**OPERATORS**

Operators: In python it provides various operators for performing operation on variables & values.

These are classified into several categories:

**1. Arithmetic Operators**

These operators perform basic on mathematical operations like addition, subtraction, multiplication, etc.

* + (Addition): Adds two operands.
* - (Subtraction): Subtracts the second operand from the first.
* \* (Multiplication): Multiplies two operands.
* / (Division): Divides the first operand by the second.
* // (Floor Division): Divides and returns the largest integer smaller than or equal to the result.
* % (Modulus): Returns the remainder when the first operand is divided by the second.
* \*\* (Exponentiation): Raises the first operand to the power of the second.

Add(+)  
a=8  
b=5  
result=a+b  
print(result)

Output: 13

Sub(-)  
a=8  
b=5  
result=a-b  
print(result)

Output:3

Mul(\*)

a=8  
b=5  
result=a\*b  
print(result)

Output: 40

Div(/)

a=8  
b=5  
result=a/b  
print(result)

Output:1.6

FloorDiv(//)  
a=8  
b=5  
result=a//b  
print(result)

Output: 1

Modulus(%)

a=8  
b=5  
result=a%b  
print(result)

Output:3  
  
Exponentation(\*\*)

a=8  
b=5  
result=a\*\*b  
print(result)

Output: 32768

**2.** **Comparison (Relational) Operators**

These operators compare two values and return a boolean result (True or False) (0 or 1).

* == (Equal to): It returns True if two values are equal.
* != (Not equal to): It returns True if two values are not equal.
* > (Greater than): It returns True if the first value is greater than the second value.
* < (Less than): It returns True if the first value is less than the second value.
* >= (Greater than or equal to): It returns True if the first value is greater than or equal to the second value.
* <= (Less than or equal to): It returns True if the first value is less than or equal to the second value.

a=30  
b=18  
print(a<b)  
print(a>b)  
print(a<=b)  
print(a>=b)  
print(a==b)  
print(a!=b)

Output: False

True

False

True

False

True

**3. Logical Operators**

These operators are used to combine conditional statements.

* and: It returns True if both values are True.
* or: It returns True if at least one value is True.
* not: Reverses the logical state (i.e., returns True if the operand is False, and vice versa).

a=True  
b=False  
result=aandb  
print(result)

result=aorb  
print(result)

result=nota  
print(result)

result=notb  
print(result)

Output: False

True

False

True

**4. Assignment Operators**

These operators are used to assign values to variables.

* =: It Assigns the value of the right value to the left value.
* +=: It adds the right value to the left value and assigns the result to the left value.
* -=: It subtracts the right value from the left value and assigns the result to the left value.
* \*=: It multiplies the left value by the right value and assigns the result to the left value.
* /=: It divides the left value by the right value and assigns the result to the left value.
* //=: In Floor divides the left value by the right value and assigns the result to the left value.
* %=: It takes the modulus of the left operand by the right value and assigns the result to the left value.
* \*\*=: It raises the left value d to the power of the right value and assigns the result to the left value.

total=10  
total+=4  
total-=5  
total\*=6  
total/=7  
print(total)

Output: 7.714285714285714

**5. Bitwise Operators**

Bitwise operators are used to perform operations on binary numbers.

* & (AND): Performs a bitwise AND.
* | (OR): Performs a bitwise OR.
* ^ (XOR): Performs a bitwise XOR (exclusive OR).
* ~ (NOT): Performs a bitwise NOT (inverts the bits).
* << (Left shift): Shifts the bits to the left by the specified number of positions.
* >> (Right shift): Shifts the bits to the right by the specified number of positions.

a=14  
b=6  
result=a&b  
print(result)

result=a^b  
print(result)

result=a/b  
print(result)

result=nota  
print(result)

result=a<<2  
print(result)

result=a>>2  
print(result)

Output: 6

8

2.3333333333333335

False

56

3

**6. Membership Operators**

These operators test whether a value is in a sequence (like a list, tuple, string, etc.).

* **in:** It returns True if the value is found in the sequence.
* **not in:** It returns True if the value is not found in the sequence.

a=[1,2,3,4,5]

print(4 in a)

Output:True

print(9 not in a)

Output: True

**7. Identity Operators**

These operators compare the memory locations of two objects.

* **is:** It returns True if both values refer to the same object in memory.
* **is not:** It returns True if both values refer to different objects.

c=10  
d=20  
print(aisb)  
print(a is not b)

Output:

True

False