21 May

Python Basic - 2

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

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

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

**Answer –**

**Input**

# Create two int type variables

first\_variable = 10

second\_variable = 5

# Perform addition, subtraction, multiplication, and division

addition = first\_variable + second\_variable

subtraction = first\_variable - second\_variable

multiplication = first\_variable \* second\_variable

division = first\_variable / second\_variable

# Print the results in the desired format

print(f"First variable is {first\_variable}\t& second variable is {second\_variable}")

print(f"Addition:\t+\t=\t{addition}")

print(f"Subtraction:\t-\t=\t{subtraction}")

print(f"Multiplication:\t\*\t=\t{multiplication}")

print(f"Division:\t/\t=\t{division}")

**Output**

First variable is 10 & second variable is 5

Addition: + = 15

Subtraction: - = 5

Multiplication: \* = 50

Division: / = 2.0

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

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

**Answer –**

(i) The '/' operator performs floating-point division, while the '//' operator performs floor division or integer division.

(ii) The '\*\*' operator is used for exponentiation, whereas the '^' operator is the bitwise XOR operator and not used for exponentiation in Python.

* 1. **List the logical operators.**

**Answer –**

The logical operators in Python are:

1. and: Returns True if both operands are True.
2. or: Returns True if at least one of the operands is True.
3. not: Returns the opposite of the operand's logical value (True becomes False, and False becomes True).
   1. **Explain right shift operator and left shift operator with examples.**

**Answer –**

The right shift operator (>>) and left shift operator (<<) are used to shift the bits of a number to the right or left, respectively.

Right shift (>>) divides the number by 2 for each shift to the right.

Left shift (<<) multiplies the number by 2 for each shift to the left.

Example:

Right shift: 10 >> 2 shifts the bits of 10 two positions to the right, resulting in 2.

Left shift: 10 << 2 shifts the bits of 10 two positions to the left, resulting in 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.**

**Answer –**

**Input**

# Created a list of integers

my\_list = [5, 8, 3, 12, 10, 17, 22, 10, 6, 14, 19, 27, 10, 9, 2]

# Checking if 10 is present in the list

if 10 in my\_list:

print("10 is present in the list.")

else:

print("10 is not present in the list.")

**Output**

10 is present in the list.