Chapter-8

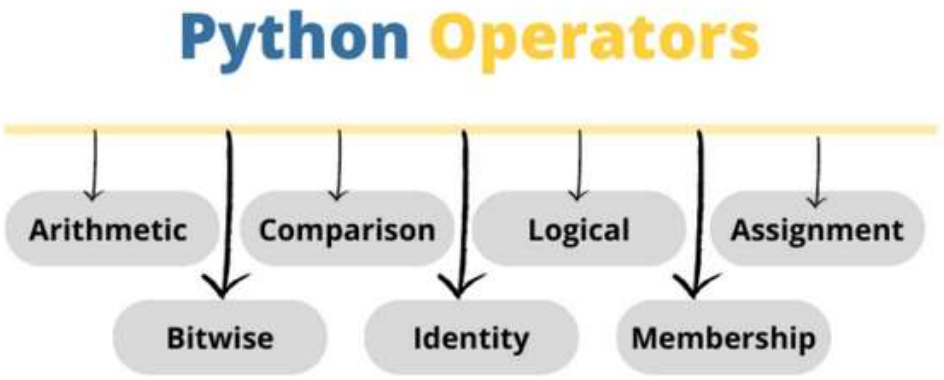
DATA OPERATORS

In the last chapter we learnt about

- **Variables and Values**

We also used **"="** **operator** to **assign** a **value** into a **variable**. Operators are used by every programming language.

In this chapter we will look into some more common operators used in Python.



Definitions –

- **Operators** are used to ***perform operations*** on **variables** and **values**.
- These are ***standard symbols*** used for the **purpose of operations**.
- Arithmetic, Comparison, Logical, and Assignment **operators** are the most commonly used. However, there are other types as well.

Arithmetic Operators

- Arithmetic operators are used to performing mathematical operations like ***addition, subtraction, multiplication, and division***.
- In other words, **Arithmetic operators** are used with **numeric values** to perform common **mathematical operation**
- In other words, with ***int or float*** data types

Operator	Description	Syntax
+	Addition: adds two operands	x + y
–	Subtraction: subtracts two operands	x – y
*	Multiplication: multiplies two operands	x * y
/	Division (float): divides the first operand by the second	x / y
//	Division (floor): divides the first operand by the second	x // y
%	Modulus: returns the remainder when the first operand is divided by the second	x % y
**	Power: Returns first raised to power second	x ** y

Arithmetic Operators Examples with variables a=9 and b = 4

```
# Addition of numbers
add = a + b

# Subtraction of numbers
sub = a - b

# Multiplication of number
mul = a * b

# Division(float) of number
div1 = a / b

# Division(floor) of number
div2 = a // b

# Modulo of both number
mod = a % b

# Power
p = a ** b
```

```
# Lets run this code

# print results
print(add)
print(sub)
print(mul)
print(div1)
print(div2)
print(mod)
print(p)
```

Output

```
13
5
36
2.25
2
1
6561
```

Comparison Operators

- **Comparison or Relational** operators are used to **compare** the values.
- It either returns **True or False** according to condition
- Comes handy in **control structures** like ***condition, iteration and sequence.***

Operator	Description	Syntax
>	Greater than: True if the left operand is greater than the right	x > y
<	Less than: True if the left operand is less than the right	x < y
==	Equal to: True if both operands are equal	x == y
!=	Not equal to – True if operands are not equal	x != y
>=	Greater than or equal to True if the left operand is greater than or equal to the right	x >= y
<=	Less than or equal to True if the left operand is less than or equal to the right	x <= y

Comparison Operators Examples with variables a=13 and b = 33

```
# a > b is False
print(a > b)

# a < b is True
print(a < b)

# a == b is False
print(a == b)

# a != b is True
print(a != b)

# a >= b is False
print(a >= b)

# a <= b is True
print(a <= b)
```

Output

```
False
True
False
True
False
True
```

Assignment Operators

- Assignment operators are used to **assign** values to the variables.

Operator	Description	Syntax
=	Assign value of right side of expression to left side operand	x = y + z
+=	Add AND: Add right-side operand with left side operand and then assign to left operand	a+=b a=a+b
-=	Subtract AND: Subtract right operand from left operand and then assign to left operand	a-=b a=a-b
*=	Multiply AND: Multiply right operand with left operand and then assign to left operand	a*=b a=a*b
/=	Divide AND: Divide left operand with right operand and then assign to left operand	a/=b a=a/b
%=	Modulus AND: Takes modulus using left and right operands and assign the result to left operand	a%=b a=a%b
//=	Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand	a//=b a=a//b
**=	Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand	a**=b a=a**b

```
# Assign value
b = a
print(b)

# Add and assign value
b += a
print(b)

# Subtract and assign value
b -= a
print(b)

# multiply and assign
b *= a
print(b)
```

Assignment Operators Examples with
a = 10

Output
10
20
10
100

Logical Operators

- Logical operators are used to check the **Logical AND**, **Logical OR**, and **Logical NOT** operations
- It is used to combine conditional statements.

Operator	Description	Syntax
and	Logical AND: True if both the operands are true	x and y
or	Logical OR: True if either of the operands is true	x or y
not	Logical NOT: True if the operand is false	not x

Logical Operators Examples

```
# Examples of Logical Operator
a = True
b = False

# Print a and b is False
print(a and b)

# Print a or b is True
print(a or b)

# Print not a is False
print(not a)
```

Output

```
False
True
False
```

We have covered the basic building blocks for learning coding. Journey of programming has just begun. In the next book we will use these concepts together and build software programs.

Thankyou !