

## Bitwise Operators

Bitwise operators are used to perform bitwise operations (using 0's and 1's). These operators are applied on binary values.

### Applications of bitwise operators

1. Operating system for memory management
2. Bitwise operators are used in low level programming
  - 0 off
  - 1 on
3. Logic Gates
4. Encryption and Decryption

Operator	Description
>>	<p>Right shift operator</p> <p>This operator is used for shifting number of bits towards right side. By shifting number of bits towards right side the value get decremented. The number of bits shifted towards right side is deleted.</p> <p>Shift operators are binary operator</p> <p>value&gt;&gt;n</p> <p><b>Example:</b></p> <pre>&gt;&gt;&gt; a=10 &gt;&gt;&gt; b=a&gt;&gt;2 &gt;&gt;&gt; print(a,b) 10 2 &gt;&gt;&gt; print(bin(a),bin(b)) 0b1010 0b10</pre> <div><div><p>a=10 b=a&gt;&gt;2 print(a) print(b)</p><div><div>2</div><div>10</div><div>2</div><div>5-0</div><div>2</div><div>2-1</div><div>1-0</div></div></div><div><div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div><div>1</div><div>0</div></div><div><div>&gt;&gt;</div><div>2</div></div><div><div>2</div><div>2</div><div>2+0=2</div></div></div></div> <p>Formula → num//2 pow n</p>

**Example:**

# find output

```
num=157
x=num>>4
print(num,x)
print(bin(num),bin(x))
```

Output:

157 9

0b10011101 0b1001

&lt;&lt;

Left shift operator

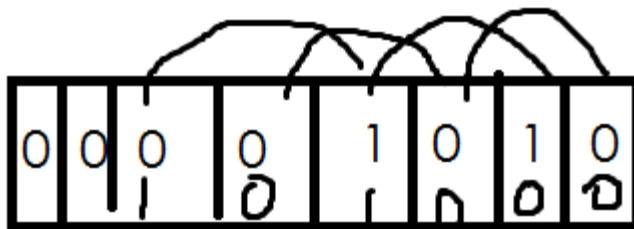
This operator is used to shift number of bits towards left side. By shifting number of bits towards left side the value get incremented.

Formula:  $\text{num} * 2^{\text{pow } n}$ 

```
>>> a=10
>>> b=a<<2
>>> print(a,b)
10 40
>>> print(bin(a),bin(b))
0b1010 0b101000
>>> c=5
>>> d=c<<1
>>> print(c,d)
5 10
>>> print(bin(c),bin(d))
0b101 0b1010
```

a=10

b=a&lt;&lt;2



&

Bitwise and operator

This operator is used to apply AND gate

Truth table of & operation

Opr1	Opr2	Opr1 & Opr2
1	1	1
0	1	0
1	0	0
0	0	0

>>> a=0b101

>>> b=0b100

>>> c=a&b

>>> print(bin(a),bin(b),bin(c))

0b101 0b100 0b100

>>> n1=5

>>> n2=4

>>> n3=n1&n2

>>> print(n1,n2,n3)

5 4 4

>>> print(bin(n1),bin(n2),bin(n3))

0b101 0b100 0b100

|

Bitwise or operator

This operator is used for applying or gate

Truth table or operator

Opr1	Opr2	Opr1   Opr2
1	0	1
0	1	1
0	0	0
1	1	1

>>> a=0b1010

>>> b=0b1111

>>> c=a|b

>>> print(a,b,c)

10 15 15

>>> print(bin(a),bin(b),bin(c))

0b1010 0b1111 0b1111

>>> a=10

	<pre>&gt;&gt;&gt; b=15 &gt;&gt;&gt; c=a b &gt;&gt;&gt; print(a,b,c) 10 15 15 &gt;&gt;&gt; print(bin(a),bin(b),bin(c)) 0b1010 0b1111 0b1111 &gt;&gt;&gt; c=a or b &gt;&gt;&gt; print(c) 10</pre>															
^	<p>Bitwise XOR operator</p> <p>This operator is used to apply xor gate</p> <table><tr><th>Opr1</th><th>Opr2</th><th>Opr1 ^ Opr2</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table> <pre>&gt;&gt;&gt; a=0b101 &gt;&gt;&gt; b=0b111 &gt;&gt;&gt; c=a^b &gt;&gt;&gt; print(a,b,c) 5 7 2 &gt;&gt;&gt; print(bin(a),bin(b),bin(c)) 0b101 0b111 0b10</pre>	Opr1	Opr2	Opr1 ^ Opr2	1	0	1	0	1	1	1	1	0	0	0	0
Opr1	Opr2	Opr1 ^ Opr2														
1	0	1														
0	1	1														
1	1	0														
0	0	0														
~	<p>Bitwise not operator</p> <p>Formula for applying bitwise not operator</p> <p>-(opr+1)</p> <pre>&gt;&gt;&gt; a=15 &gt;&gt;&gt; b=~a &gt;&gt;&gt; print(a) 15 &gt;&gt;&gt; print(b) -16 &gt;&gt;&gt; c=-16</pre>															

<pre>&gt;&gt;&gt; d=~c &gt;&gt;&gt; print(c) -16 &gt;&gt;&gt; print(d) 15</pre>
---

## Assignment Operators

**a=10**

**a,b=10,20 → tuple**

**a,b,c,d=10,20,30,40**

**a,b,c,d=10,20,30,40,50,60 # Error**

**=**

**+=**

**-=**

**\*=**

**/=**

**//=**

**%=**

**\*\*=**

**&=**

**|=**

**^=**