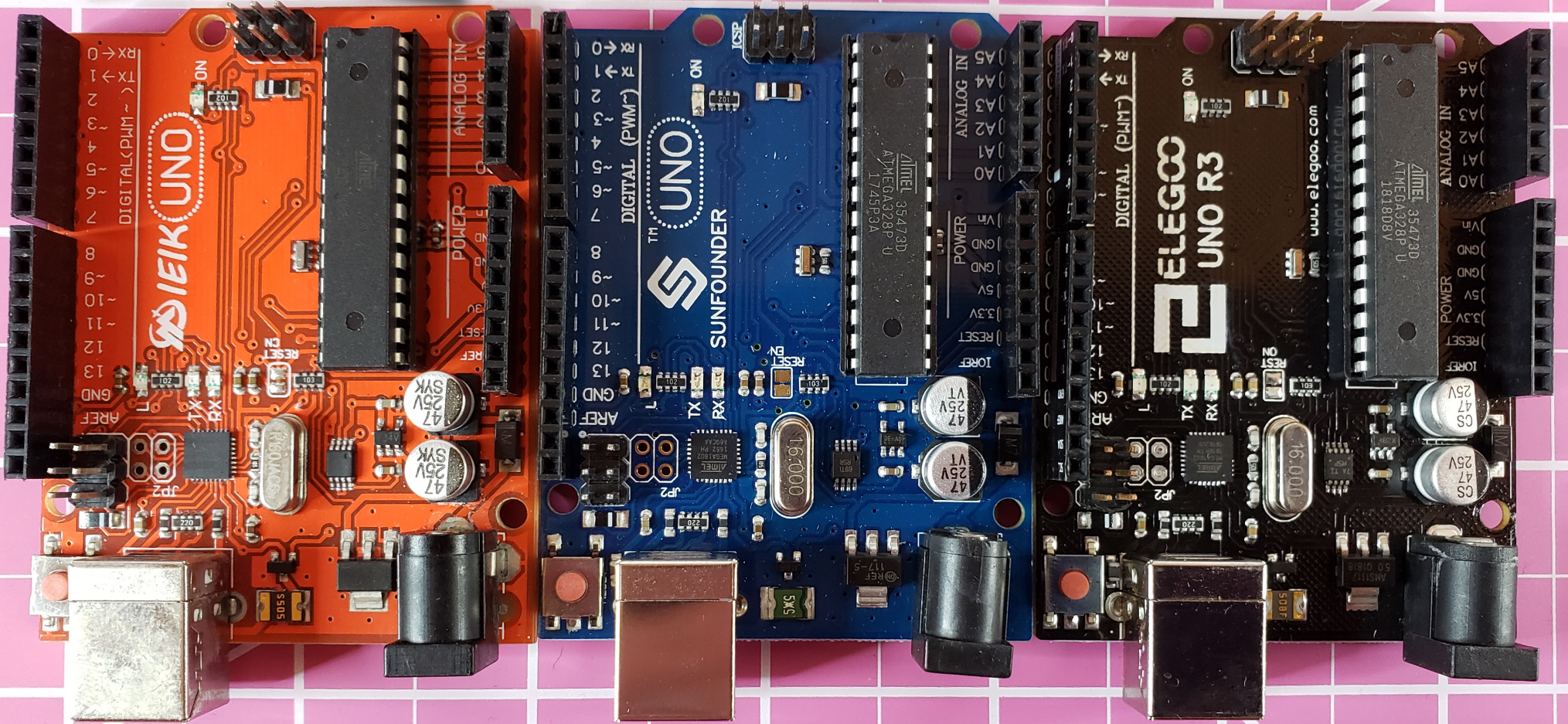


Ch 1.

(Arduino + You)



# Arduino! A microcontroller for everyone.

**You** are not flying around in your flying suit because, well, you have not yet built a flying suit. Over the last few years artists, tinkerers, students and inventors have seen more of their visions come to life to because of the \$5 microcontroller we call Arduino. This little device can allow an artist to create a light instillation with millions of colors that are reactive to the people looking at the instillation, it allows a robotic kitten to judge the distance to an obstacle and, yes it allows those who are compelled to attach small jet engines to a wearable wing to actually fly. This book is a journey through several Arduino projects that will allow you build a set of skills to tackle your own dream projects. I wrote this book so you can build a machine that will make you smile.

The projects in this book are designed to be completed in order. There is nothing stopping you from building any project this book as a standalone project, but the order of projects was carefully thought out to build the reader's confidence across the broad spectrum of devices that can be used with the Arduino.

The way to understand the Arduino at a high level is to look at the board as an "if this happens, then do that" machine. *If this input happens then make this other output happen.* Typically Arduino is looking at some kind of input event (like a switch pressed or a countdown reaches zero) and using the input event as a programmed trigger for an output event like a motor turning on or an LED illuminating.

## Materials used in Ch1:

- + Arduino R3 Uno board
- + USB Type 2.0 Printer Cable
- + A Laptop or Chromebook

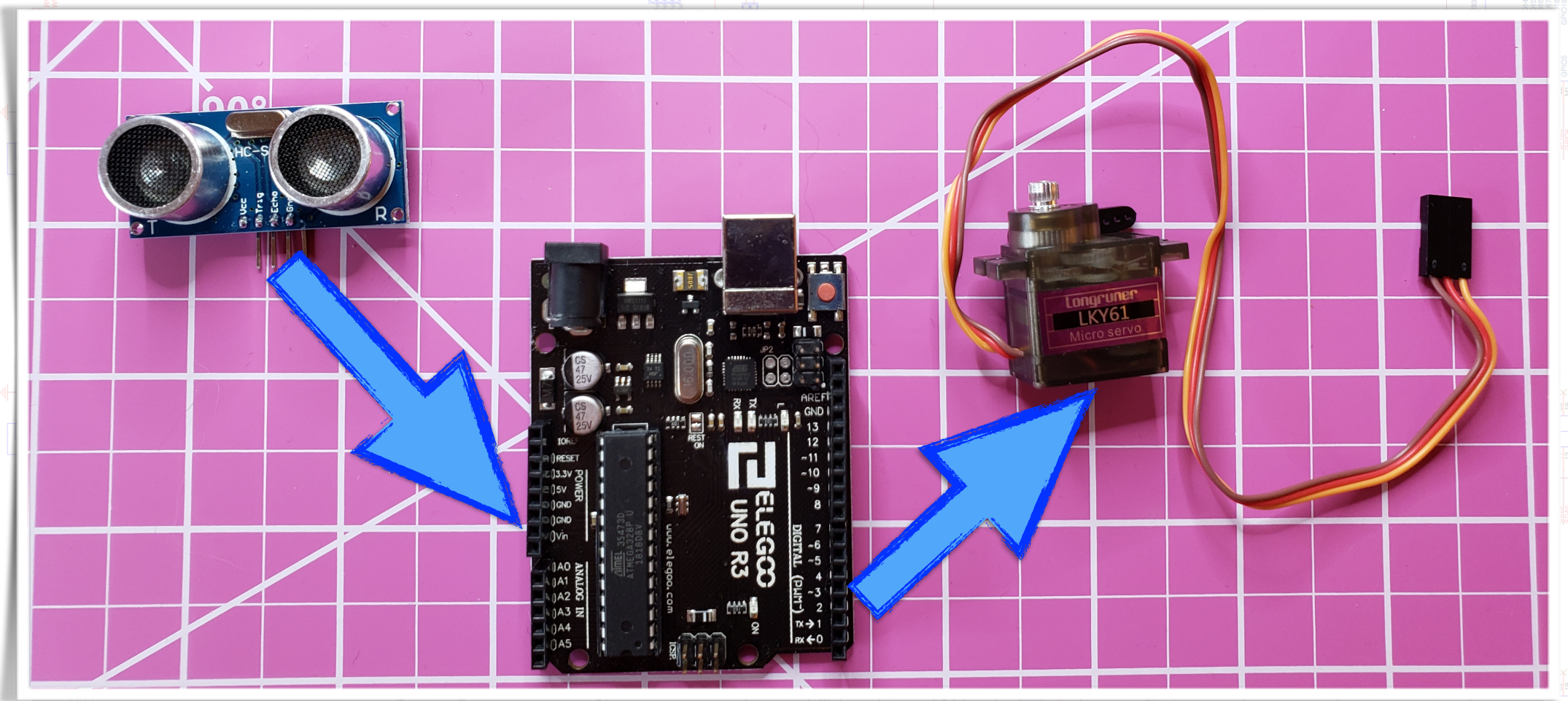
Input → Arduino → Output

Button Press → Arduino → Motor On

Timer Up → Arduino → LED On

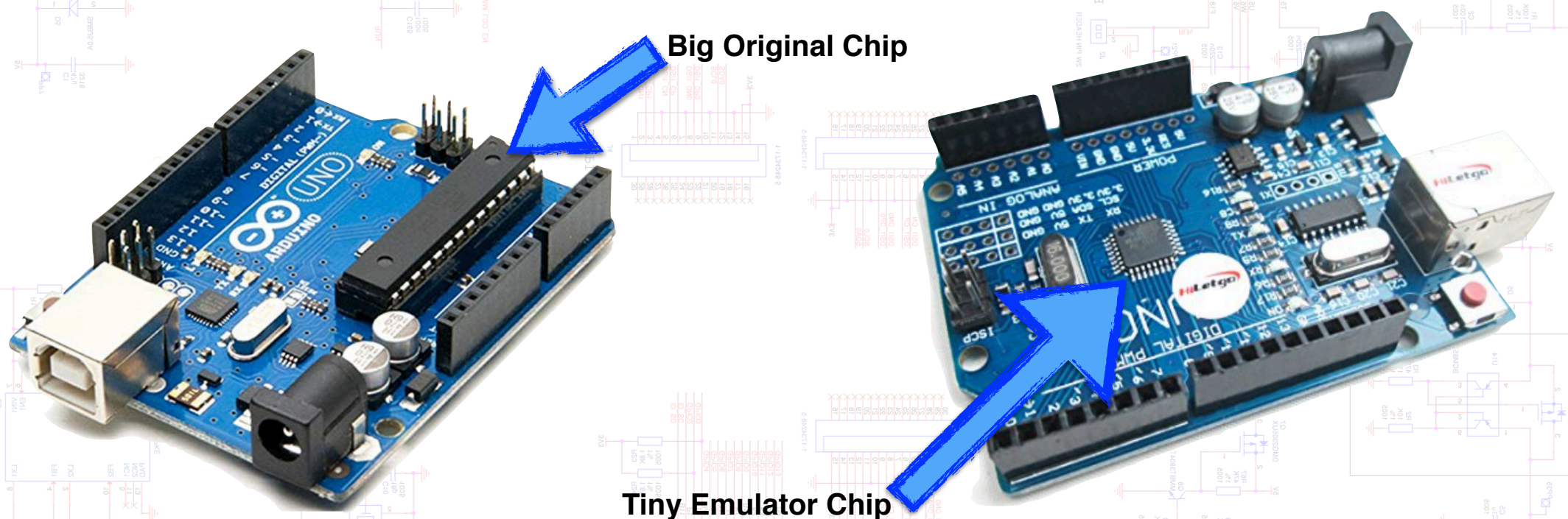
Listed above are the materials used in chapter 1. The first thing that you'll notice when shopping for an Arduino on Amazon is a wide variety of circuit boards that pop up under the search term. This is because the folks who invented Arduino ([www.arduino.cc](http://www.arduino.cc)) made the incredibly generous decision to release their product as open-source hardware and software. This means that they're freely released schematics design data to everyone, so everyone is making their own version of the microcontroller.





**Fig 1.1) Pictured above is an ultrasonic distance sensor (left) that can act as an input device for the Arduino and a servo motor (right) that can act as an output device.**

To add to the confusion in choosing your first board there are several variants of Arduino designs. Technically we are looking for the variant named an Arduino Uno. Other variants are called the Arduino Leonardo, YUN and Mega to name a few of the 20 most common flavors. The Uno was the first design released and is by far the most common model. In the picture on the previous page shows three copies of the Uno that I have you successfully in my classroom over the last five years. The most reliable boards other than an authentic Arduino (\$20) come from Elegoo and SunFounder about \$5-\$7. If you're an Amazon user I would suggest typing in "Arduino Uno R3" and choosing any starter kit that's rated at four stars or above. The only caveat I can add is to look at the picture and make sure microchip on the board looks similar to the one in Fig 1.2. This is because some manufactures are you using microchips that emulate the functionality of the original chip and those sometimes require a separate software driver to be installed on your computer.



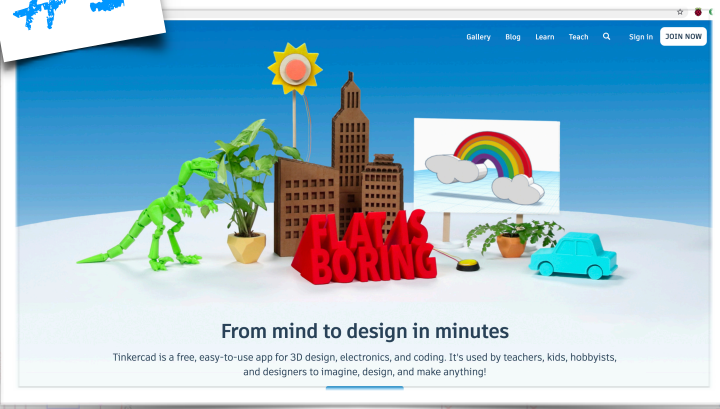
**Fig 1.2) Pictured above are two different flavors of the Arduino Uno boards, both will function identically, but the version with tiny emulator can acquire extra steps to install the Software driver to work correctly on some laptops. When in doubt look for the big bulky microchip.**



# Getting Started

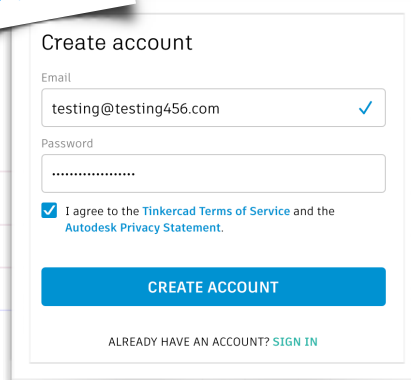
We we will actually start our adventure and become familiar with the basics of using an Arduino before we even physically pick one up! A company called Autodesk just makes a wonderful tool that will allow us to explore the basic functionality of an Arduino with an online simulator. The reason that we want to start here is to first focus on becoming successful going through the basic steps to write code for your Arduino and simulating a successful circuit. After this we will separately explore setting up your laptop in building that same circuit with the physical Arduino board. This is the approach that I use with students to set them up for success before they face some of the common hurtles in starting an Arduino based project. To get started navigate to [www.TinkerCad.com](http://www.TinkerCad.com)

#1



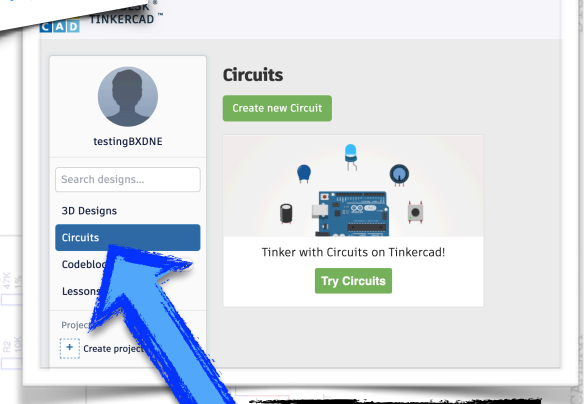
Go to [www.TinkerCad.com](http://www.TinkerCad.com) and click Join Now

#2



Create an Account.

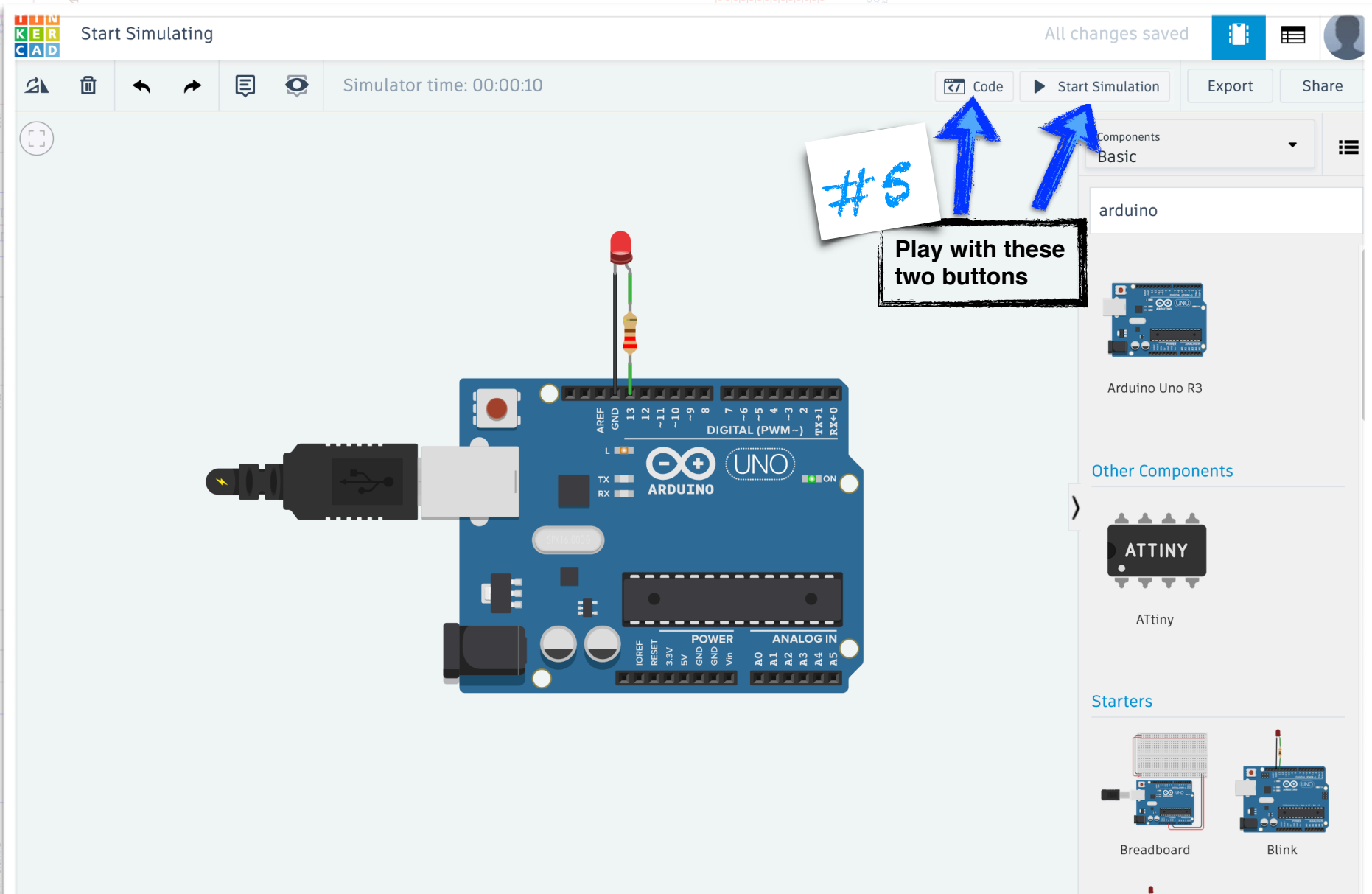
#3



Click "Circuits"

#4

Type "Arduino" under components and select "blink" under Starters



#5

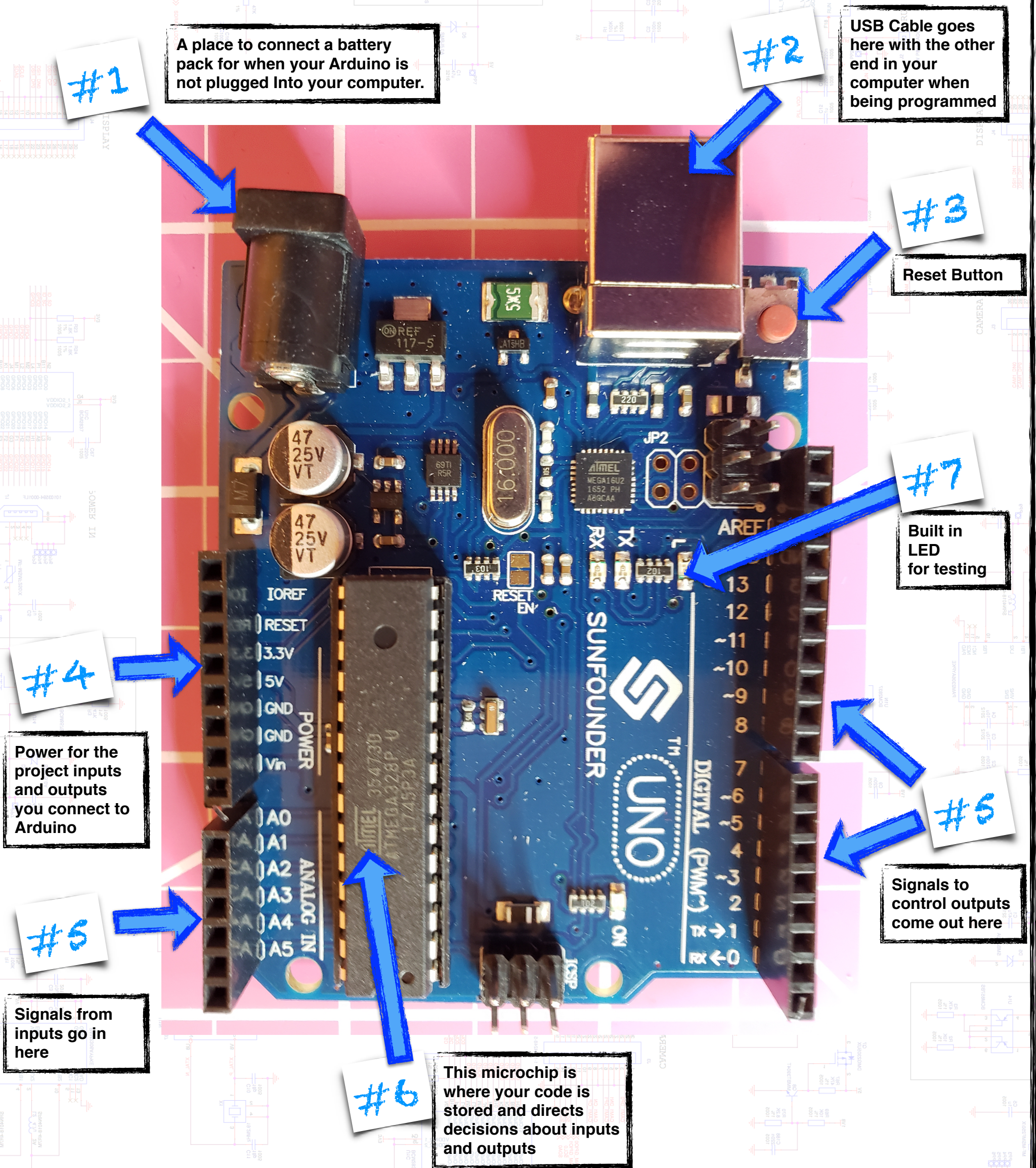
Play with these two buttons

Once you have the "Blink" example program up and running we are ready to start with the physical Arduino board!



# Your Board, Know it Inside Out

Fig 1.3) One of the things that I really dislike is when an explanation is overly technical when simple words will suffice. My goal in this section is for you to gain a basic confidence on how and why the Arduino microcontroller is laid out the way it is. If you are examining a naked circuitboard for the first time this is the diagram for you. Earlier I described Arduino as the part of the machine that lives between inputs and outputs. The diagram below build on that understanding.



As complicated as some people might like to make electronics sound, the truth is a microcontroller is simply a board that allows us the plug-in inputs and outputs into a chip that stores and runs the programs that we write.



# Installing and using the Arduino Interface

We will need an interface to program our Arduino with. Programmers call this interface an Integrated Development Environment (IDE). I just call it a place you type your code and then upload it to the Arduino board. There are four options here for you to choose from. The first three options are specific to the kind of laptop that you are using. The fourth option is to use an online interface that is platform agnostic. Take a moment to review the pros and cons of each option below and choose the one that makes sense to you.

#1

## Arduino on Windows

### What Works:

The Arduino IDE on a Windows machine is stable and backwards compatible with very version of Windows since XP. You can also choose to download an Windows app instead of a program for those of you who like a Windows 10 style tablet experience.

### Frustrations:

This is Windows, land of the force update and missing software drivers. Expect to spend time chasing down bugs when the USB port is not showing up correctly

#2

## Arduino on Mac

### What Works:

The Arduino IDE on a Mac is simple. You can even choose to run the program directly from the downloads folder in the Finder window. The Beta (early preview editions of new versions of the IDE) are usually stable and I have never gone wrong downloading the most current version of the Arduino IDE

### Frustrations:

Users should be reasonably good at using the Finder window. Each Arduino program you run will need it's own folder and if you move a file from one folder to another you can trip up.

#3

## Arduino on Chrome book

### What Works:

The Arduino IDE on a Chrome book is a very lightweight install that does not take up very much memory or require much processing power which can be an issue with other programs on Chromebooks. This is a great option for folks who are already comfortable on Chromebooks.

### Frustrations:

Like many Chrome apps you need an Internet connection. Things just won't work in this browser based experience without a reliable internet connection.

#4

## Arduino Online

### What Works:

The Arduino Web editor does a great job simulating the download versions as a browser plugin. This is the way to get Arduino to work on any computer and to be able to save and access your work from more than one computer

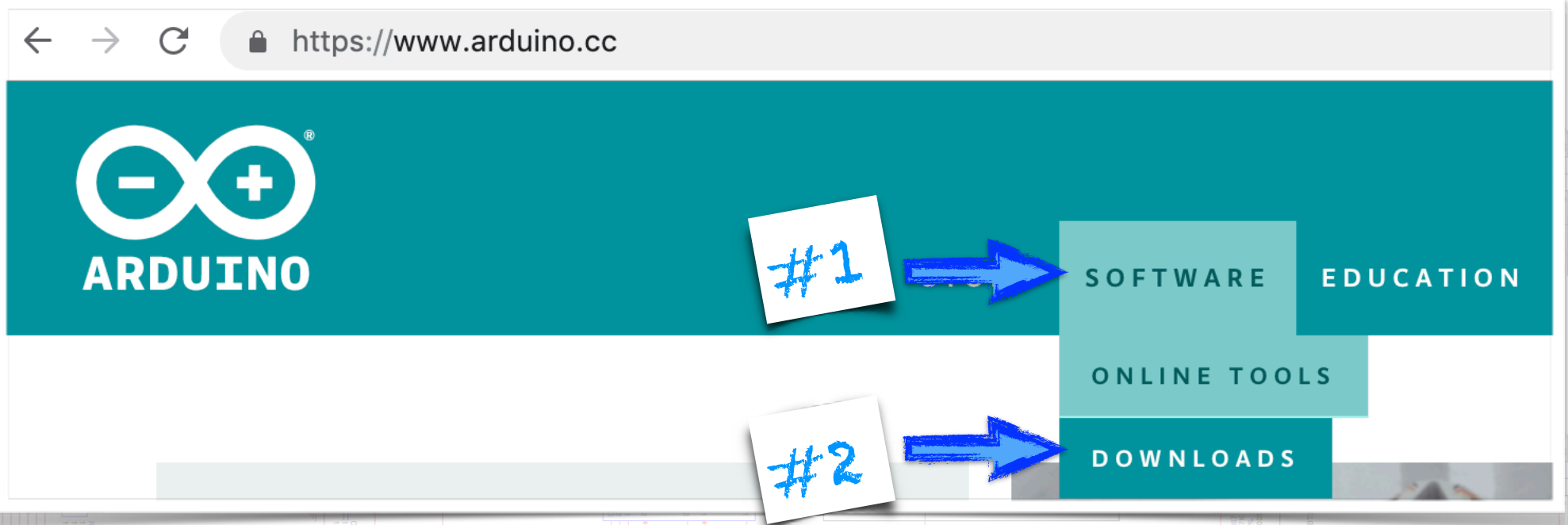
### Frustrations:

Unlike a downloaded IDE you are total dependent on an internet connection at all times when programming your Arduino. You are also trusting the programs you write to an online service that can go down or loose your files.



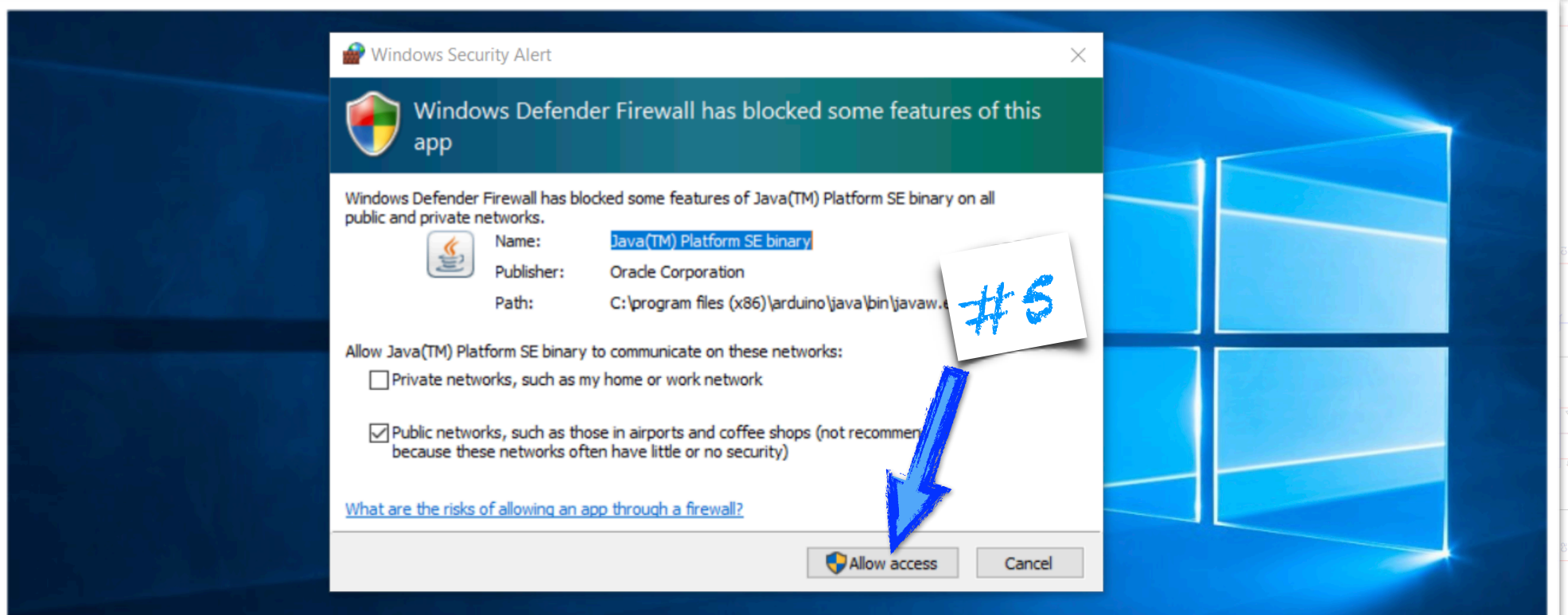
# Installing Arduino on Windows

Start by going to [www.arduino.cc](http://www.arduino.cc). Select 1) "Software" then 2) "Downloads". Click either the link for the traditional program on 3) Windows machines running an operating system XP or newer, or 4) choose to download an app for Windows 8.1 or Windows 10. Don't sweat the version number of the Arduino IDE. That changes quite often and newer versions are always backward compatible with older versions. 5) Allow access for the program to run in the Windows Defender pop-ups.



## Download the Arduino IDE

A screenshot of the Arduino IDE download page. The page title is "ARDUINO 1.8.9". The text describes the software as open-source and available for Windows, Mac OS X, and Linux. Hand-drawn annotations include a blue arrow labeled "#3" pointing to the "Windows Installer, for Windows XP and up" link, and another blue arrow labeled "#4" pointing to the "Windows app Requires Win 8.1 or 10" link. The page also lists download links for Mac OS X, Linux 32 bits, Linux 64 bits, Linux ARM 32 bits, and Linux ARM 64 bits.



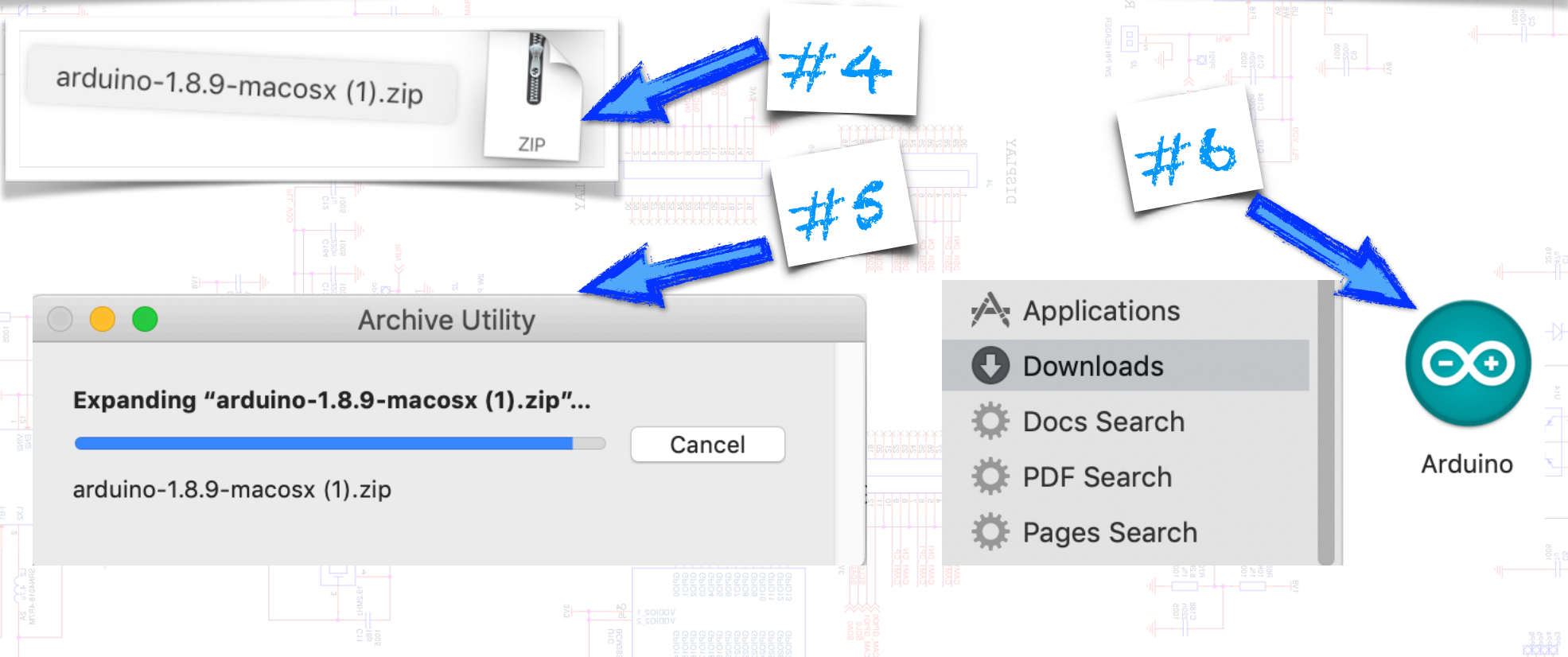
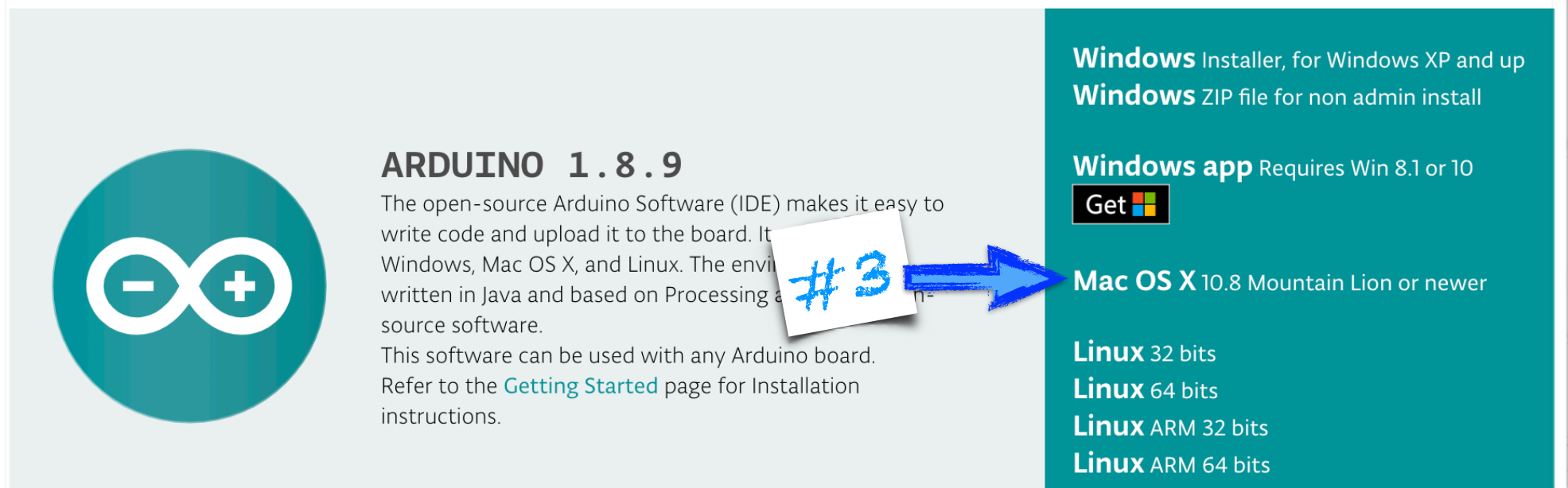


# Installing Arduino on a Mac

Start by going to [www.arduino.cc](http://www.arduino.cc). Select 1) "Software" then 2) "Downloads". 3) Click the link for the OSX Mountain Lion or newer. Don't sweat the version number of the Arduino IDE. That changes quite often and newer versions are always backward compatible with older versions. 4) Open the downloaded file. 5) The file should unzip in less than 30 seconds. 6) Locate the unzipped file in Downloads and open the program.



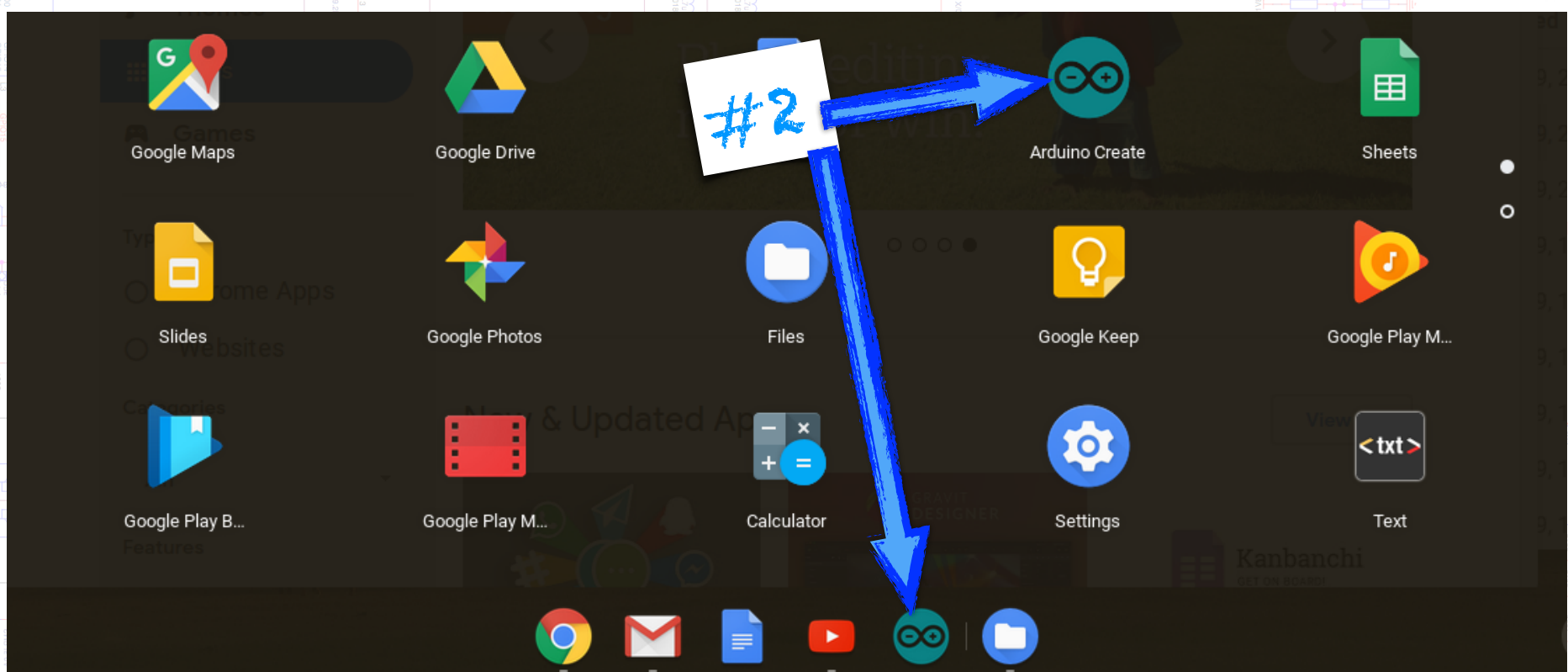
## Download the Arduino IDE





# Installing Arduino on a Chromebook

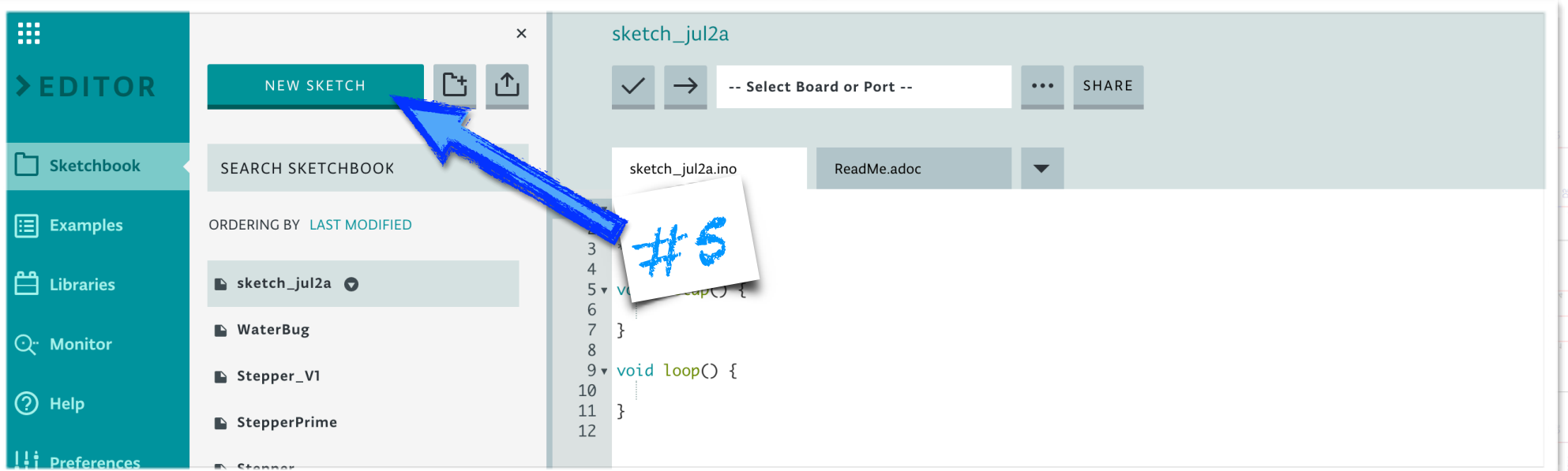
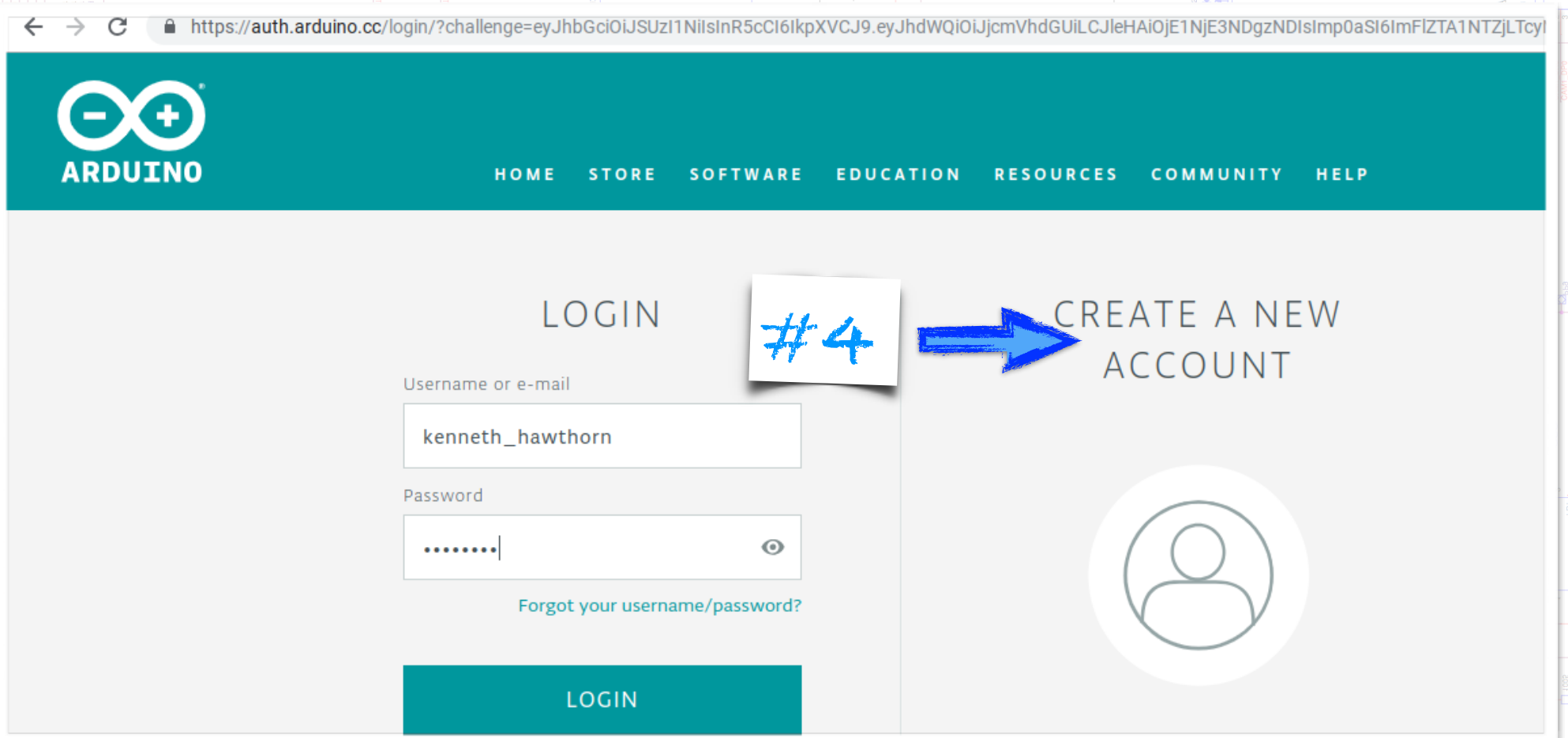
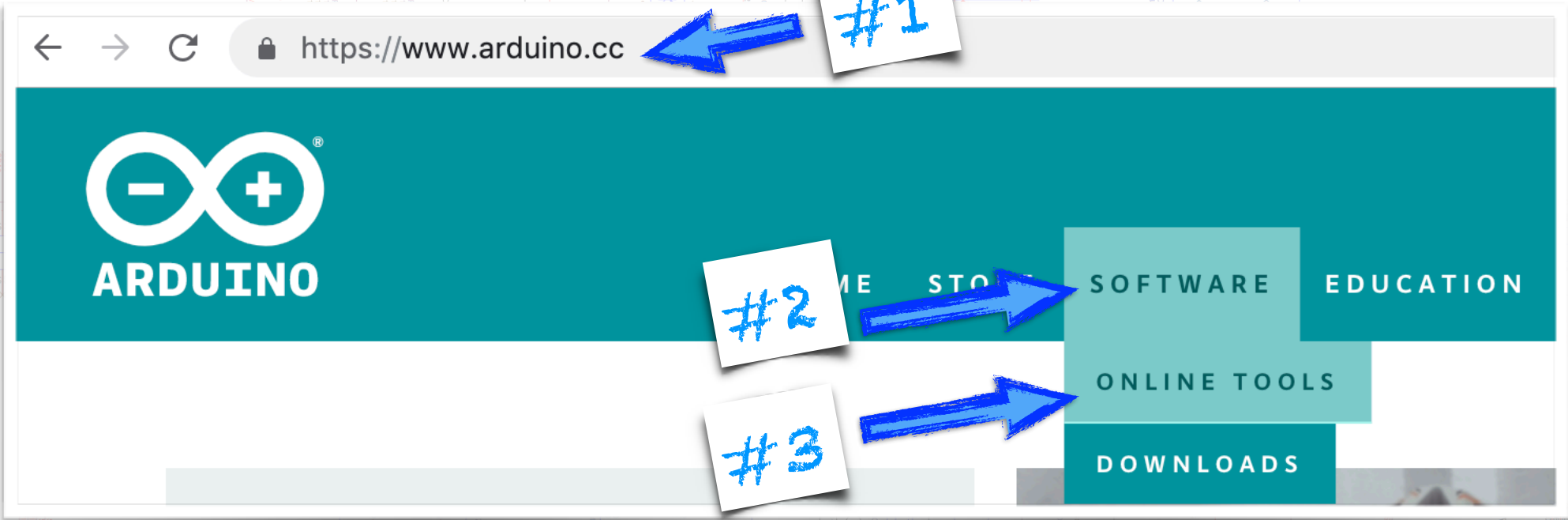
Open the Chrome Web Store app on your Chromebook. Search for Arduino create. Click on the launch app button. At the icon for Arduino create to the app launch bar. Double click the Arduino icon and it will open a Chrome browser tab.





# Arduino with the Online Web Editor

1) Open the a web browser. Navigate to [www.Arduino.cc](http://www.Arduino.cc) 2) click on software then 3) click on Online Tools. Click on Arduino Web Editor, then 4) click "Create New Account". 5) Open the Arduino Web Editor and click New Sketch.

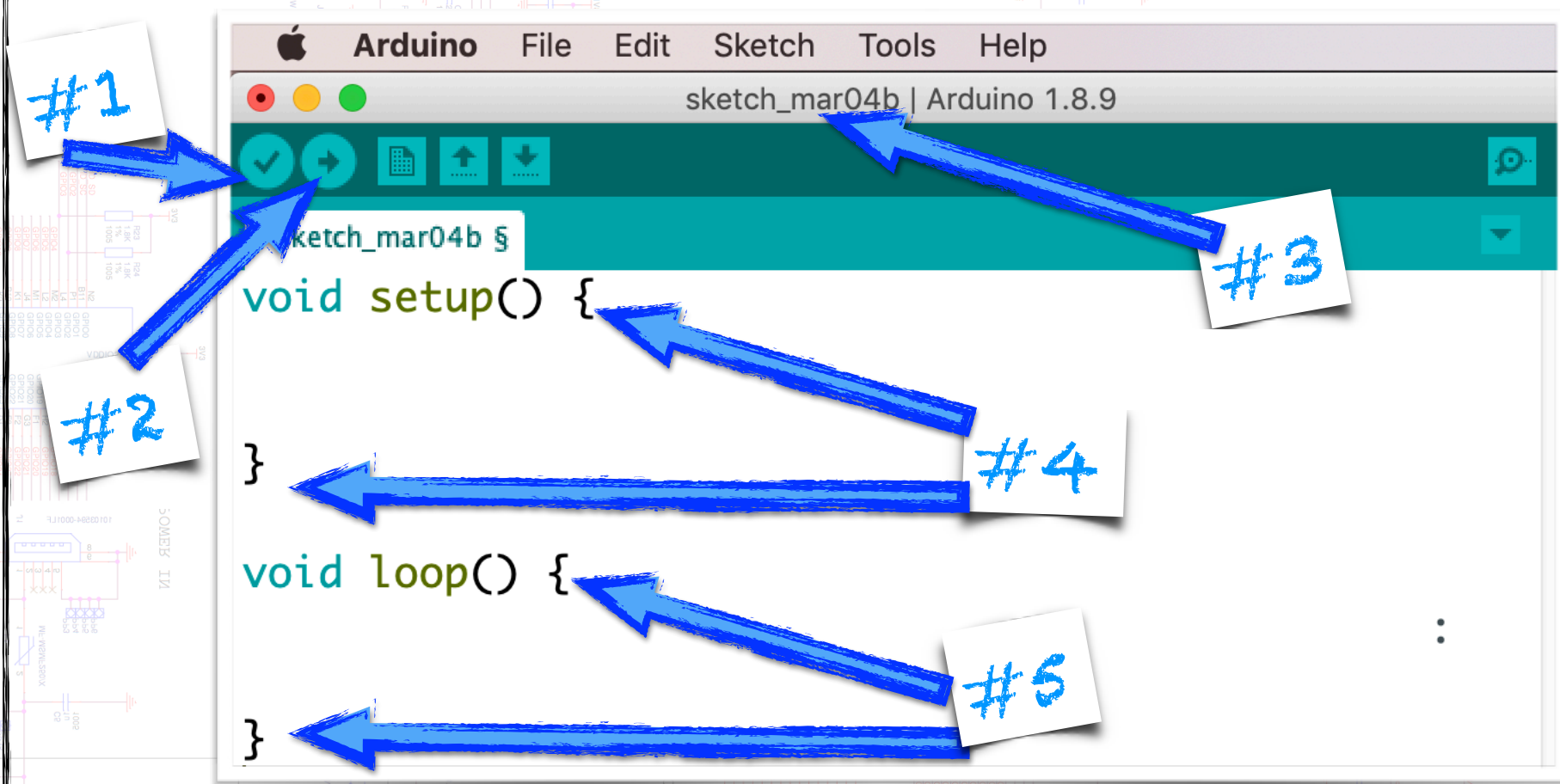




# Understand and Upload Your First Code

Regardless of the Arduino IDE you have chosen to use for your first project, all of these interfaces have exactly the same function and very similar layouts. For this book I have chosen to use The Arduino IDE install a Mac. Everything that we cover Will be almost identical in the Chrome book, Windows or online editor options.

When you're working within Arduino land the formal name for a program that you write is called "Sketch". In the screenshot below you can see how a blank sketch opens within the Arduino IDE. This is exactly parallel opening up a new Microsoft Word document. I have labeled the most important parts of the Arduino IDE below.



At this point I expect the Arduino IDE to look completely foreign to you and maybe even a little bit intimidating. Inside this window you have 1) A Check button that you can press to literally check your code for errors. 2) the second button with the arrow as an icon will upload your code from the Arduino IDE to the physical Arduino Uno board. 3) Every Arduino sketch automatically names of itself "sketch\_....." and the date it was created. There are just two sections to every bit of code you will right foreign Arduino. The first section of code 4) addresses things that your program Will do just once at the start. The second section of code 5) contains commands that you would like to keep doing as long as the program is running. In both cases each of these two sections code will be encapsulated between a pair of {} curly brackets.

At this point please note that the word "void" appears in a teal color well the words "setup" and "loop" appear in a dark green. This is significant because the Arduino IDE actually changes the colors of words we typed as it interprets the meaning of each word. In the same way in the English language we have verbs, nouns and adjectives in programming languages we have words that are treated with different purposes. Functions and variables along with a host of other special words are treated in similar manner. For now, just know that the word void is usually associated with a function so it appears in teal while the words setup and loop denote that the function Will occur once or more than once. Don't worry too much about this now, just know that it is the Arduino IDE that is changing the color of the text in order to better clarify with you how the words that you're typing Will be interpreted in the sketch you upload to your Arduino microcontroller