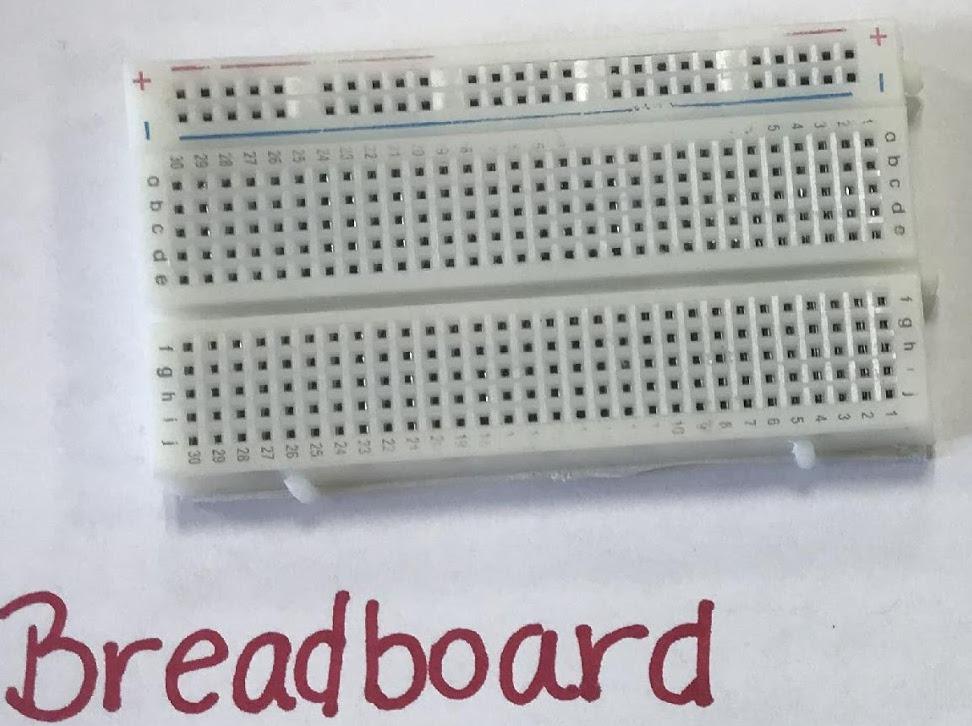
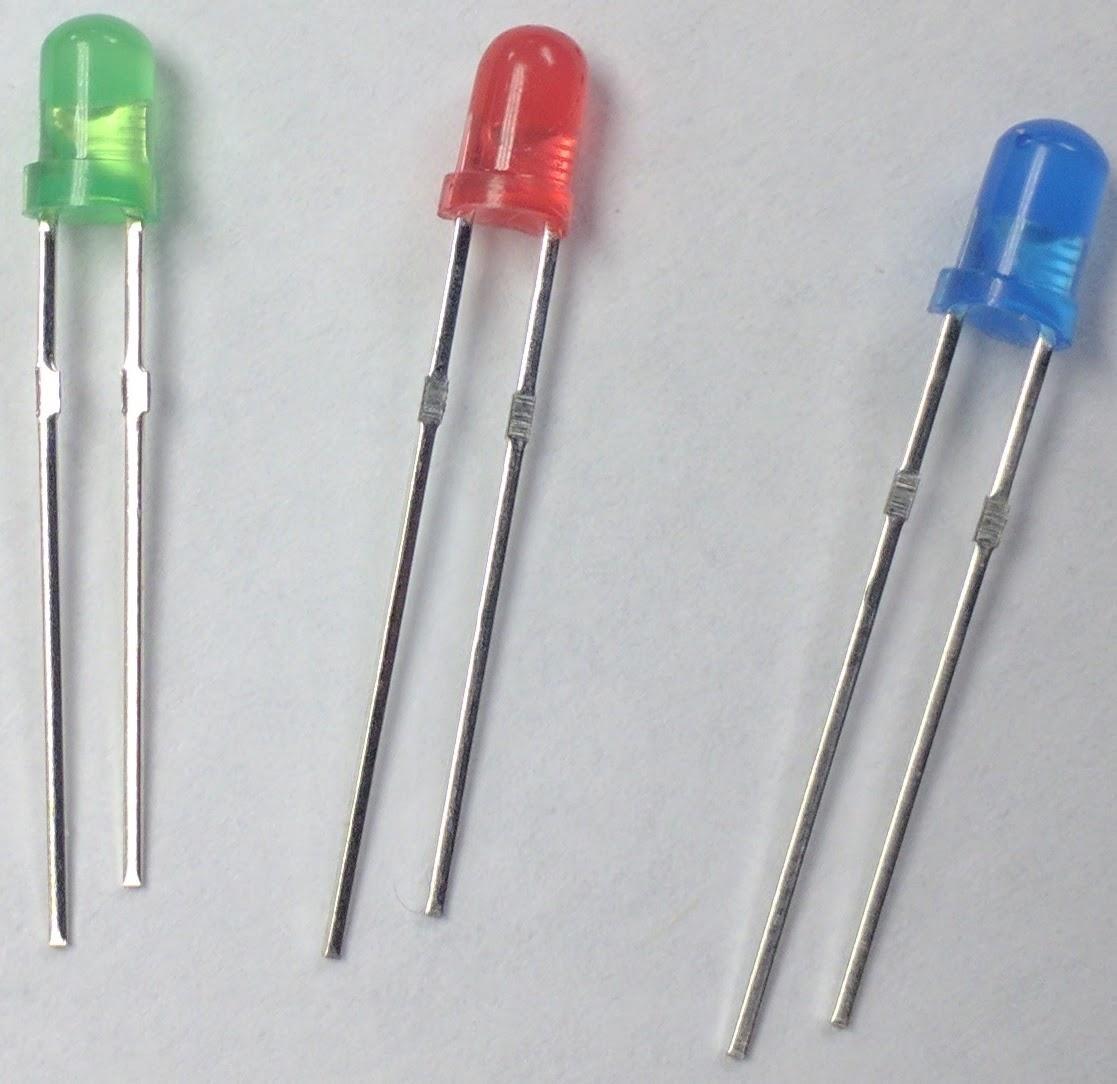
**iCREAT I: Module 3 - LAB 2 -**

**Introduction to Coding with the Arduino**

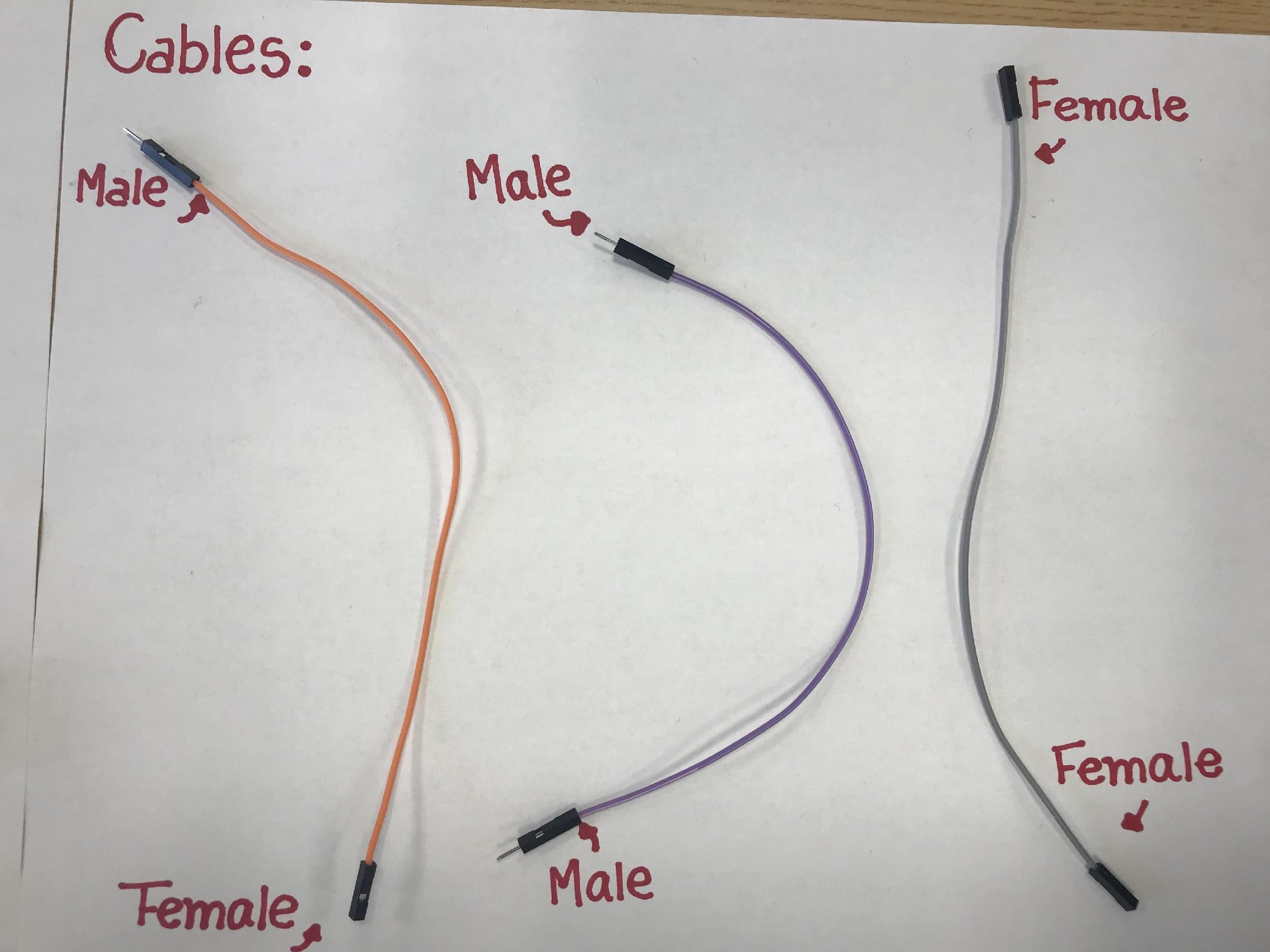
horizontal line

# pic Arduino.PNG

# Objectives

* Learn about the Arduino and its potential
* Learn about the components we will control with the Arduino
* Learn about the Arduino IDE and Tinkercad
* Learn to design and simulate your circuit
* Light up LEDs and make music with the buzzer

# Required Resources:

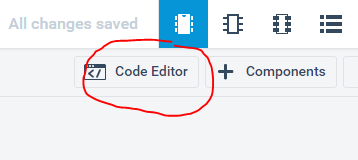
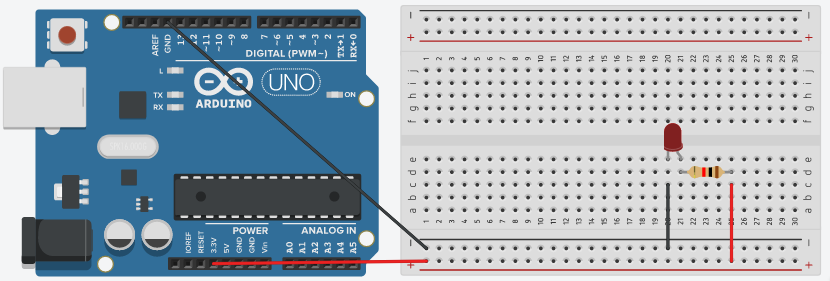
* Access to the Arduino IDE and Tinkercad
* An Arduino UNO connected to a computer
* LEDs, resistors (330Ω, 100Ω), a piezo buzzer, jumper cables, a breadboard, 2-pin screw terminal, screwdriver
* Access to iCREAT and enough time to have fun!

**Reminder: Setting Up the Arduino IDE and Tinkercad**

Connect the board to the computer and click on the Arduino icon to get started .

1. Under the **Tools->Board:** menu select the Arduino/Genuino Uno board.
2. Under **Tools->Port:** select the correct serial port assigned by your system. This may vary from computer to computer or day.
3. Once you have selected board and port start Tinkercad [https://www.Tinkercad.com](https://www.tinkercad.com) the simulation tool.
4. Tinkercad allows you to design your circuit and to simulate it by programming it with text or blocks. It works very much like [Scratch](https://scratch.mit.edu/), but it controls the Arduino and its components. Tinkercad will generate Arduino text-based code that will run as a simulation while you debug your circuit and code. Once it all works well you can copy the text code to the Arduino IDE and upload it to your board. If you change the Arduino code, it will **NOT** be reflected on your Tinkercad simulation code. Make sure you keep them synchronized.

**Part 1. Simulating A circuit to Control any LED with the Arduino**

When we connect an LED to the Arduino, we must use a resistor to limit the amount of electrical current that flows through it. Make sure you select the **correct resistor**. If you use a resistor with large resistance the amount of current flowing will be small and the LED will be very dim or not light up at all. A resistor with small resistance may burn the LED. 

Which resistor to use? We should look at the data sheet for the LEDs but typically a small LED can handle 20mA. The output from our Arduino will be 5V at HIGH. Using the Ohm’s law you learned you should calculate it: V = I \* R or R = V / I.

We want to control when to let current flow through a pin. For this purpose we will use any of the numbered pins (1-13) which we can program to control the flow of current.

1. **Use the simulation tool to simulate your circuit.**
2. **Follow the diagram to create** a circuit for 1 LED connected to **pin #2** in the Arduino. Notice that you must use **Ground GND**, and pin #2 to have a closed circuit.
3. In the simulation tool click on the **Code Editor** button on the top right to modify the generated code. Check the pin number you will use in the Arduino. Use the correct pin number in your code.
4. Run the simulation until it blinks the LED correctly.
5. **Implement your circuit and Program** it using the Arduino IDE and sketch. Remember that the longest LED leg is the **positive**! The shortest is negative and must be connected to **ground**.
6. Remember to include a delay each time you send a signal to the pin (whether it is LOW or HIGH).

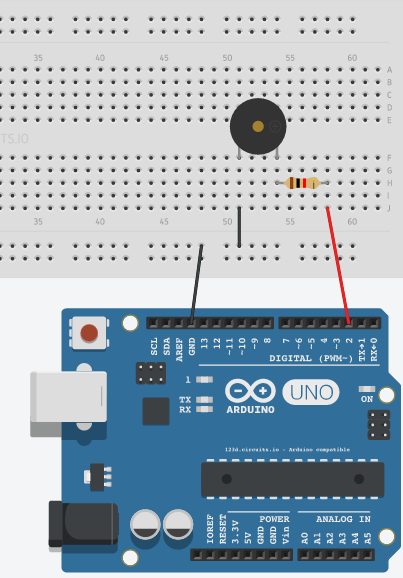
**Part 2. A circuit to Control 3 LEDs with the Arduino**

1. Modify your circuit to include 3 LEDs. Make them blink in different ways!!
2. **Use Tinkercad to simulate your circuit first**.
3. Remember to modify the generated code in the simulation tool to make it work with 3 LEDs.
4. **Once your simulation is successful proceed to implement the circuit using the provided components.**
5. **Show us your** completed diagrams and programsfor theblinking LEDs before you move the next section.

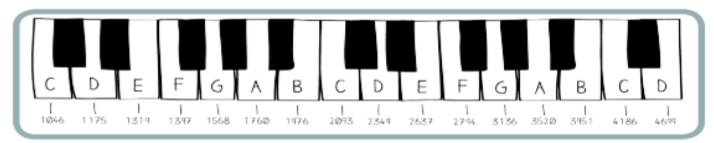
**Part 3. A circuit to Control the buzzer**

For this exercise you will use a piezo buzzer, 100 ohm resistor, and the breadboard.

In Tinkercad you can use the block



to send tone frequencies to the speaker on pin 8. Make sure you have a delay after each note to play it correctly.

1. **Follow** this diagram to create a circuit that uses the buzzer on pin 8. **Simulate** it first!
2. **Program** the Arduino to play a few frequencies (values between 400 and 5000).
3. **Show** us when your buzzer makes a sound.
4. **Program** your Arduino to have the buzzer **play a short Do-RE-MI melody.** Use the frequency guide below.
5. Tutorials, Examples and help at <https://www.arduino.cc/en/Tutorial/BuiltInExamples> 

For more info watch: [Using the Arduino tone() function with a piezo speaker](https://youtu.be/1_LMAgO14z0)