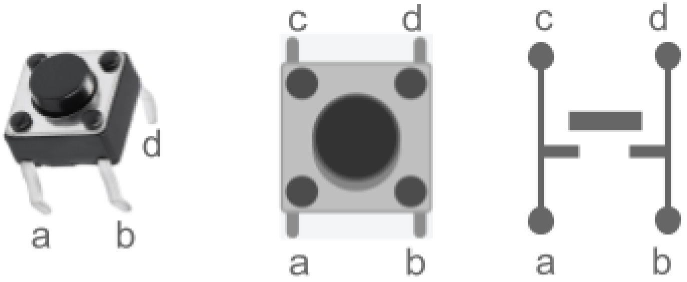
AZIA Robotics
Lesson 8 - Push Buttons

website: aziarobotics.github.io
tinkercad.com class code: **ALD 5WB V3J**

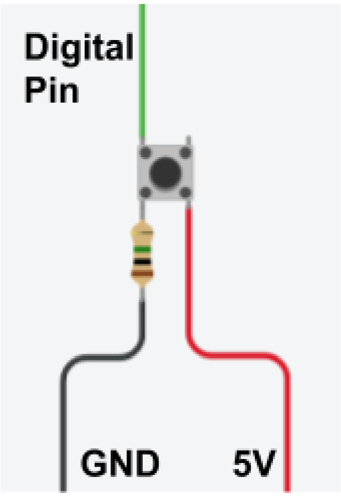
In this course we will learn about push buttons and ways of using it to add input to our circuits.

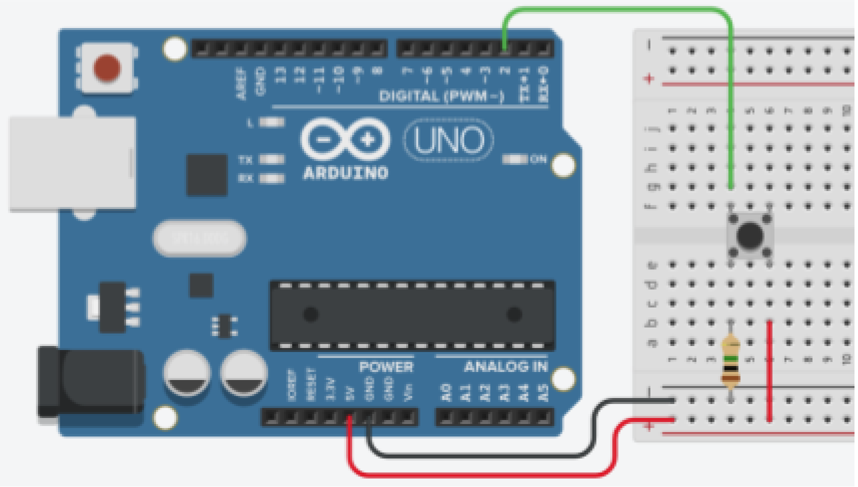
This is called a push button. It has 4 pins. Pin-a is always connected to pin-c, and pin-b is always connected to pin-d. But when the button is pushed, all 4 pins get connected together.



To use a push button, connect pin-a to Uno's GND using a resistor, pin-b to Uno's 5V pin, and pin-c to a digital Uno pin, for example Uno pin 2.

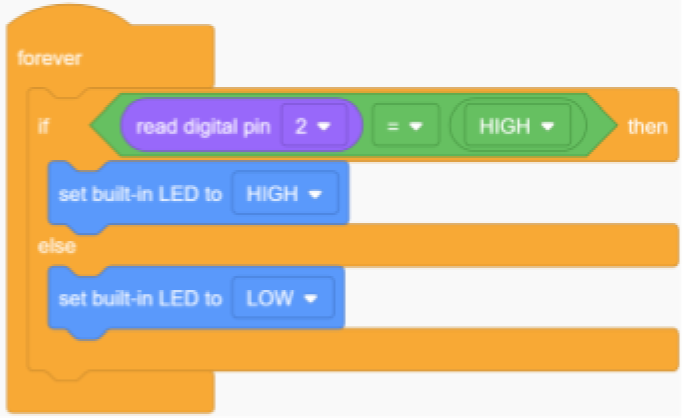
Then use the read digital pin to check if the pin connected to the push button is HIGH.

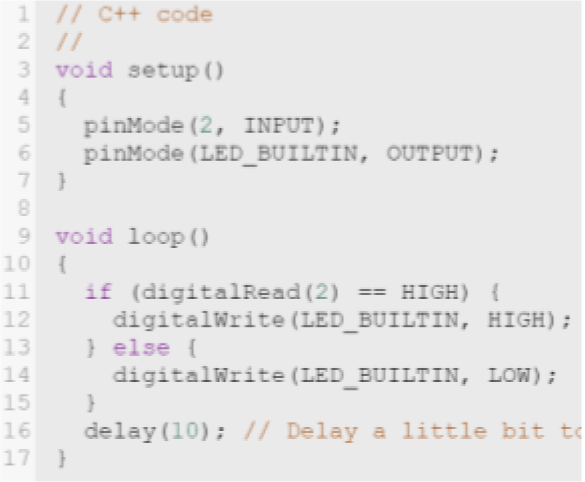




Activity 1 - Push Button LED

Use the circuit with a push button to turn the built-in LED on when the button is pushed.





```
// C++ code
//
void setup()
{
  pinMode(2, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  if (digitalRead(2) == HIGH) {
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10); // Delay a little bit to
}
```

Activity 2 - Use Push Button as a Switch

You can also use a push button like a switch that toggles on and off. So if the light is off, pushing the button once turns the light on, and then if the button is pushed again it will turn it off.

To do so, in your code you need to know if you previously pushed the button to turn the LED on or off. This can be done in the code by using a <u>**variable**</u>. A variable is used in the code to store values, such as numbers.

A variable has a name made of alphabet letters, numbers and the underscore _ character. It has to start with an alphabet letter.
For example these are variable numbers that can be used:

toggle	*Toggle1*	*FirstName*	*abc_123*	*Price*

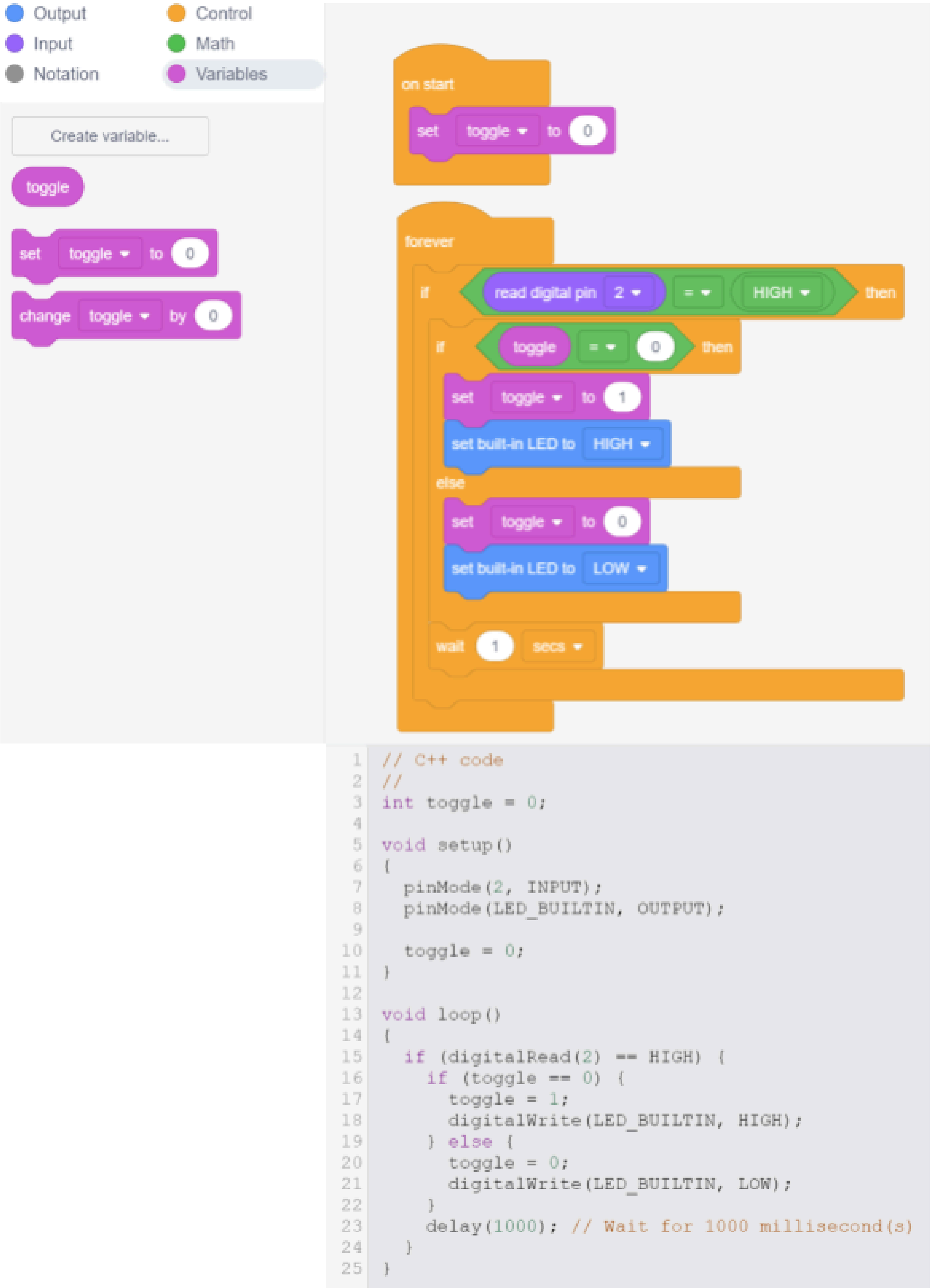
But these can NOT be used for naming the variables:

A toggle	*toggle-1*	*First Name*	*123_abc*	*$Price*

To make a push button work like a switch, you can add a variable and set its value to 0. When the button is pushed, use an **if-else** block to check the value of the variable:

- **if** the value of the variable is 0, change the value of the variable to 1 and set the LED to high.
- otherwise (**else**), it means that it was previously changed to 1. So this time you change it back to 0 and set the LED to low.

Without Changing your circuit, change the code to this to make the push button into a switch. Here we have added a variable and named it “**toggle**”.

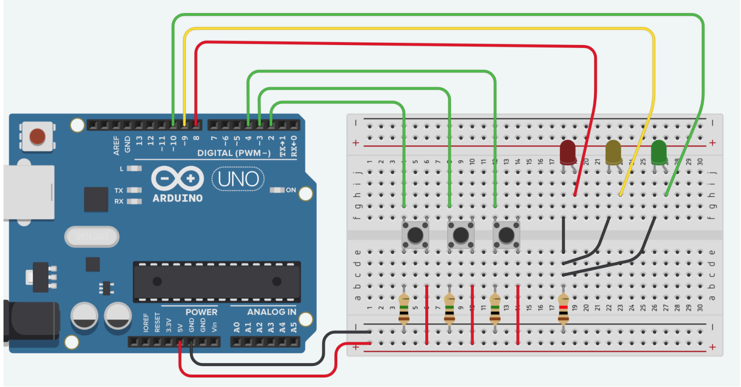


```
// C++ code
//
int toggle = 0;

void setup()
{
  pinMode(2, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  toggle = 0;
}

void loop()
{
  if (digitalRead(2) == HIGH) {
    if (toggle == 0) {
      toggle = 1;
      digitalWrite(LED_BUILTIN, HIGH);
    } else {
      toggle = 0;
      digitalWrite(LED_BUILTIN, LOW);
    }
    delay(1000); // Wait for 1000 millisecond(s)
  }
}
```

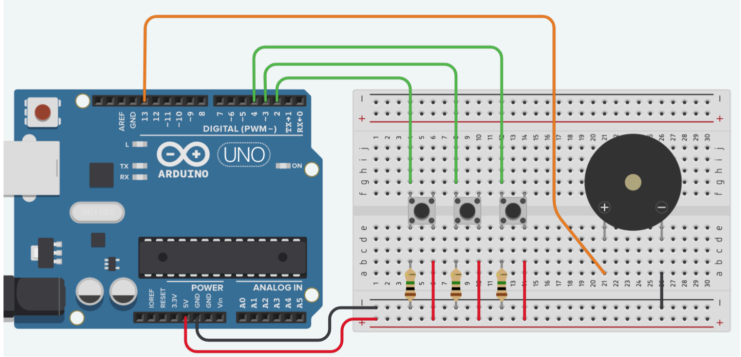
Activity 3 - Use multiple push buttons

Now that you know how to use a push button, use multiple of them (3 or more) to control some of the electronics you have been using in the previous sessions. Here are some ideas:

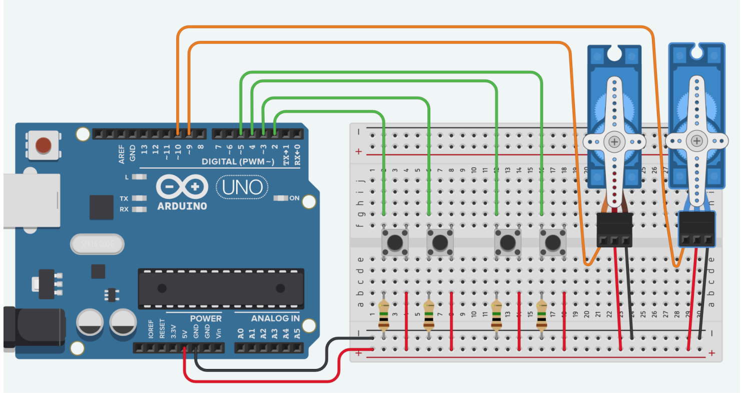
Idea 1: Have multiple LEDs with different colors. Pushing each button will turn one of the LEDs on.



Idea 2: Use a passive buzzer with multiple push buttons to create a mini-piano. Pressing each push button will play a different tune.



Idea 3: use the push buttons to control one or two servo motors. For example one push button to turn a servo to 180 degrees and another to turn it to 0. Use another two push buttons with a second servo.



What else can you come up with ?!