Binary numbers

As humans, we typically represent numbers in the **decimal** system. Counting to ten is as simple as 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

As we just learned, computers represent all information in **bits**. In order to represent numbers with just 0 and 1, computers use the **binary** number system. Here's what it looks like when a computer counts to ten:

0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010

Refresher: Decimal numbers

Before exploring how the binary system works, let's revisit our old friend, the decimal system. When you learned how to count, you might have learned that the right-most digit is the "ones' place", the next is the "tens' place", the next is the "hundreds' place", etc.

Another way to say that is that the digit in the right-most position is multiplied by 1, the digit one place to its left is multiplied by 10, and the digit two places to its left is multiplied by 100.

Let's visualize the number 234:

2	3	4
hundreds' place	tens' place	ones' place
100	10	1

When we multiply each digit by its place, we can see that 234 is equal to (2 x 100) + (3 x 10) + (4 x1).

We can also think of those places in terms of the powers of ten. The ones' place represents multiplying by 10^0, the tens' place represents multiplying by 10^1, and the hundreds' place represents multiplying by 10^2. Each place we add, we're multiplying the digit in that place by the next power of 10.

2	3	4
hundreds' place	tens' place	ones' place
100	10	1
10^2	10^1	10^0

Binary numbers

The binary system works the same way as decimal. The only difference is that instead of multiplying the digit by a power of 10, we multiply it by a power of **2**.

Let's look at the decimal number 1, represented in binary as 0001:

0	0	0	1
8	4	2	1
2^3	2^2	2^1	2^0

That's the same as $(0\times8)+(0\times4)+(0\times2)+(1\times1)$.

Okay, perhaps you could have guessed that one — now for a bigger number!

The decimal number 10 is represented in binary as 1010:

1	0	1	0
8	4	2	1
2^3	2^2	2^1	2^0

That's the same as $(1\times8)+(0\times4)+(1\times2)+(0\times1)$, or 8 + 0 + 2 + 0. Indeed, binary 1010 equals the decimal 10.

Now you try it: How would you represent the decimal number 6 in binary?

Answer: 0110

Converting decimal to binary

Here's my favorite way to convert decimal numbers to binary:

1. Grab a piece of paper.
2. Draw dashes for each of the bits. If the number is less than 16, draw 4 dashes. Otherwise, for numbers up to 255, draw 8 dashes. Bigger numbers than that require more bits and take a while to do by hand, so let's focus on the smaller numbers.
3. Write the powers of 2 under each dash. Start under the right-most dash, writing 1, then keep multiplying by 2.
4. Now start at the left-most dash and ask yourself "Is the number greater than or equal to this place value?" If you answer yes, then write a 1 in that dash and subtract that amount from the number. If you answer no, then write a 0 and move to the next dash.
5. Keep going from left to right, keeping track of how much remainder you still need to represent. When you're done, you'll have converted the number to binary!
   Here's what that looks like for the decimal number 6:

"Hmm, 6 is less than 16, so 4 bits is plenty..."

$$\frac{}{8}\ \frac{}{4}\ \frac{}{2}\ \frac{}{1}$$

"Well, 6 is less than 8, so I'll write a 0 first..."

$$\frac{0}{8}\ \frac{}{4}\ \frac{}{2}\ \frac{}{1}$$

"6 is bigger than 4, so I'll write a 1 next..."

$$\frac{0}{8}\ \frac{1}{4}\ \frac{}{2}\ \frac{}{1}$$

"Ok, 6 - 4 = 2, so I still need to represent 2. Let me note that..."

$$\frac{0}{8}\ \frac{1}{4}\ \frac{}{2}\ \frac{}{1}\ \text{(Remainder: 2)}$$

"2 is equal to 2, so I'll write a 1 next..."

$$\frac{0}{8}\ \frac{1}{4}\ \frac{1}{2}\ \frac{}{1}$$

"2 - 2 = 0, so there's nothing left to represent!"

$$\frac{0}{8}\ \frac{1}{4}\ \frac{1}{2}\ \frac{}{1}\ \text{(Remainder: 0)}$$

"I'll fill a 0 in the last bit, since I'm all done now..."

$$\frac{0}{8}\ \frac{1}{4}\ \frac{1}{2}\ \frac{0}{1}$$

In case you're wondering, there's only one way to represent any given number in binary, just like there's only one way to represent any given number in decimal. Any technique that you use for converting a decimal to binary number should yield the same number.