

How Color Works in Your Digital Camera

In your camera, and on your computer, there is a language that has been developed to translate the digital information captured in our photos into the colors that we see. This language is called the Color Mode and our cameras, computers, and software, like Photoshop Elements, use the RGB Color Mode.

The RGB Color mode is named after the three primary colors that are the foundation of the color mode's "language"; Red, Green and Blue.

In the images that our cameras capture, every color you see is created by mixing some amount of Red, Green and Blue together.

It looks something like this:

In this color wheel you can see the Red, Green, and Blue circles. Where the circles overlap and mix, you get new colors.

Here are the mixes:

Red + Blue = Magenta

Blue + Green = Cyan

Red + Green = Yellow

Red + Blue + Green = White

no Red + no Blue + no Green = Black



When looking at this wheel however, you should be wondering how you get all the colors you see in your digital photos using just Red, Green and Blue.

The trick is to mix together different variations of Red, Green and Blue.

This color wheel shows each of the primary colors (Red, Green & Blue) in their purest form. The Red is 100% Red, the Green is 100% Green and the Blue is 100% Blue.

To illustrate, here are the above color equations restated with percentages:

100% Red + 100% Blue + 0% Green = Magenta

100% Blue + 100% Green + 0% Red = Cyan

100% Red + 100% Green + 0% Blue = Yellow

100% Red + 100% Green + 100% Blue = White

0% Red + 0% Green + 0% Blue = Black

We get more colors by using different percentages of the primary colors. You don't just mix 100% of Red Green and Blue together. Sometimes you mix 25% Red, 75% Green and 10% Blue. In fact, here is the color that results from that mixture:

25% Red + 75% Green + 10% Blue =



It's this method of mixing varying percentages of Red, Green and Blue that allows our cameras and computers to create the infinite variety of colors in our digital photos.

But there's a catch. The variety of colors we can capture using the RGB Color Mode is not infinite. It's actually limited to 16.8 million colors.

Why is the number of colors limited?

For a computer to represent color, each color needs to have a unique identifier. So for the color Red, every variation of Red from from 0% to 100% needs to be assigned a unique identifier. You might think that means you only need 100 unique identifiers, but that's not true, because between 0% and 100% there are an infinite number of variations. Anything from 0% to 64.9835267443% to 99.99999999999999999999% to 100% is valid. This infinite variety of color between 0% and 100% is impossible to use because our computers and cameras would have to hold an infinite number of unique identifiers, and trying to hold an infinite number of unique identifiers would give us enormous image files that would make working with them impossible.

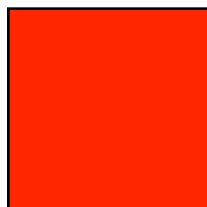
To make things reasonable for our cameras and computers to handle, the number of variations for each primary color is limited to a specific set of unique identifiers, and these unique identifiers are created using bits.

A bit is a bit of computer terminology (see what I did there?) that refers to a single unit of information. Bits, like a switch, only have two options. So a bit is either on (0), or off (1).

The number of bits used to create the identifier for a color limits the number of variations of that color available to be used.

If one bit is used to identify all the varieties of Red available, you get two colors.

Color	Red
Bit Identifier	0
Percentage	100%

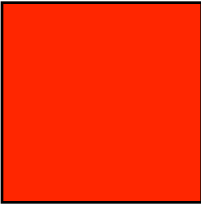



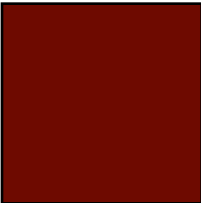

Color	Red
Bit Identifier	1
Percentage	100%



Two colors are clearly not enough to give us all the colors we capture in our digital images, so we need to add more bits.

If we add one bit to the identifier, giving us a total of two bits, we get four varieties per color. Again for the Red, we would now have:

Color	Red		Color	Red	
Bit Identifier	00		Bit Identifier	01	
Percentage	100%		Percentage	66.6%	

Color	Red		Color	Red	
Bit Identifier	10		Bit Identifier	11	
Percentage	33.3%		Percentage	0%	

It's important to note here that when creating a unique identifier using bits, the order matters. In the example above, we have two bit sequences that use 1 and 0 (10 and 01). If we just look at the occurrences of a combination of 0 and 1, those colors would be the same, but since the order matters, each creates a unique variation of Red.

Each bit you add to the identifier increases the possible variations of a color. To calculate the number of possible variations, you multiply the number of states possible per bit (2) by the number of bits being used.

For one bit $2 \times 1 = 2$ color variations. (0, 1)

For two bits $2 \times 2 = 4$ color variations. (00, 01, 10, 11)

An easier way to show this is with exponentials.

So $2 \times 1 = 2^1$, $2 \times 2 = 2^2$ and so on.

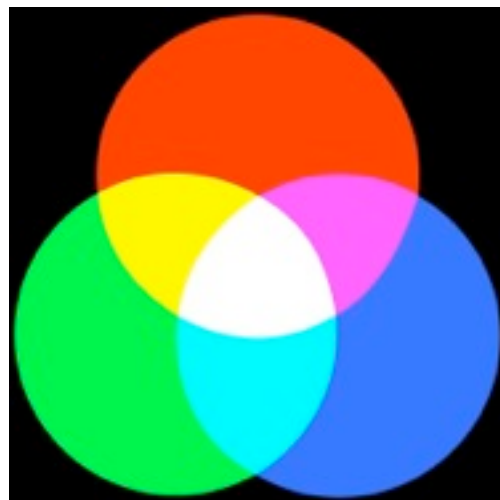
Now up to this point we've been looking at the variations of a single color, Red. To figure the total number of colors in the RGB Color Mode we also have to factor in the Green and Blue variations.

With RGB Color, if you only use one bit to identify each color, the total colors available would be eight.

This is figured by multiplying the number of variations per color together (2 Red, 2 Green and 2 Blue).

$$2 \times 2 \times 2 = 8$$

Bringing us back to this color wheel, and the eight colors available in it.



Why 16.8 Million Colors?

If we do the math bit by bit it's easy to see why we use 16.8 million colors the RGB Color Mode.

Let's take it bit by bit:

1 bit: $2^1 = 2$ variations per color. $2 \times 2 \times 2 = 8$ total colors

2 bits: $2^2 = 4$ variations per color. $4 \times 4 \times 4 = 64$ total colors

3 bits: $2^3 = 8$ variations per color. $8 \times 8 \times 8 = 512$ total colors

4 bits: $2^4 = 16$ variations per color. $16 \times 16 \times 16 = 4096$ total colors

You can see the exponential increase each time a bit is added, so let's skip ahead.

7 bits: $2^7 = 128$ variations per color. $128 \times 128 \times 128 = 2.1$ million total colors

8 bits: $2^8 = 256$ variations per color. $256 \times 256 \times 256 = 16.8$ million total colors

It's by using 8 bits to create the unique identifiers for Red, Green and Blue that we get our 16.8 million colors.

Why 8 bits and 16.8 million colors? The human eye is able to see approximately 10 million different colors, but as you can see from the math, you can't create exactly 10 million different colors using bit identifiers, so we have to use the bit system because that's what the entire foundation of computing is built on.

If we need at least 10 million colors, 7 bits falls way short, and the next available option is 8 bits with 16.8 million colors, so that is what we use. Quite honestly, you wouldn't want exactly 10 million colors anyway. 16.8 million colors gives us some breathing room, providing us with more than enough options to create richly colorful photographs with our digital cameras.