

Images

 coursera.org/learn/programming-fundamentals/supplement/DFE8U/images

Your computer frequently displays images—whether it's the windows and icons on your screen, or the lolcats you view in your web browser. These may seem like they are not numbers, however, they are actually just many numbers put together. The first step to representing an image as a number is to represent a color as a number.

While there are many ways to represent a color as a number, the most common is RGB encoding, which encodes each color by specifying how much red, green, and blue they contain. Typically, this encoding is done with each component being represented on a scale from 0 to 255. The RGB values for the color red are: R=255, G=0, B=0. Orange is R=255, G=127, B=0. If you search the Internet, you will find many online tools that will let you select a color, and then tell you its corresponding RGB encoding.

Once we can encode a single color numerically, an image is encoded as a 2D grid of colors. Each “dot” in this grid is called a pixel. As with strings, understanding how to store a 2D sequence requires an understanding of pointers, which will come later. However, for now, it suffices to understand that an image can be encoded as many numbers organized in a logical format.

You may have noticed that computers typically have a variety of image formats, such as JPG, BMP, PNG, TIFF, and many others. Each of these encodes the image numerically, however, the specific details differ between the formats. Some image formats compress the image data—performing math on the colors (after all, the colors are just numbers!) to encode the image data in fewer bits, reducing the size of the data that must be stored on disk and/or transferred across the Internet.