CIS 211 Class Encore
Spring 2022
Jose Renteria

Bits and Booleans

Terms:

- **Bit**: Smallest unit of binary data, representing single binary digit, 0 or 1
- **Nibble**: a nibble consists of 4 bits. Commonly used to represent hexadecimal digits.
- **Byte**: A byte consists of 8 bits
    - 1 kilobit (Kb) = 1,000 bits
    - 1 kilobyte (KB) = 1,024 bytes
    - 1 megabit (Mb) = 1,000,000 bits
    - 1 megabyte (MB) = 1,048,576 bytes
- **Bitwise**: refers to any operation performed on individual bits of binary data.
    - &&, ||, <<, >>, ==, != , XOR, different than

1. Binary Representation
    a. Binary is in base two and is represented with most significant bits first, i.e. on the right. This means any leading zeros can essentially be dropped, but they can be useful for ease of manipulation.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

11111111 = 255

Essentially just the sum of every bit that is “on”/True/Equal to 1:

$$\sum 2^i \text{ , for each } 2^i == 1$$

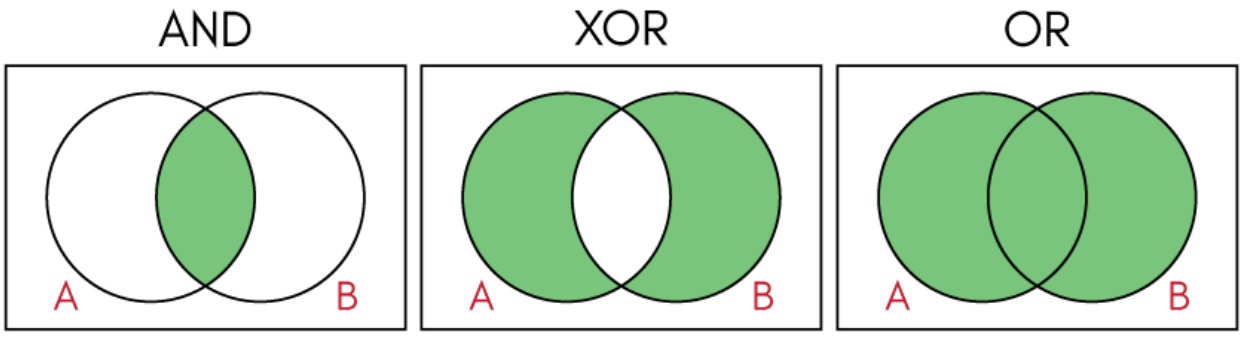
Practice:

Find the decimal representation of this binary representation

1	1	1	0	0	1	0	1

2. Bitwise operations

A	B	!A	A && B	A || B	A != B
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0



Bit shifting,

Left Bit Shift		
Syntax	Binary	Decimal
a = 7	00000111	7
a <<= 1	00001110	14
a <<= 3	01110000	112
a <<= 2	11000000	192 (448 in py)

What is going on here? What does each shift signify in its binary representation? Why might this be useful?

Hint: Think about the values at each each bit position

Right Bit Shift		
Syntax	Binary	Decimal
a = 56	00111000	56
a >>= 1	00011100	28
a >>= 2	111	7
a >>= 3	0000	0

Why did a >> 2 cause our binary representation to change its length?