



Unit 2: for loops with the TI-Innovator™ Hub

Skill Builder 1: Looping the Light

In this lesson, you will be introduced to the concept of a **for** loop in the context of the TI-Innovator Hub

Objectives:

- Make a **for** loop using the **range()** function
- Input decimal values using **float()**
- Create a custom blinking light

To loop through a set of code a *specified number of times*, we can use a **for** loop with a **range()** function.

The **range()** function returns a sequence of numbers, starting from 0 and incrementing by 1 (by default), and ends at a specified number.

```
for index in range(size):
    ♦♦block
```

The function **light.blink()** does not give you enough control over the blink cycle. How long is the light on and how long is it off between blinks? For example, can you have it blink 3 times in 10 seconds?

This lesson develops a program that gives you this control and more direct control over the total number of blinks.

Teacher Tip: A **for** loop in Python is also used for iterating over a sequence (that is, either a list, a tuple, a dictionary, a set, or a string). This will be demonstrated in a later lesson.

The next step demonstrates a coding example of ‘composition of functions’: **float(input(...))**

The **blink()** functions for light and color only let you control the blinks per second.

1. Start with a **<New>** Hub Project template. We named it BLINKER.

Import the **light** module using **[math] ti_hub... Hub Built-in devices...**

Write three **input** statements:

- One for the total number of **blinks**
- One for the **onTime** (the time that the LED is on during a blink)
- One for the **offTime** (the time between blinks)

You may want to have blink times that are not whole numbers. Rather than using the **int()** function, you can use **float()** for the **onTime** and **offTime**. This allows you to enter numbers with decimals. Rather than have separate statements for the input and the type conversion, the two actions can be combined into one statement:

```
blinks = int(input("how many blinks?"))
```

First get the **float()** function (**<Fns...> Type**) and then, inside the parentheses, paste the **input()** function (**<Fns...> I/O**). Note the two right parentheses at the end of this statement.

Tip: Since the three statements are similar, this is a good place to try **<Tools> Copy Line** and **<Tools> Paste Line Below**. Then edit each line to suit.

```
EDITOR: BLINKER
PROGRAM LINE 0008
# Hub Project
from ti_system import *
from time import *
import light

blinks=int(input("how many blinks?"))
onTime=float(input("on time?"))
offTime=float(a
```



2. Insert this **for** statement:

```
for i in range(size):
```



found on <Fns...> Ctl. There are several different for loop options. Choose the first one.

The placeholder **size** is shown on the menu but is not pasted into your code.

Notice the two indentation spaces (the gray diamond symbols). The for loop 'block' is the set of statements that are all indented these two spaces (like the **while** loop in the previous lesson).

```
EDITOR: BLINKER
PROGRAM LINE 0009
from time import *
import light

blinks=int(input("how many blinks?"))
onTime=float(input("on time?"))
offTime=float(input("off time?"))
)
for i in range(size):
**
```

Teacher Tip: The menu screen shows the word 'size' inside the **range(size)** function, but this word is not pasted into the code. The variable **i** is used by default but the programmer can change it to any valid variable name.

3. Type the variable **blinks** in the parentheses of the **range()** function:

```
for i in range( blinks ):
```

Move your cursor to the next line using the [downarrow] key. Do not press [enter]. Be sure that the **for** statement still contains the colon (:) at the end of the line.

Important Tip: if your program gets stuck in a long or 'infinite' loop, you can 'break' the program by pressing the [on] key.

```
EDITOR: BLINKER
PROGRAM LINE 0010
from time import *
import light

blinks=int(input("how many blinks?"))
onTime=float(input("on time?"))
offTime=float(input("off time?"))
)
for i in range(blinks):
**
```

4. The **for** loop 'block' is the set of statements that operate the **light** (the red LED on the TI-Innovator Hub). Use **light.on()**, **light.off()** and **sleep()** statements. Try it yourself now. Remember to use the variables for timing that you used in the input statements. Be sure that each statement is indented two spaces (provided for you by the Editor).

You can use decimal values for all three **input** statements, but the number of blinks will be converted to an integer. There's no such thing as 1/2 of a blink!

The next step shows the completed program.





5. Your program should resemble this:
- ```
blinks = int(input("how many blinks? "))
onTime = float(...
offTime = float(...
for i in range(blinks):
♦♦ light.on()
♦♦ sleep(onTime)
♦♦ light.off()
♦♦ sleep(offTime)
```

```
EDITOR: BLINKER
PROGRAM LINE 0014
blinks=int(input("how many blink
s?"))
onTime=float(input("on time?"))
offTime=float(input("off time?"
))
for i in range(blinks):
♦♦light.on()
♦♦sleep(onTime)
♦♦light.off()
♦♦sleep(offTime)
♦♦
♦♦
Fns... | a A # | Tools | Run | Files
```

6. <Run> the program and enter your values and then watch the red LED on the TI-Innovator Hub. When the program ends, is the light on or off?

The LED blinks but nothing happens on the calculator screen. Add a **print** statement *inside* the **for** loop to *display* the current blink number:

**print(i)**

**print()** is found on <Fns...> I/O. Type the variable **i** inside the parentheses.

Running the program now shows the value of the index variable **i** on the Shell screen as the program makes the LED blink to your specifications. Do you notice anything unusual about the numbers on the screen?

```
EDITOR: BLINKER
PROGRAM LINE 0010
blinks=int(input("how many blink
s?"))
onTime=float(input("on time?"))
offTime=float(input("off time?"
))
for i in range(blinks):
♦♦print(i)
♦♦light.on()
♦♦sleep(onTime)
♦♦light.off()
♦♦sleep(offTime)
♦♦
♦♦
Fns... | a A # | Tools | Run | Files
```

**Teacher Tip: print()** is a useful debugging tool, too.

What's unusual? Python loops for using **range(size)** start with 0 and end with (size-1). So, with **range(5)** the five numbers processed are 0, 1, 2, 3, and 4.

```
PYTHON SHELL
>>> # Running BLINKER
>>> from BLINKER import *
how many blinks?5
on time?2
off time?1
0
1
2
3
4
>>> |
Fns... | a A # | Tools | Editor | Files
```

Python **for** loops using **range(start,stop)** start with the start value, and stop with (stop - 1).



7. Instead of the standard Python `print()` function you could use a special Texas Instruments developed `disp_at()` function that the Hub Project template imports.

From **[math] > ti\_hub... > Commands**, select  
**`disp_at( row , "text" , "align" )`**

You will be prompted for the align value (“left”, “center”, or “right”). Choose “center”. You can always change it later.

Use row **9**.

In place of “text” (the second argument), use the expression `str(i)` without quotes which is found on **<Fns... > Type** and converts the integer value of `i` into a string. You must supply the variable `i` inside the parentheses.

Turn the `print(i)` statement into a comment by typing `#` in front of it using **[2<sup>nd</sup>] [3]**.

Your code should resemble this:

```
print(i)
disp_at(9, str(i), "center")
```

8. **<Run>** the program again. You should now see the numbers displayed near the bottom center of the screen, all in the same position. Other `disp_` functions are available on the **ti\_hub > Commands** menu to make your screen prettier. Use `disp_clr()` to clear the screen before and after the loop.

*What values do you need to enter to get the light to blink 3 times in 10 seconds? This challenge is left as an exercise.*

Tip: There is another version of `disp_at`:

**`disp_at( row, column, "text" )`**

*There are 11 rows numbered 1..11 and 32 columns numbered 1..32.*

```
EDITOR: BLINKER
PROGRAM LINE 0012
#import light

blinks=int(input("how many blink
s?"))
onTime=float(input("on time?"))
offTime=float(input("off time?")
)
for i in range(blinks):
print(i)
disp_at(9,str(i),"center")
--
_
Fns... | a A # | Tools | Run | Files
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running BLINKER
>>> from BLINKER import *
how many blinks?5
on time?1
off time?1

4>>> |
Fns... | a A # | Tools | Editor | Files
```