21 May

Python Basic - 2

* 1. Create two int type variables, apply addition, subtraction, division and multiplications and store the results in variables. The:n print the data in the following format by calling the variables:

First variable is & second variable is . Addition: + =

Subtraction: - = Multiplication: \* = Division: / =

num1= 20

num2= 30

print(f"""First variable is {num1} & Second variable is {num2}

Addition:{num1}+{num2}={num1+num2}

Subtraction:{num1}-{num2}={num1-num2}

Multiplication:{num1}\*{num2}={num1\*num2}

Division:{num1}/{num2}={num1/num2}

""")

* 1. What is the difference between the following operators:

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

And: (i) The first one is Float Division("/"): it divides left hand operand by right hand operand and give the result with decimal places.

the second is Floor Division ("//"): it divides left hand operand by right hand operand and it removes the decimal places and returns the

* 1. List the logical operators.

And, Or, Not

* 1. Explain right shift operator and left shift operator with examples.

right shift: Shifts the bits of the number to the right and fills 0 on voids left( fills 1 in the case of a negative number) as a result. Similar effect as of dividing the number with some power of two.

Example 1:

a = 10 = 0000 1010 (Binary)

a >> 1 = 0000 0101 = 5

Example 2:

a = -10 = 1111 0110 (Binary)

a >> 1 = 1111 1011 = -5

left shift: Shifts the bits of the number to the left and fills 0 on voids right as a result. Similar effect as of multiplying the number with some power of two.

Example 1:

a = 5 = 0000 0101 (Binary)

a << 1 = 0000 1010 = 10

a << 2 = 0001 0100 = 20

Example 2:

b = -10 = 1111 0110 (Binary)

b << 1 = 1110 1100 = -20

b << 2 = 1101 1000 = -40

* 1. Create a list containing int type data of length 15. Then write a code to check if 10 is present in the list or not.

list = [1,2,3,4,5,6,5,7,8,11, 20,3,45,5]

if 10 in list:

print(f'Number 10 is present at index {i} in list')

else:

print('number 10 is not present in the list')