

There are 10 rocks.

Oh, you must be  
using base 4. See,  
I use base 10.

No. I use base 10.  
What is base 4?



Every base is base 10.

# Binary

- Computers represent everything as bits
- Recall: a byte is 8 bits
- Int: 4 bytes (32 bits)

# Binary

- Computers represent everything as bits
- Recall: a byte is 8 bits
- Int: 4 bytes (32 bits)
- What's the largest int we can represent?

# Binary

- Computers represent everything as bits
- Recall: a byte is 8 bits
- Int: 4 bytes (32 bits)
- What's the largest int we can represent?

$2^{32} - 1$

(unsigned)

# Hexadecimal (base 16)

- Binary takes up a lot of space
- Hexadecimal takes few digits but can easily be converted to binary (and vice versa)
  - Hex uses digits 0-9 and a-f
  - 1 hex digit = 4 bits
  - It would be helpful to use binary...but it would take up too much space
- 0000 0000 0000 0001 1101 0011 0101 1011
- 1d35b

# In C

- Format ints
  - %d for decimal
  - %b for binary
  - %x for hex
- Assign ints
  - 0b for binary (ex: 0b11011 is 27)
  - 0x for hex (ex: 0x83fa9 is 540585)

# Bitwise Operators

- You know logical operators...&&, ||, !
- We will now learn &, |, ~, ^, <<, >>
- These operate at the bit level

&

a	b	a & b
1	1	1
1	0	0
0	1	0
0	0	0



|

a	b	a   b
1	1	1
1	0	1
0	1	1
0	0	0

$\wedge$

a	b	$a \wedge b$
1	1	0
1	0	1
0	1	1
0	0	0

$\wedge$

a	$\sim a$
1	0
0	1

# Operators on multiple bits

AND

	0110
&	1100
	----
	0100

OR

	0110
	1100
	----
	1110

XOR

	0110
^	1100
	----
	1010

NOT

	1100
~	
	----
	0011