Bitwise Operators

And,Or ,XOr(&,|,^)

In [2]:
```
a = 12
b = 14
```

In [3]:
```
print(a)
print(b)
```

```
12
14
```

In [4]:
```
bin(a)
```

Out[4]: '0b1100'

In [5]:
```
bin(b)
```

Out[5]: '0b1110'

In [6]:
```
a & b     #takes 12 and 13 in binary form and then and operation will perform
```

Out[6]: 12

In [7]:
```
bin(a & b)
```

Out[7]: '0b1100'

In [8]:
```
c = 25
d = 30
```

In [9]:
```
print(c)
print(d)
```

```
25
30
```

In [10]:
```
print(bin(c & d))
```

```
0b11000
```

In [11]:
```
a | b
```

Out[11]: 14

In [12]:
```
bin(a)
```

Out[12]: '0b1100'

In [13]:
```
bin(b)
```

Out[13]: '0b1110'

In [14]:
```
print(bin(a | b))
```

```
0b1110
```

In [15]:
```
print(c)
print(d)
```

```
25
30
```

In [16]:
```
c ^ d                #takes c and d values in binary
                       # perform xor operation(any binary value is 1/0 of both
                       # for the answer do (from right to left)(2 power zero in
```

Out[16]: 7

In [17]:
```
bin(7)
```

Out[17]: '0b111'

In [18]:
```
# right shift(>>),left shift(<<)
```

In [19]:
```
a << 2
```

Out[19]: 48

In [20]:
```
print(a)
```

```
12
```

In [21]:
```
bin(a)
```

Out[21]: '0b1100'

In [22]:
```
a<<23              # add 2 zero to left side to the binary value
```

Out[22]: 48

In [23]:
```
bin(48)
```

Out[23]: '0b110000'

In [24]:
```
print(b)
```

```
14
```

In [25]:
```
bin(b)
```

Out[25]: '0b1110'

In [26]:
```
b>>1                    # remove 1 value from the binary value
```

Out[26]: 7

In [27]:
```
bin(7)
```

Out[27]: '0b111'

In [28]:
```
a >> 2
```

Out[28]:
```
3
```

In [29]:
```
bin(3)
```

Out[29]:
```
'0b11'
```

In [ ]:
```

```