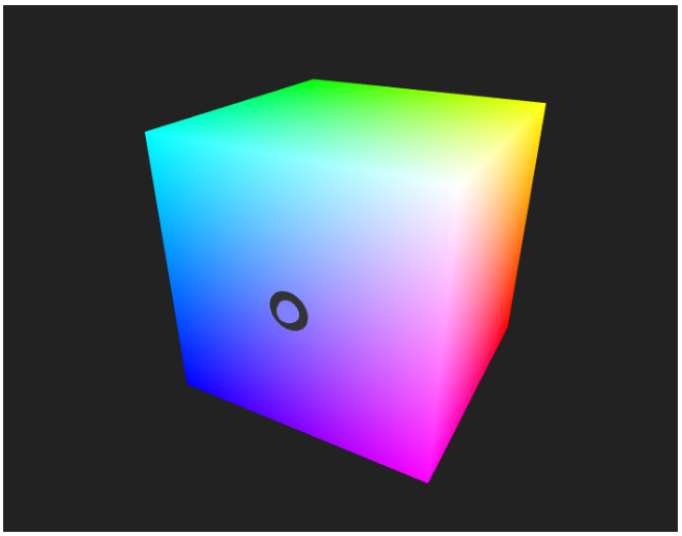
Color_Model

See more : https://github.com/KongphopTongdee/Working_At_Yanix/tree/main/What_you_have_learned

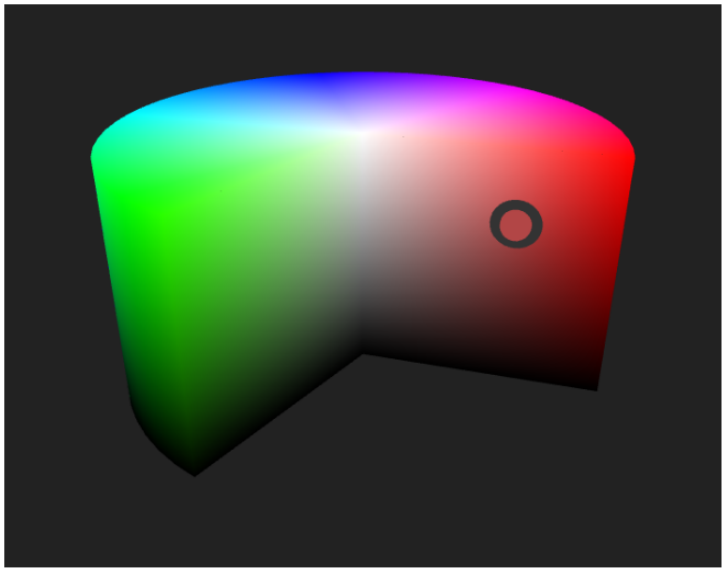
1. Color Models : A color model is a visualization that depicts the color spectrum as a multidimensional model. In the following, we will look at the RGB, HSV, and HSL color models, which are all prevalent in current digital design tools and programming languages.
    - RGB : is a color model with three dimensions – red, green, and blue – that are mixed to produce a specific color. The RGB color model is often depicted as a cube by mapping the red, green, and blue dimensions onto the x, y, and z axis in 3D space.



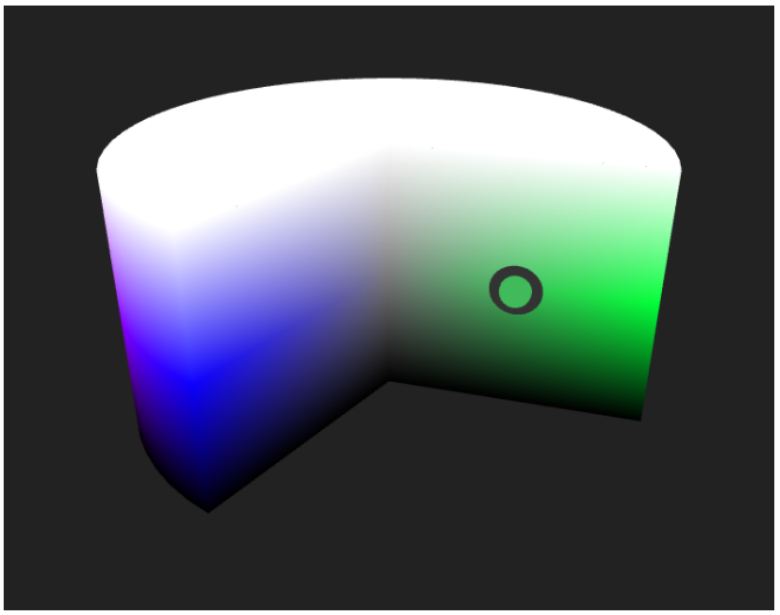
The RGB color model is not an especially intuitive model for creating colors in code. Because you might not be able to guess the combination of values to use for some colors.

    - HSV : is a cylindrical color model that remaps the RGB primary colors into dimensions that are easier for humans to understand. Like the Munsell Color System, these dimensions are hue, saturation, and value.
        - Hue : specifies the angle of the color on the RGB color circle. A 0° hue results in red, 120° results in green, and 240° results in blue.
        - Saturation : controls the amount of color used. A color with 100% saturation will be the purest color possible, while 0% saturation yields grayscale.
        - Value : controls the brightness of the color. A color with 0% brightness is pure black while a color with 100% brightness has no black mixed into the color. Because this dimension is often referred to as brightness, the HSV color model is sometimes called HSB, including in P5.js.

It is important to note that the three dimensions of the HSV color model are interdependent.

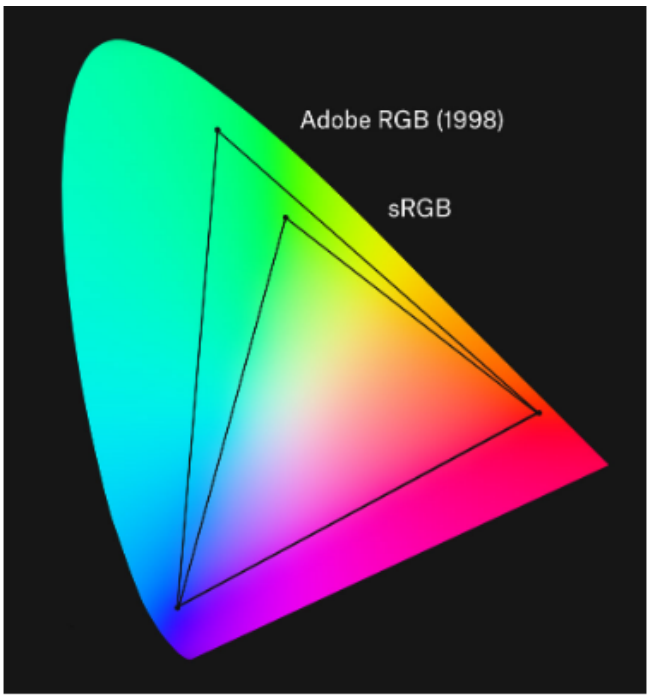


- HSL : is another cylindrical color model that shares two dimensions with HSV, while replacing the value dimension with a lightness dimension.
    - Hue : specifies the angle of the color on the RGB color circle, exactly like HSV.
    - Saturation : controls the purity of the color, exactly like HSV.
    - Lightness : controls the luminosity of the color. This dimension is different from the HSV value dimension in that the purest color is positioned midway between black and white ends of the scale. A color with 0% lightness is black, 50% is the purest color possible, and 100% is white.



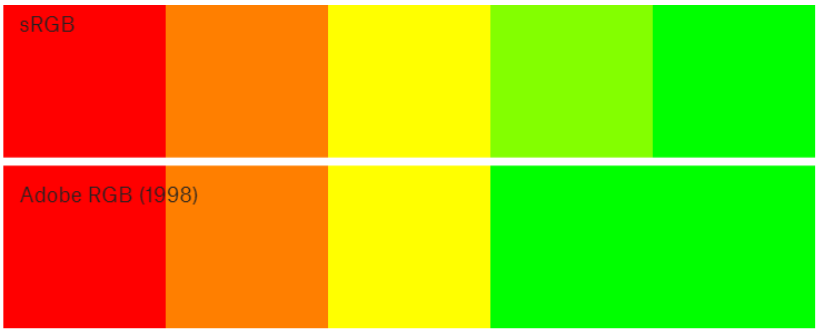
There are plenty of other ways to visualize the color spectrum in a multi-dimensional space. The CMYK color model has four dimensions, which means that one has to use either animation or multiple 3D shapes to visualize the states of the model. Another color model called CIELAB is modeled on the opponent-process theory of human perception with two of three dimensions representing scales from red to green and yellow to blue – two opponent color pairs that humans cannot perceive simultaneously.

2. Color spaces : Color spaces refers to a specific way of representing colors using numerical values. It defines the range of colors that can be displayed or printed, as well as the mathematical rules for converting between color representations. Color models are inadequate when it comes to defining and displaying colors on computer screens. This chromaticity diagram was created by the International Commission on Illumination (CIE). It was based on a number of vision experiments on human subjects in the 1930's, and it accurately defines the relationship between the wavelength of a color and the perceived effect on the human eye. This diagram – which is also a color space called CIEXYZ – is very important as all modern color spaces define their absolute range of colors (called a color gamut) in relation to this color space. The two triangles inside the curved shape indicate the color gamuts of two popular color spaces: sRGB and Adobe RGB (1998). The corners of each triangle define the primary colors of each color gamut, and you might notice that while the two color spaces share the same red and blue primaries, the green primary color is different between the two.



The CIE chromaticity diagram showing the color gamuts of the Adobe RGB (1998) and sRGB color spaces.

The sRGB color space has the smallest color gamut of the two color spaces, which means that it covers the smallest range of colors. It was created for use by computer monitors, and the smaller gamut reflects the exact colors of the primary lights in most HDTVs and computer monitors. This also means that the sRGB color space is easy to adapt for hardware manufacturers, which is why it has become the most widely used color space for digital files. Whenever you come across a color or an image on a website, it is most likely an sRGB color. Even though sRGB is a great color space for the range of colors that can be shown on a screen, the color gamut is not wide enough to support colors printed in ink – especially in the green-blue parts of the spectrum.



A simulation of how most monitors render the two color spaces. The last two colors in the Adobe RGB (1998) color space will look identical.

It is important to note that although color models are abstract mathematical concepts, it is impossible to visualize a color model without an accompanying color space. The RGB, HSV, and HSL color model examples from above are all visualized within the sRGB color space, because that is the default color space of the internet.

3. Color profiles : A digital image can adhere to a specific color space by embedding a color profile in its metadata. The program read the image pixel values that are stated according to a particular color space, and images without a color profile are often assumed to be sRGB. Color profiles are important in order to correctly reproduce identical colors across multiple devices. As an example, if you paste an image with an Adobe RGB (1998) profile into a Photoshop file with a sRGB profile, Photoshop will interpret the pixel values to be within the smaller color gamut, changing the colors of your pasted image. Because of this, most digital design tools have built-in commands to convert between color spaces, and Photoshop actually does a good job of alerting the user before reinterpreting color profiles.

The left-hand side of this Paul Klee painting was correctly converted from Adobe RGB (1998) to sRGB, while the right-hand side wrongly reinterpreted the colors into sRGB without conversion.

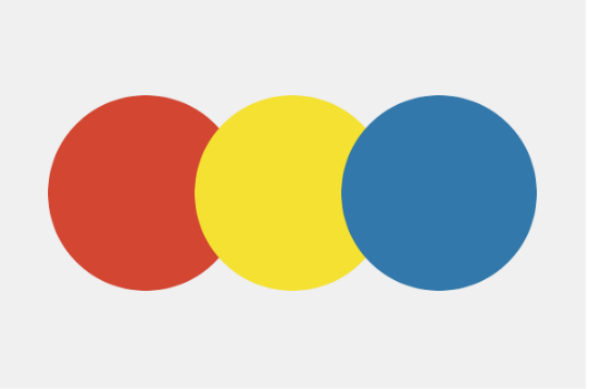
If a digital image uses a color profile with a wide color gamut, it is almost guaranteed to lose colors on most screens because most screens can only show colors within the sRGB gamut.

4. Color in P5.js : As a browser-based JavaScript library, all color values in P5.js adhere to sRGB, the standard color space for the internet. You can define these colors in all three of the aforementioned color models: RGB, HSV (called HSB), and HSL. The default color model in P5.js is RGB, which means that the fill() and stroke() functions expect three numbers between 0 and 255, indicating the amount of red, green, and blue to use for the color. The reason behind this specific range is that a maximum of 256 values can be stored in a single byte (8 bits), allowing each RGB color to only take up 24 bits. Even though 256 different amounts of red, green, and blue might not sound like much, this can produce 16,777,216 distinct colors which is actually much more than what the human eye can perceive.

```
noStroke();
fill(210, 70, 50);
ellipse(150, height/2, 200, 200);

fill(245, 225, 50);
ellipse(300, height/2, 200, 200);

fill(50, 120, 170);
ellipse(450, height/2, 200, 200);
```

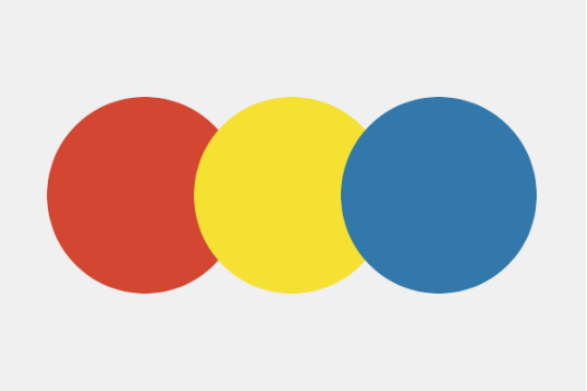


P5.js also allows you to use an alternative hexadecimal syntax known from web design for specifying colors in the RGB color model.

```
noStroke();
fill("#d24632");
ellipse(150, height/2, 200, 200);

fill("#f5e132");
ellipse(300, height/2, 200, 200);

fill("#3278aa");
ellipse(450, height/2, 200, 200);
```

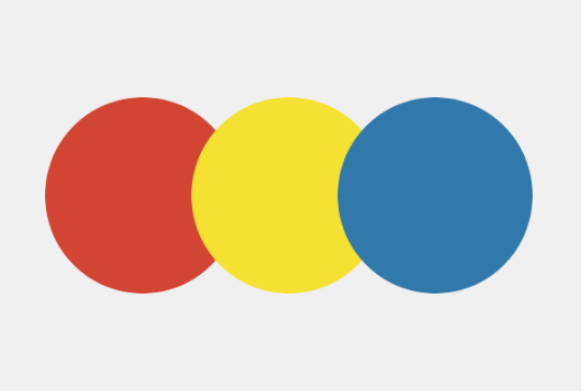


The colorMode() function in P5.js can be used to switch to another color model.

```
noStroke();
colorMode(HSB, 1, 1, 1);
fill(0.0195, 0.76, 0.82);
ellipse(150, height/2, 200, 200);

fill(0.15, 0.80, 0.96);
ellipse(300, height/2, 200, 200);

fill(0.569, 0.71, 0.67);
ellipse(450, height/2, 200, 200);
```



Reference(https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html,
https://www.geeksforgeeks.org/computer-graphics-the-rgb-color-model/)