CS50 Hexadecimal

Overview

Recall that our computers break everything down into combinations of 0s and 1s (**binary**), this includes ASCII symbols as well as all of your source code. Those 0s and 1s are not that efficient when it comes to expressing large numbers. Just to express the decimal number 15, we need four place values in binary, 1 1 1 1. One solution computer scientists settled on, to represent those larger numbers, was **hexadecimal**, a number system of base 16.

Key Terms

- binary
- hexadecimal
- RGB values

Hexadecimal

In the decimal system (base 10) we have 10 digits, 0-9, similarly in hexadecimal (base 16) we use 0-9 for the first 10 digits and the letters A-F for the remaining 6. We can think of A as 10, B as 11, and so forth. In base 10, each place value represents the next power of 10. So the n^{th} place value can be calculated by taking 10^{n-1}, similarly with binary (base 2) we can calculate the n^{th} place value by taking 2^{n-1}. As you might guess, hexadecimal's place values are based off of powers of 16. Notice that the same numbers in the hexadecimal place values are found in the binary place values albeit more spread out. Since $2^4 = 16$, what takes 4 digits in binary can be expressed in 1 digit in hexadecimal. To convert numbers directly between binary and hexadecimal, simply block off the binary number into chunks of four digits, 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 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 way we represent colors. As you may know, every color is 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), so we need to be able to represent 16,777,216 different colors. We are able to do this in 6 digits using the hexadecimal number system. Imagine using the binary system to express that many colors. It would take 4 times as many digits.