



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

#### Skill Builder 2: Color and the Hub Project

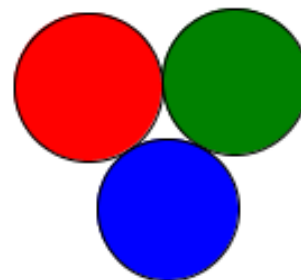
In this lesson, you will learn about controlling the color LED on the TI-Innovator Hub using a Python program template.

#### 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 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.



**Teacher Tip:** This lesson introduces the Python ‘Hub Project’ programming template. This Texas Instruments provided feature imports many different modules and special functions that are used in projects using the TI-Innovator Hub. This lesson does not address all these tools, but we will continue to incorporate this template in later TI-Innovator Hub lessons and address most of the features throughout the course.

If you are new to Python...

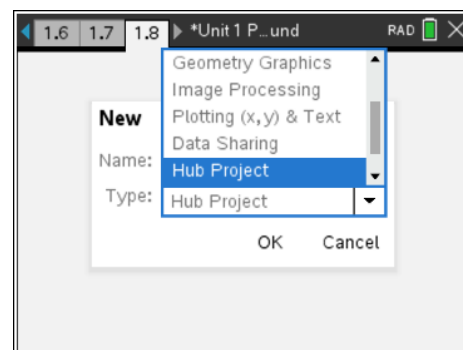
You will see three different types of import statements in the templates:

<b>from ti_system import *</b>	import everything from the ti_hub module
<b>import ti_plotlib as plt</b>	import everything from the ti_plotlib module <i>but</i> these functions must be preceded by the name plt.
<b>from time import sleep</b>	only import the sleep function from the time module.

1. Start a new TI-Nspire™ document using **home > New** or insert a new page in a document using **ctrl+doc** (+ page).

Select **Add Python > New....**

Type a name for the Python program (we used U1SB2). *Do not press enter yet!* From the **Type:** dropdown list (click the arrow at the right of the field), select **Hub Project**. Press **enter**.





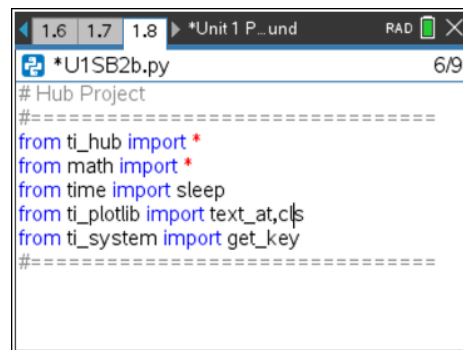
# 10 Minutes of Code - Python

## TI-NSPIRE™ CX II WITH THE TI-INNOVATOR™ HUB

### UNIT 1: SKILL BUILDER 2

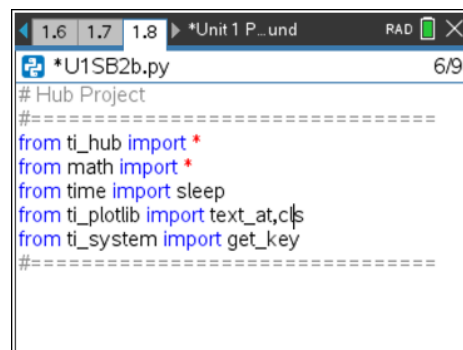
### TEACHER NOTES

2. The 'Hub Project' template provides several **import** statements at the top of the Python Editor. There are also some **#comment** statements. The **#** sign (number, hash, pound, octothorpe) indicates the start of a comment and these comments are ignored when running the program. Comments are used to add personal notes to a program.



```
1.6 1.7 1.8 ▶ *Unit 1 P...und RAD 6/9
*U1SB2b.py
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
```

3. The **import** statements provide several tools that will come in handy when working with the TI-Innovator Hub. We will not address all these tools in this lesson, but they will be used in later lessons. In the first lesson you used the **ti\_hub** module and the **time** module (for the **sleep()** function) by selecting those import statements from different menus. The template gathers several useful tools together for you. There are several other templates available for various types of Python programming projects on the TI-Nspire CX II.



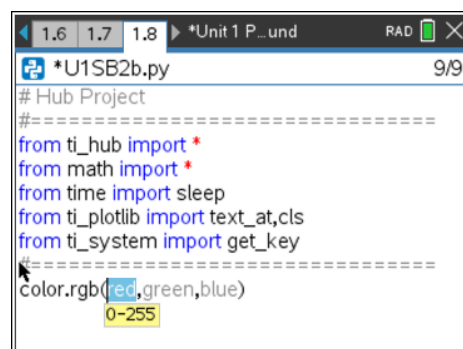
```
1.6 1.7 1.8 ▶ *Unit 1 P...und RAD 6/9
*U1SB2b.py
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
```

4. Place your cursor at the bottom of the program (below the bottommost **#=====** comment line).

Press **menu > TI Hub > Hub Built-in Devices > Color Output**. Select **rgb(red,green,blue)** which places the statement

**color.rgb(red, green, blue)**

into your program. The three 'inline prompts' (red, green, and blue) must be replaced with numbers, each between 0 and 255 as the hovering 'tool tip' indicates. Choose three numbers, pressing **tab** or the **right arrow** to go from field to field.



```
1.6 1.7 1.8 ▶ *Unit 1 P...und RAD 9/9
*U1SB2b.py
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
color.rgb(red,green,blue)
0-255
```

**Teacher Tip:** 'inline prompts' and 'hovering tool tips' appear when the statement is selected from the menus. They do not appear if the statement is typed in manually and will not return if the typed information is deleted.

How many colors are possible?  $256^{*3} = 256^3 = 16,777,216$

About **color.rgb()**: 'color' is an instance of a class and '.rgb()' is a method of that class.

Three arguments to the **.rgb()** function are *required*.



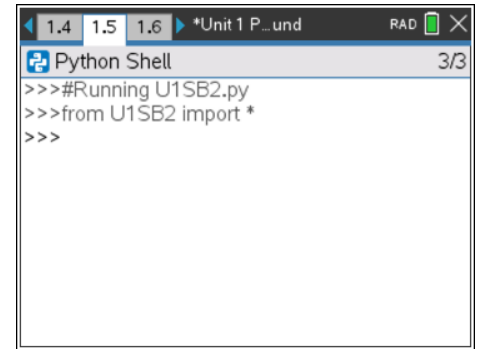
# 10 Minutes of Code - Python

TI-NSPIRE™ CX II WITH THE TI-INNOVATOR™ HUB

## UNIT 1: SKILL BUILDER 2

### TEACHER NOTES

5. Run your program. The Shell app appears on the screen and the color LED will light up on the TI-Innovator Hub. Now go back one page (**ctrl+left arrow**) in the document to the Python Editor and try other numbers for the three color channels.



```
< 1.4 1.5 1.6 ▶ *Unit 1 P...und RAD [X]
Python Shell 3/3
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>
```

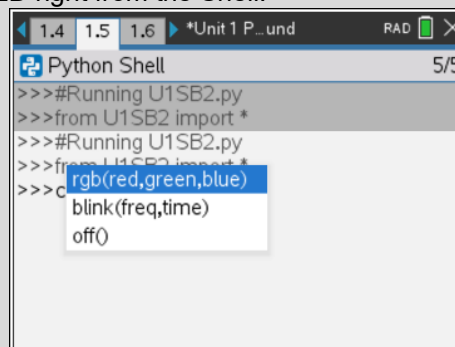
6. Two other functions on the **Color Output** menu are **blink(frequency, time)** and **off()**.

1 Color Output	1 rgb(red, green, blue)
2 Light Output	2 blink(frequency, time)
3 Sound Output	3 off()

Can you guess what they do? Try adding them to the program to test your guess.

Remember to save your work.

**Teacher Tip:** After running the program in the Shell, the Shell 'knows' the **color.rgb()** function. At the command prompt (>>>) you can type the word 'color.' (including the period) and the three methods appear in a selection list. Completing the commands and pressing **enter** controls the color LED right from the Shell:



```
< 1.4 1.5 1.6 ▶ *Unit 1 P...und RAD [X]
Python Shell 5/5
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>color.
  rgb(red, green, blue)
  blink(freq, time)
  off()
```