

Bit operations

```
In [1]: x = 67
        bx = format(x, '08b')
        print("x: decimal {}, binary {}".format(x, bx))

x: decimal 67, binary 01000011
```

```
In [2]: y = 11
        by = format(y, '08b')
        print('y: decimal {}, binary {}'.format(y, by))

y: decimal 11, binary 00001011
```

```
In [3]: z = x & y
        bz = format(z, '08b')
        print('x & y: decimal {}, binary {}'.format(z, bz))

x & y: decimal 3, binary 00000011
```

```
In [4]: z = x | y
        bz = format(z, '08b')
        print('x | y: decimal {}, binary {}'.format(z, bz))

x | y: decimal 75, binary 01001011
```

```
In [5]: z = ~x
        bz = format(z, '08b')
        print('~x: decimal {}, binary {}'.format(z, bz))

~x: decimal -68, binary -1000100
```

```
In [6]: z = x ^ y
        bz = format(z, '08b')
        print('x ^ y: decimal {}, binary {}'.format(z, bz))

x ^ y: decimal 72, binary 01001000
```

```
In [7]: z = x << 1
        bz = format(z, '08b')
        print('x << 1: decimal {}, binary {}'.format(z, bz))

x << 1: decimal 134, binary 10000110
```

```
In [8]: z = x >> 1
        bz = format(z, '08b')
        print('x >> 1: decimal {}, binary {}'.format(z, bz))

x >> 1: decimal 33, binary 00100001
```

Hexadecimal

0,1,...,9,A,B,C,D,E,F

- 0 => 0000
- 1 => 0001
- ...
- A => 1010
- ...
- F => 1111

```
In [9]: x = 67
        bx = format(x, '#08b')
        hx = format(x, '#02x')
        print("x: binary {}, hex {}".format(bx, hx))

x: binary 0b1000011, hex 0x43
```

```
In [10]: y = 75
         by = format(y, '#08b')
         hy = format(y, '#02x')
         print("y: binary {}, hex {}".format(by, hy))

y: binary 0b1001011, hex 0x4b
```

Big endian vs little endian

```
In [11]: num = 258
         print(format(num, '016b'))

00000000100000010
```

```
In [12]: print(num.to_bytes(2, 'big'))
         print(num.to_bytes(2, 'little'))

b'\x01\x02'
b'\x02\x01'
```

```
In [13]: print(num.to_bytes(4, 'big'))
         print(num.to_bytes(4, 'little'))

b'\x00\x00\x01\x02'
b'\x02\x01\x00\x00'
```