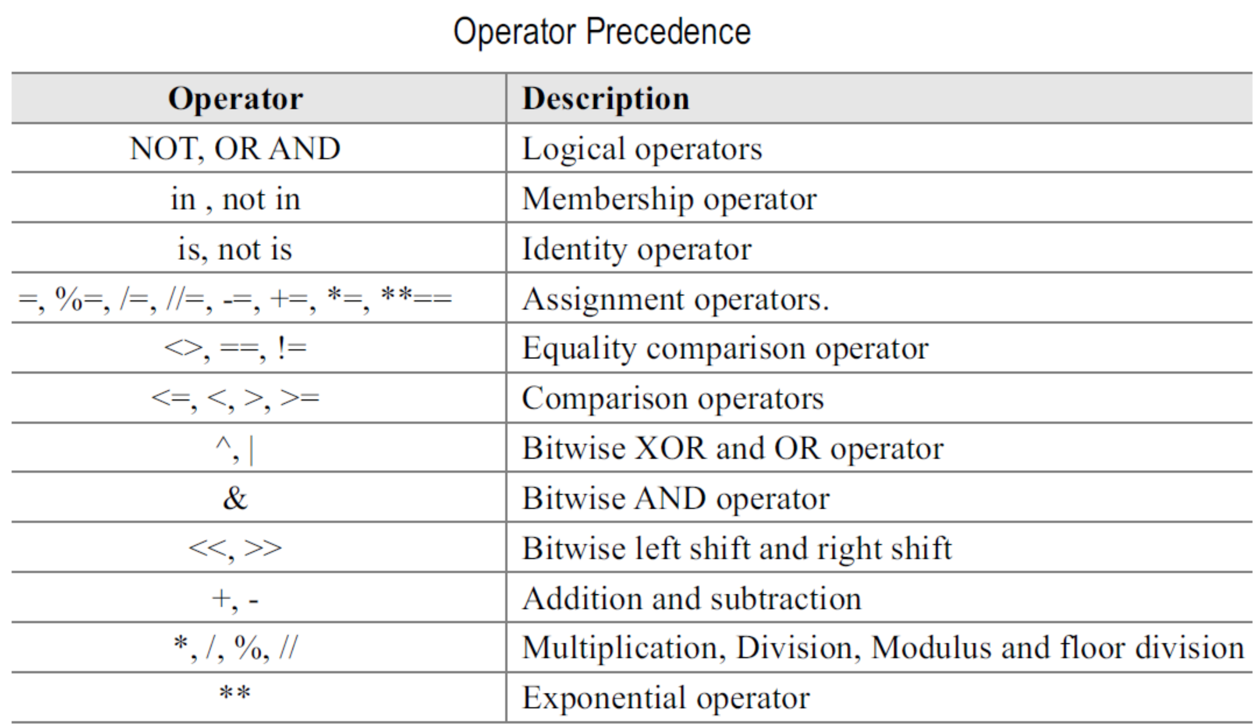
When an expression has two or more operators, we need to identify the correct sequence to evaluate these operators. This is because the final answer changes depending on the sequence thus chosen.

Let us look at an example of a mathematical expression:

10 + 5 / 5

When the given expression is evaluated left to right, the final answer becomes 3. However, if the above expression is evaluated right to left, the final answer becomes 11. This shows that changing the sequence in which the operators are evaluated in the given expression also changes the solution. Therefore, in order to remove this problem, a level of precedence is associated with the operators. Precedence is the condition that specifies the importance of each operator relative to the others.

Table in the right hand side to display operator precedence from lower precedence to higher:



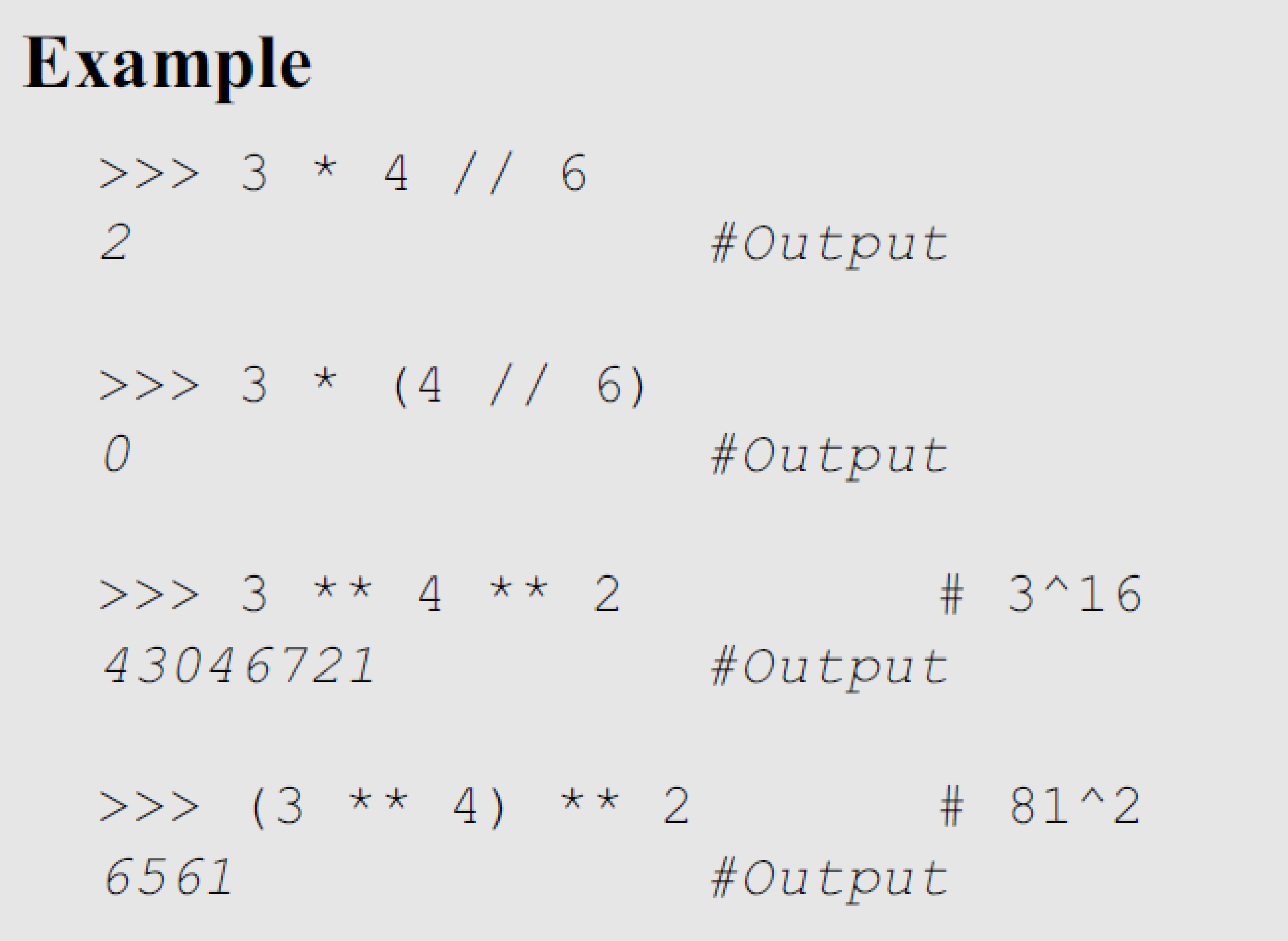
Operator Precedence

Operator	Description
NOT, OR AND	Logical operators
in , not in	Membership operator
is, not is	Identity operator
=, %=, /=, //=, -=, +=, *=, **==	Assignment operators.
<>, ==, !=	Equality comparison operator
<=, <, >, >=	Comparison operators
^, |	Bitwise XOR and OR operator
&	Bitwise AND operator
<<, >>	Bitwise left shift and right shift
+, -	Addition and subtraction
*, /, %, //	Multiplication, Division, Modulus and floor division
**	Exponential operator

ASSOCIATIVITY

- Associativity decides the order in which the operators with same precedence are executed.
- There are two types of associativity. One is left-to-right and other is right-to-left. In left-to-right associativity, the operator of same precedence are executed from the left side first and in right-to-left associativity, the operator of same precedence are executed from the right side first. Most of the operators in Python have left-to-right associativity. Examples for left-to-right associative operators are multiplication, floor division, etc and ** operator is right-to-left associative.

ASSOCIATIVITY



Example

```
>>> 3 * 4 // 6
2                 #Output

>>> 3 * (4 // 6)
0                 #Output

>>> 3 ** 4 ** 2          # 3^16
43046721          #Output

>>> (3 ** 4) ** 2        # 81^2
6561              #Output
```

PYTHON PROGRAM

- **Write a Python program to solve a quadratic equation.**
- **Write a Python program to display calendar.**
- **Write a Python Program to check if a Number is Positive, Negative or Zero.**
- **Write a Python Program to find the Factorial of a Number.**

Thank You