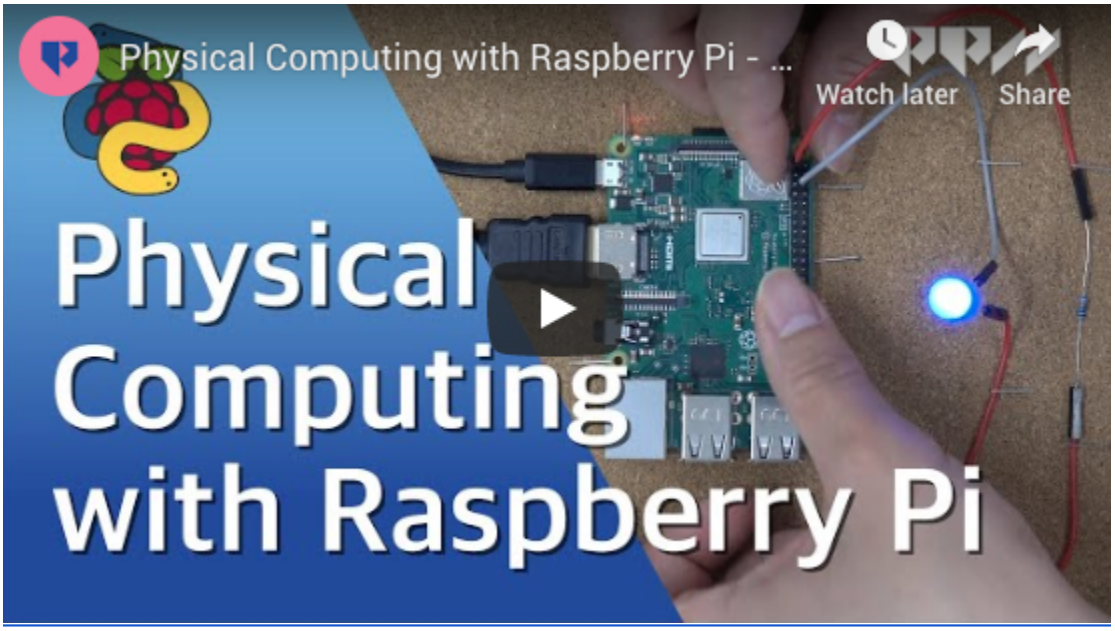
Physical Computing with Raspberry Pi



We have spent time working on Python code on the Pi and now it is time bring in physical objects into the mix. We are going to start using LEDs. Light Emitting Diodes (LED) are a simple way to bring physical computing to life. We will work to turn the light one and make it blink. Once you understand how to do this, an entire world of physical computing will be open to you.

Materials

- Raspberry Pi Model 3 B+
- 8GB Micro SD Card
- Micro USB Power Supply

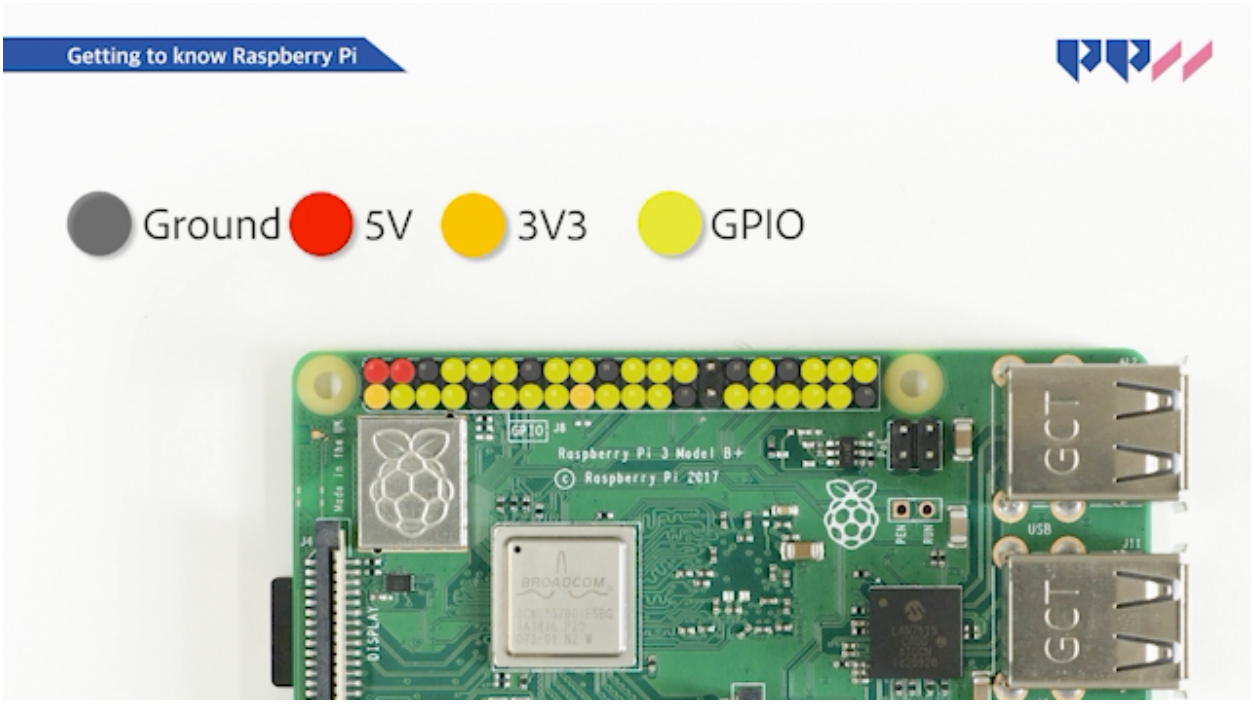
- Mouse
- HDMI Cord Keyboard
- HDMI Monitor Computer
- SD Card Reader Wifi Access
- LED 330ohm Resistor

Key Concepts

- GPIO
- Time
- Sleep

Raspberry Pi

The Raspberry Pi

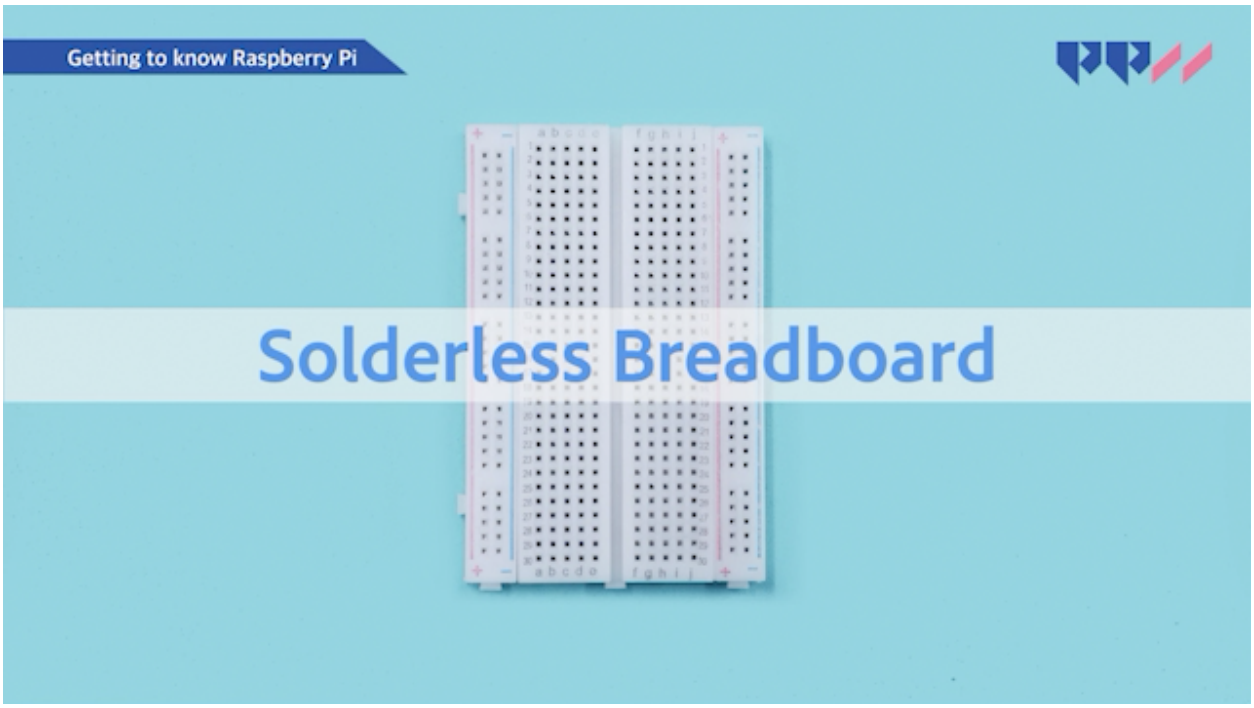


One of the best things about Raspberry Pi and Python is using the GPIO pins to create just about anything you want. GPIO (General Purpose Input Output) are the 40 male pins on the side of the Raspberry Pi. Each pin has a specific purpose that can be used

for ground, power, connecting to physical computing projects and more. Here is a look at the pins.

These pins serve many purposes and you want to make sure that you take good care of them. They can bend and break if you are not careful. Bent pins could make it difficult to add other components to the Pi later.

Solderless Breadboard



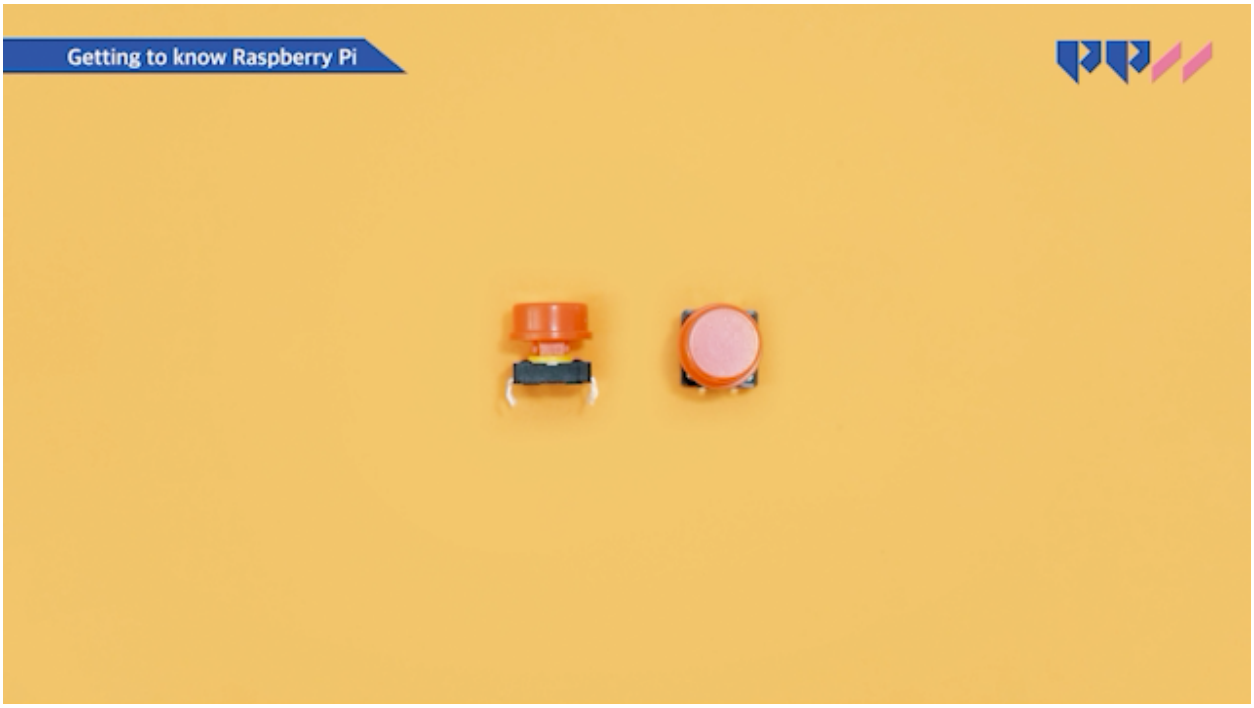
A breadboard allows makers to easily connect multiple components of a project without the need to solder. There are hidden metal tracks beneath the breadboard that connects all of the components together. You do not have to have a breadboard to build physical computing projects, but it really helps when you are prototyping ideas.

Jumper Wires

Female-Female Male-Male Male-Female

There are three types of jumper wires; male/male, female/female, and male/female. You will use the different wires for different parts of a project. You will need a female end to connect to Pi and a male end to use the breadboard. You would need to male/male to go from one spot on the breadboard to another. Have a collection of each type of jumper in different lengths is always helpful when working on physical computing projects.

Button Switch

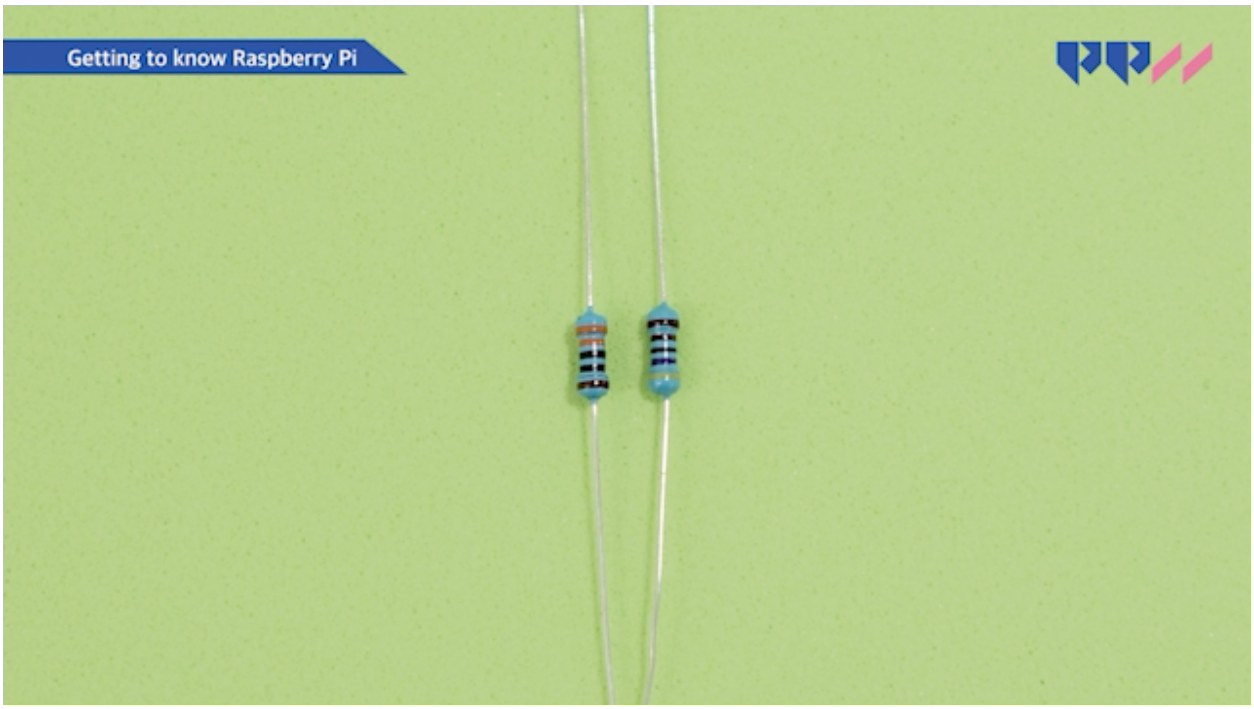


The button switch is a simple physical computing component that is an input device. You would tell the code to be on the lookout for the button press and then do something when it is pressed.

Light-Emitting Diode (LED)

The LED is an output device. It is controlled by the program you have created. You could code a program to turn the light on when the button is pushed. LEDs come in all shapes, sizes, and colors, but not all are good to use with Raspberry Pi. The LEDs we need to use have to be designed for 5 or 12 volts. You will also notice that one leg is longer than the other. The long leg is the **Positive** leg and the short leg is the **Negative** leg. You will need to know then when you are starting putting together your circuits.

Resistors



Resistors help control the flow of electrical current. They come in different values that are measured in ohms (Ω). The higher the number, the more resistance it will offer. You will most often use resistors around 330Ω for LEDs. If you do not use resistors with LEDs on your Pi, you risk damaging or burning out the LED completely. Here is how you can read resistors and their values.

Buzzer

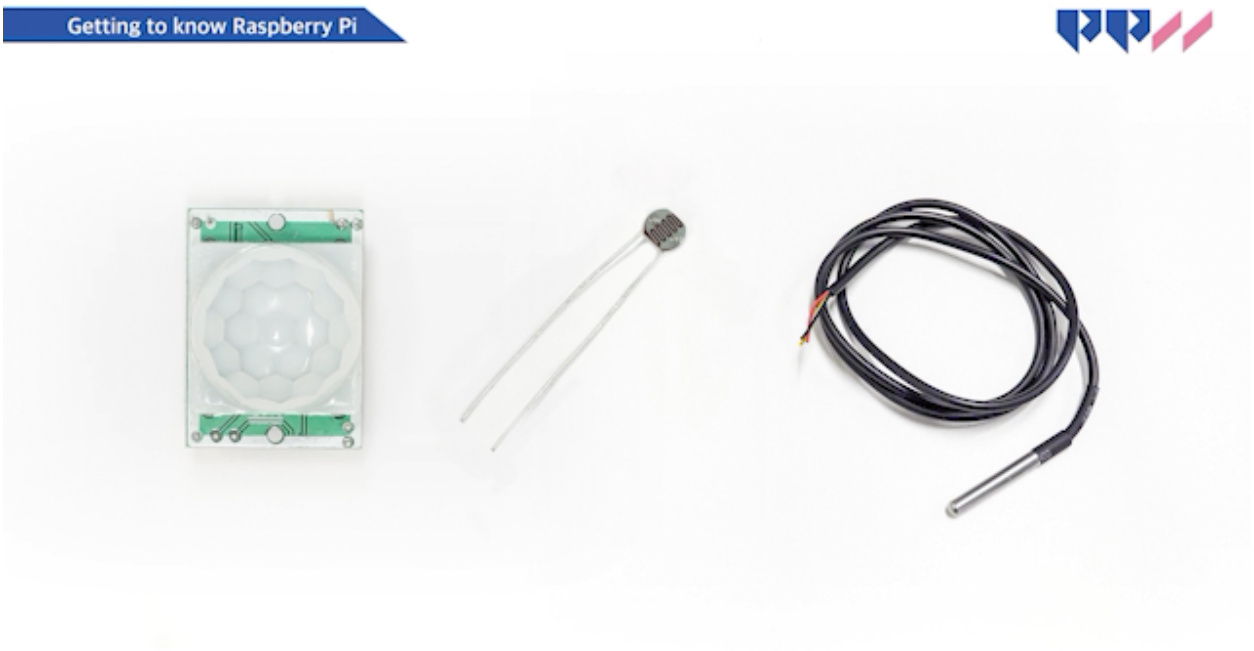
Getting to know Raspberry Pi

active

passive

The full name of a buzzer is the piezoelectric buzzer another output device. There are two different types of buzzers; active and passive buzzer. Make sure you get the **active** buzzer because they are much easier to work with on projects.

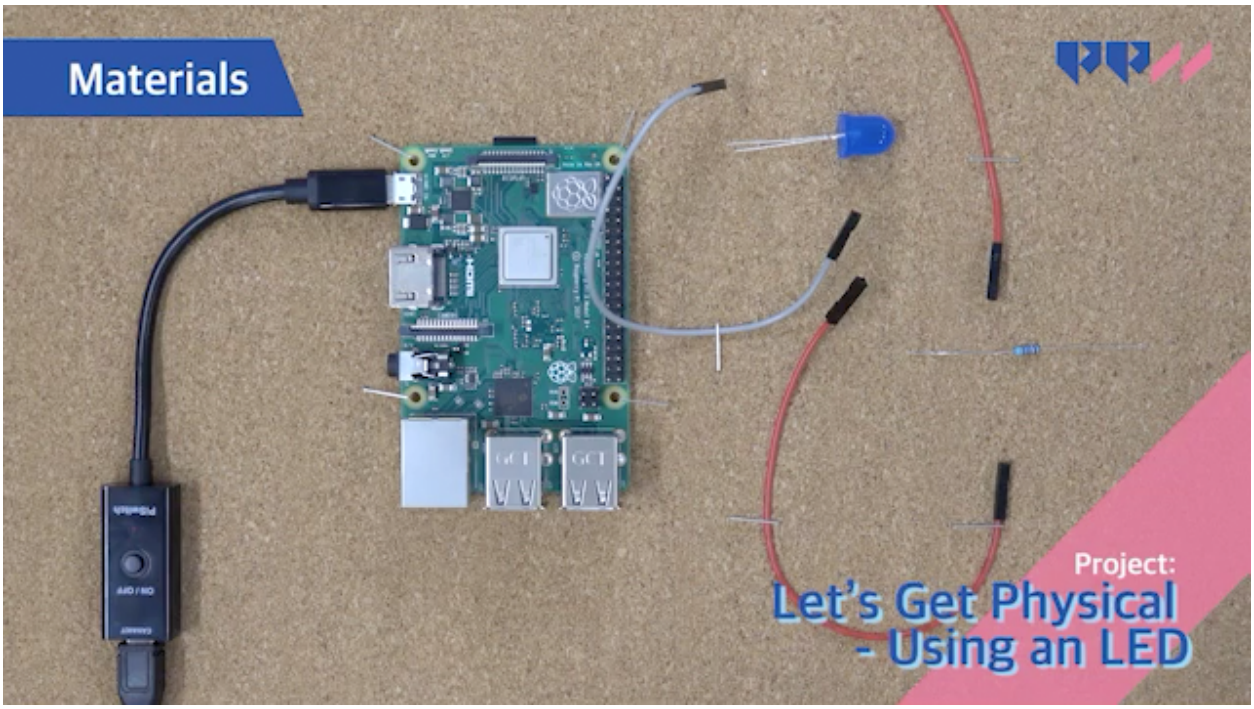
Sensors



There are a variety of sensors that can be used with Raspberry Pi for physical computing projects. There are inputs devices. The Pi will take in information from the sensors and do something based on the program you have written.

- **Motion Sensor** - This sensor uses infrared to detect motion. Perfect if you are planning on creating a trap or an alarm for your room.
- **Temperature Sensor** - Measures the temperature of the area around the sensor. This would be good in gardening projects or projects that require a specific temperature.
- **Humidity Sensor** - Measures the general humidity of a room or soil. Perfect for predicting whether or checking to see if plants need to be watered.
- **Light Sensor** - This sensor checks to see if there is sufficient light or darkness (depending on what you program) and will trigger something based on the code you have written. This could be used to create a nightlight that turns on when the room goes dark.

Hello World! LEDs are Here.



Materials:

- LED
- 3 female/female jumper wires
- 1 330Ω resistor
- Raspberry Pi

The Hardware

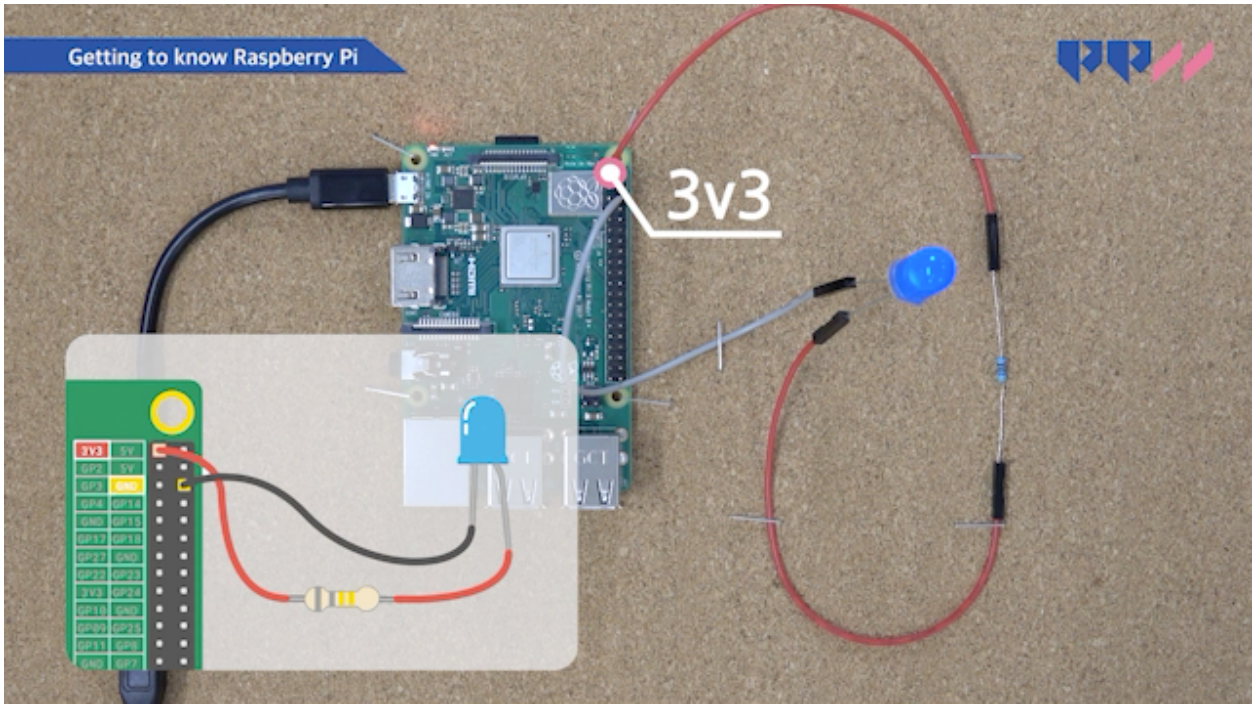
1. The first thing you are going to do is make sure that your Raspberry Pi is turned on.

2. Next, you are going to attach one end of a female/female jumper to your resistor and the other end to the long leg (positive leg) of the LED.

3. Take another female/female jumper wire and connect it to the other side of the resistor and then plug that jumper wire into the 3v3 GPIO pin on the Raspberry Pi.

4. Take the last female/female jumper wire and attach it to the Ground GPIO pin on the GPIO and the other end to the short leg (negative leg) of the LED. You should

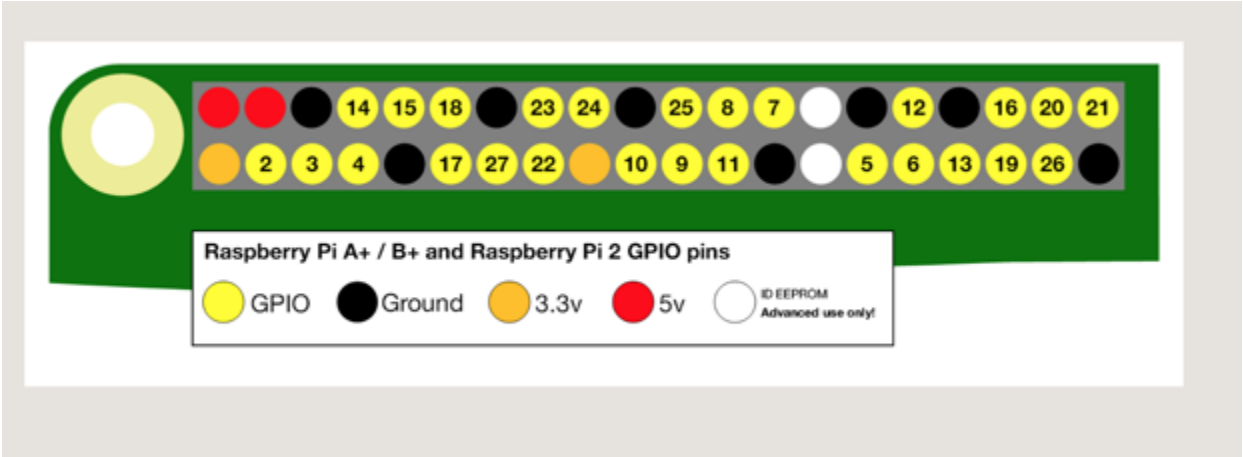
have a circuit that looks like this,



5. Once you wired everything together, the light should turn on. If it does not, make sure that all the wires are connected correctly. The mistake that most often occurs is that the LED legs are connected to the wrong wires. If it still does not light up, try a different LED.

6. The reason that the LED lights up without any code is that the 3V3 pin is always on. That means it is always sending power out. By connecting the LED to the 3V3 GPIO pin and the ground, you are completing the circuit.

7. We are now ready to add code to make the light turn on.



Code-1: LED control

1. Take the female jumper wire connected to the 3V3 GPIO pin and move it down to the **GPIO 25** pin. This is the 11th pin from the top on the outer row of GPIO pins.
2. Open a new Thonny tab and save it as First Light.
3. Enter the following lines of code,

```
from gpiozero import LED
led = LED(25)
led.on()
led.off()
```

We are naming led as LED(25)#
The 25 in LED(25) is the GPIO pin we are going to use to control the LED
To control the LED on and off, type one of the two following lines of code,

```
led.on()
led.off()
```

You can use the # to comment out one of the two lines depending on whether you want the led to turn on or off. Try out both.

Code-2: Blinky Blinky

Now that you have turned the light on and off, it is much easier to have the program do that for you. Add the following code to make your LED blink,

```
`
sleep(1)
```

- We are adding the sleep function from the time library and we are adding a sleep line three times. Note that the number in the parenthese (1) is the amount of time the code will pause there. Run this code and see what happens.
- The light should turn on, off, and then on again. You did it! You have made the light blink.

Challenge

- Do you think you could make the light blink forever using a forever loop?
- What would that look like? Can you add another led and code it to blink as well?