



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 by default, and increments by 1 (by default), and ends at a specified number.

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? Can you have it blink 3 times in 10 seconds?

This lesson will develop a program that gives you this control and also 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.

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

1. Start with a New Python Hub Project template.

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)

Caution: You may want to have blink times that are not whole numbers.

Rather than using **int()** around the **input()** function, you can use **float()**. This allows you to enter numbers with decimals.

float() and **int()** are found on **menu > Built-ins > Type** (along with others).

```
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at_cls
from ti_system import get_key

blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
```

2. Insert the statement

for index in range(size):
block

selected from the **menu > Built-ins > Control**.

The inline prompts 'index', 'size', and 'block' must be replaced next.

```
from time import sleep
from ti_plotlib import text_at_cls
from ti_system import get_key

blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for index in range(size):
    ♦♦block
```



Teacher Tip: 'index' can be any valid variable. 'size' must be an integer greater than 0. 'block' is the block of code that is processed in each iteration of the loop. Each statement in the block is indented.

3. Replace 'index' with any variable (*i* is commonly used). Press the **tab** key to move to the next field.

Replace 'size' with the variable **blinks** that was used in the first input statement. Press **tab** again to highlight 'block'.

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 13/27
*u2sb1.py
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
    *block
  
```

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.

You can use decimal values for all the input statements, but the number of blinks will be converted to an integer. You cannot make 1/2 of a blink!



The next step shows the completed program.

5. Your program should resemble this:

```

blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
    light.on()
    sleep(onTime)
    light.off()
    sleep(offTime)
  
```

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 17/31
*u2sb1.py
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
    *light.on()
    *sleep(onTime)
    *light.off()
    *sleep(offTime)
  
```

6. When you run the program, enter your values, and then watch the LED. When the program ends, is the light on or off?

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

print(i)

print() is found on **menu > Built-ins > I/O**.

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?

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 13/32
*u2sb1.py
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
    *print(i)
    *light.on()
    *sleep(onTime)
    *light.off()
    *sleep(offTime)
  
```

