



# 10 Minutes of Code - Python

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

UNIT 1: SKILL BUILDER 1

STUDENT ACTIVITY

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

## Skill Builder 1: Light It Up

In this lesson, you will learn the basics of writing and running a Python program and using the 'light' (the red LED) on the TI-Innovator Hub.

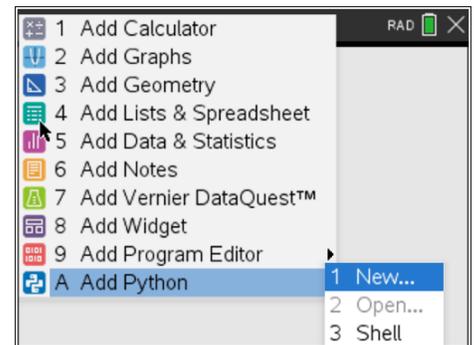
### Objectives:

- Create and run a Python program
- Control the light on the TI-Innovator Hub

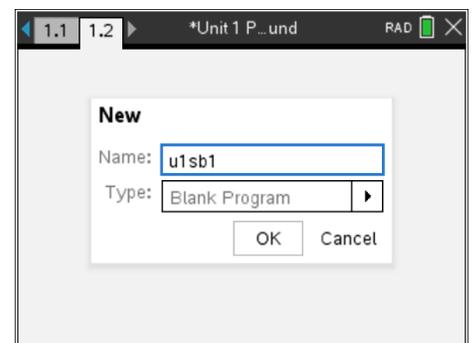
Welcome to the world of TI-Innovator Hub programming using Python on your TI-Nspire CX III! Your first program will operate the red LED on the TI-Innovator Hub circuit board. It is hard to see on the board, but when you turn it on you will know it!



1. Open a new TI-Nspire™ document. The available applications are listed. Select **Add Python** and then select **New...**



2. Type a name for the Python program (we use **u1sb1**) and press **enter**. We will look at the 'Type:' menu later.





# 10 Minutes of Code - Python

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

### UNIT 1: SKILL BUILDER 1

### STUDENT ACTIVITY

3. You are now in the Python Editor. Press the **menu** key. Each of these menu items contains related Python programming tools. Our main interest for now is the **TI Hub** menu. Select the top item from that menu:

**from ti\_hub import \***

This Python command gives you the tools (commands) needed to operate the devices on (or connected to) the TI-Innovator Hub.

Also, *this statement will check to see if a TI-Innovator Hub is connected. If not, then the program will not run.*

4. The next statement you will use is:

**light.on()**

Can you guess what it does?

This statement is found on

**menu > TI Hub > Hub Built-in Devices > Light Output > on().**

All Hub-related features are on the **TI Hub** menu.

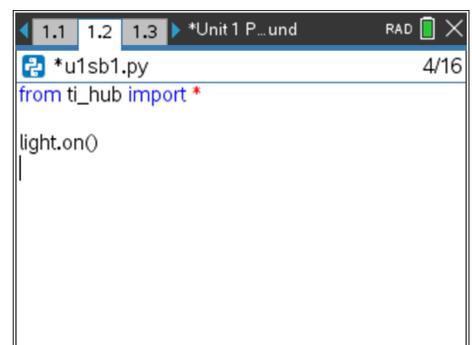
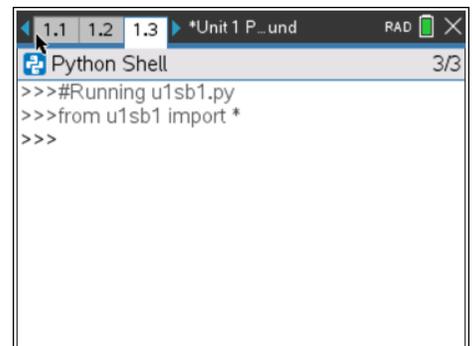
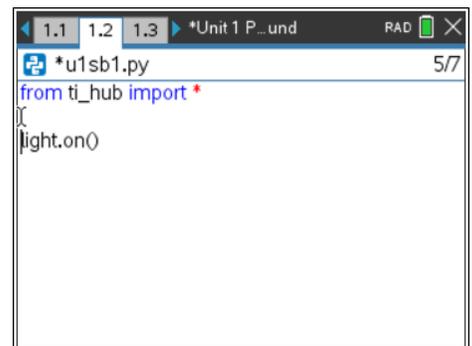
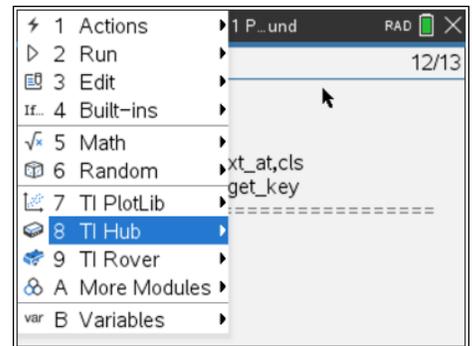
5. You are now ready to run this amazing program. You could use

**menu > Run > Run (Ctrl+R)**

or simply press **ctrl+R** on your keypad. Your TI-Nspire CX II screen will look like this one. Pressing **ctrl+R** added a page to your document and placed a Python Shell app on it. You are now using the Python Shell. The Shell is similar to the Calculator application on the TI-Nspire. This is the place where Python programs are executed (actually, the only place).

The **>>>** symbol is the Python command prompt. It is waiting for the next command. But if you look at the TI-Innovator Hub you will see a red LED lit up. That's the result of the **light.on()** statement in your program.

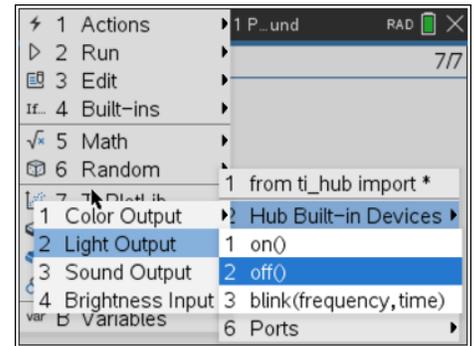
6. Your Python program is now on the page before this Shell page. To go back to editing your program, press **ctrl+left arrow**. Notice on the TI-Innovator Hub that the red LED is still on.





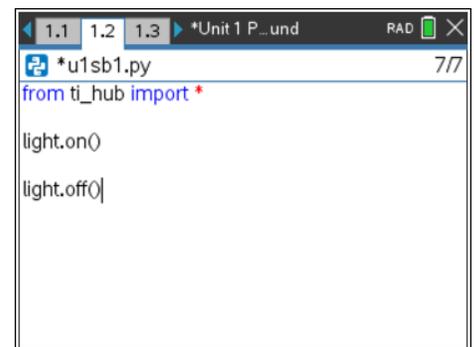
7. Can you guess what command turns the light off? You can find it on:

**menu > TI Hub > Hub Built-in Devices > Light Output > ...**



8. Add the **light.off()** command after the **light.on()** command. It is okay to skip lines in the Python Editor. They have no effect on the execution and they make the code easier to read. Run the program again. Do you see the LED blink quickly? Too fast?

In the next few steps, we add a feature that lets you control just how long the LED stays lit.



9. Place your cursor on the line below **from ti\_hub import \***.

Press **menu > More Modules > Time** and select:

**from time import \***

Between the **light.on()** and **light.off()** statements add the statement

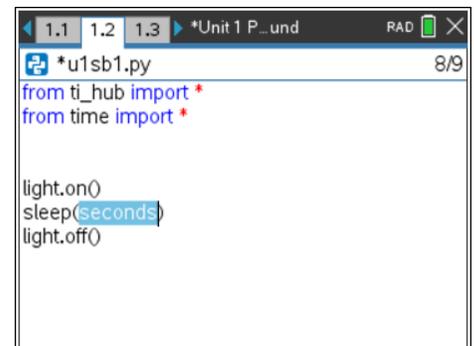
**sleep(seconds)**

found on the **Time** menu as well.

The word 'seconds' is a placeholder.

Replace it with a number, like 2 or 3.

The **sleep()** function tells the computer to wait or pause for that many seconds before going to the next statement in the program.



When you run the program now (press **ctrl+R**), the LED will stay lit for your chosen number of seconds before turning off.



## 10 Minutes of Code - Python

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

### UNIT 1: SKILL BUILDER 1

### STUDENT ACTIVITY

10. To make the light blink, you could repeat the sequence of statements in the program or....

on the Light Output menu there is also a **blink()** function that causes the LED to blink. **light.blink()** has two parameters: *frequency* and *time*. Replace both with numbers and figure out the pattern. Pay attention to the pop-up tool tip!



```
*Unit 1 P...und RAD 9/11
*u1sb1.py
from ti_hub import *
from time import *

light.on()
sleep(seconds)
light.off()

light.blink(frequency,time)
```