



Unit 3: brightness, if and while with the TI-Innovator™ Hub

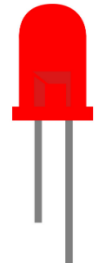
Skill Builder 2: Make a Light Switch

In this lesson, you will learn to use the brightness sensor to control the light (the red LED) automatically.

Objectives:

- Set up and monitor the brightness sensor
- Introduce the **if** statements
- Control the TI-Innovator Hub light (the red LED) using the brightness sensor.

Now that you can monitor the light coming into the TI-Innovator Hub, use that information to cause the onboard light (the red LED) to switch on/off when the brightness value changes.



Teacher Tip: Working with a copy of the program ensures that changes won't modify the original program.

1. Make a copy of the program you used in Skill Builder 1 of this Unit. To copy a program (from the Editor or Shell), select **<Files>** to go to the File Manager, use the arrow keys to point the selection arrow on the screen to your BRIGHTA program (shown) and select **<Manage>**. Choose **Replicate Program** and type the new program's name, BRIGHTB [enter].

Your new program appears in the Editor and will appear in the **<Files>** list later.



2. The **if** statements:

On the **<Files>** **Ctl** menu there are three **if...** statements shown at the right. Each version is used in different programming situations and all of them depend on one or more *logical conditions* (expressions that are either **True** or **False**):

if <condition>:	if <condition>:	if <condition1>:
block	block	block
	else:	elif <condition2>:
	block	block
		else:
		block

elif is Python's version of 'else if'.

Note that **if :** and **elif :** require a condition between the word and the colon. **else:** does not.

```
1:if ..
2:if .. else ..
3:if .. elif .. else
```



3. Immediately after reading the brightness in your program, insert the **if..else** structure from <Files> Ctl.

The <condition> placed after the word **if** will depend on the brightness variable **b**.

◆◆ **if b > 25:**
◆◆◆◆ **TRUE block**
◆◆ **else:**
◆◆◆◆ **FALSE block**

The 'greater than' operator (>) is found on the [test] menu ([2nd] [math]) along with all the other *relational and logical operators*. Be sure to leave the colon (:) at the end of the **if** statement.

The value 25 is only a sample value. You will change it to adapt to your lighting situation.

Note: **if** and **else** are indented two spaces because they are inside the **while** loop. But the **if :** and **else:** blocks are indented *another* two spaces. Indentation is the Python way of indicating these special blocks of code. See the <Tools> menu for **Indent** and **Dedent** (un-indent) options or just place spaces (or delete spaces) at the beginning of a line to control indentation. The gray diamonds are this Editor's method of indicating the indentation. Improper indentation can result in either a syntax error or a programmer error.

4. In order to control the light on the TI-Innovator Hub you must **import light**. Place this statement at the top of your code along with the other import statements. Find import light on <math> ti_hub... Built-in devices.

```
EDITOR: BRIGHTB
PROGRAM LINE 0014

brightns.range(0,255)

while not escape():
    b=brightns.measurement()
    if b>25:
        *****
    else:
        *****
    disp_at(6,"brightness= "+str(b),
            "left")

Fns... a A # Tools Run Files
```

```
EDITOR: BRIGHTB
PROGRAM LINE 0006

# Hub Project
from ti_system import *
from time import *
import brightns
import light

disp_clr()
disp_at(11,"Press esc to end","c
enter")
disp_cursor(0)

Fns... a A # Tools Run Files
```



5. Complete the two missing (**if** and **else**) blocks: One will turn the **light** (the red LED) on and the other will turn it off. Study the logic of the structure to decide which action goes in which block. In the screen to the right, we leave the answer up to you. There are ??? in the screen image for you to replace with the proper functions.

After entering the proper statements, **<Run>** the program. Change the brightness reaching the sensor. You may have to adjust the threshold value of **25** in the **if** statement to suit your lighting situation and your **brightns.range()** setting.



```
EDITOR: BRIGHTB
PROGRAM LINE 0019

brightns.range(0,255)

while not escape():
    b=brightns.measurement()
    if b>25:
        light.???
    else:
        light.???
    disp_at(6,"brightness= "+str(b), "left")

Fns... a A # Tools Run Files
```

Teacher Tip: When it is bright (if $b > 25$), turn the LED off by using **light.off()**. Else, when it is dark, turn the LED on by using **light.on()**.

It's also a good idea to add a **light.off()** statement at the end of the program (outside the while loop... un-indented) to make sure the light goes off when the program ends.