

1. Demonstrate the use of all types of operators.

Ans:- There are 7 types of operators in python:-

1. Arithmetic Operators:- Arithmetic operators perform basic mathematical operations.

Code:-

1. Addition

```
a = 10 + 5
```

```
print(a)
```

Output: 15

2. Subtraction

```
b = 20 - 8
```

```
print(b)
```

Output: 12

3. Multiplication

```
c = 6 * 4
```

```
print(c)
```

Output: 24

4. Division

```
d = 21 / 3
```

```
print(d)
```

Output: 7.0

2. Assignment Operators:- Assignment operators are used to assign values to variables.

Code:-

```
x = 10 # Assigning 10 to variable x
```

# Additional assignment operators

```
x += 3 # Equivalent to x = x + 3
```

```
y -= 2 # Equivalent to y = y - 2
```

```
x *= 2 # Equivalent to x = x * 2
```

```
y /= 4 # Equivalent to y = y / 4
```

3. Comparison Operators:- Comparison operators compare values and return a Boolean result (True or False).

Code:-

1. Greater than

```
print(a > b)
```

Output: True

2. Less than

```
print(a < b)
```

Output: False

3. Equal to  
print(a == b)  
Output: False

4. Not equal to  
print(a != b)  
Output: True

5. Greater than or equal to  
print(a >= b)  
Output: True

6. Less than or equal to  
print(a <= b)  
Output: False

4. Logical Operators:-Logical operators perform logical operations on Boolean values.

Code:-  
p = True  
q = False

1.Logical AND  
print(p and q)  
Output: False

2.Logical OR  
print(p or q)  
Output: True

3.Logical NOT  
print(not p)  
Output: False

5. Bitwise Operators:-Bitwise operators perform operations on binary representations of integers.

Code:-  
a = 10  
Binary: 1010  
b = 5  
Binary: 0101

1.Bitwise AND

```
print(a & b)
Output: 0
```

2.Bitwise OR  

```
print(a | b)
```

  
Output: 15

3.Bitwise XOR  

```
print(a ^ b)
```

  
Output: 15

6. Identify Operators:-Identity operators are used to compare the memory locations of two objects.

Code:-

```
x = 10
y = 10
z = [1, 2, 3]
```

7. Membership Operators:-Membership operators are used to test if a value is present in a sequence.

Code:-

```
numbers = [1, 2, 3, 4, 5]
```

1.In operator

```
print(3 in numbers)
```

Output: True

2.Not in operator

```
print(6 not in numbers)
```

Output: True