Operators:

An Operator is a symbol which is used for performing certain operations on given object Data /values and gives result

Assignment Operator:

The assignment operator is represented by the equal sign (=). It is used to assign a value or the result of an expression to a variable.

variable = Expression

- variable is the name of the variable to which you want to assign a value.
- '=' is the assignment operator. *expression is the value or expression that you want to assign to the variable.

```
a=10 #asign value 10to variable a
```

compound assignment operators :

- += : Add and assign
- -= : Subtract and assign
- *= : Multiply and assign
- /= : Divide and assign
- %= : modulus and assign
- **= : Exponentiate and assign

```
x=6
x+=4 #x = x + 4
print(x) #It adds the value on the right to the variable on the left
and assigns the result to the variable on the left.
```

```
10
```

```
x=6
x-=4 #x = x - 4
print(x) #It subtracts the value on the right from the variable on the
left and assigns the result to the variable on the left.
```

```
2
```

```
x=4
x*=5 #x=x*5
print(x) #It multiplies the variable on the left by the value on the
right and assigns the result to the variable on the left.
```

```
20
```

```
x=10
x/=2 #x=x/2
print(x)
```

```
5.0
```

```
x=4
x**=2 #x=x**2
print(x)
```

```
16
```

Bitwise Operator:

- The Bitwise operator are applicable on integer data but not on floating data point because floating point values does not contain certainity
- Bitwise operator are used for performing operator on Integer data in the form of Bit by Bit
- '&' (Bitwise AND)
- '|' (Bitwise OR)
- '^' (Bitwise XOR)
- '~' (Bitwise NOT)
- '<<' (Bitwise Left Shift)
- '>>' (Bitwise Right Shift)

Bitwise AND " & " 0 & 1 ---------->0 0 & 0 ---------->0 1 & 0 ---------->0 1 & 1 ---------->1

```
a=10 # 1010
b=15 # 1111
c=a&b# 1010
print(c)
```

```
10
```

Bitwise OR " | " 0 | 0 ---------->0 1 | 0 ---------->1 0 | 1 ---------->1 1 | 1 ---------->1

```
a=10 # 1010
b=15 # 1111
c=a|b# 1111
print(c)
```

```
15
```

Bitwise XOR " ^ " 0 ^ 0 ---------->0 1 ^ 0 ---------->1 0 ^ 1 ---------->1 1 ^ 1 ---------->0

```
a=10 # 1010
b=15 # 1111
c=a^b# 0101
print(c)
```

```
5
```

Bitwise NOT " ~ "

- It inverts the bits
- Inverting the bits nothing but 0 becomes 1 and 1 become 0
- The formula for bitwise complement operator (~) is that ~ value= -(value+1)

```
a=10 # 1111
b=~a
print(b)
```

```
-11
```

```
a=4 #0100
~a  #-(1011)+1
```

```
-5
```

Bitwise Left shift (<<)

```
a=10 #1010
b=a<<3
b
```

```
80
```

```
#formula = Given number * 2 power of no of bits
a=10
b=a<<3
10*2**3
```

```
80
```

Bitwise Right shift(>>)

```
a=10 #1010
b=a>>3
b
```

```
1
```

```
#formula : varname=given number / 2 power of no of bits
a=10
b=10>>3
b=10/2**3
```

```
b=10//8
b
```

```
1
```