OPERATORS

- An operator is a symbol that represents an operations that may be performed on one or more operands.
- An operand is a value that a given operator is applied to.
- **Example: 4+(3*k)**

    +, * are operator and 4,3,k are operands

Different forms of operator

- **Unary Operator:**
  - Unary arithmetic operators perform mathematical operations on one operand only. The ‘+’ and ‘-‘’ are two unary operators.
  - **Example:**

```
>>> x = -5  #Negates the value of X
>>> x
-5
```

- **Binary operator:**
  - A Binary operator operates on two operands
  - **Example:**

```
>>> 3 + 10
>>>13
>>> 10 – 7
>>> 3
```

Types of Operators

1. Arithmetic operator
2. Relational operator
3. Logical operator
4. Bitwise operator
5. Assignment operator
6. Special operator

1. Arithmetic operator

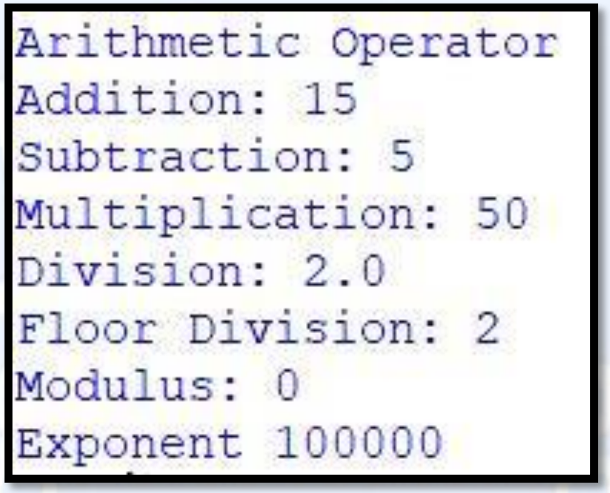
- Arithmetic operators are basic mathematical operations.

Operator	Meaning	Example	Result
+	Addition	C=12+1	C=13
-	Subtraction	C=12-1	C=11
*	Multiplication	C=12*1	C=12
/	Division	C=12/1	C=12
//	Floor division	C=12//10	1
%	Modulus	C=12%10	C=2
**	Exponentiation	C=10**2	C=100

Example of Arithmetic Operator

```
print("Arithmetic Operator")
a=10
b=5
print("Addition:",a+b)
print("Subtraction:",a-b)
print("Multiplication:",a*b)
print("Division:",a/b)
print("Floor Division:",a//b)
print("Modulus:",a%b)
print("Exponent",a**b)
```

Output:



```
Arithmetic Operator
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
Floor Division: 2
Modulus: 0
Exponent 100000
```

2. Relational operator

- Relational operators are also called as Comparison operators
- It is used to compare values.
- It either returns True or False according to condition.

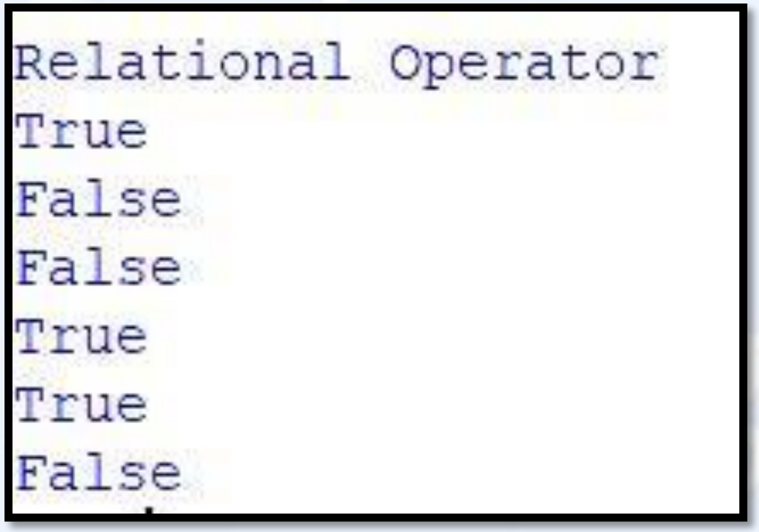
Operator	Meaning	Example	Result
>	Greater than	5>6	False
<	Less than	5<6	True
==	Equal to	5==6	False
!=	Not equal to	5!=6	True
>=	Greater than or equal to	5>=6	False
<=	Less than or equal to	5<=6	True

Example of Relational Operator

```
print("Relational Operator")

a=10
b=5
print(a>b)
print(a<b)
print(a==b)
print(a!=b)
print(a>=b)
print(a<=b)
```

Output:



```
Relational Operator
True
False
False
True
True
False
```

3. Logical operator

- Logical operator are typically used with Boolean(logical) values.
- They allow a program to make a decision based on multiple condition.

Operator	Meaning	Example	Result
and	True if both the operands are true	10<5 and 10<20	False
or	True if either of the operands is true	10<5 or 10<20	True
not	True if operands is false (complements the operand)	not (10<20)	False

Example of Logical Operator

```
print("Logical Operator")
print(10<5 and 10<20)
print(10<5 or 10<20)
print(not(10<20))
```

Output:

```
Logical Operator

False
True
False
```

4. Bitwise operator

- Bitwise operators act on operands as if they are string of binary digits.
- It operates bit by bit.

Operator	Meaning	Example
&	Bitwise AND	a & b
|	Bitwise OR	a | b
~	Bitwise NOT	a ~ b
^	Bitwise XOR	a ^ b
>>	Bitwise right shift	a >> 2
<<	Bitwise left shift	a << 2

5. Assignment operator

- Assignment operators are used to assign values to variables.

Operator	Meaning	Example
=	Assign a value	a=5
+=	Adds and assign the result to the variable	a+=1 (a=a+1)
-=	Subtracts and assign the result to the variable	a-=1 (a=a-1)
*=	Multiplies and assign the result to the variable	a*=5 (a=a*5)
/=	Division and assign the result to the variable	a/= (a=a/5)
//=	Floor division and assign the result to the variable	a//=5(a=a//5)
%=	Find modulus and assign the result to the variable	a%=5 (a=a%5)
**=	Find Exponentiation and assign the result to the variable	a**=5 (a=a**5)

Operator	Meaning	Example
&=	Find Bitwise AND and assign the result to the variable	a&=5(a=a&5)
|=	Find Bitwise OR and assign the result to the variable	a|=5(a=a|5)
^=	Find Bitwise XOR and assign the result to the variable	a^=5(a=a^5)
>>=	Find Bitwise right shift and assign the result to the variable	a>>=5 (a=a>>5)
<<=	Find Bitwise left shift and assign the result to the variable	a<<=5 (a=a<<5)

6. Special operator

- Python offers some special operators like identity operator and the membership operator.
- **Identity Operator:**
  - **is** and **is not** are the identity operator

Operator	Meaning	Example
is	True if the operands are identical	a is true
is not	True if the operands are not identical	a is not true

Example of Identity Operator

```
a1=5
b1=5
a2="Hello"
b2="Hello"
a3=[1,2,3]
b3=[1,2,3]

print(a1 is not b1)
print(a2 is b2)
print(a2 is b3)
```

Output:

```
False
True
False
```

- **Membership Operators:**
  - **in** and **not in** are the membership operators.

Operator	Meaning	Example
in	True if value/ variable is found in the sequence	5 in a
not in	True if value/ variable is not found in the sequence	5 not in a

Example of Membership Operator

```
a="Hello world"
b={1,"a","b",2}

print("H" in a)
print("hello" in a )
print(1 in b)
print("b" in b)
Print(“c” not in b)
```

Output:

```
True
False
True
True
True
```