CS50 Hexadecimal

Overview

Recall that our computers break everything from ASCII symbols to source code down into combinations of 0s and 1s (**binary**). Those 0s and 1s are not that efficient when it comes to expressing large numbers. To express the decimal number 15, for instance, we need four place values in binary: 1 1 1 1. Because four digits of binary can represent 16 values, computer scientists settled on hexadecimal, a number system of base 16, to represent those larger numbers.

Key Terms

- binary
- hexadecimal
- RGB values

Hexadecimal

In the decimal system (base 10), we have ten digits, 0-9, and each place value represents the next power of 10. So the n^{th} place value can be calculated by taking 10_{n-1}, like in binary (base 2), where we could calculate the n^{th} place value by taking 2_{n-1}.

Similarly, in **hexadecimal** (base 16), we use 0-9 for the first ten digits and the letters A-F for the remaining six. We can think of A as 10, B as 11, and so forth. As you might guess, hexadecimal's place values are based on powers of 16. Note that all the hexadecimal place values are found in binary, albeit more spread out. This makes sense when we remember that $2^4 = 16$ and that what takes 4 digits to express in binary can be expressed in 1 digit in hexadecimal.

To convert numbers directly from binary to hexadecimal, simply block off the binary number into chunks of four digits and express what they represent as a single hexadecimal digit. For example, 0 0 0 0 in binary would be a 0 in hexadecimal, and a 1 1 1 1 in binary would be converted into an F (which represents 15) in hexadecimal. This optimization allows us to represent much larger numbers using fewer characters.

Decimal System

3	1	9
100s	**10s**	**1s**
(3 x 100) +	(1 x 10) +	(9 x 1)

300 + 10 + 9
319

Hexadecimal System

1	3	F
256s	**16s**	**1s**
(1 x 256) +	(3 x 16) +	(15 x 1)

256 + 48 + 15
319

Binary System

1	0	0	1	1	1	1	1	1
256s	**128s**	**64s**	**32s**	**16s**	**8s**	**4s**	**2s**	**1s**
(1 x 256) +	(0 x 128) +	(0 x 64) +	(1 x 32) +	(1 x 16) +	(1 x 8) +	(1 x 4) +	(1 x 2) +	(1 x 1)

256+ 0 + 0 + 32 + 16 + 8 + 4 + 2 + 1
319

Hex Colors

One application of the hexadecimal system is the representation of colors. As you may know, all colors are made up of varying levels of red, green, and blue. We refer to these as the **RGB values**. Each of the three colors can have a value between 0 and 255 (16^2-1), which means we need to be able to represent 16,777,216 different colors. And using the hexadecimal number system, we are able to do this in only 6 digits! Imagine using the binary system to express that many colors. It would take 4 times as many digits.