

## Use of Color in Graphics

- [Alyce](#)

[Use of Color in Graphics](#) | [Three Commands for Graphic Color](#) | [COLOR](#) | [BACKCOLOR](#) | [FILL](#) | [NAMED COLORS](#) | [BUTTONFACE](#) | [RGB](#) | [BONUS QUESTION!](#) | [CUSTOM COLORS](#) | [DEMO](#)

## Three Commands for Graphic Color

There are three commands that are used in graphicboxes and graphics windows to specify color. They are:

- color
- bgcolor
- fill

The color commands are not case sensitive, so color, Color, COLOR and CoLoR are all acceptable.

### COLOR

Color sets the foreground color. This is the color of the pen used to draw the borders of all drawn objects, like lines, circles and boxes. It is also the color of graphic text.

### BACKCOLOR

BackColor sets the color that will fill drawn objects. For instance, a "boxfilled" will be filled with the color set by the BACKCOLOR command. Backcolor is also the color that appears behind graphics text.

### FILL

Fill covers the graphics area - either the entire graphicbox or the entire workspace of a graphics window - with the color specified. Issuing a FILL command will cover over everything previously drawn on the graphics area.

### NAMED COLORS

There are sixteen named colors to use with the three COLOR commands:



Here is a tiny program that uses the color commands with named colors. It illustrates that text is backed by the BACKCOLOR and not by the FILL color. Run the program and notice that the part of the text that is over the cyan interior of the circle blends in with the circle, but the part of the text that is on the yellow-filled background still displays a cyan BACKCOLOR. (Tutorials on drawing objects and text will follow later in the series of graphics tutorials.)

```
nomainwin
open "Colors" for graphics_nsb_nf as #1
print #1, "down"
print #1, "color red"
print #1, "backcolor cyan"
print #1, "fill yellow"
print #1, "place 100 100"
print #1, "circlefilled 80"
print #1, "place 30 30"
print #1, "\Hello World"
wait
```

Running the program above will show you a window that contains in addition to the text, a circle bordered in red and filled with cyan, against a window that is filled with yellow. A single line of code has been added to make the program below. A FILL command is issued after the other graphics commands, but before the WAIT statment. Running the new version of the program produces a graphics window filled with blue, and nothing else! The FILL command covers all previous graphics.

```
nomainwin
open "Colors" for graphics_nsb_nf as #1
print #1, "down"
print #1, "color red"
print #1, "backcolor cyan"
print #1, "fill yellow"
print #1, "place 100 100"
print #1, "circlefilled 80"
print #1, "place 30 30"
```

```
print #1, "\Hello World"  
print #1, "fill blue"  
wait
```

## BUTTONFACE

There is one more named color. It is **BUTTONFACE**. This will match the system's buttonface color. In a standard Windows color scheme, this will be lightgray. In other color schemes, this color might be blue, or green, or very pale gray, or tan, or any color specified by the user in a custom color scheme. It is good to know this color so we can match our graphics to the user's system. We could fill a graphicbox with lightgray, which would match many user's color schemes, but it would look very wrong to the user who has an "all blue" color scheme on his system. If instead, we fill with **BUTTONFACE**, then the users with the standard color scheme will see lightgray, but the guy with an "all blue" color scheme will see a blue box that matches all the other colors on his system. Here is a modified demo that uses **BUTTONFACE**:

```
nomainwin  
open "Colors" for graphics_nsb_nf as #1  
print #1, "down"  
print #1, "color red"  
print #1, "backcolor cyan"  
print #1, "fill buttonface"  
print #1, "place 100 100"  
print #1, "circlefilled 80"  
wait
```

## RGB

RGB stands for Red, Green, Blue. Each color has a red component, a green component and a blue component. Each of these components is in the range of 0 - 255. If, for instance, the red component of a color is 255, then there is total saturation of red in the color. If instead it is 0, then there is no red in the color at all. Numbers in between indicate varying amounts of a color, with 128 being the halfway point.

You specify an RGB color differently than a named color. To specify a **RED FILL** with a named color:

```
print #1, "fill red"
```

An RGB color statement requires a number for red, one for green and one for blue.

```
print #graphics, "fill red(0-255) green(0-255) blue(0-255)"
```

To fill with red as an RGB color as it would actually appear in a program:

```
print #1, "fill 255 0 0 "
```

The statement above makes the color have a total saturation of red, and no green or blue. To create a "darkred" color, use a half saturation of red, like this:

```
print #1, "fill 128 0 0"
```

Yellow is a combination of red and green, so setting the red and green values both to 255 creates yellow. The following two statements are equivalent:

```
print #1, "fill 255 255 0"  
print #1, "fill yellow"
```

Cyan is a combination of green and blue with no red. Pink is a combination of red and blue with no green. (Some people refer to this color as magenta.) If the red, green and blue components are equal, the color will be a shade of gray, ranging from black to white:

```
print #1, "color 0 0 0"           'black  
print #1, "color 128 128 128"     'darkgray  
print #1, "color 192 192 192"     'lightgray  
print #1, "color 255 255 255"     'white
```

## BONUS QUESTION!

Can you work out the RGB values for all 16 named Liberty BASIC colors?

## CUSTOM COLORS

Custom colors are easily created, so feel free to experiment.

```
print #1, "fill 255 200 140"      'orange
```

```
print #1, "backcolor 95 240 160"    'mint green
print #1, "color 255 240 210"       'beige
```

## DEMO

Here is a little demo that shows many possible color combinations that can be created with RGB colors. It fills the screen with random RGB colors at one second intervals using the TIMER. The red, green and blue components are chosen randomly with Liberty BASIC's RND() function. Please look in the helpfile for more on the use of the RND() function. Remember that variables are placed outside of quote marks in graphics statements. The resulting values for red, green and blue are variables and are outside of the quote marks when used below, and blank spaces are preserved!

```
nomainwin
open "Random RGB Colors" for graphics_nsb_nf as #1
print #1, "trapclose [quit]"
print #1, "down"

timer 1000, [change]
wait

[quit]
timer 0
close #1:end

[change]
red = int(rnd(0)*255)+1
green = int(rnd(0)*255)+1
blue = int(rnd(0)*255)+1
print #1, "fill ";red;" ";green;" ";blue
wait
```

[Use of Color in Graphics](#) | [Three Commands for Graphic Color](#) | [COLOR](#) | [BACKCOLOR](#) | [FILL](#) | [NAMED COLORS](#) | [BUTTONFACE](#) | [RGB](#) | [BONUS QUESTION!](#) | [CUSTOM COLORS](#) | [DEMO](#)