



Unit 1: Getting Started with the TI-Innovator™ Hub

Skill Builder 2: Color and the Hub Project

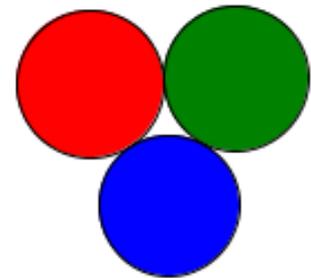
In this lesson, you will learn about controlling the color LED on the TI-Innovator Hub using a Python program template.

Objectives:

- Use a Python project template
- Control the color LED on the TI-Innovator Hub

The color LED has three color ‘channels’: red, green, and blue. This is often referred to as an “RGB LED”. Computer screens, phone screens and TV screens all use many of these LEDs to create color images.

To get a unique color, mix the correct amounts of red, green, and blue. Many colors are possible with the right mix of these three primary colors of light.

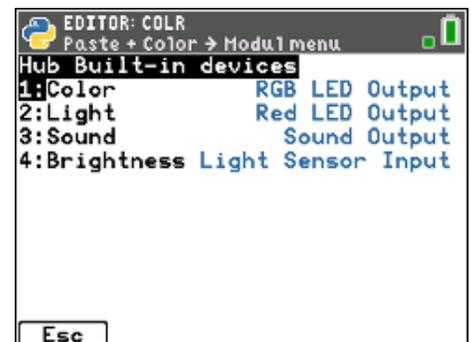
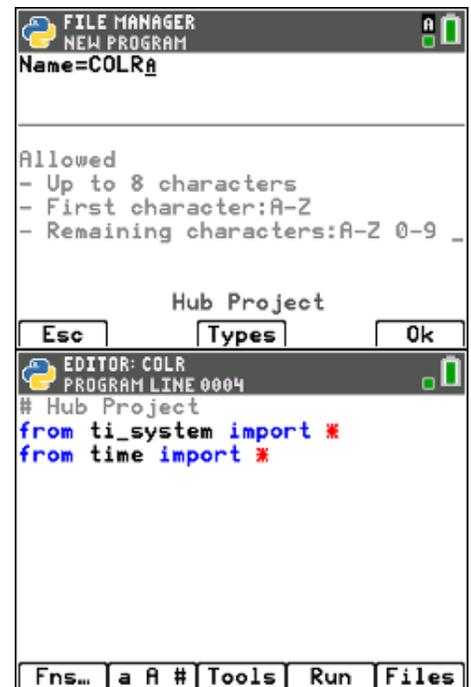


1. Begin this lesson with a new Python file. If you are not using the Python App, press **[prgm]** and select **Python App**. If you are using the Python App and are using the Editor or Shell, go to the File Manager by selecting the **<Files>** soft key.

Select **<New>** and first select the **<Types>** soft key. Select **Hub Project** from the list and name the program **COLR** (‘COLOR’ is already being used). Notice that the screen to the right now says ‘Hub Project’ right above the **<Types>** soft key. Press **<enter>** to begin the Editor.

2. The Hub Project template provides two **import** statements. These modules provide some functions like **sleep()** that will come in handy later. The top line of code is a **#comment** since it starts with **#** (number sign, hash, pound, or octothorpe). The entire line is colored light gray. **#comments** are ignored when the program is run and are a handy way to document your program and to help in debugging (removing errors). Press **[2nd] [3]** for the **#** symbol or find it on the **<a A #>** Character Map.

3. We need one more **import** statement. Place your cursor below the last import statement and press **[math] > ti_hub... > Hub Built-in devices...** and select **Color**.





10 Minutes of Code – Python

TI-84 PLUS CE PYTHON WITH THE TI-INNOVATOR™ HUB

UNIT 1: SKILL BUILDER 2

STUDENT ACTIVITY

4. This adds the statement **import color** to your code.

```

EDITOR: COLR
PROGRAM LINE 0005
# Hub Project
from ti_system import *
from time import *
import color

```

Fns... a A # Tools Run Files

5. With your cursor on a new line below **import color**, press **[math]** and select the new menu item at the bottom: **Color...**

There are three functions on the **Color** menu. The first one,

rgb(r, g, b)

establishes the color of the LED using the three channels: red, green, and blue. Each of these three arguments must be an *integer* in the range 0 to 255 as indicated on this screen. Values outside this range will produce an error when the program is run.

blink(,) and **off()** work like the corresponding **light** functions from the previous Skill Builder.

```

EDITOR: COLR
Color
1:rgb(r,g,b) 0-255
2:blink(freq,time)
3:off()

```

Esc

6. Enter three numbers, each between 0 and 255 in the parentheses. The commas are provided for you. We chose 255,100, 0.

<Run> the program.

Watch the TI-Innovator Hub and see your custom color appear.

```

EDITOR: COLR
PROGRAM LINE 0005
# Hub Project
from ti_system import *
from time import *
import color
color.rgb(255,100,0)

```

Fns... a A # Tools Run Files

If you encounter an error, make sure that the Hub is properly attached to your calculator (has a green power light) and that all the code is exactly correct. A misplaced comma, missing parenthesis, or misspelling can cause an error. To return to the Editor select **<Editor>** in the Shell. If all goes well, the color LED will remain lit and you can return to the Editor and try different r, g, and b values. Select **<Run>** again after editing.

7. Two other functions on the **[math] Color** menu are

blink(freq, time)

and

off()

Can you guess what they do? Try adding them to the program to test your guess. You can also incorporate the **sleep()** function (that's why the **time** module is imported) into your code as you did in Skill Builder 1.

Challenge: Try making a program to display various colors like the rainbow or your school colors.

```

EDITOR: COLR
Color
1:rgb(r,g,b) 0-255
2:blink(freq,time)
3:off()

```

Esc