

A photograph of a Tesla coil experiment. The coil is a tall, vertical metal structure with a glowing orange-red top section. It is surrounded by a dense, branching network of bright purple and white electrical discharges (sparks) that fill the dark sky. In the background, a person wearing a bright yellow shirt and dark shorts stands on a platform, looking towards the coil. The scene is set outdoors at night.

Getting started with electricity and Arduino!

A crash course:
Electricity
Buttons

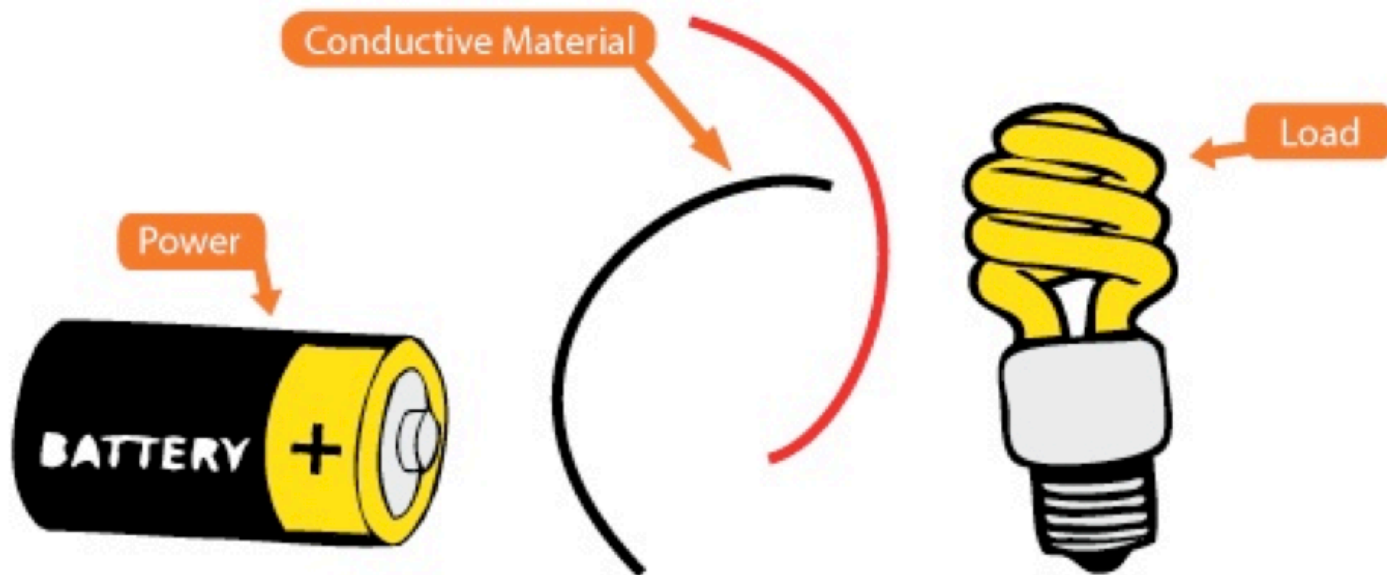
What is electricity?

- general term referring to the flow of **electrons** or an electric charge
- electrons are **charged particles** which we can direct in specific directions through a **circuit**



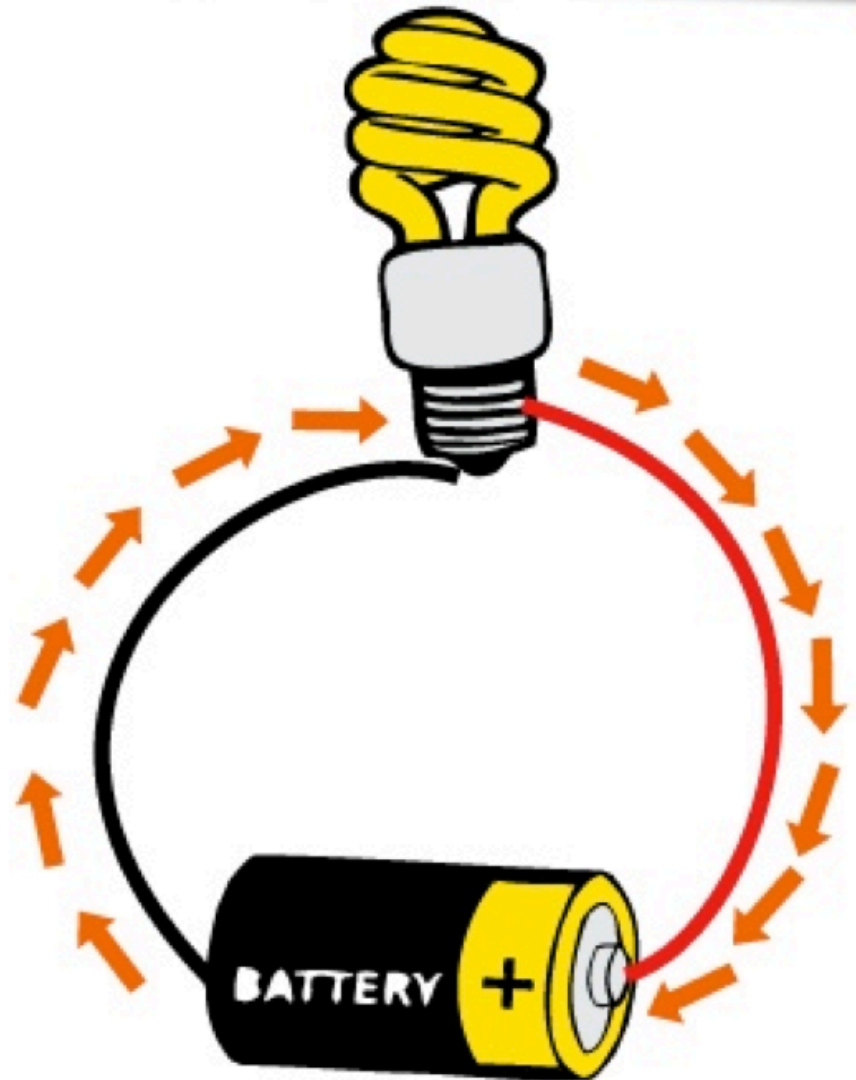
Circuits

- consist of a **power supply**, **conductive material** and a **load**
- we'll be using wire for our conductor, but there are many conductive materials out there! fabric, thread, water, your skin...



Simple circuit

- electrons **move in only 1 direction:** from the ground to the load and to the positive lead of the power source
- 'circuit' comes from the latin word for circle...



Measuring electricity

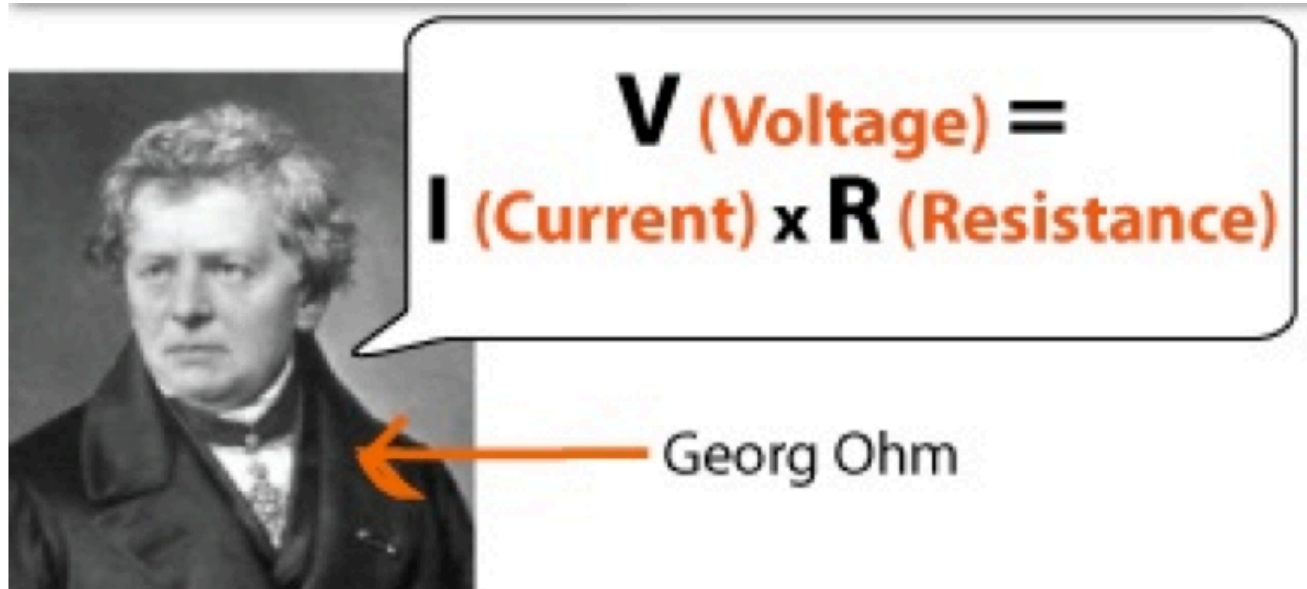
- we use **Volts** to measure voltage
- **Amperes** (amps) to measure current
- **Ohms** (Ω) to measure resistance

Ohm's Law:

$$V = I \times R$$

$$I = V / R$$

$$R = V / I$$

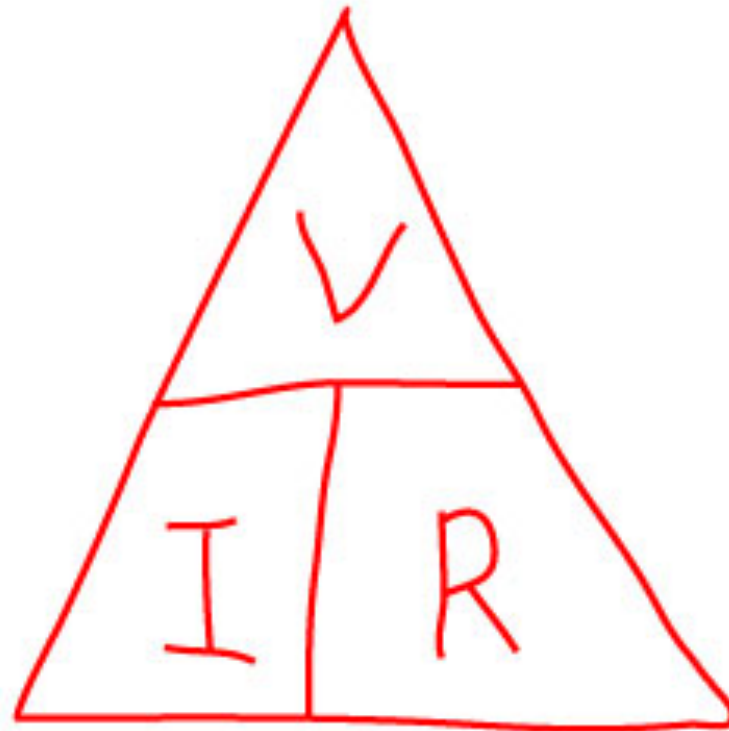


Georg Ohm

Let's watch an awesome video about Ohm's Law!!!!



Ohm's law



volts vs. amps

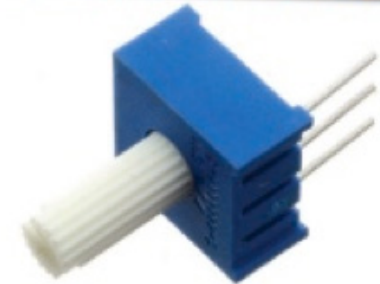
- one way to understand is with the classic 'water analogy'
- the **power** source is like the **water pump**
- **voltage** is the **water pressure**
- **current** is the actual **flow of water**
- **resistance** is the thinning of a **water pipe**

Resistance

- resistance is a force that limits current. it slows down the electrons, and lets less of them through
- some components like LEDs, speakers, or motors need a specific, smaller amount of current...
- we can add resistance to our circuit with several components



POTENTIOMETER



PHOTORESISTOR



Reading a resistor

RESISTOR COLOR CODES

0

1

2

3

4

5

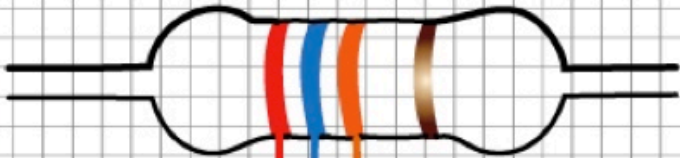
6

7

8

9

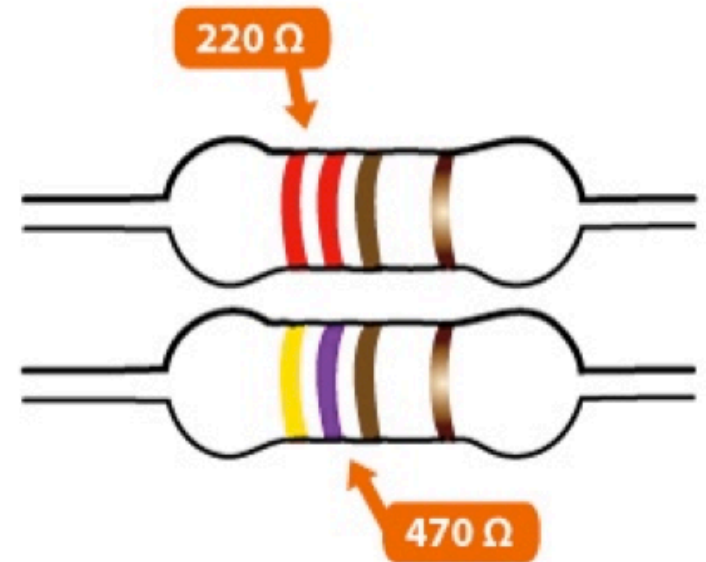
Use this chart to read the Ohms on your resistors using the color bands.



First Digit

Second Digit

Number of Zeros after the first and second digits



Arduino

- **open-source hardware and software** prototyping platform
- will allow us to **control** the flow of electricity **and** let us **measure** electricity through **CODE**

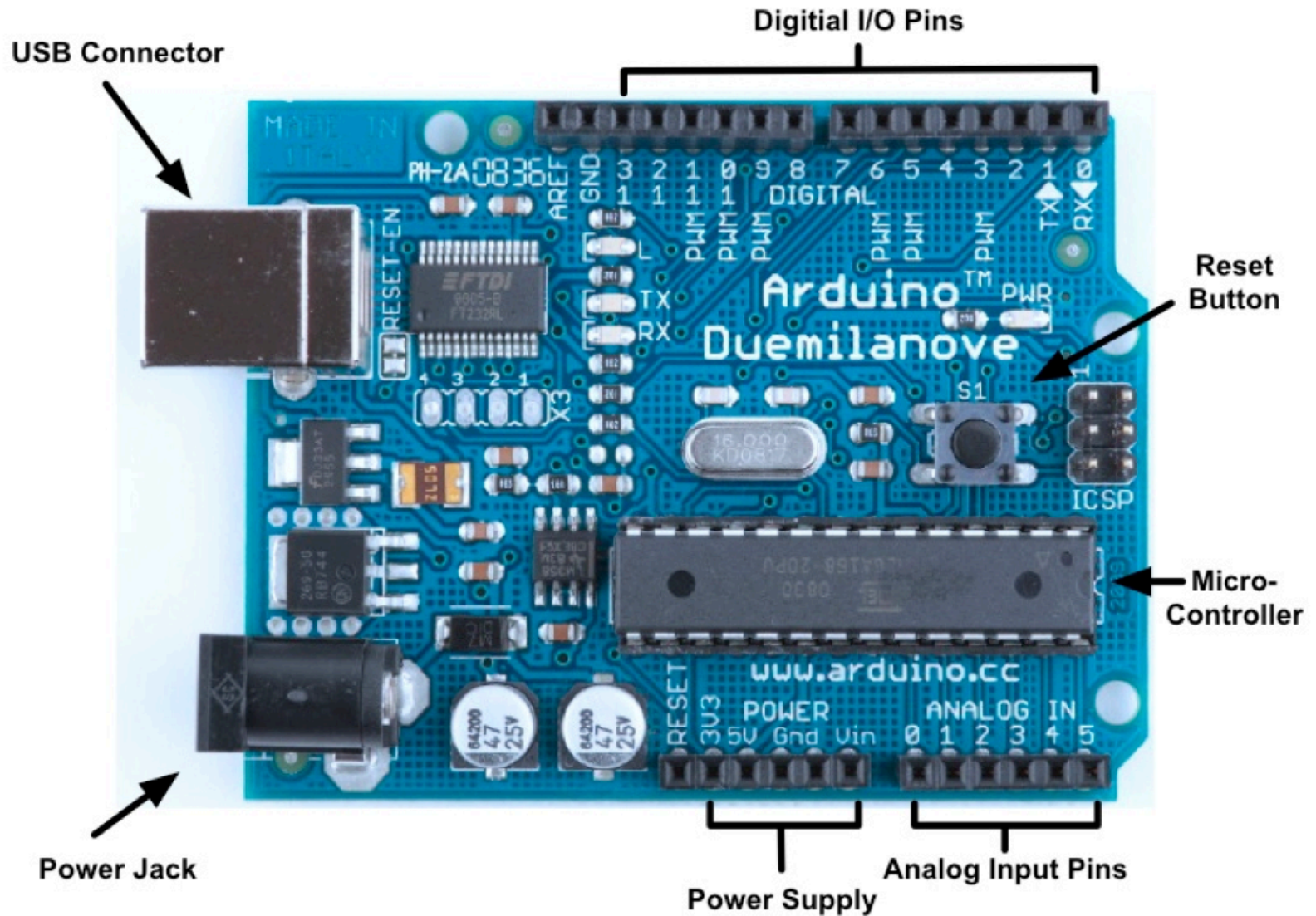


A really tiny computer

- Arduino is basically a breakout board for a **microcontroller**. it's a tiny computer that we can **program** it.
- it has **memory**, **inputs**, and **outputs**.

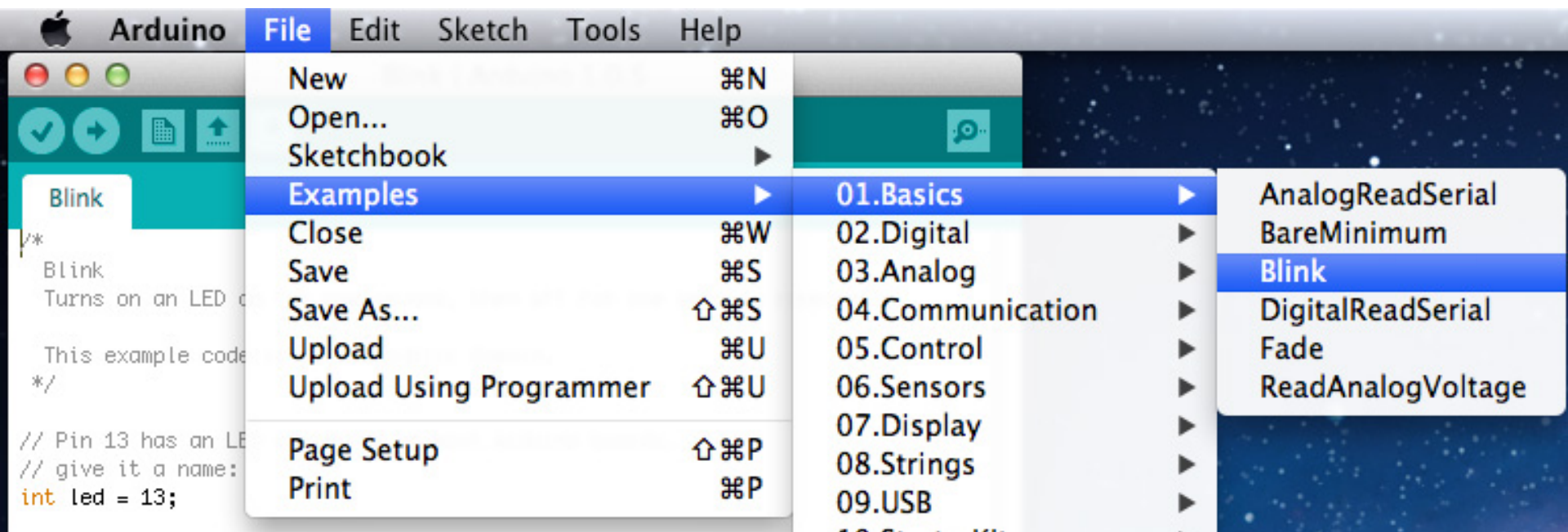


Break it down



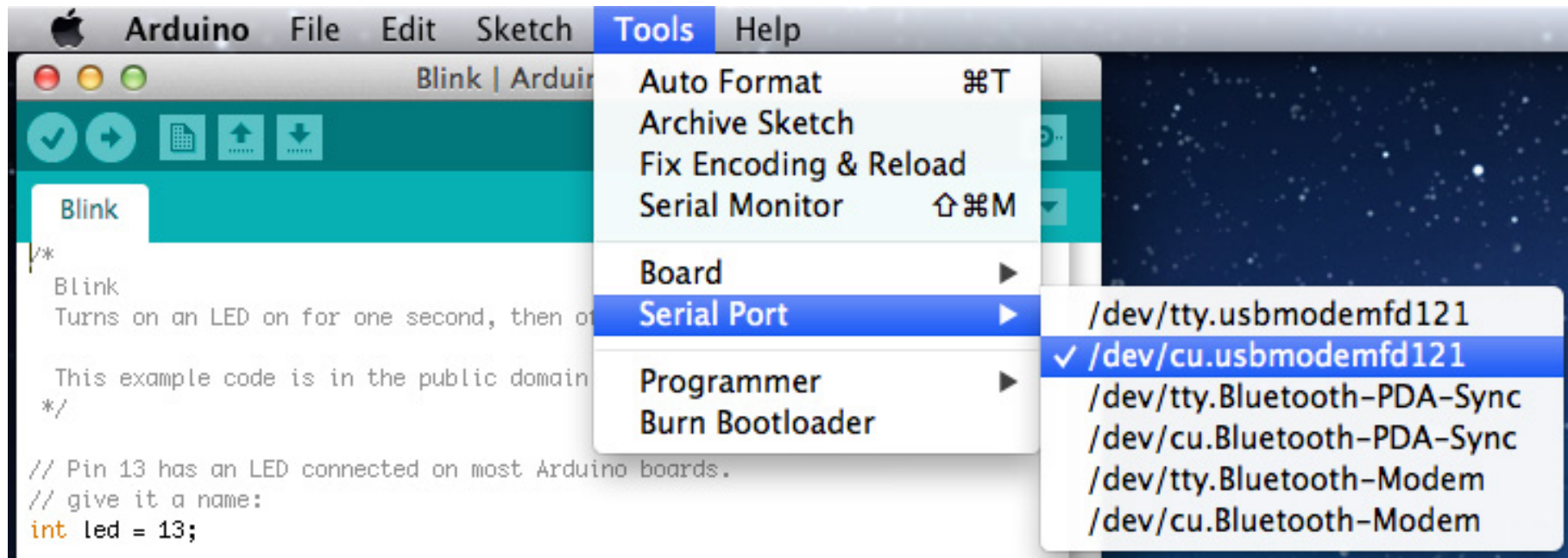
Programming Arduino

1. connect your Arduino to computer with USB cable
2. open a sketch



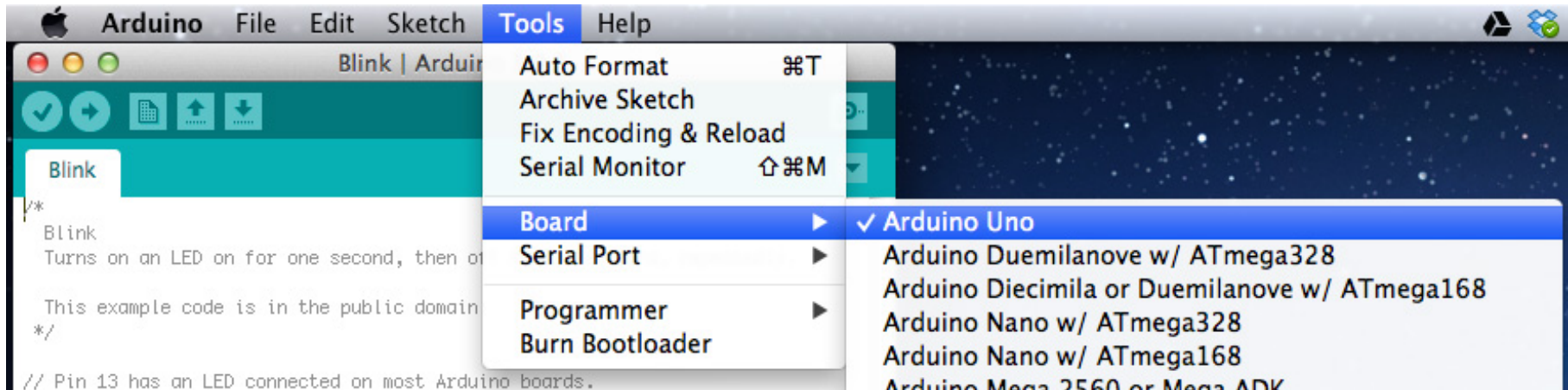
Programming Arduino

3. Select the correct **Serial Port**



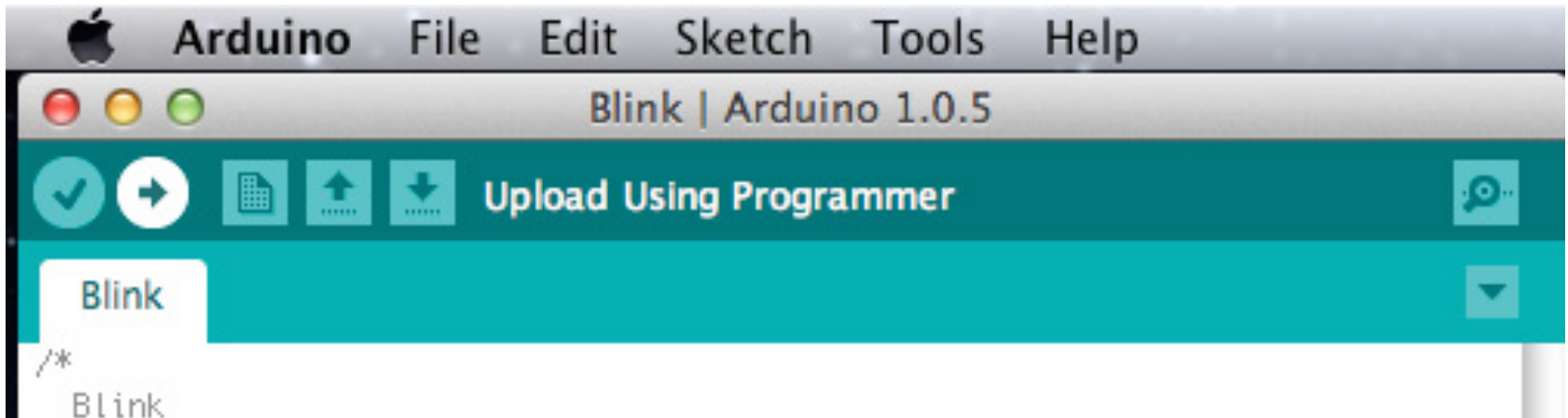
Programming Arduino

4. Select the correct **Board**



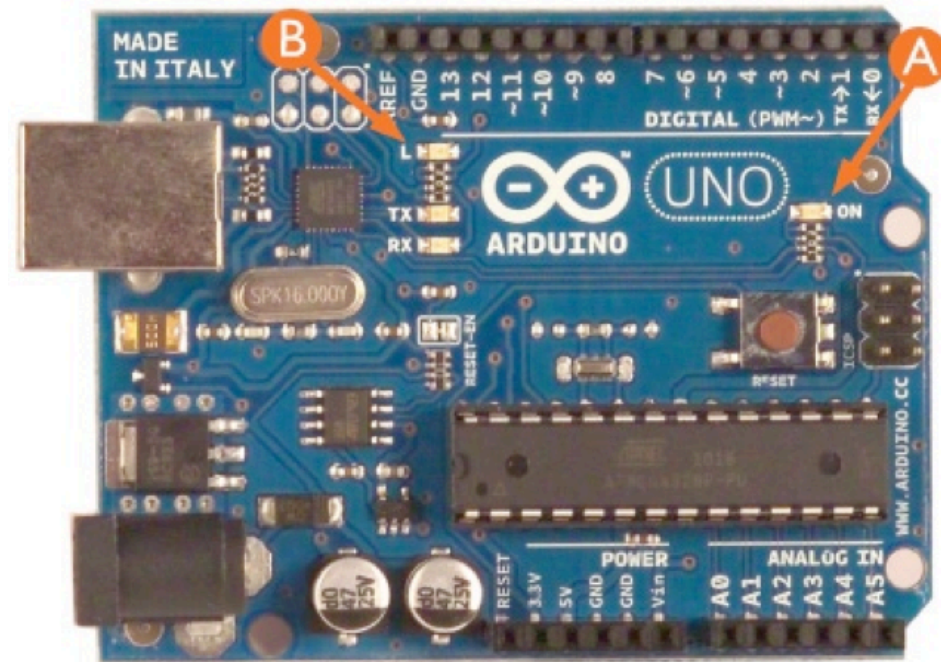
Programming Arduino

5. **Upload** your code to the Arduino board
6. **BLINKKKIINNNGGGG!!!**



The UNO has several LEDs built-in

- **A** is pointing to the power indicator LED
- **B** is pointing to an LED connected to digital pin 13
- **RX** and **TX** pins have activity LEDs



note: these are smaller LEDs, called **surface mount** components

Arduino vs. Processing

```
void setup() {                                << familiar
  // initialize the digital
  // Pin 13 has an LED conr
  pinMode(13, OUTPUT);                       << digital pin mode
                                              (INPUT or OUTPUT)
}

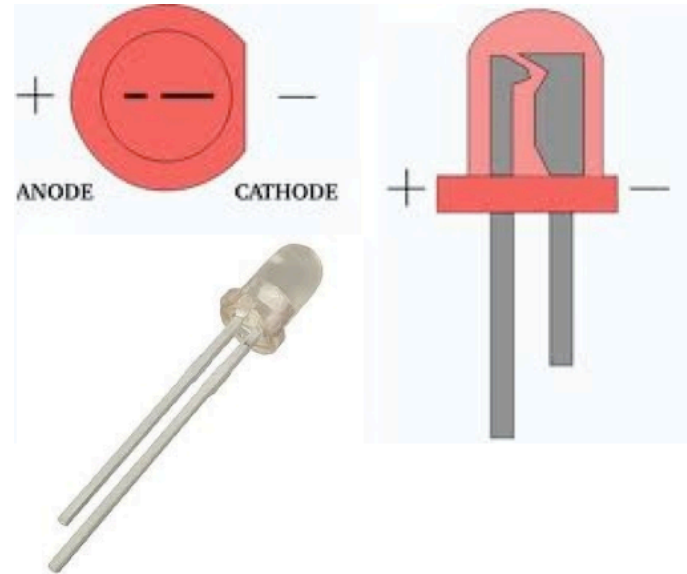
void loop() {                                  << draw is now called loop
  digitalWrite(13, HIGH);                    << turn pin 13 ON
  delay(1000);                                << pause application in ms
  digitalWrite(13, LOW);                     << turn pin 13 OFF
  delay(1000);                                << another pause before looping
}
```


LEDs



light emitting diodes

components that allow current to travel in one direction, and light up when the right amount is passed through

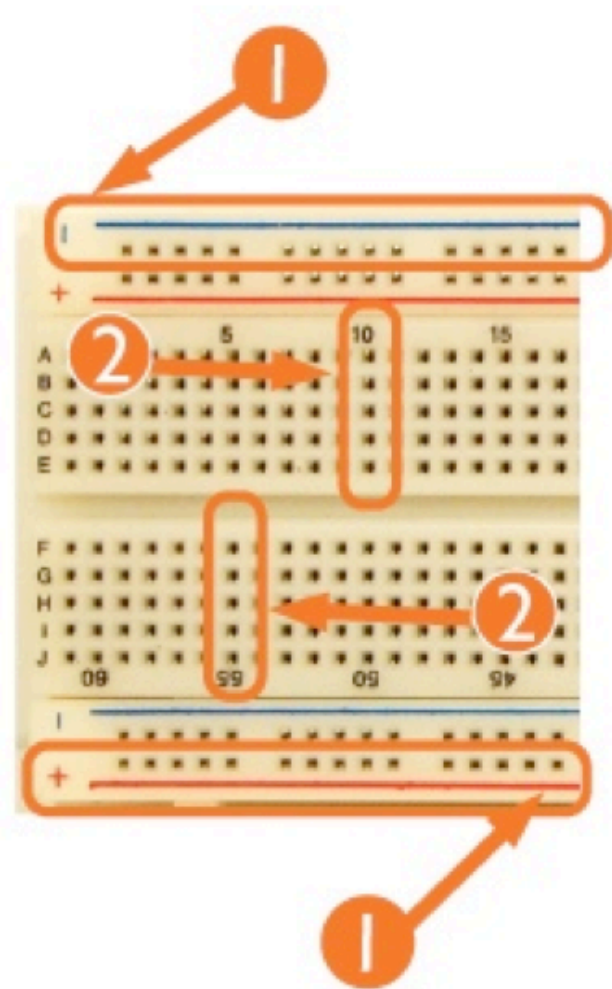


-short leg goes to ground

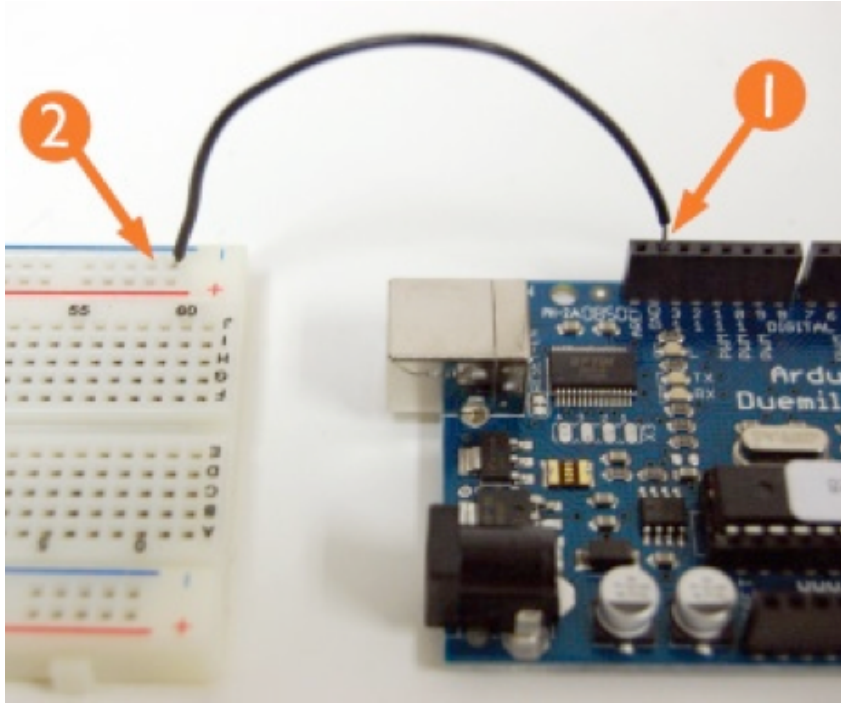
+ long leg goes to power

Using a breadboard

- holes in a breadboard are connected to each other in specific ways
- (1) are the **power rails**. they are connected horizontally, and color coded for your convenience!
- (2) are the **terminal strips**. they are connected vertically and where you will place most of your components

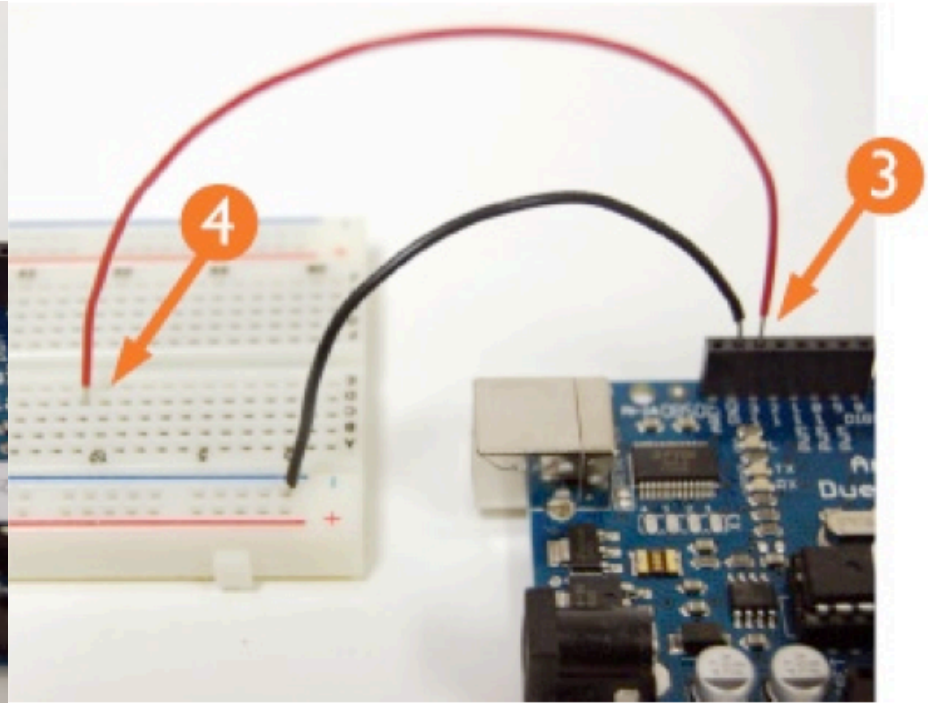


Connect Arduino to a breadboard



1 – jumper wire to **GND**

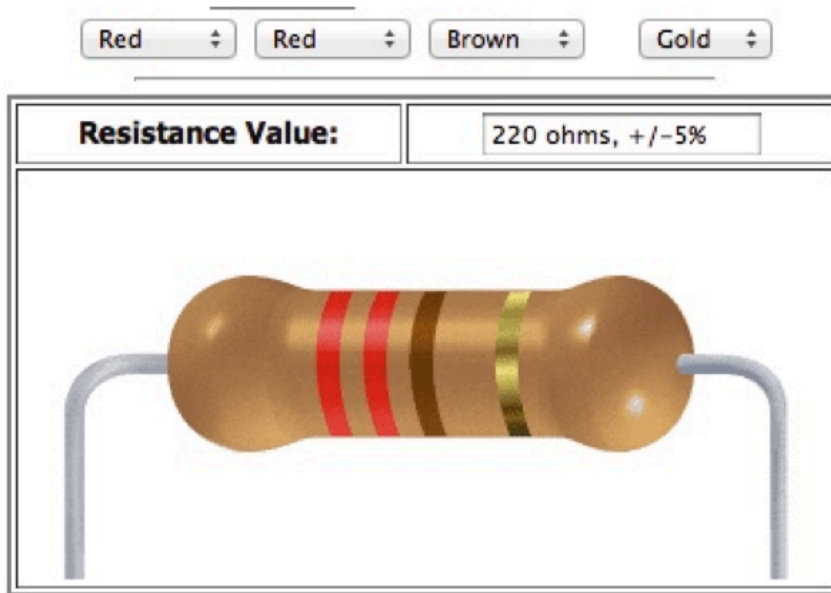
2 – connect **GND** to power rail



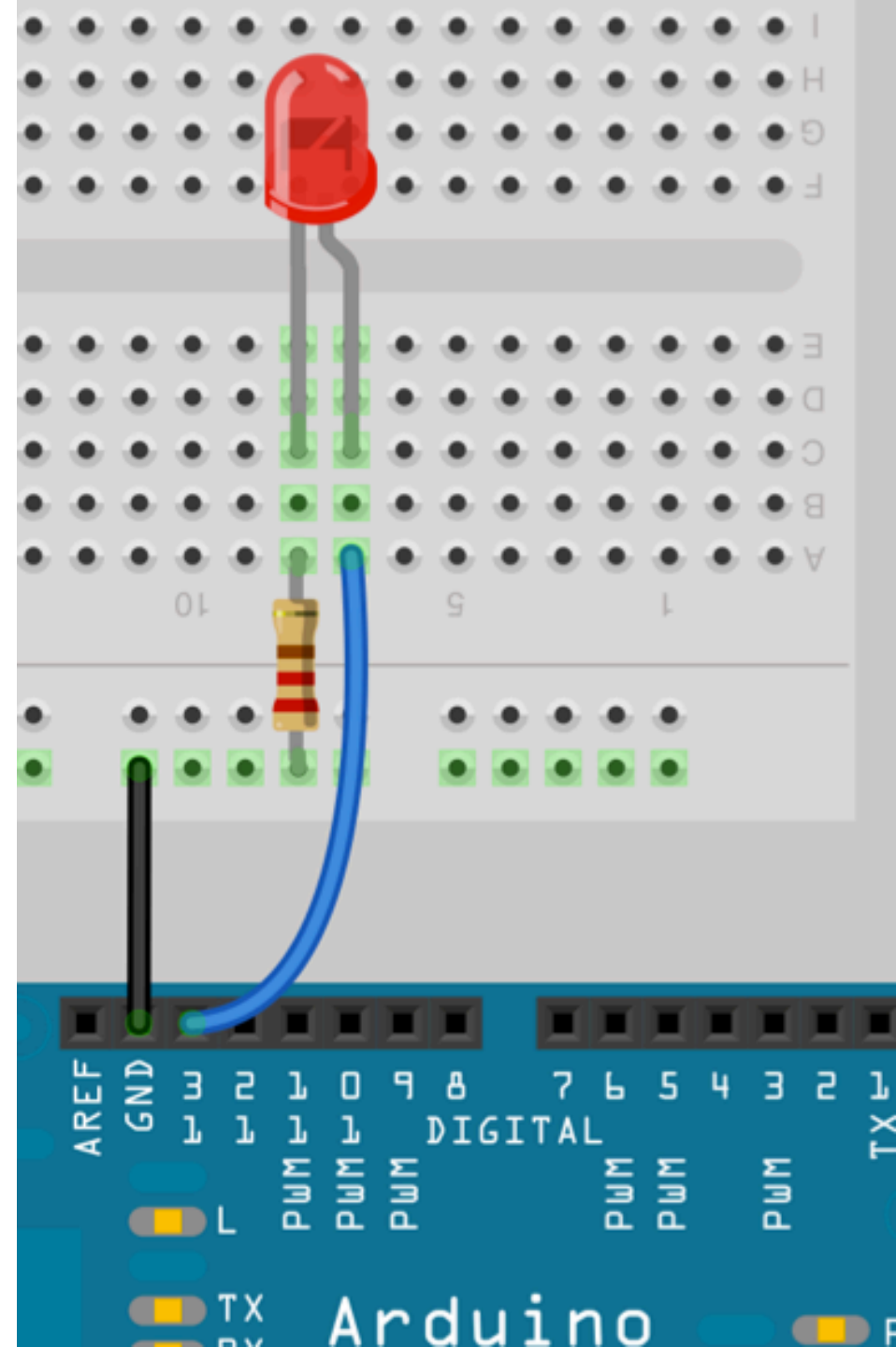
3 – jumper wire to **PIN 13**

4 – PIN 13 to any **terminal row**

LED in breadboard

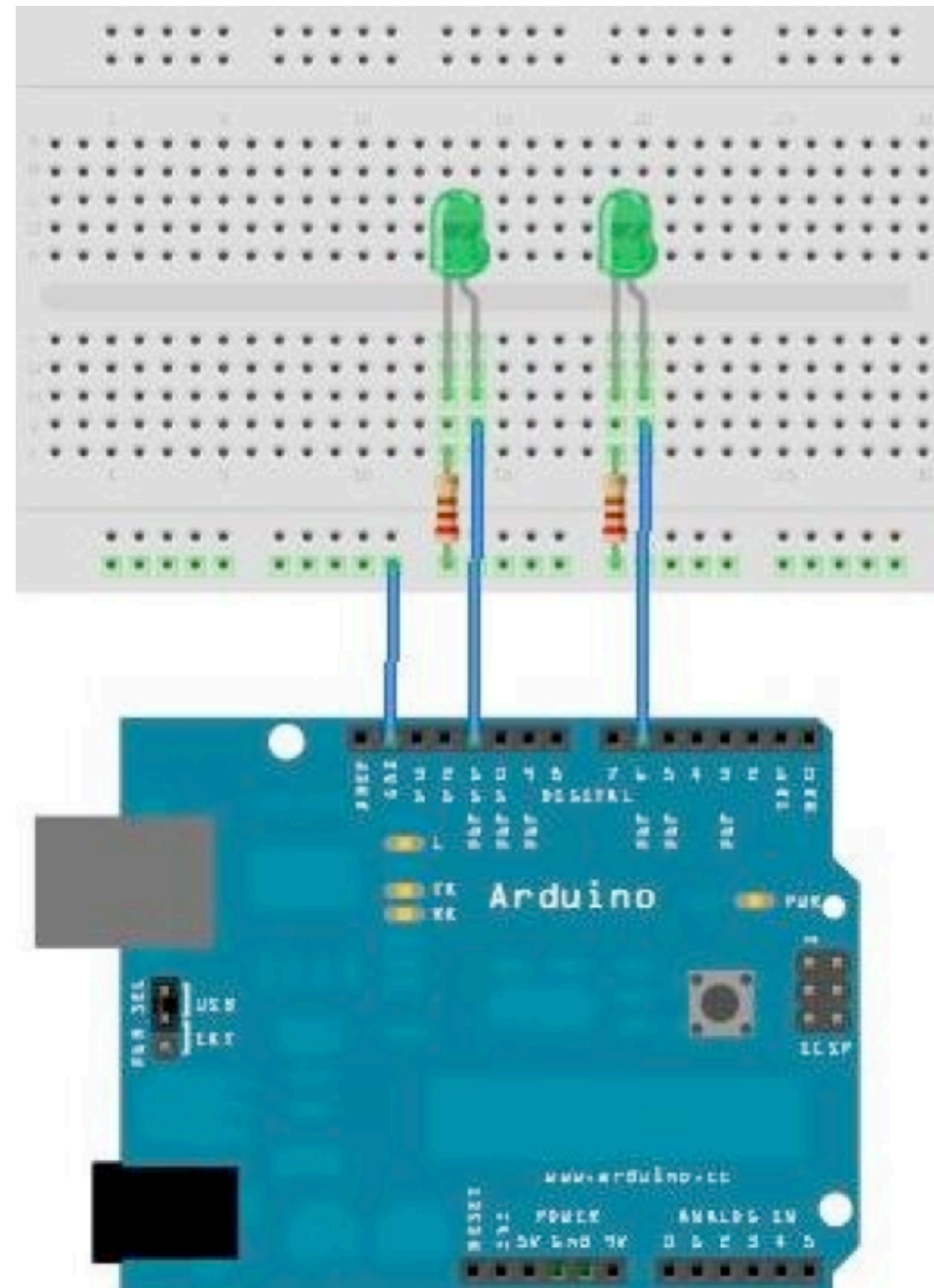


- **220 Ω resistor** between **Pin 13** and **long leg** of LED
- wire connecting **GND** to **short leg** of LED



Multiple LEDs

- each digital pin can power it's own LED
- **resistor** can go on **either** the **GND** leg **or** the **POSITIVE** leg of an LED
- **do it up!**



The Code

```
// Pin 13 has an LED connected on most Arduino boards.
// give your LEDs a name:
int led1 = 11;
int led2 = 06;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led1, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100);               // wait for a second
  digitalWrite(led1, LOW);  // turn the LED off by making the voltage LOW
  delay(100);               // wait for a second
  {
    digitalWrite(led2, HIGH);
    delay(100);
    digitalWrite(led2, LOW);
    delay(100);
  }
}
```

Digital pins

- can **either** be initialized as an **INPUT** or an **OUTPUT** pin
- **OUTPUT** could be to an **LED**, a **speaker**, a **motor**, another device.... **anything!**
- **INPUT** mode **senses** the **electricity** being sent to that pin from your circuit.
- **only** use **digitalPin 0** or **digitalPin 1** for serial communication

Digital pin functions

```
pinMode(ledPin, OUTPUT);  
pinMode(buttonPin, INPUT);
```

- **pinMode**: initialize as INPUT or OUTPUT

```
digitalWrite(ledPin, HIGH);
```

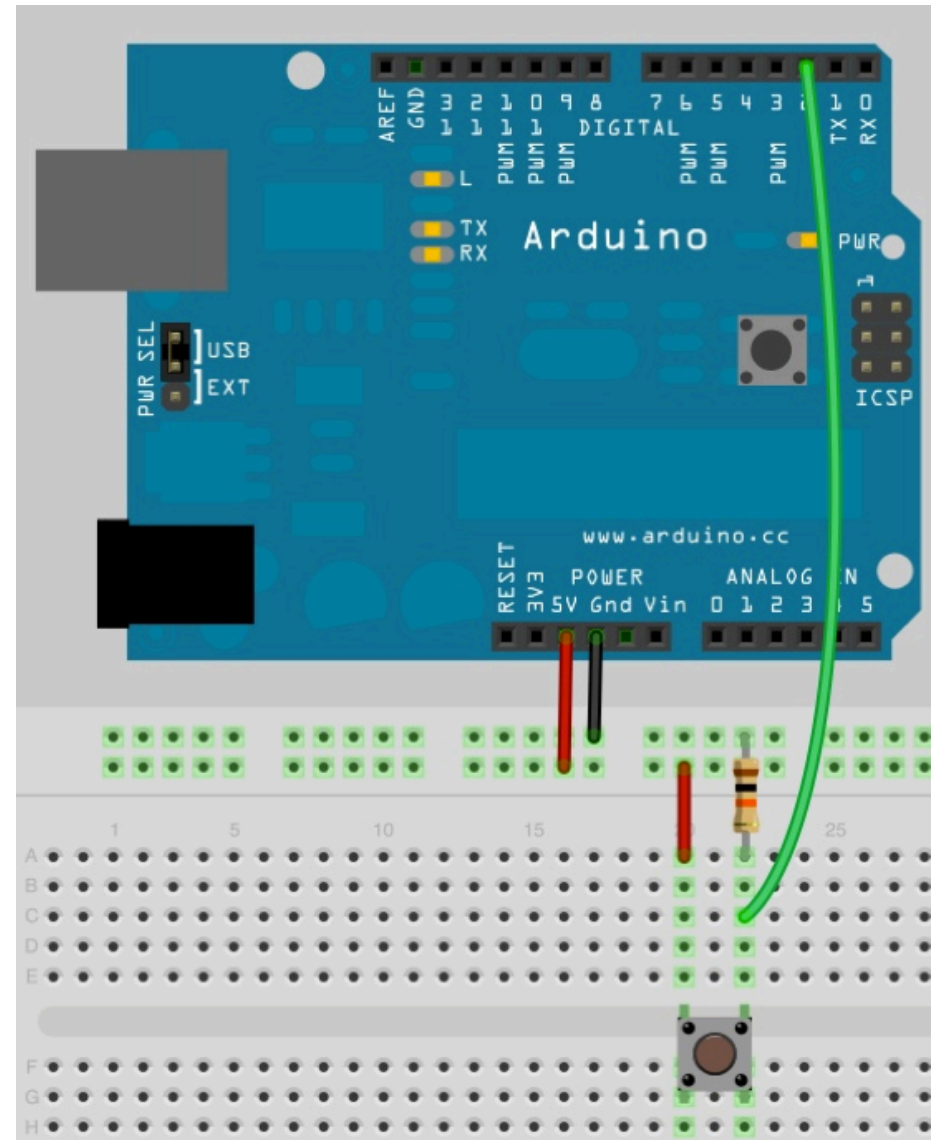
- **digitalWrite**: turn pin **on** (HIGH) or **off**

```
buttonState = digitalRead(buttonPin);
```

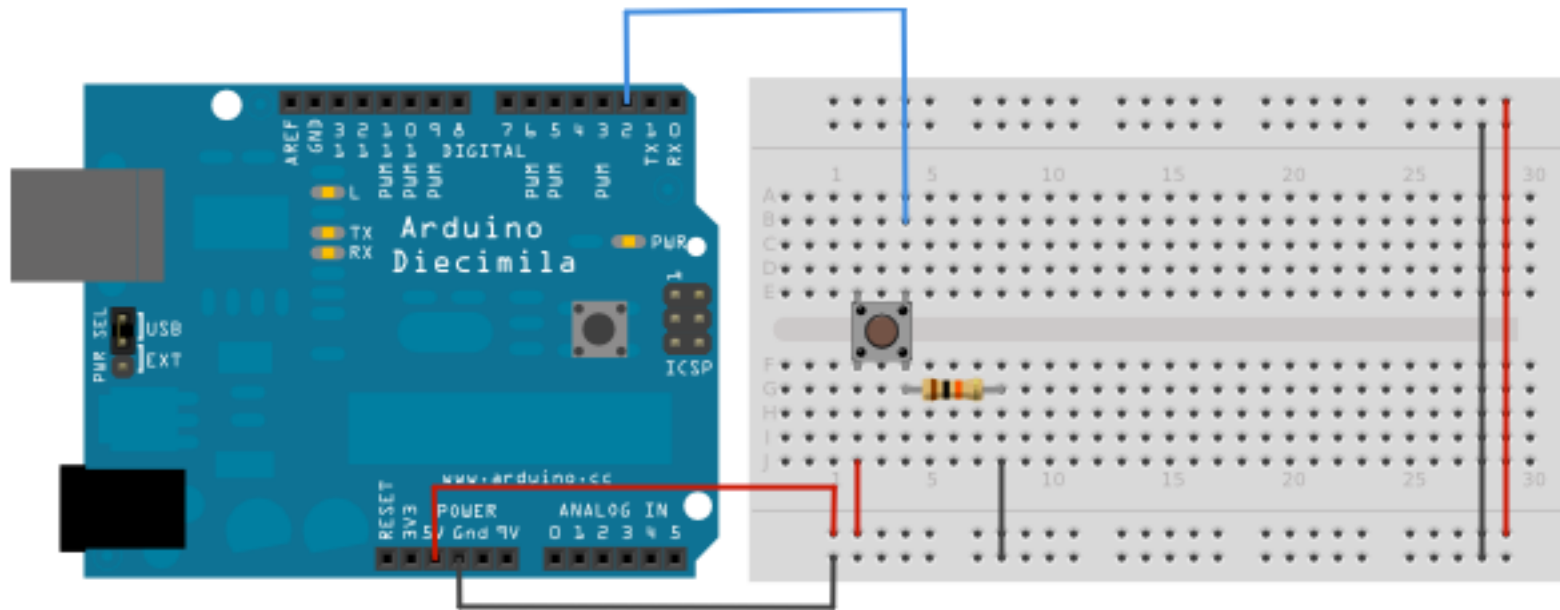
- **digitalRead**: return **HIGH** or **LOW**
depending on what it is sensing (5v or GND)

Digital pins as INPUTs

- **buttons** are perfect as a **digital input sensor** -- WHY?
- a “pull-down” **resistor** makes sure when the button is **not pressed**, the input **pin feels GND**
- when **pressed**, 5v overpowers GND and sends a **HIGH** voltage to the pin



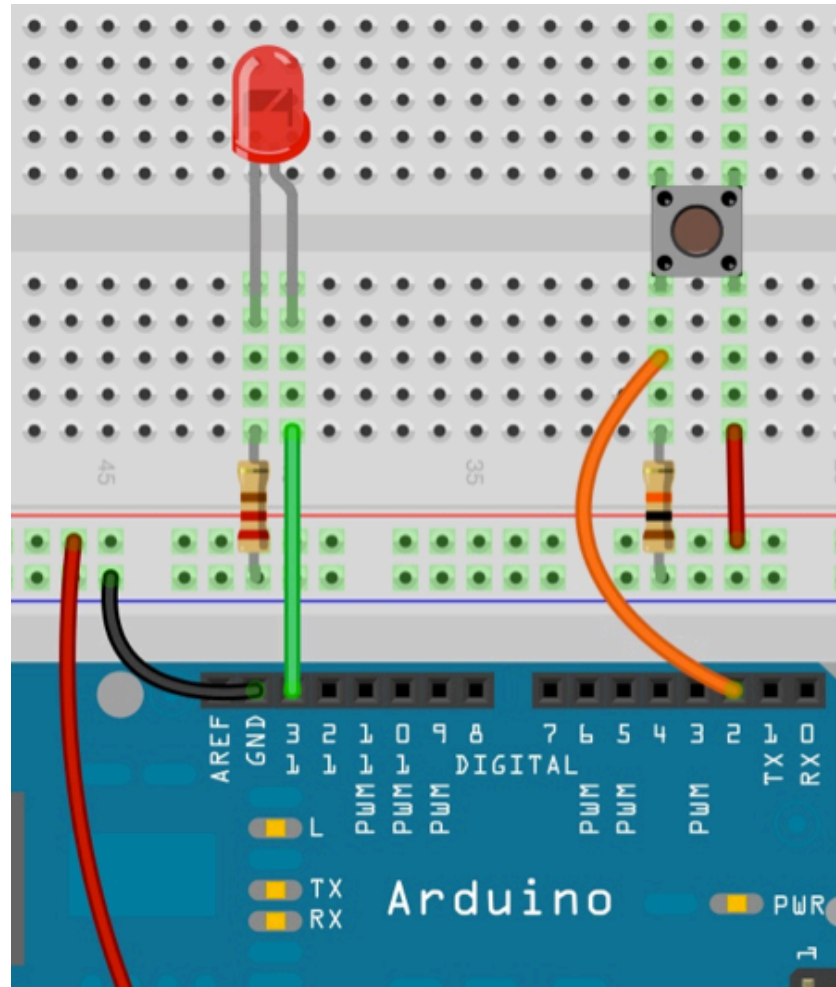
Wire up button



Load up some code

- File > Examples > 2.Digital > Button
 - check out that **if statement!!**
- **Upload** code to your Arduino
- **Press** that **buttonnnnn**

Button and LED



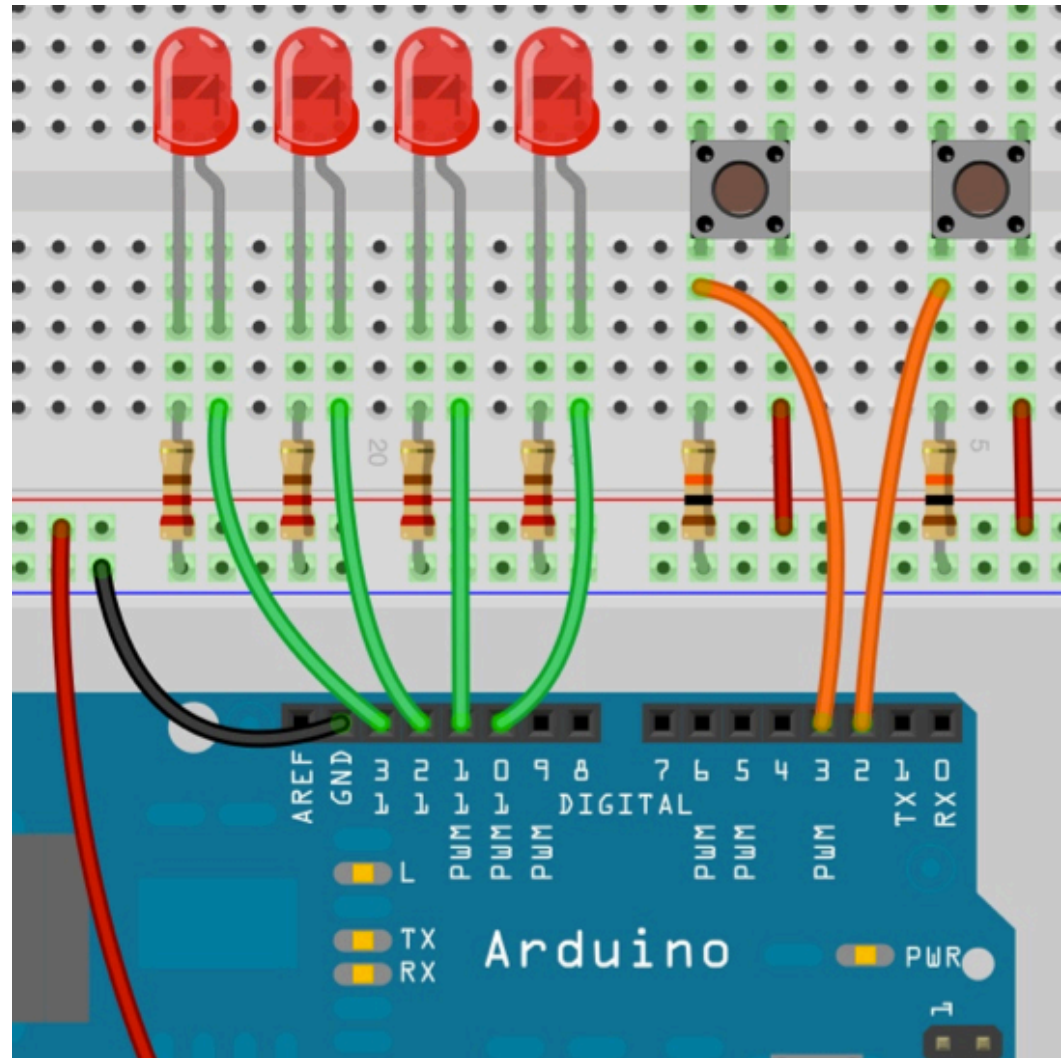
RED wire to **5V** Pin



HOMEWORK!

Button and LED

More
More
More
More
More
More
More
More
More
More
More
More!!!!!!!



RED wire to **5V** pin

Homework

- 1. CATCH UP ON YOUR HOMEWORK!**
2. Get 3 LEDs blinking
 1. Post a video to the blog
 2. Post code and references to the blog
 3. Draw the schematic or use Fritzing and post
3. Control LEDs with at least 2 buttons and 2 LEDs
 1. Post a video to the blog
 2. Post code and references to the blog
 3. Draw the schematic or use Fritzing and post

References

<http://fritzing.org/>

<http://arduino.cc/en/Tutorial/HomePage>

<http://www.instructables.com/index>