

Brief introduction to color in VPython

MIXING COLORS

Color on television screens and computer monitors is created by mixing different amounts of the 3 primary additive colors: red, green, and blue. Red is simply made by making the red pixels on the screen bright, while keeping the green and blue dim. Similarly, it is easy to see how green and blue are made.

If you look at bright yellow on a monitor very closely (with a magnifying glass, e.g.), you will see little red and green pixels fully lit. Similarly, magenta is blue and red and cyan is a mixture of green and blue.

More subtle shades are made from other mixtures of red (R), green (G), and blue (B). For example, mixing maximum intensity red and *half* maximum intensity green and zero intensity blue gives a orange color.

VPython

The visual library allows you to control the color of an object in several ways. These methods are quite similar to controlling the position of an object in 3D space. For example, you can set the **x** attribute of an object to move it along the *x*-axis direction. Or you can modify the whole three coordinates using the **pos** attribute. So to move a cube named **a** to position **x=0, y=1, z=0.5**, you could say

```
a.x = 0  
a.y = 1  
a.z = 0.5
```

or you could say

```
a.pos = (0, 1, 0.5)
```

where the order is always **x, y, z**.

The three axes of color we will use are **red, green** and **blue**. To make an object orange, you can say

```
a.red = 1  
a.green = 0.5  
a.blue = 0
```

or you could say

```
a.color = (1, 0.5, 0)
```

where the order is always R, G, B.

You will want to experiment with this, say in a shell. Just make a cube, then set its **red, green, blue,** and **color** attributes to different values, between 0 (dark) and 1 (maximum.)

Python provides some defined colors. **color.blue** is nothing but the three numbers (0,0,1.) (this is also called a tuplet.) **color.green** is (0,1,0) and **color.magenta** is (1,0,1). So when you say

```
a.color = color.magenta
```

this is equivalent to

```
a.color = (1, 0, 1)
```

For example, try entering these statements one at a time, in a shell:

```
from visual import *
a = box()
a.red = 0
a.green = 0
a.blue = 0
a.color = (1., 0.7, 0.3)
print color.blue
print color.magenta
print color.red
print color.yellow
```

You can make the color dynamic, same as the position. Here is a program to move a box in space from $x=0$ to $x=1$:

```
from visual import *
a = box(color=color.green)
for count in range(100):
    rate(0.1)
    a.x = count * 0.01
```

A simple modification (just replacing $a.x$ with $a.red$) will not move it, but instead change its color by mixing in more and more red:

```
from visual import *
a = box(color=color.green)
for count in range(100):
    rate(0.1)
    a.red = count * 0.01
```

You can use color to indicate parameter values. For example, you might indicate the temperature of a box using a parameter **temperature**.

```
from visual import *
a = box(color=color.red)
temperature = 1.
for count in range(100):
    rate(10)
    temperature = temperature * 0.99
    a.color = (temperature, 0, 1-temperature)
```

If the temperature is high, the color will be reddish. As **temperature** decreases, the red component will be small and the blue component will be high, giving the box a bluish color.