“PYTHON OPRETORS”

Python Operators

1. Arithmetic Operators
These operators are used to perform basic mathematical operations.

- **Addition (+)**: Adds two values.
  Example: `a + b` (if `a = 5` and `b = 3`, then `a + b = 8`)
- **Subtraction (-)**: Subtracts one value from another.
  Example: `a - b` (if `a = 5` and `b = 3`, then `a - b = 2`)
- **Multiplication (*)**: Multiplies two values.
  Example: `a * b` (if `a = 5` and `b = 3`, then `a * b = 15`)
- **Division (/)**: Divides one value by another, returns a float.
  Example: `a / b` (if `a = 5` and `b = 3`, then `a / b = 1.6667`)
- **Floor Division (//)**: Divides and returns the integer part of the result.
  Example: `a // b` (if `a = 5` and `b = 3`, then `a // b = 1`)
- **Modulus (%)**: Returns the remainder of the division.
  Example: `a % b` (if `a = 5` and `b = 3`, then `a % b = 2`)
- **Exponent (**)**: Raises one value to the power of another.
  Example: `a ** b` (if `a = 5` and `b = 3`, then `a ** b = 125`)

---

2. Comparison Operators
These operators compare two values and return `True` or `False`.

- **Equal to (==)**: Checks if two values are equal.
  Example: `a == b` (returns `False` if `a = 5` and `b = 3`)
- **Not equal to (!=)**: Checks if two values are not equal.
  Example: `a != b` (returns `True` if `a = 5` and `b = 3`)
- **Greater than (>)**: Checks if one value is greater than another.
  Example: `a > b` (returns `True` if `a = 5` and `b = 3`)
- **Less than (<)**: Checks if one value is less than another.
  Example: `a < b` (returns `False` if `a = 5` and `b = 3`)
- **Greater than or equal to (>=)**: Checks if one value is greater than or equal to another.
  Example: `a >= b` (returns `True` if `a = 5` and `b = 3`)
- **Less than or equal to (<=)**: Checks if one value is less than or equal to another.
  Example: `a <= b` (returns `False` if `a = 5` and `b = 3`)

---

3. Assignment Operators
These operators assign values to variables and can also perform operations during assignment.

- **Assign (`=`)**: Assigns the value on the right to the variable on the left.
  Example: `a = 10`
- **Add and assign (`+=`)**: Adds the right value to the left and assigns the result to the left.
  Example: `a += 5` (if `a = 10`, then `a = 15`)
- **Subtract and assign (`-=`)**: Subtracts the right value from the left and assigns the result to the left.
  Example: `a -= 5` (if `a = 10`, then `a = 5`)
- **Multiply and assign (`*=`)**: Multiplies the left value by the right and assigns the result.
  Example: `a *= 5` (if `a = 10`, then `a = 50`)
- **Divide and assign (`/=`)**: Divides the left value by the right and assigns the result.
  Example: `a /= 5` (if `a = 10`, then `a = 2`)

---

4. Logical Operators
These operators are used to combine conditional statements.

- **AND (`and`)**: Returns `True` if both conditions are true.
  Example: `x > 5 and x < 10` (returns `True` if `x = 7`)
- **OR (`or`)**: Returns `True` if at least one condition is true.
  Example: `x > 5 or x < 3` (returns `True` if `x = 7`)
- **NOT (`not`)**: Reverses the result of a condition.
  Example: `not(x > 5)` (returns `False` if `x = 7`)

---

5. Membership Operators
These operators are used to test if a sequence contains a certain value.

- **in**: Returns `True` if the value is found in the sequence.
  Example: `x in list` (returns `True` if `x` is in `list`)
- **not in**: Returns `True` if the value is not found in the sequence.
  Example: `x not in list` (returns `True` if `x` is not in `list`)

---

6. Identity Operators
These operators compare objects to see if they are the same object in memory.

- **is**: Returns `True` if both variables point to the same object.
  Example: `x is y` (returns `True` if `x` and `y` point to the same object)

- **is not**: Returns `True` if the variables point to different objects.
  Example: `x is not y` (returns `True` if `x` and `y` point to different objects)