

# Exercise: Programming with the LED and Color Sensor

These exercises are designed to help you develop your programming skills in a fun and hands-on way. For this series, you will use a hardware kit that includes a color sensor and an RGB LED.

## Task 1: Smoke Test

Let's start by checking that everything is set up correctly.

1. Open the Brick Viewer and connect your hardware kit to your computer. Make sure the devices are recognized.
2. Write a Python program that connects to both the LED and the color sensor.
3. As a simple demonstration, set the LED to green.
4. Continuously read the color sensor values and print the red, green, and blue (RGB) values to the console.

## Task 2: Color Mimic

Now let's make the LED respond to the environment.

1. Create a new Python program. Reuse your connection code from Task 1.
2. Add a loop that continuously reads the RGB values from the color sensor.
3. Print the RGB values to the console, normalized to the 0–255 range.
4. Update the LED color in real-time so that it reflects the color detected by the sensor.

### Task 3: Color Memory Game

It's game time! Let's test your skills with a fun challenge.

1. Start a new program and copy in your device connection code from Task 1.
2. Define a list of color names: red, green, blue, yellow, purple, orange, white.
3. Write a function that picks a random color from the list.
4. Create another function that takes a color name as input and sets the LED to that color. (You will need to map each color name to RGB values.)
5. In the main loop, wait for the user to press "ENTER".
  - a. When "ENTER" is pressed, pick a random color and set the LED accordingly.
  - b. Start a timer.
6. The player must find an object that matches the LED color and place it in front of the sensor.
  - a. Read the RGB values from the sensor.
  - b. Compare them to the LED color and calculate the deviation.
  - c. If the deviation is small enough, stop the timer.
7. Print the following:
  - a. The RGB values of the LED
  - b. The RGB values detected by the sensor
  - c. The color deviation
  - d. The time it took the player to match the color