Operators

Objectives:

- Arithmetic operators
- Comparison operators
- Identity operators
- Membership operators
- Logical operators

1. What is an Operator?

Operators: Symbols that perform operations on variables and values.

Operands: Variables or values which are passed as inputs to an operator.

For example: `10 - 4`

- `-` is an operator which performs `minus` operation
- `10` and `4` are operands

2. Operators

2.1 Arithmetic Operators

Arithmetic operators perform common mathematical operations on values.

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
//	Floor division	x // y
%	Modulus	x % y
**	Exponentiation	x ** y

Create 2 variables `x = 5` and `y = 3`

In [ ]:
```
1 x, y = 5, 3
```

Addition `+` , Subtraction `-` , Multiplication `*` , Division `/`

In [3]:
```
x + y
x - y
x * y
x / y
```

Out[3]: 1.6666666666666667

Floor Division //

In [4]:
```
x // y
```

Out[4]: 1

Modulus Division %

In [5]:
```
x % y
```

Out[5]: 2

Exponential **

In [6]:
```
x ** y
```

Out[6]: 125

2.2 Comparison Operators

Comparison operators are used to compare two values.

- It returns a boolean value.

Operator	Name	Example
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y
==	Equal	x == y
!=	Not equal	x != y

Greater than > , Greater than or equal to >=

In [8]:
```
x > y
x >= y
```

Out[8]: True

Euqal == , Not equal !=

In [9]:
```
x == y
x != y
```

Out[9]: True

2.3 Logical Operators

Sometimes we need to make decisions based on multiple conditions. Logical operators are used to combine conditional statements.

- Operands shall be conditions which can result in a boolean value.
- The outcome of such an operation is either true or false too.

Operator	Description	Example
and	Returns True if both statements are true	x > 0 and x < 10
or	Returns True if one of the statements is true	x < 0 or x > 10
not	Reverse the result, returns False if the result is true	not x==5

In [10]:
```
x > 0 and x < 10
```

Out[10]: True

In [11]:
```
x < 0 or x > 10
```

Out[11]: False

In [12]:
```
not x == 5
```

Out[12]: False

2.4 Identity Operators

Identity Operators check whether two objects/variables are identical, i.e. whether they point to same memory locations.

Operator	Description	Example
is	Returns true if both variables are the same object	x is y
is not	Returns true if both variables are not the same object	x is not y

In [9]:
```
x = 1000
y = x
z = 1000
print(y is x)
print(z is x)
```

```
True
False
```

Identity Operator can also be used to determine whether a value is of a specific class or type.

In [16]:
```
x = 1
type(x) is int
```

Out[16]: True

2.5 Membership Operators

Membership operators enable us to test whether a value is a member of other Python objects such as strings, lists, or tuples.

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

In [17]:
```
x = 5
y = [1,3,5,7]
x in y
```

Out[17]: True

String is a collection of characters. It behaves very much like a list.

In [8]:
```
x = 'Hello'
'h' not in x
```

Out[8]: True