

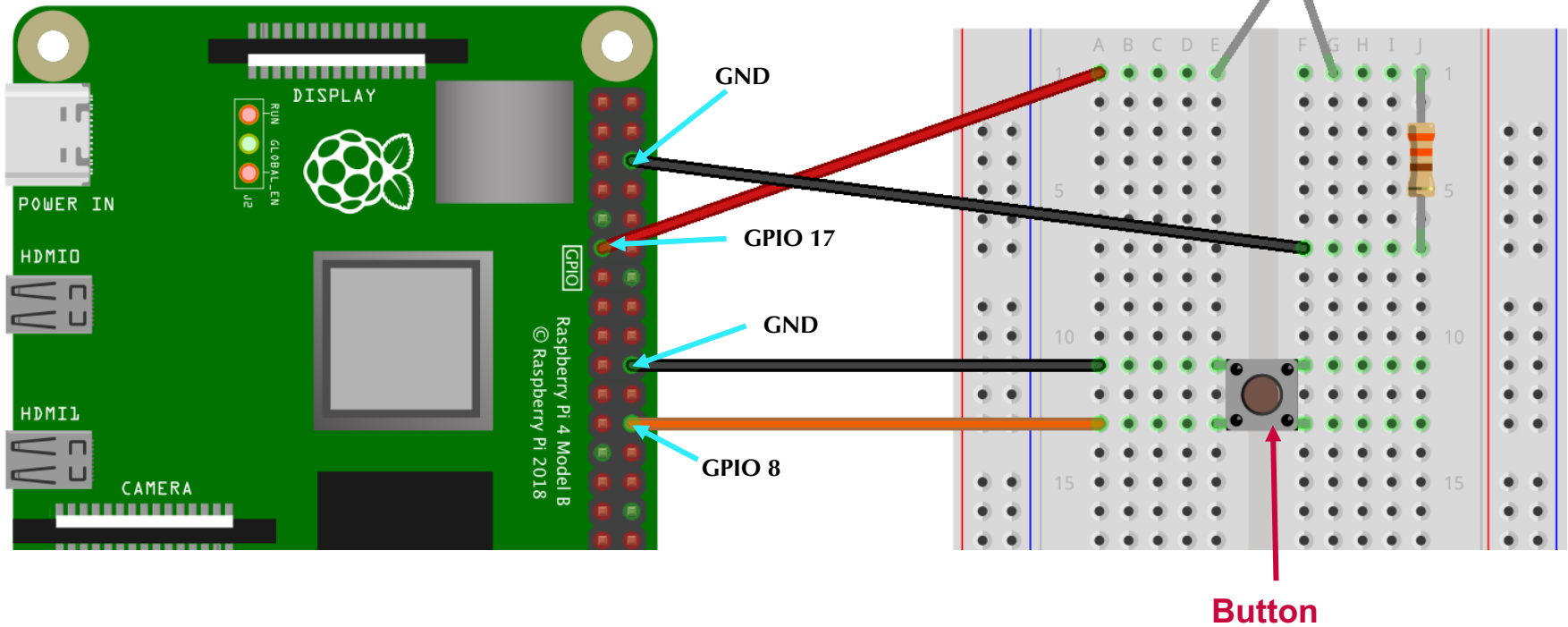
Part 3

Python: LED + Button

Adapted with permission from Raspberry Pi Picademy

LED + Button Circuit

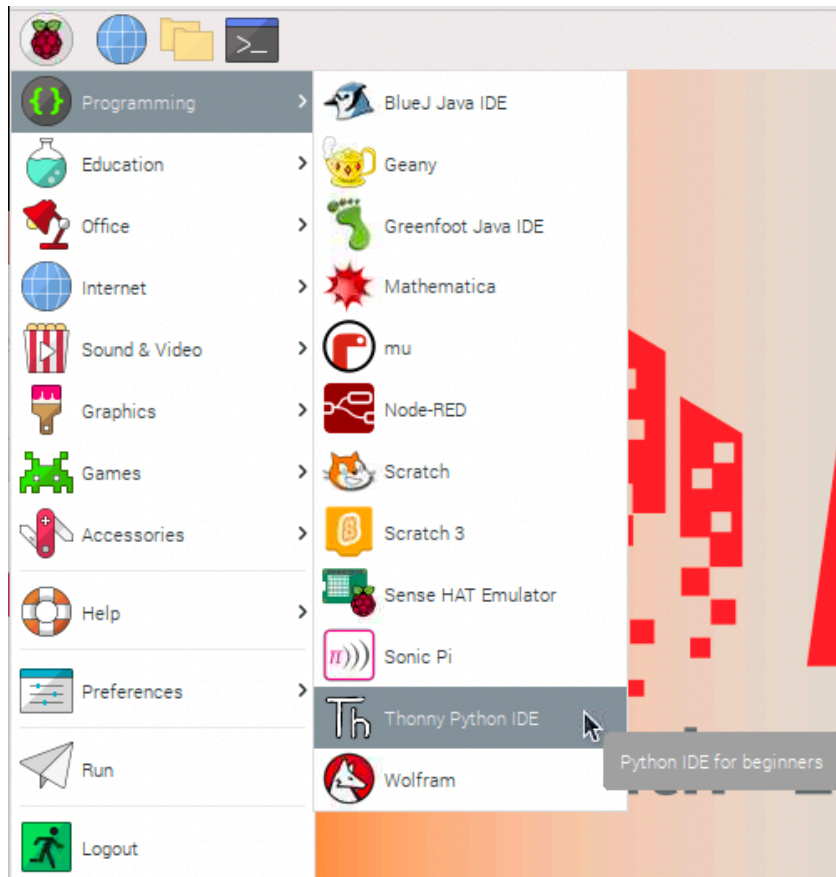
For this module, you will need to have this LED and button circuit that you built in Part 2.





Open Thonny Python

- Click on Raspberry Pi logo to show menu
- Select 'Programming'
- Select 'Thonny Python IDE'






Blink LED Once

Inside the top left space, write the following lines of code:

```
1 from gpiozero import LED
2 from time import sleep
3
4 myled = LED(17)
5
6 myled.on()
7 sleep(1)
8 myled.off()
```

To run your program, click on . You can also press F5 on your keyboard. If your program isn't already saved, you will be prompted to do so first before it runs.

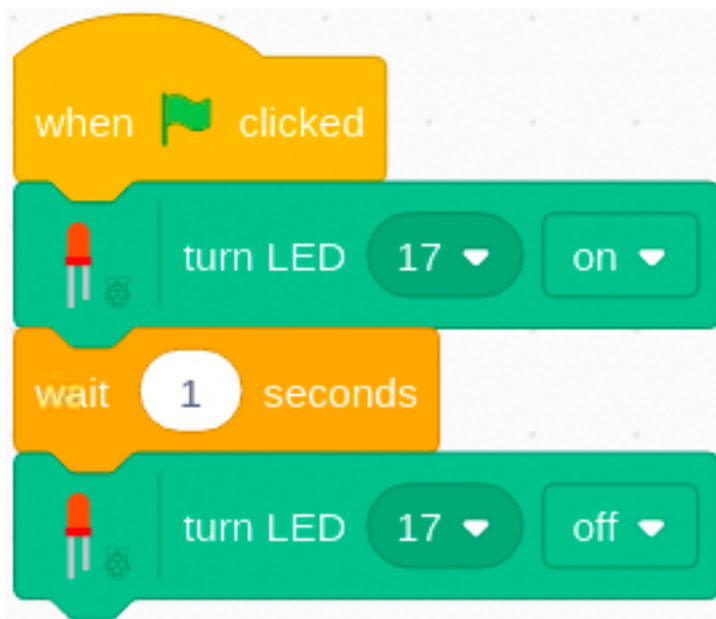
- Lines 1-2: In Python you always have to 'import' libraries of commands/functions that you will use in your code such as 'LED' and 'sleep'
- Line 4: create a variable, in this example it's 'myled' and assign it to the GPIO where your LED is connected. In this example it's GPIO17.
- Lines 6-8: .on() and .off() are used to output on and off signals where 'myled' is connected
- Line 7: 'sleep' in Python is the equivalent of a pause or wait and the number inside represents seconds



Scratch vs Python

LED On-Off

Block-based



Text-based

```
1 from gpiozero import LED
2 from time import sleep
3
4 myled = LED(17)
5
6 myled.on()
7 sleep(1)
8 myled.off()
```



Blink LED Continuously (Loop)

Save a new version of the previous code or click  to start a new file.

```
1 from gpiozero import LED
2 from time import sleep
3
4 myled = LED(17)
5
6 while True:
7     myled.on()
8     sleep(1)
9     myled.off()
10    sleep(1)
```

- Lines 6-10: to make a loop, use the 'while True:' text. The lines that are indented are part of this loop.
- Line 10: Why is this line necessary? What happens if you erase it and run the program again?

Click  or press F5 to Save and Run your program.



Blink Your LED - alternate

The LED library contains other commands that you can use to control the LED. Try them out by editing this code:

```
1 from gpiozero import LED
2
3 myled = LED(17)
4
5 myled.blink()
```

Challenge

- Can you determine what these commands do?
 - myled.blink(5)
 - myled.blink(2, 0.5)
 - myled.blink(0.1, 10)
 - myled.blink(0.5, 0.5, 5, False)

Other things to try

led.on() - Switches the Pin high

led.off() - Switches the Pin low

led.blink() - Makes the LED blink

led.toggle() - Change the LED state

led.pin.number - Gives the pin number

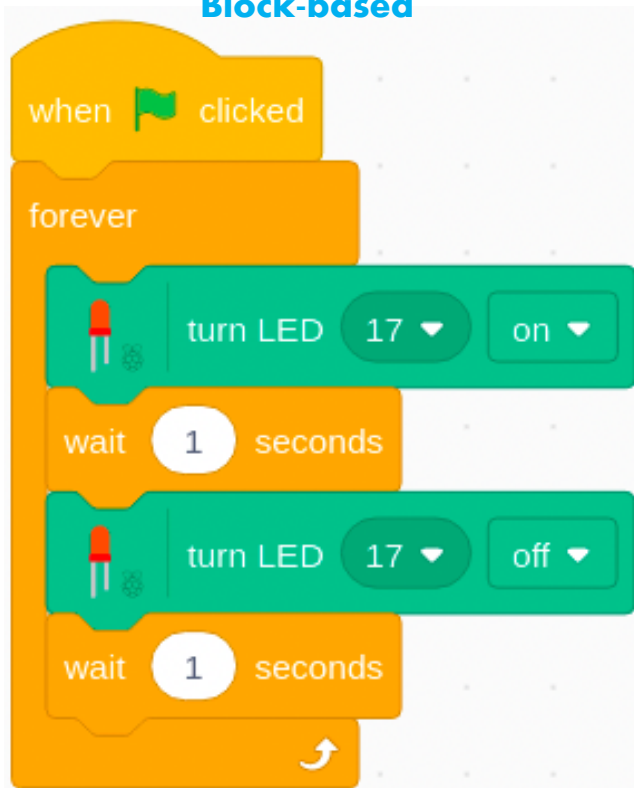
led.is_lit - Gives the current state



Scratch vs Python

Blinking LED

Block-based



Text-based

```
1 from gpiozero import LED
2 from time import sleep
3
4 myled = LED(17)
5
6 while True:
7     myled.on()
8     sleep(1)
9     myled.off()
10    sleep(1)
```




Button: Let's test it first...

Save a new version of the previous code or click  to start a new file.

```
1 from gpiozero import Button
2
3 button = Button(8)
4
5 button.wait_for_press()
6 print("The button was pressed.")
```

Click  or press F5 to Save and Run your program.



```
Shell X
Python 3.7.3 (/usr/bin/python3)
>>> %Run 'button test.py'
    The button was pressed.
>>>
```

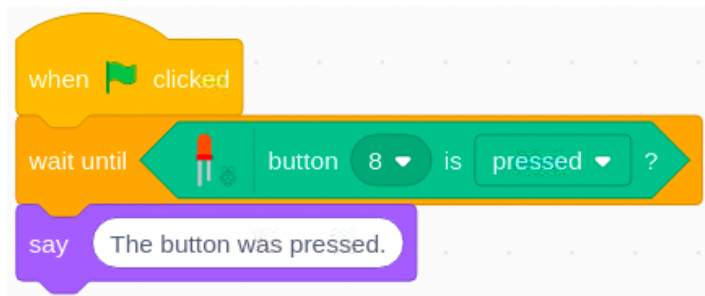
- Lines 1: first import a new element 'Button' from the gpiozero library.
- Line 4: create a variable, in this example it's 'button' and assign it to the GPIO where your Button is connected. In this example it's GPIO8.
- Lines 5-6: the program will wait until it detects that the button was pressed and print the statement in the Shell at bottom of the Thonny screen.



Scratch vs Python

Button Test

Block-based



Text-based

```
1 from gpiozero import Button
2
3 button = Button(8)
4
5 button.wait_for_press()
6 print("The button was pressed.")
```



Button: Combine with LED

Save a new version of the previous code or click  to start a new file.

```
1 from gpiozero import LED, Button
2 from time import sleep
3
4 blue = LED(17)
5 button = Button(8)
6
7 button.wait_for_press()
8 print("button pressed")
9 blue.on()
10 sleep(2)
11 blue.off()
```

- Lines 1: you can import multiple elements separated by a comma
- Line 4-5: 'blue' is the name of the LED at GPIO 17. 'button' is connected to GPIO8.
- Lines 7: the program will not continue until the button is pressed
- Line 8: print the statement in the Shell
- Lines 9-11: turns the blue LED on for two seconds and then turns it off

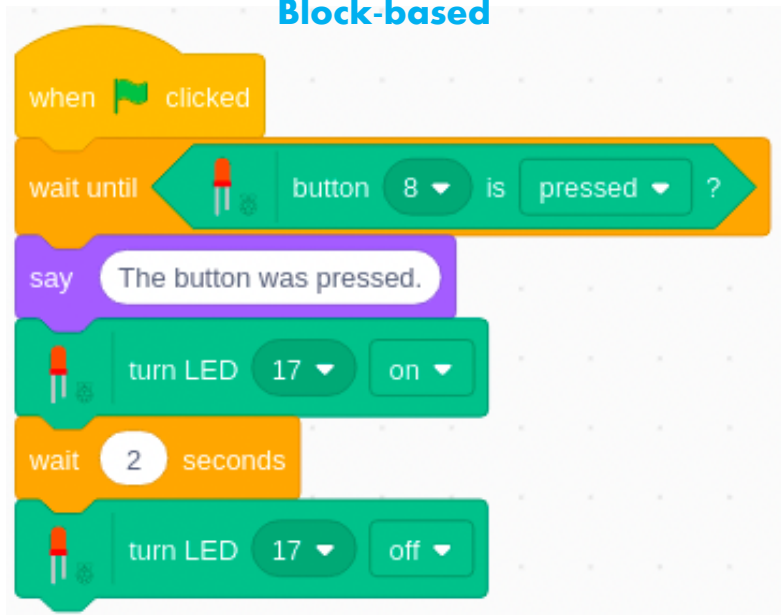
Click  or press F5 to Save and Run your program.



Scratch vs Python

Button + LED

Block-based



Text-based

```
1 from gpiozero import LED, Button
2 from time import sleep
3
4 blue = LED(17)
5 button = Button(8)
6
7 button.wait_for_press()
8 print("button pressed")
9 blue.on()
10 sleep(2)
11 blue.off()
```



Button: Make a light switch

Save a new version of the previous code or click  to start a new file.

```
1 from gpiozero import LED, Button
2
3 blue = LED(17)
4 button = Button(8)
5
6 while True:
7     blue.off()
8     button.wait_for_press()
9     blue.on()
```

Click  or press F5 to Save and Run your program.

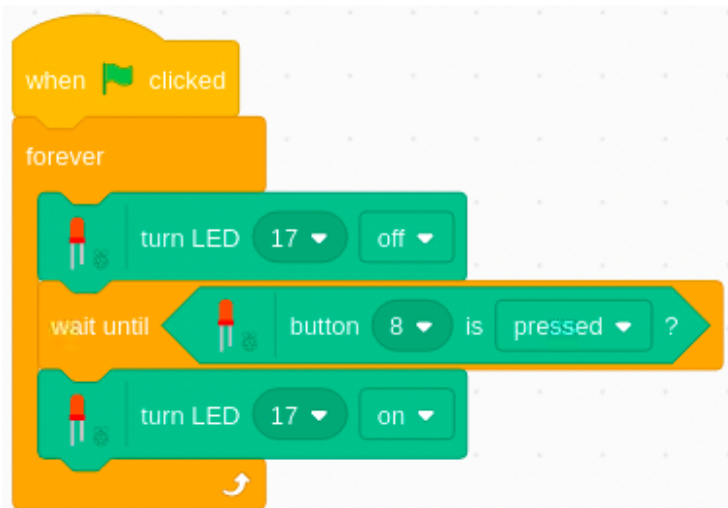
- Line 1: you can import multiple elements separated by a comma
- Lines 3-4: 'blue' is the name of the LED at GPIO 17. 'button' is connected to GPIO8.
- Line 6: sets up a continuous loop containing all the indented lines underneath it
- Line 7: the loop starts with setting the 'blue' LED to the off state
- Line 8: the loop waits for the button to be pressed before continuing
- Lines 10: only when the button is pressed will the 'blue' LED turn on and the loop will restart
- As long as the loop is running and the button is pressed, the LED will turn on, otherwise it will turn off



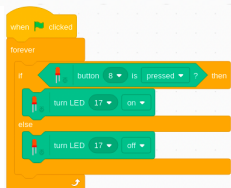
Scratch vs Python

Button Light Switch

Block-based



How is this different than the first Scratch program you wrote at the end of Part 2?



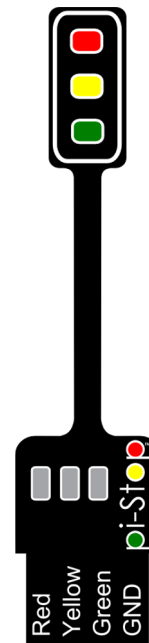
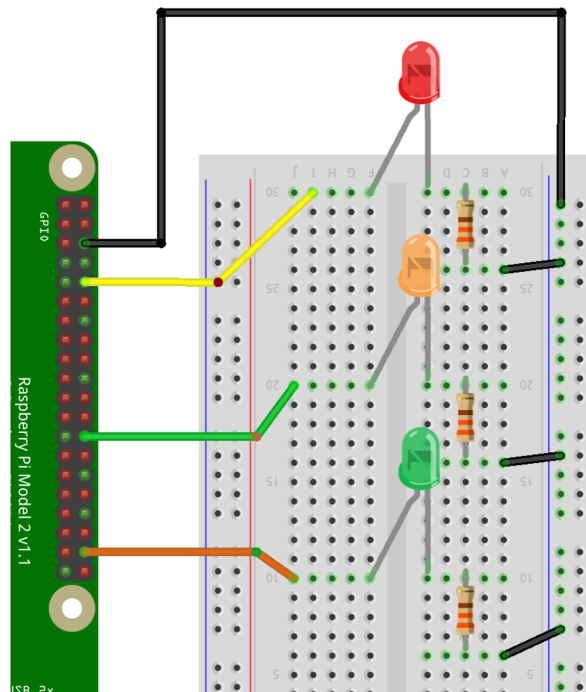
Text-based

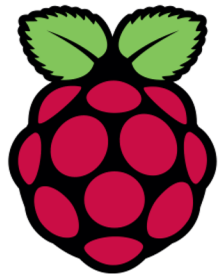
```
1 from gpiozero import LED, Button
2
3 blue = LED(17)
4 button = Button(8)
5
6 while True:
7     blue.off()
8     button.wait_for_press()
9     blue.on()
```



Ok, now what?

- Add more LEDs
- Add more Buttons
- Make a traffic light
- Use buttons to control multiple LEDs
- Just keep track of your jumper wires and the GPIO pin numbers
- What other projects could you do with buttons and LEDs?





END

PART 3