⇒ OPERATORS :-

1. Arithmetic Operators [+, -, *, %, /, //, **]
2. Assignment Operators
3. Comparison Operators
4. Logical Operators
5. Bitwise Operators
6. Special Operators

1) ARITHMETIC OPERATORS :-

- Addition (+) : Add two operands
- Subtraction (-) : Subtracts the right operand from the left operand.
- Multiplication (*) : Multiplies Two Operands.
- Division (/) : Divides the left Operand by The right operand (returns a float).

* Floor Division (//) :- Divides the left operand by the right operand and rounds down to the nearest integer [returns an integer].

* Modula (%) :- Returns the remainder of the division.

* Exponentiation (**) :- Raises the left operand to the power of the right operand.

Assignment Operators :

→ Assignment (=) :- Assignment [Assigns] the value on the right to the variable on the left.

→ Add and Assign (+ =) :- Adds the right operand to the variable on the left and assigns the result to the variable.

→ Subtract and Assign (- =) :- Subtracts the right operand from the variable on the left and assigns the result to the variable.

→ Multiply and Assign (* =) :- Multiplies the variable on the left by the right operand and assigns the result to the variable.

→ Divide and Assign (/=) :- Divides the variable on the left by the right operand and assigns the result to the variable.

→ Floor Divide and Assign (//=) :- Floor divides the variable on the left by the right operand and assigns the result to the variable.

→ Modulo and Assign (%=) :- Calculates the remainder of the division and assigns it to the variable.

→ Exponentiate and Assign (**=) :- Raises the variable on the left to the power of the right operand and assigns the result to the variable.

Comparision Operators

1. Equal to (==) :- Checks if two values are equal.
2. Not equal to (!=) :- Checks if two values are not equal.
3. Greater than (>) :- Checks if the left operand is greater than the right operand.
4. Less than (<) :- Checks if the left operand is less than the right (op).
5. Greater than or equal to (>=) :- Checks if The left operand is grater than or equal to the right operand.
6. Less than or equal to (<=) :- Checks if the left operand is less than or equal to the right operand.

Logical Operators

i. and logical (AND) :- Returns True if both conditions are True.

ii. or Logical (OR) :- Returns True if at least one condition is True.

iii. not Logical (NOT) : Returns True if the condition is false, and vice versa.

Membership Operators

1. Membership (in) :- Returns True if the value is present in the sequence.
2. (not in) Negated Membership : Returns True if the value is not present in the sequence.

Identity Operators

1. (is) identity :- Returns True if both variables point to the same object.
2. (is not) Negated identity :- Returns True if the variables point to different objects.

Bitwise Operators

i. Bitwise AND (&) :- Sets each bit to 1 if both bits are 1.

ii. Bitwise OR (|) :- Sets each bit to 1 if at least one of the bits is 1.

iii. Bitwise XOR (^) :- Sets each bit to 1 if only one of the bits is 1.

iv. Bitwise NOT (~) :- Inverts the bits, changing 1 to 0 and 0 to 1.

V. Bitwise Left Shift (<<) :- Shifts the bits to the left by a specified number of positions.

VI. Bitwise Right shift (>>) :- Shifts the bits to the right by a specified number of positions.

Comments :- In Computer programming, Comments are hints that we use to make our code more understandable.

Comments are completely ignored by the interpreter.

Types of Comments :-

1. Single-line Comments

These Comments begin with a # Symbol and continue until the end of the line.

2. Multi-line Comments

Docstrings are enclosed in triple quotes (""" or ''') and can span multiple lines.