# Numbering system

- Binary number 0b -- base 2(0,1)
- Octal 0o -- base 8(0 to 7)
- Decimal 0x -- base 10(0to9)
- Hexadecimal 0xa,b,c -- base 16 (0to9 and a to f)

```
In [4]:  bin(25) # bin is code for binary

Out[4]:  '0b11001'

In [2]:  int(0b11001)

Out[2]:  25

In [3]:  0b11001

Out[3]:  25

In [5]:  bin(30)

Out[5]:  '0b11110'

In [6]:  0b11110

Out[6]:  30

In [11]: oct(25) # oct is code for octal

Out[11]: '0o31'

In [12]: 0o31

Out[12]: 25

In [17]: 0x19

Out[17]: 25

In [18]: hex(25)

Out[18]: '0x19'

In [20]: hex(16)

Out[20]: '0x10'

In [24]: print(0xa)
         print(0xb)
         print(0xf)
```

```
10
11
15
```

In [27]:
```
hex(1)
```

Out[27]:
```
'0x1'
```

In [28]:
```
hex(3)
```

Out[28]:
```
'0x3'
```

In [31]:
```
hex(9)
```

Out[31]:
```
'0x9'
```

In [32]:
```
hex(10)
```

Out[32]:
```
'0xa'
```

In [33]:
```
hex(15)
```

Out[33]:
```
'0xf'
```

In [34]:
```
hex(16)
```

Out[34]:
```
'0x10'
```

In [35]:
```
hex(17)
```

Out[35]:
```
'0x11'
```

In [36]:
```
hex(100)
```

Out[36]:
```
'0x64'
```

In [37]:
```
hex(256)
```

Out[37]:
```
'0x100'
```

# Bitwise operators

- AND , OR
-

In [39]:
```
12&13
```

Out[39]:
```
12
```

In [40]:
```
12 | 13
```

Out[40]:
```
13
```

In [41]:
```
bin(32)
```

```
Out[41]: '0b100000'
```

```
In [42]: 35 & 40
```

```
Out[42]: 32
```

```
In [43]: 35 | 40
```

```
Out[43]: 43
```

```
In [44]: bin(43)
```

```
Out[44]: '0b101011'
```

```
In [45]: 0b0001
```

```
Out[45]: 1
```

```
In [47]: bin(1)
```

```
Out[47]: '0b1'
```

# Xor (Xclusive or)

```
In [50]: print(12 ^ 13)
         print(25 ^ 30)
```

```
1
7
```

# Left shift (<<) and Right shift (>>)

```
In [56]: print(10<<1) # left shfit moves binary left
         print(10<<2)
         print(10<<3)
         print(10>>1) # right shfit moves binary rigth
         print(10>>2)
         print(10>>3)
```

```
20
40
80
5
2
1
```

```
In [ ]:
```