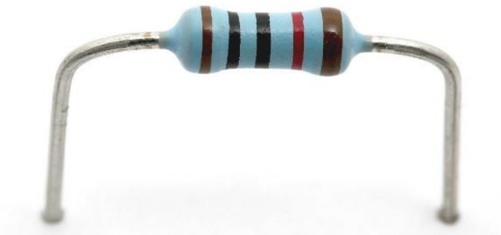
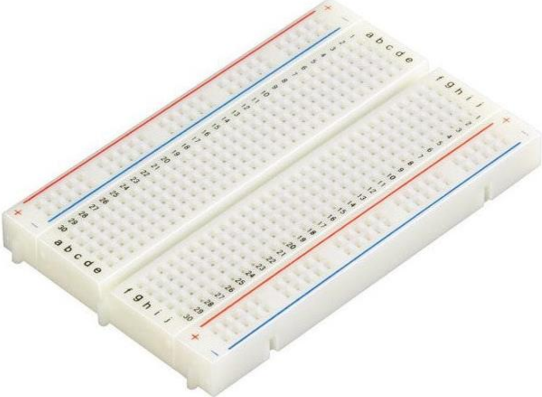
Components



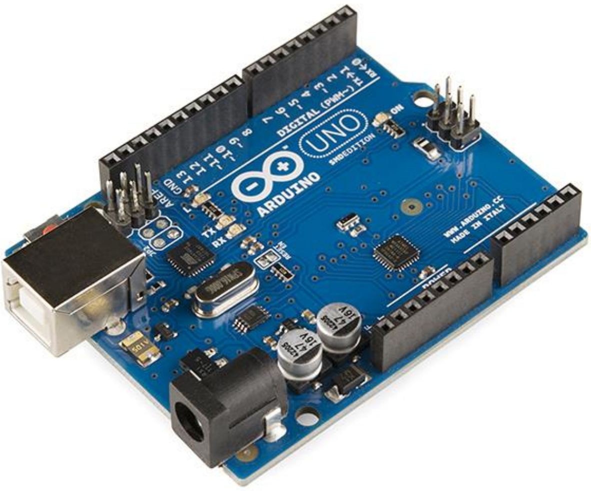
Resistors

LED



Breadboard

Pushbutton



Arduino

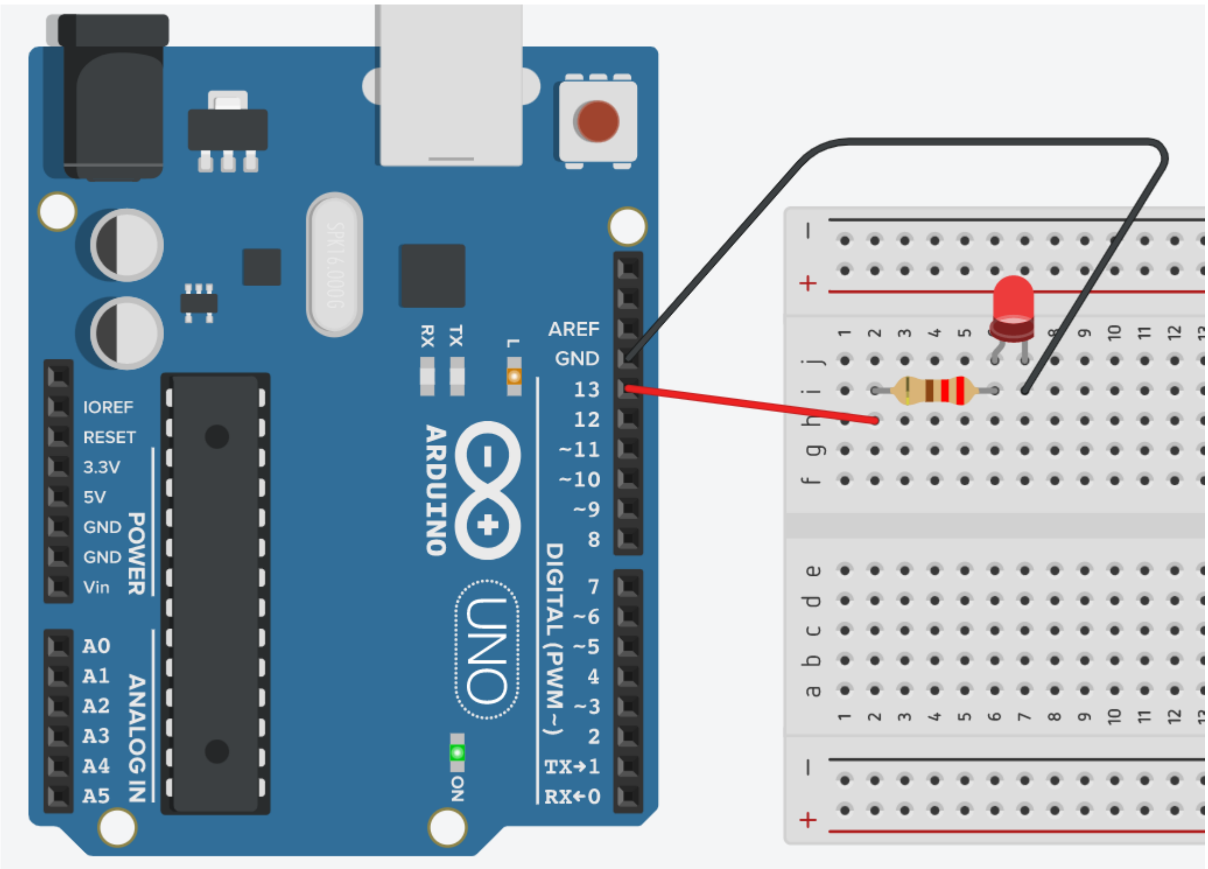
RGB LED

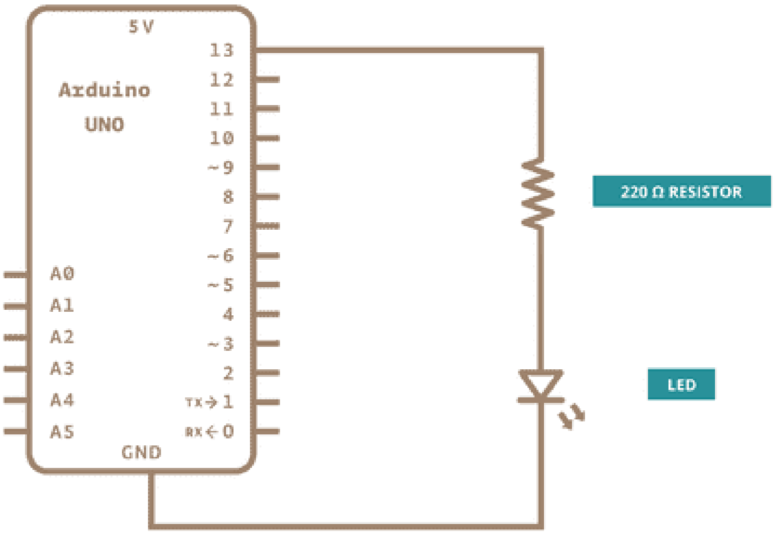
Exercise 1: Blink an LED

Build the circuit below, pay attention to where the wires plug in!

Use the 220 ohm resistor (red, red, black, black, brown)

The LED needs to be the right way around, it will have a shorter leg (called the cathode), which should be on the right side, connected to the black wire and ground (GND).

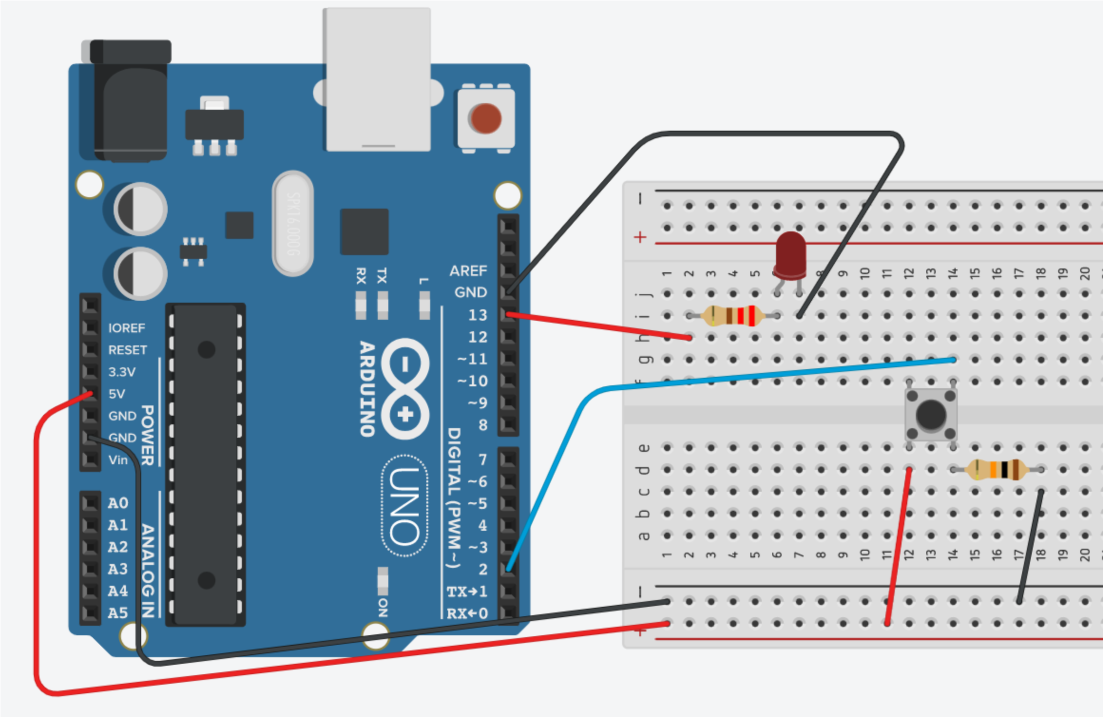




Exercise 2: Use a Push Button

Add the circuit below to your Arduino to use a push button.

Use a 10 kOhm resistor next to the button. (brown, black, black, red, brown)



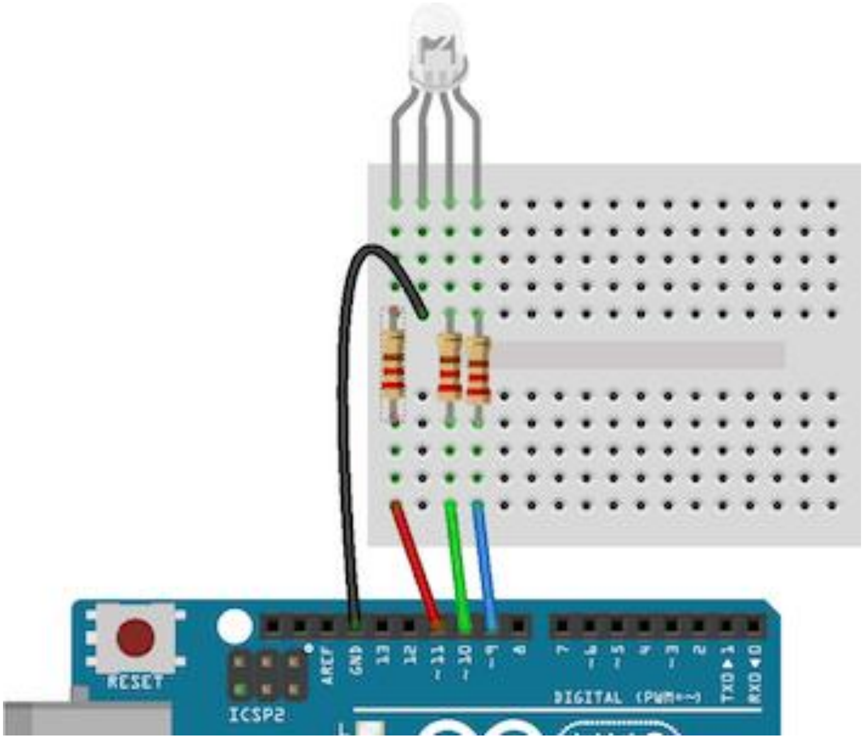
Program the Arduino with the code below

When you push the button, the LED should now light up.

```
int buttonState = 0;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  buttonState = digitalRead(2);

  if (buttonState == HIGH) {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
}
```

Exercise 3: Light an RGB

Clear your board of all the other circuits, and now build the circuit below with an RGB (Red Green Blue) LED. This is a single component with three LEDs inside.



Upload the code below to the Arduino. The LED should light up red.

The line RGB_color(255, 0, 0) sets the colour of the RGB, setting how much red, green and blue should be lit up, with a number between 0 and 255.

What happens if you change it to RGB_color(0,255,0)?

Try different combinations of numbers and see which colours you can make.

```
int red_pin= 11;
int green_pin = 10;
int blue_pin = 9;
void setup() {
  pinMode(red_pin, OUTPUT);
  pinMode(green_pin, OUTPUT);
  pinMode(blue_pin, OUTPUT);
  RGB_color(255, 0, 0);
}
void loop() {
}
void RGB_color(int red, int green, int blue) {
  analogWrite(red_pin, red);
  analogWrite(green_pin, green);
  analogWrite(blue_pin, blue);
}
```