Pin reads high and low levels

For controlling the Raspberry Pi GPIO pins, our tutorial uses the GPIO Zero library.

1. Driver library

If users have used the RPi.GPIO library and wirinPi library before, they will find that the GPIO Zero library and the RPi.GPIO library/wirinPi library handle pins differently:

Driver library	Control pin mode
GPIO Zero library	Device class: Button (input)
RPi.GPIO library/wirinPi library	Input status

The GPIO Zero library has more control methods associated with the device, while the RPi.GPIO library and wirinPi library control pins directly.

2. Button

In the GPIO Zero library, we can use the Button interface to read the high and low levels of the pin's input.

- Read the high and low levels of BCM pin number 27: print information about whether it is pressed

```
from gpiozero import Button

button = Button(27)

while True:
     if button.is_pressed:
         print("Button is pressed")
     else:
         print("Button is not pressed")
```

- Button control LED on and off: LED+Button

```
from gpiozero import LED, Button

led = LED(17)
button = Button(27)

while True:
     if button.is_pressed:
         led.on()
     else:
         led.off()
```