Arduino for EVERYONE

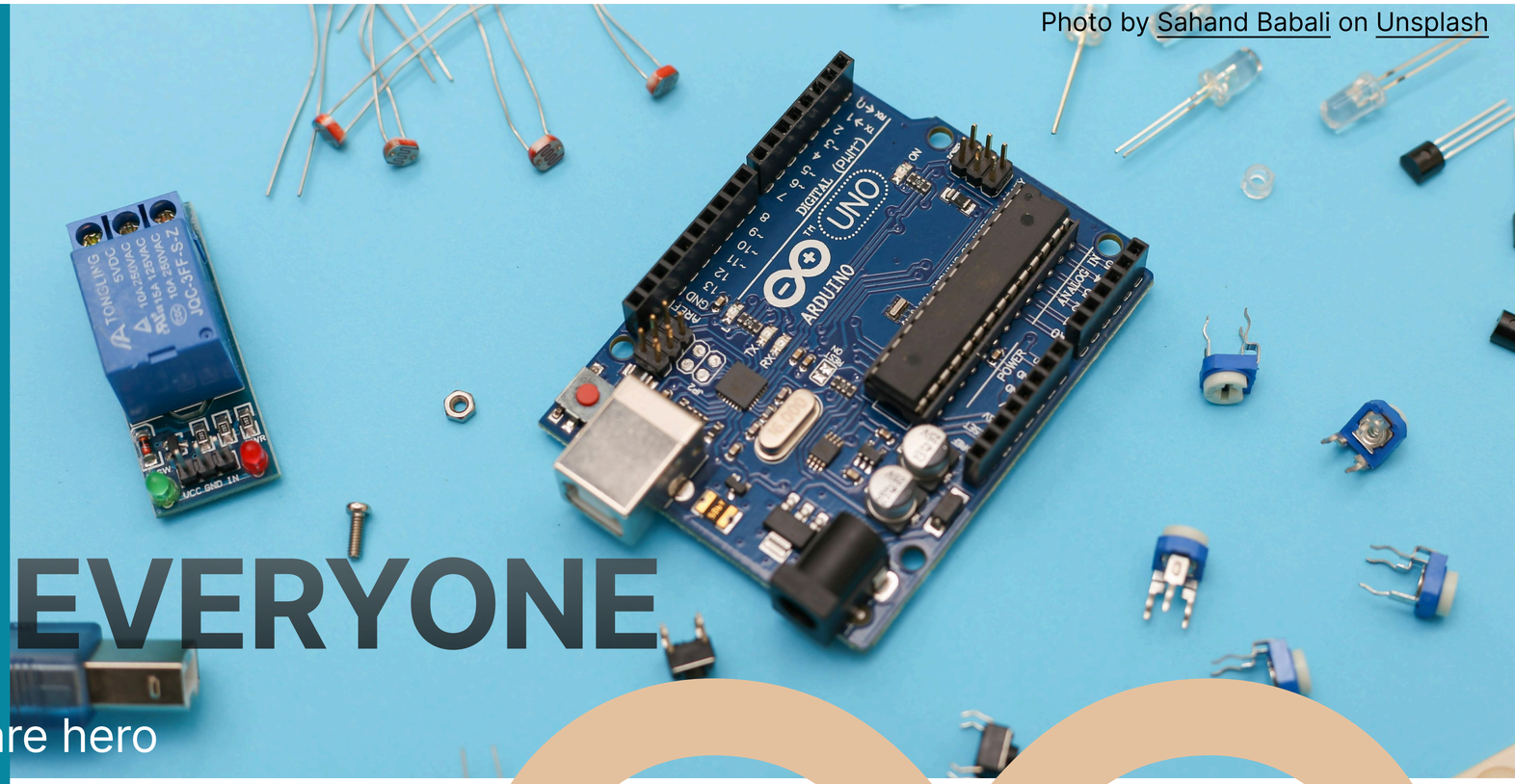


Photo by Sahand Babali on Unsplash

Chapter 1: from zero to hardware hero

LED’s

Rebekah Mugenyi

ARDUINO

alpha v.

Chapter 1:

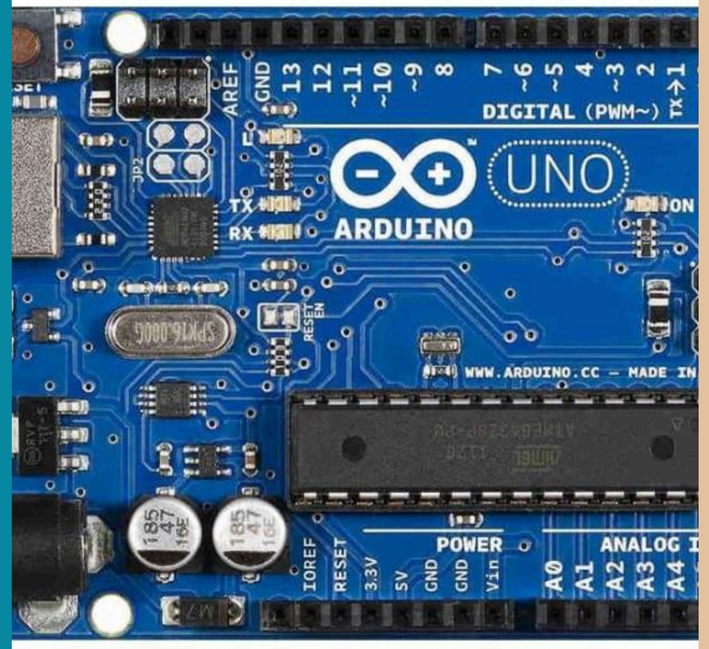
From 0 to Hardware Hero

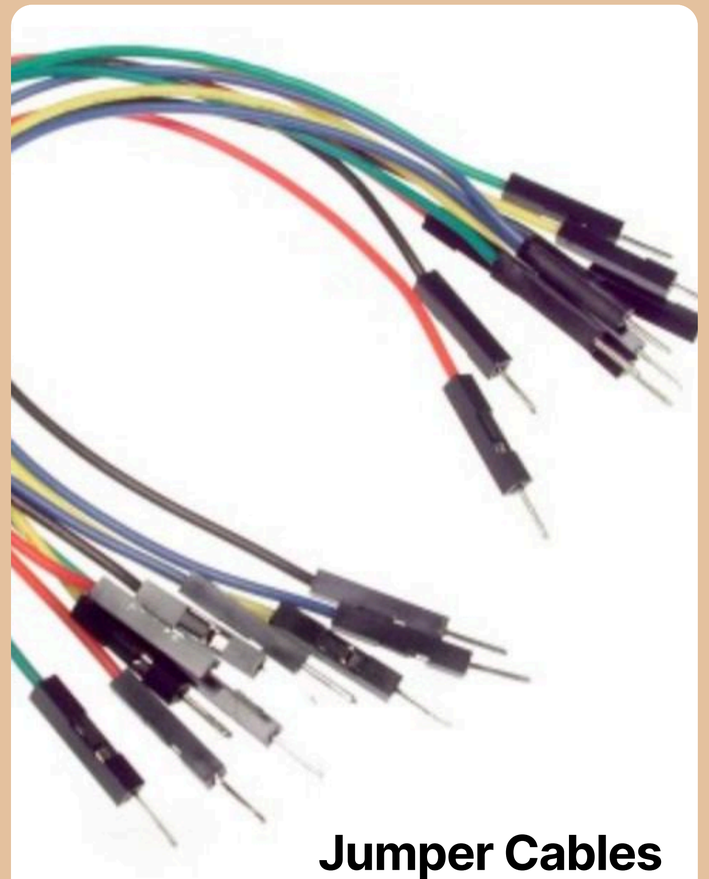
Objectives:

1. A basic overview of common components
2. An introduction to the software
3. A practical - using an LED (programming hardware)

What you need:

Arduino Board (UNO)





Jumper Cables

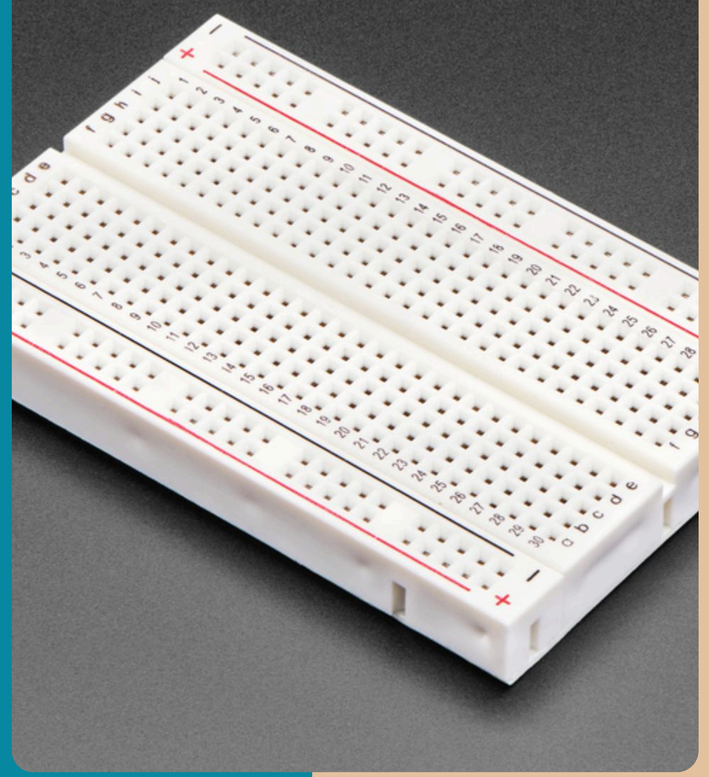
LEDs



What you need:

**(as many as the LEDs are…
also 330 ohms is acceptable.
“pay attention to the colors to tell”)**

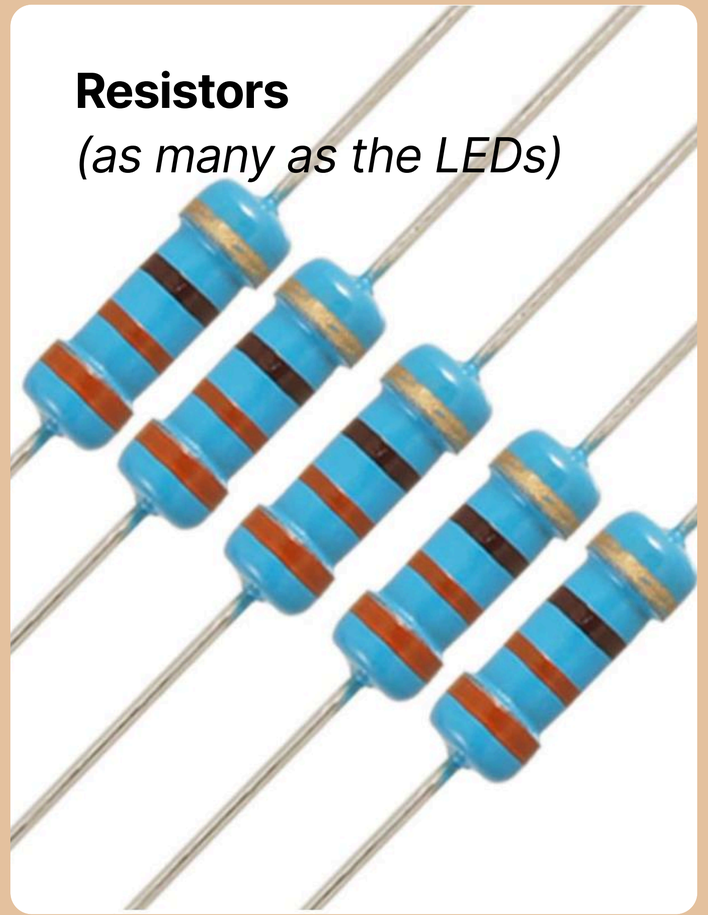
Breadboard



Laptop
(obvious 😅)



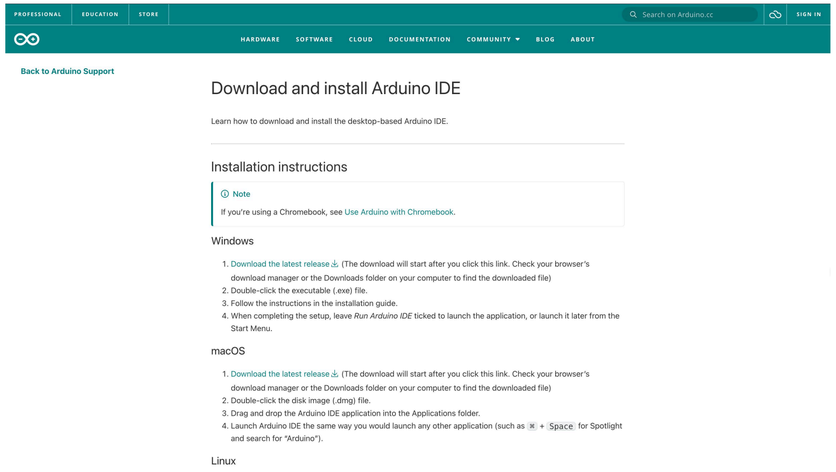
Resistors
(as many as the LEDs)



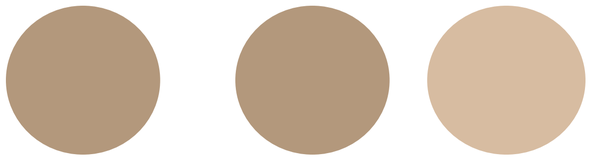
Hero’s Path:

1. The Call to Adventure: Getting the software
2. The First Trial: Understanding the components
3. The Turning Point: Making Connections (pun intended)
4. The Epic Encounter: meeting the code
5. The Final Battle: uploading the code
6. The Triumph: ...watching it work
7. Side quests

1. The Call to Adventure: Getting the software



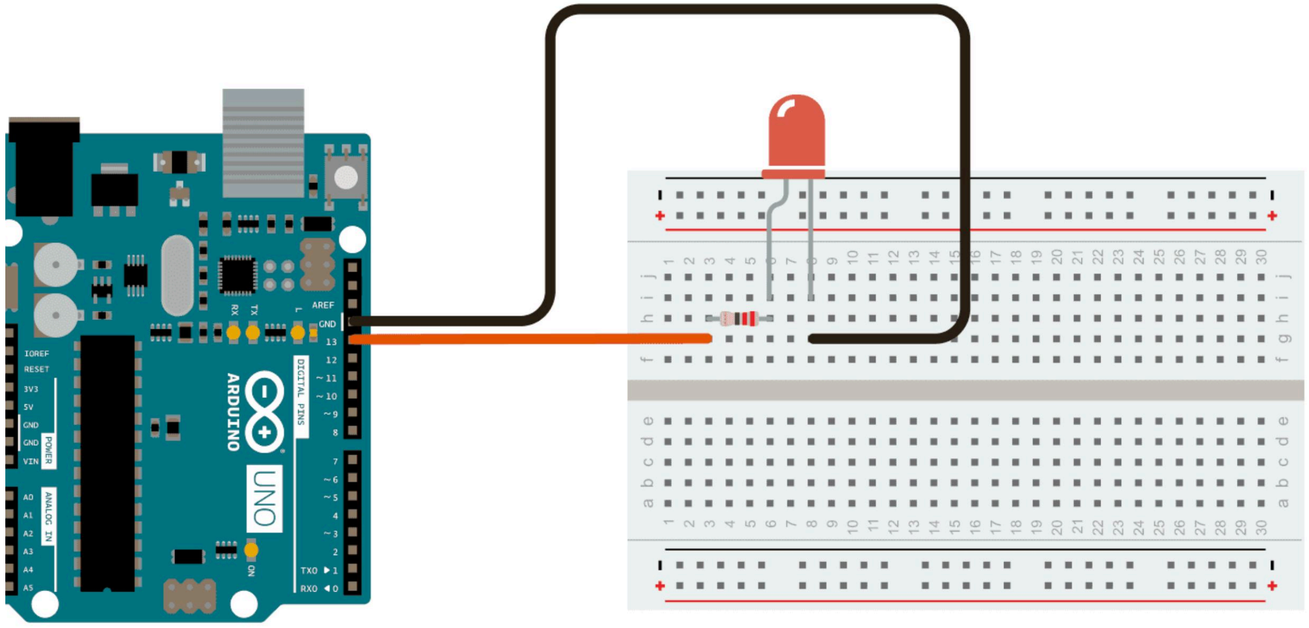
https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE





2. The First Trial: Understanding the components

1. Breadboard
2. LED (Light Emitting Diode)
3. Resistor
4. Jumper wires
5. Arduino



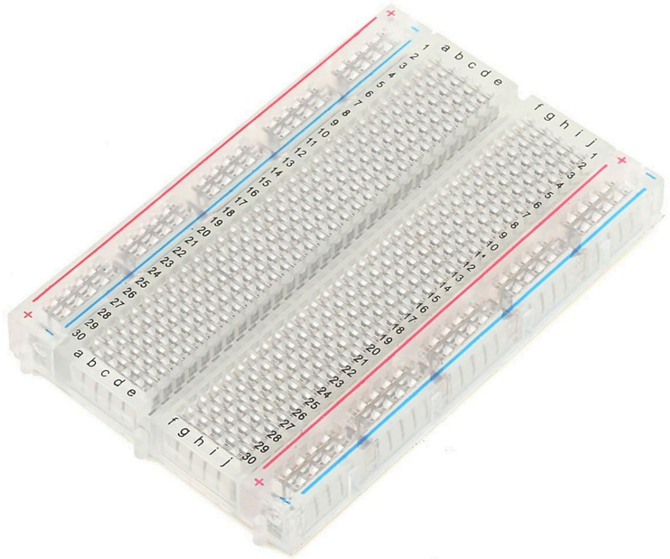
1 What’s a breadboard?



Ignore the top-right picture! 😅 The only thing in common *your* breadboard and the one in the photo have is...

...both are great...one for your stomach, and the other for your circuits.

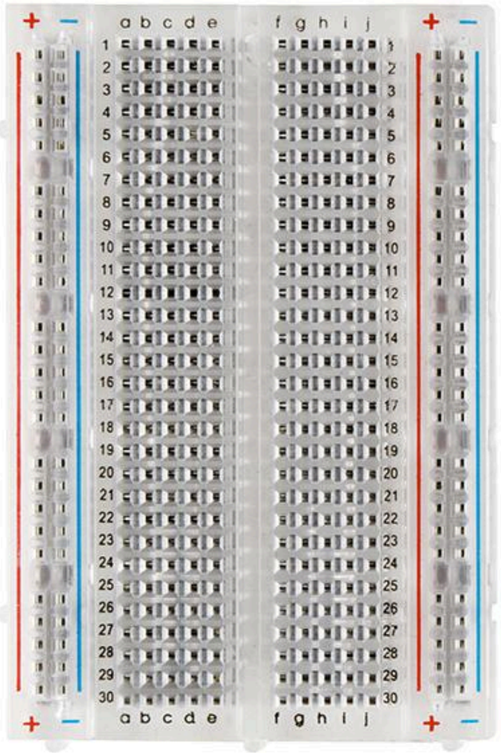
The breadboard you’ll be using is great for making prototypes (as you will soon discover).

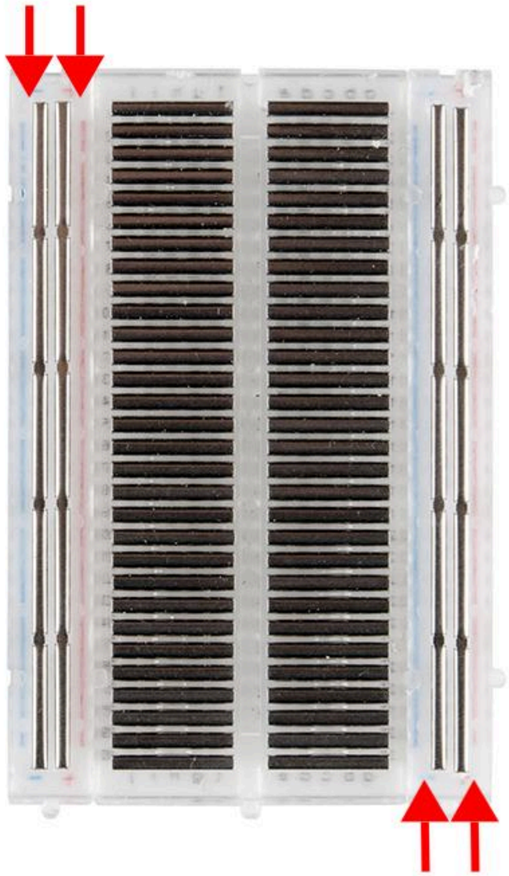


How a breadboard works.

If you turn your breadboard the other way around and removed the white material, you would see the metal rods that run along the breadboard.

This helps explain how they are connected. The general rule of thumb is that the lines of at the extreme ends are connected all the way through (top to bottom as indicated by the red arrows), and for the ones in the middle...its connected in rows (not columns). For example, 1a, 1b, 1c, 1d, and 1e are all connected, but 2a, 3a, 4a, and 5a are not.

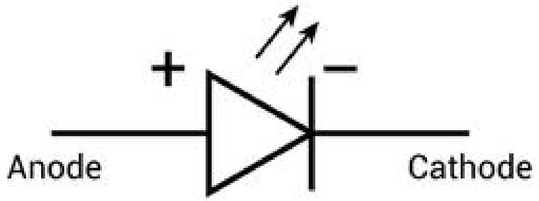




2. LEDs (Light Emiting Diodes)

It’s a light...and we’ll get into the details later. What’s important to know right now is which side is positive (the longer one) and which side is negative (the shorter one) and their representation on a circuit diagram.

Don’t worry about cramming it. First off, you can visit this page whenever you like, and secondly I’ll be sure to remind you what it is the next time it’s encountered. :)





3. Resistors

It’s almost the easiest component to explain!

It *resists* the flow of current.

You can use a multimeter to measure the resistance (measured in ohms)...but a pretty cool hack is to look at the colour of the bands.

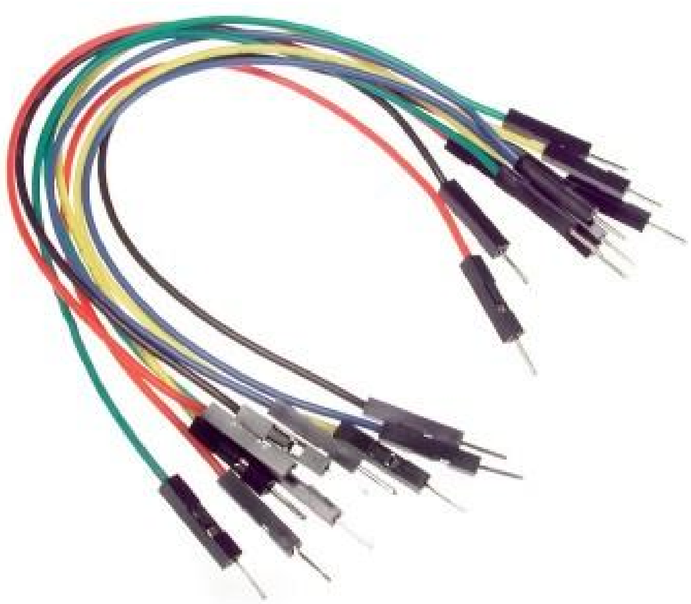


4. Jumper wires

...this ones actually the easiest.

There are different kinds of jumper wires, but they all allow current to flow through them.

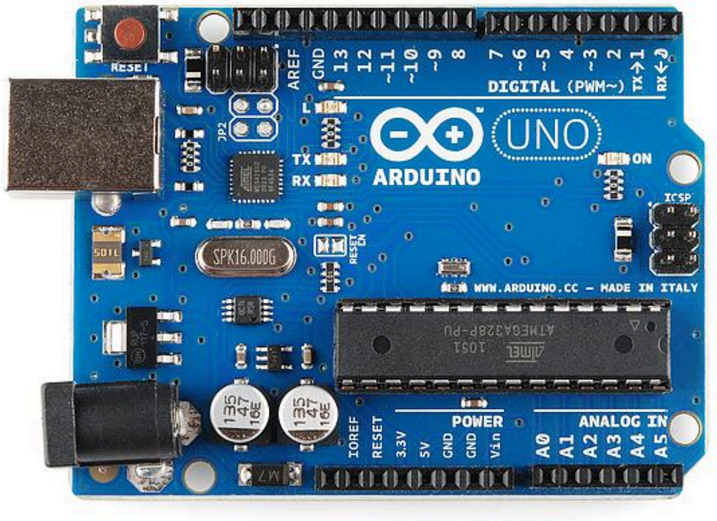
Oh, colours don’t matter...but it’s advisable to connect using a colour scheme.



5. Arduino

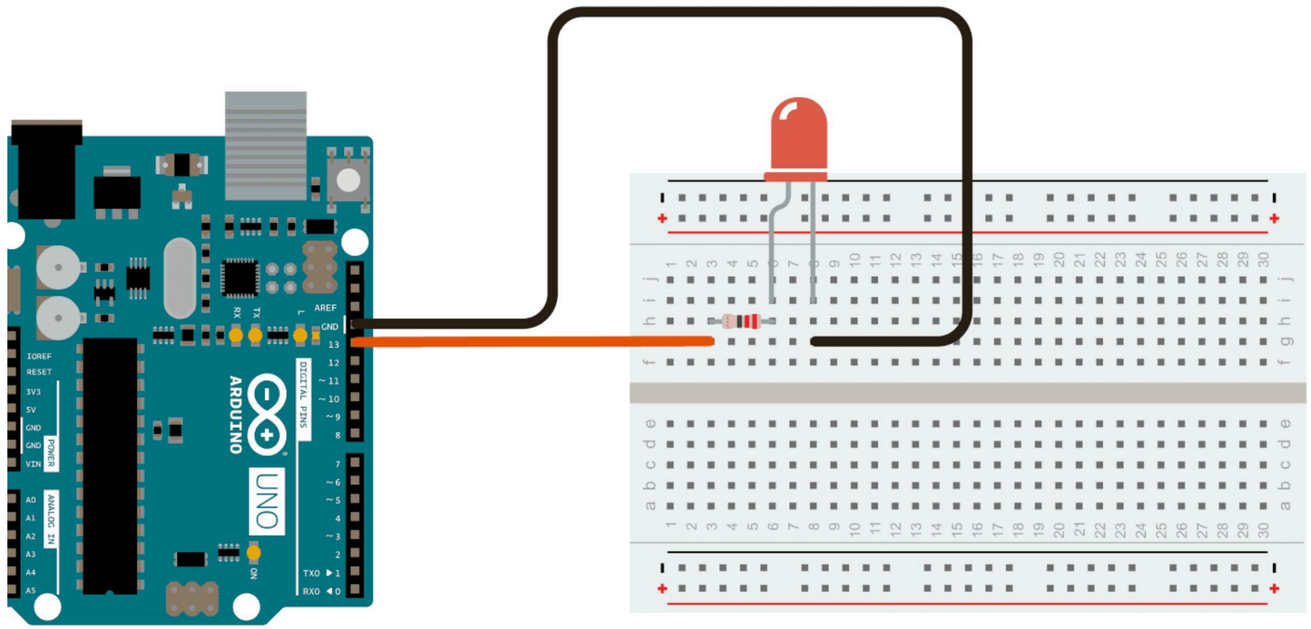
This is the brain of your entire operation.

It’s an arduino. There are many different kinds and we’ll get into later, but for now, what you need to know is that it helps your code “talk to” your hardware.

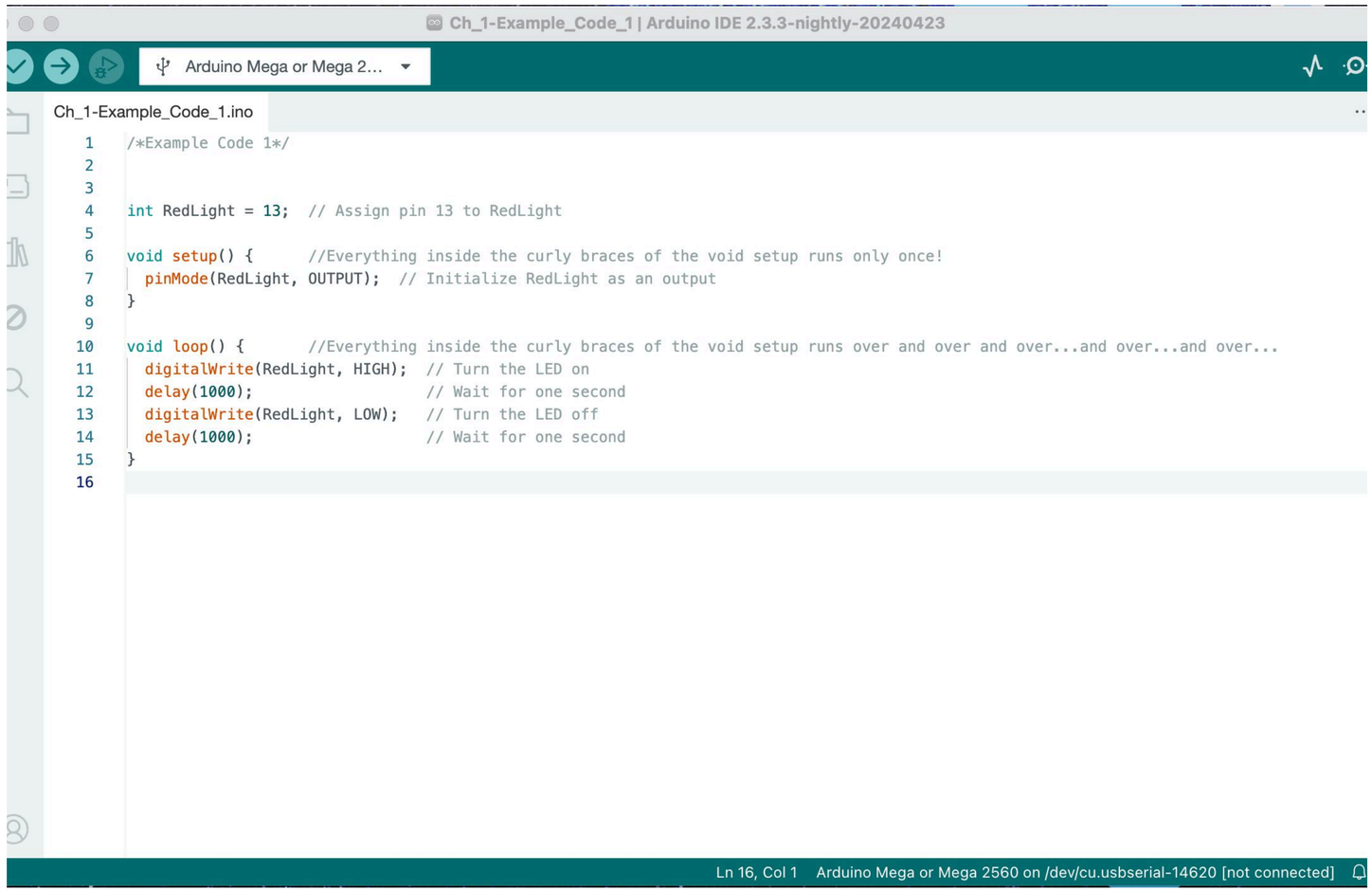


The Turning Point: Making Connections

1. Place the LED into the breadboard...the right way.
2. Connect the negative side (the shorter side of the LED) to GND (negative) on the Arduino
3. Connect the positive side of the LED to the 330 ohm resistor
4. Connect the resistor to pin 13.



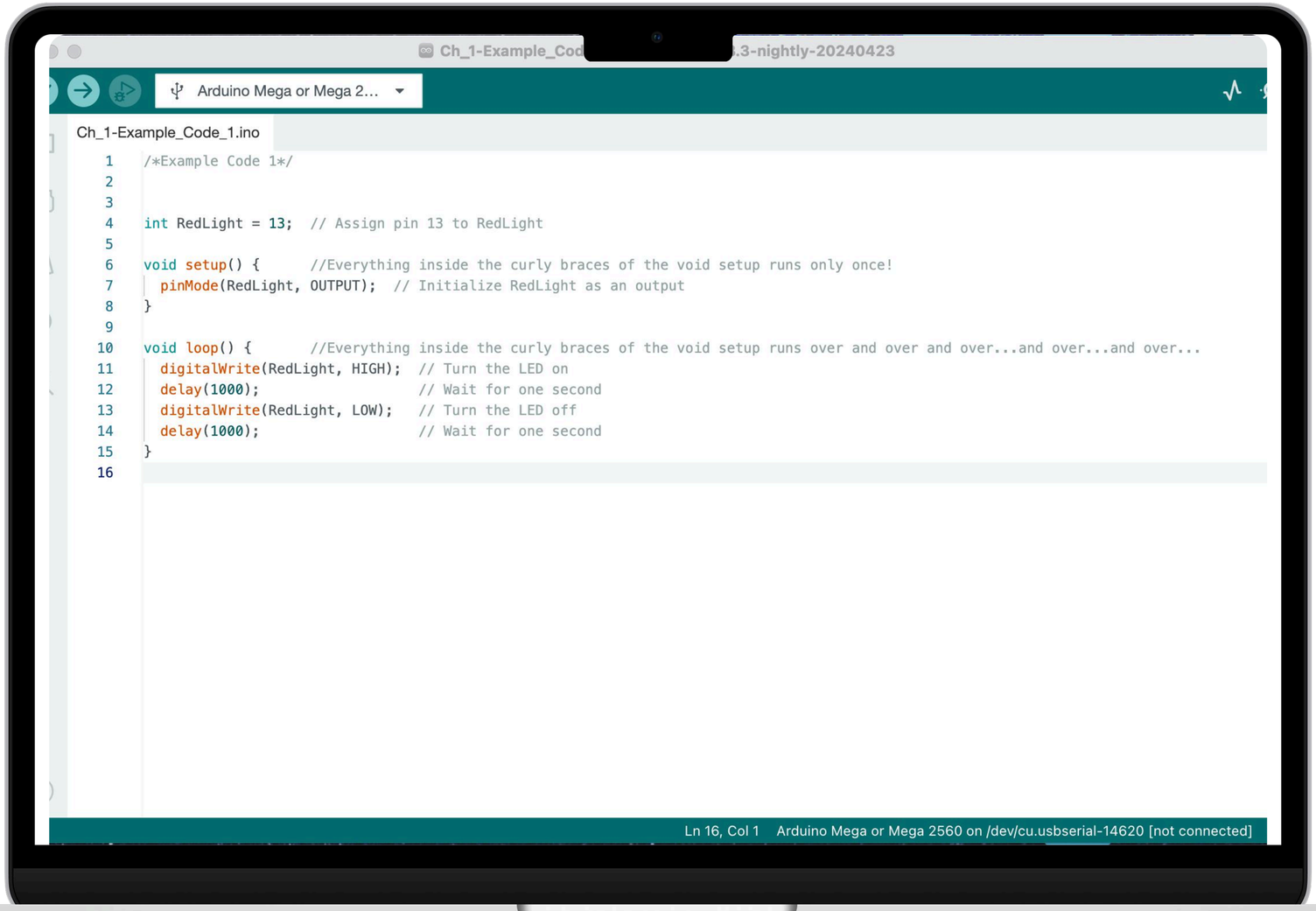
The Epic Encounter: Meeting the code.



You can copy the code from the link below and skip the next page... this section...but that's like buying a box of cake mix. You can never change anything...if you don't know what's inside.

https://github.com/mugenyirebekah/Arduino-for-everyone/tree/main/Chapter%201

Meeting the code.



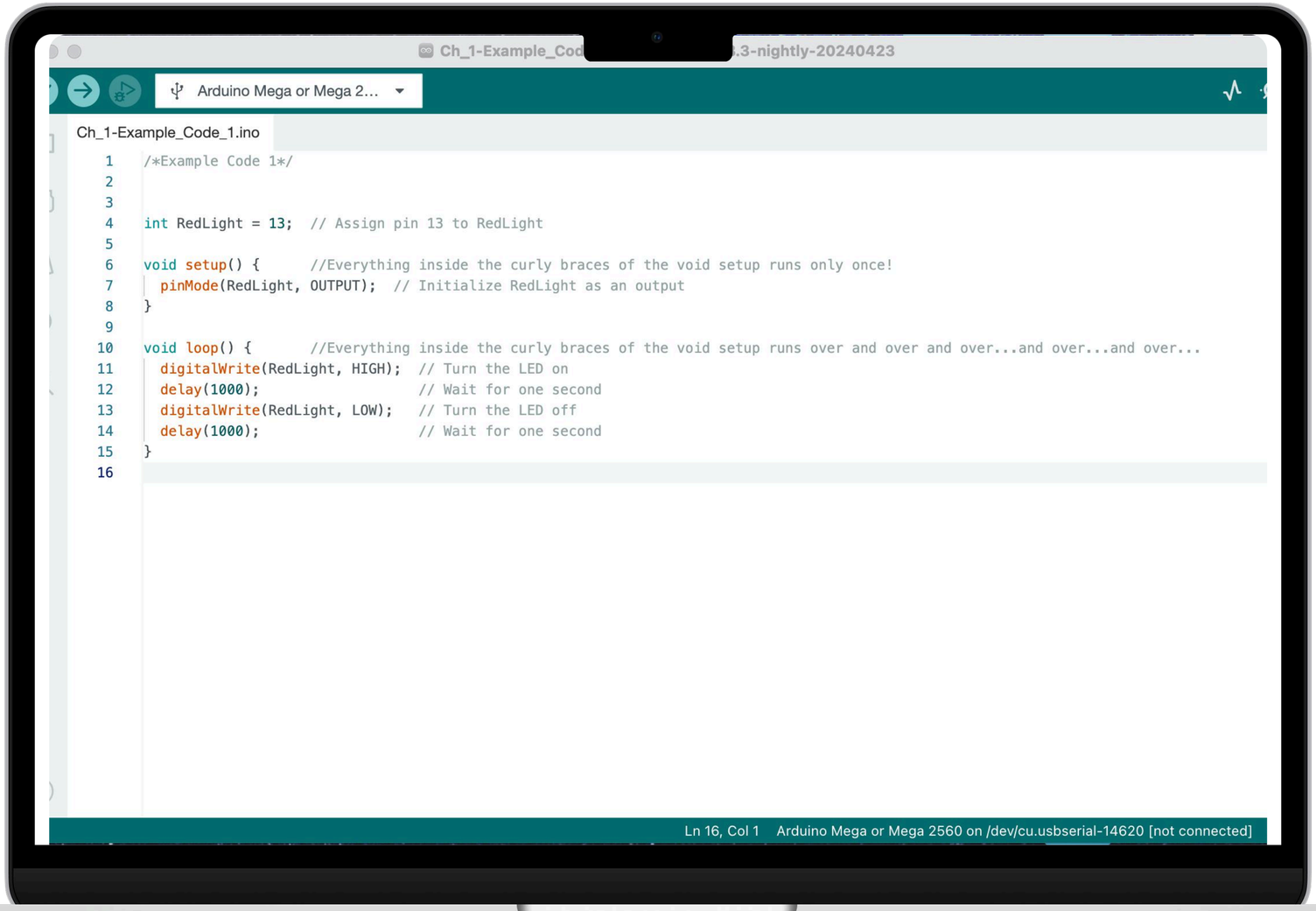
You should know that everything in grey is a “comment”. That means that everything after these specific symbols does not change any of the functionality. You can write whatever you want! Of course, the useful thing is to write explanations...😅.

//

/**/

Meeting the code.

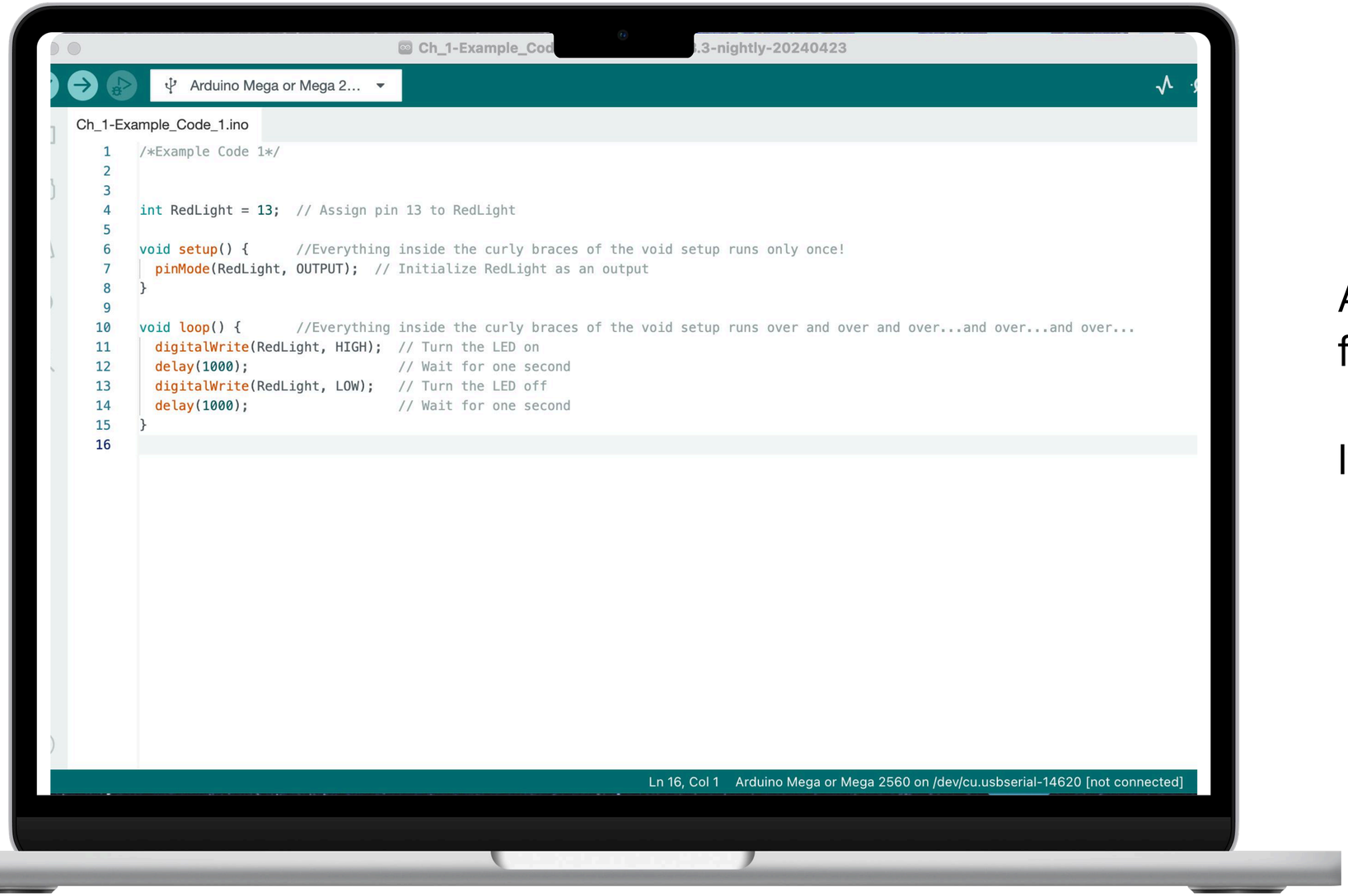
int RedLight =13;



Here we're using int to tell the Arduino that we are working with something (which is the LED) and we are connecting it to pin 13. It can be connected to other pins, but you'd also have to indicate it here.

RedLight is just a word...a variable if you want to get fancy right away... if your LED is a different color, you can change all the occurences of 'RedLight' to YellowLight, or BlueLight...or anything really...even your name!

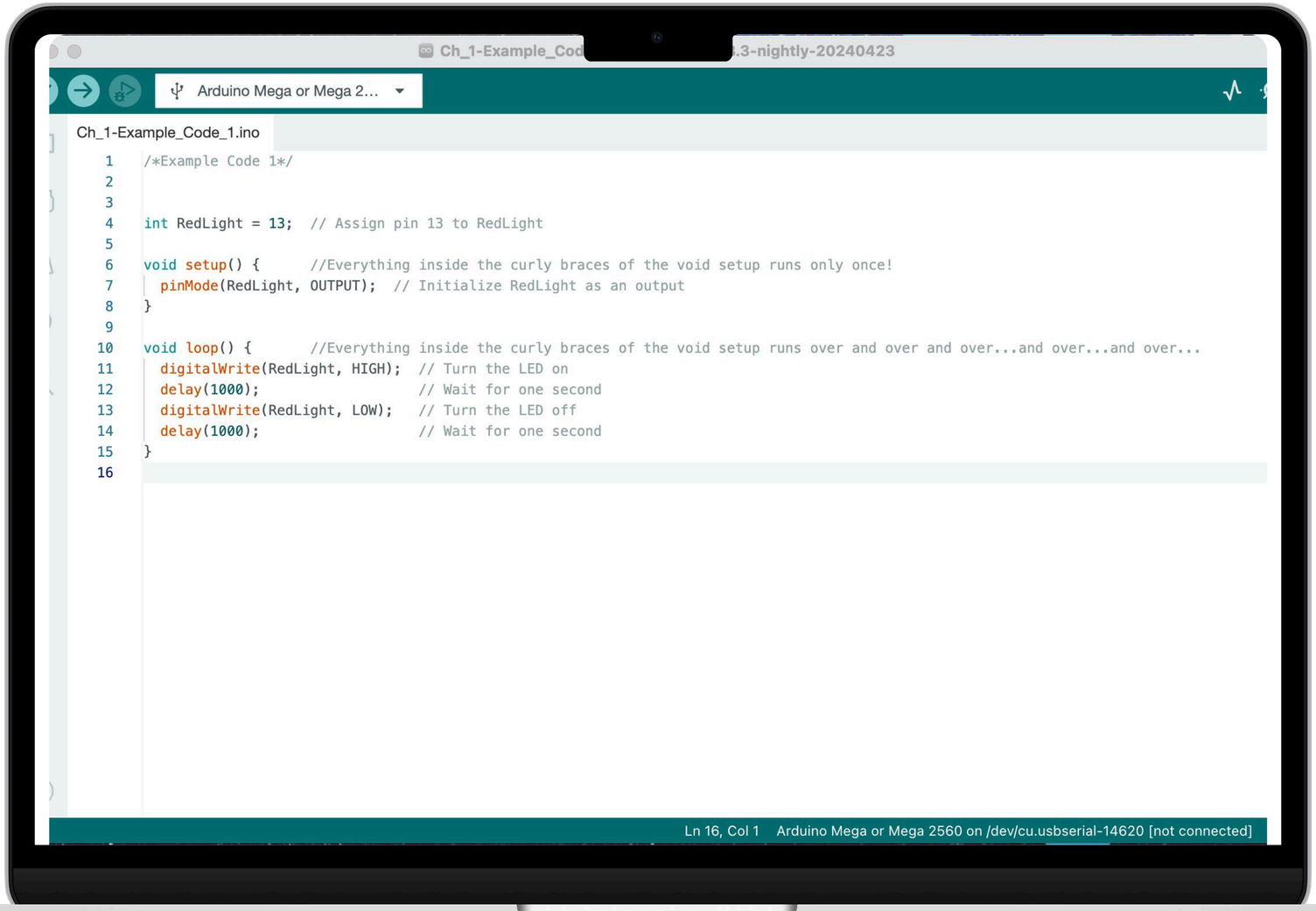
Meeting the code.



A semicolon is to Arduino as a full stop is in English.

It goes *almost* everywhere!

Meeting the code.



void setup(){

All the code written in the void setup will run once! - as soon as you upload the code.

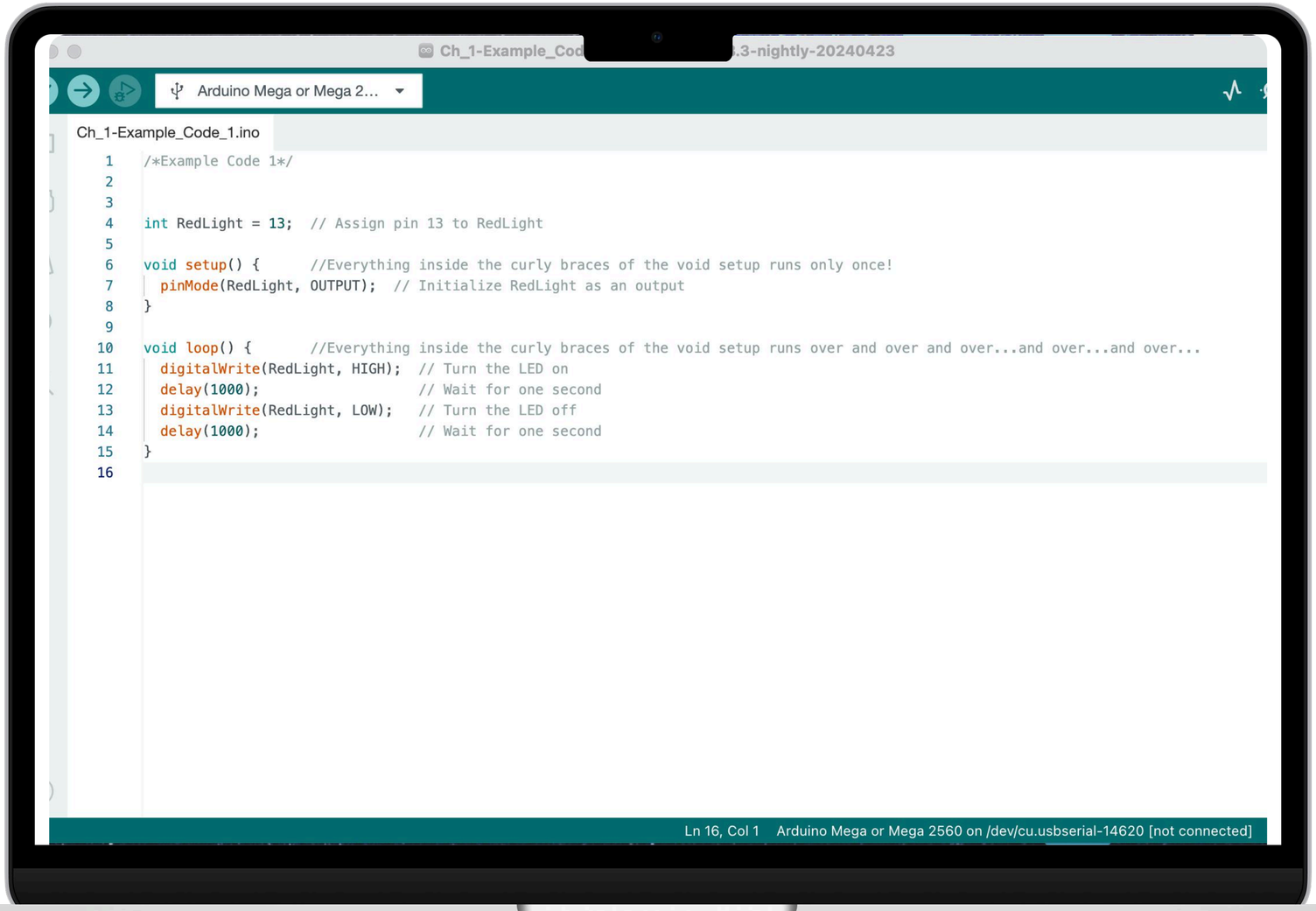
here,

pinMode(RedLight, OUTPUT);

Tells the arduino you are connecting something that gives an output (The LED emits light).

}

Meeting the code.



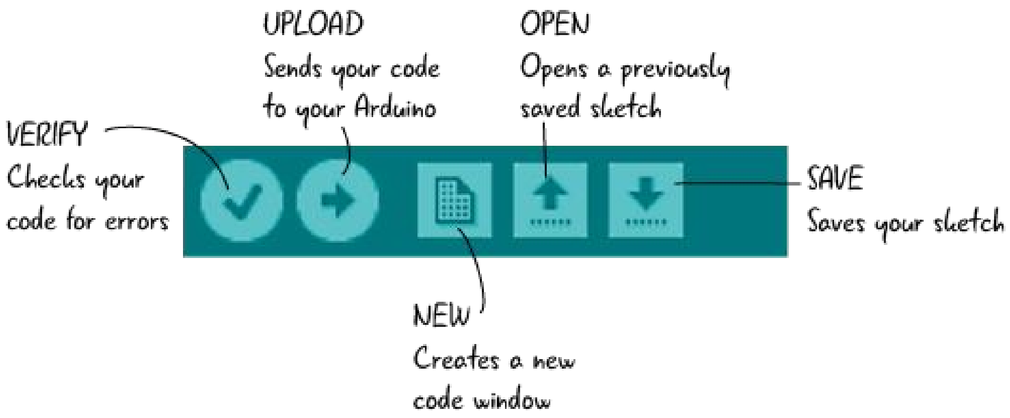
void loop(){

The code you place here keeps on running. In this example, it turns the LED on (“HIGH”) for one second (1000 milliseconds), then turns it off (“LOW”), for 1 second, then turns it on for one second, then turns it off for one second...

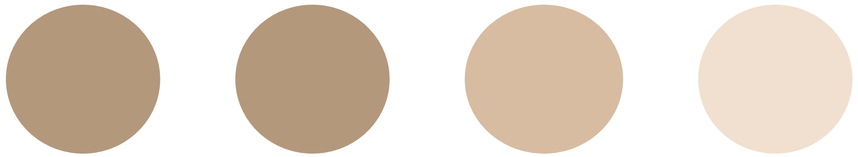
}

...then turns it on for one second, then turns it off for one second, then turns it on for one second, then turns it off for one second...

6. The “Final” battle (uploading the code)



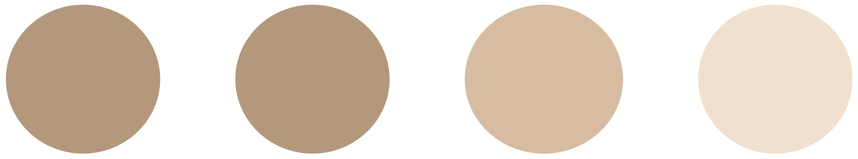
Connect the arduino to your computer and click ‘upload’.





7. The Triumph...watching it work







Side Quests

These are listed for you to try out on your own.

- **Change the delays and see what happens**

  Can you make the LED go off for one second and on for three seconds?

  How about making it blink on and off for a duration of half a second?

- **Try adding more LED’s**

  Can you add a second light and make sure that they aren’t on at the same time?

  Can you make traffic lights? (one set, then multiple?)

  What cool patterns can you come up with?

FYI

This project is commonly referred to as ‘Blink’ in the Arduino community.

You can get more information here:

https://docs.arduino.cc/built-in-examples/basics/Blink/



“Let me know if this helped.”

rebekahsystems@gmail.com

Rebekah Mugenyi