



# Arduino Basics: Getting Started with blinking lights

DIFFICULTY

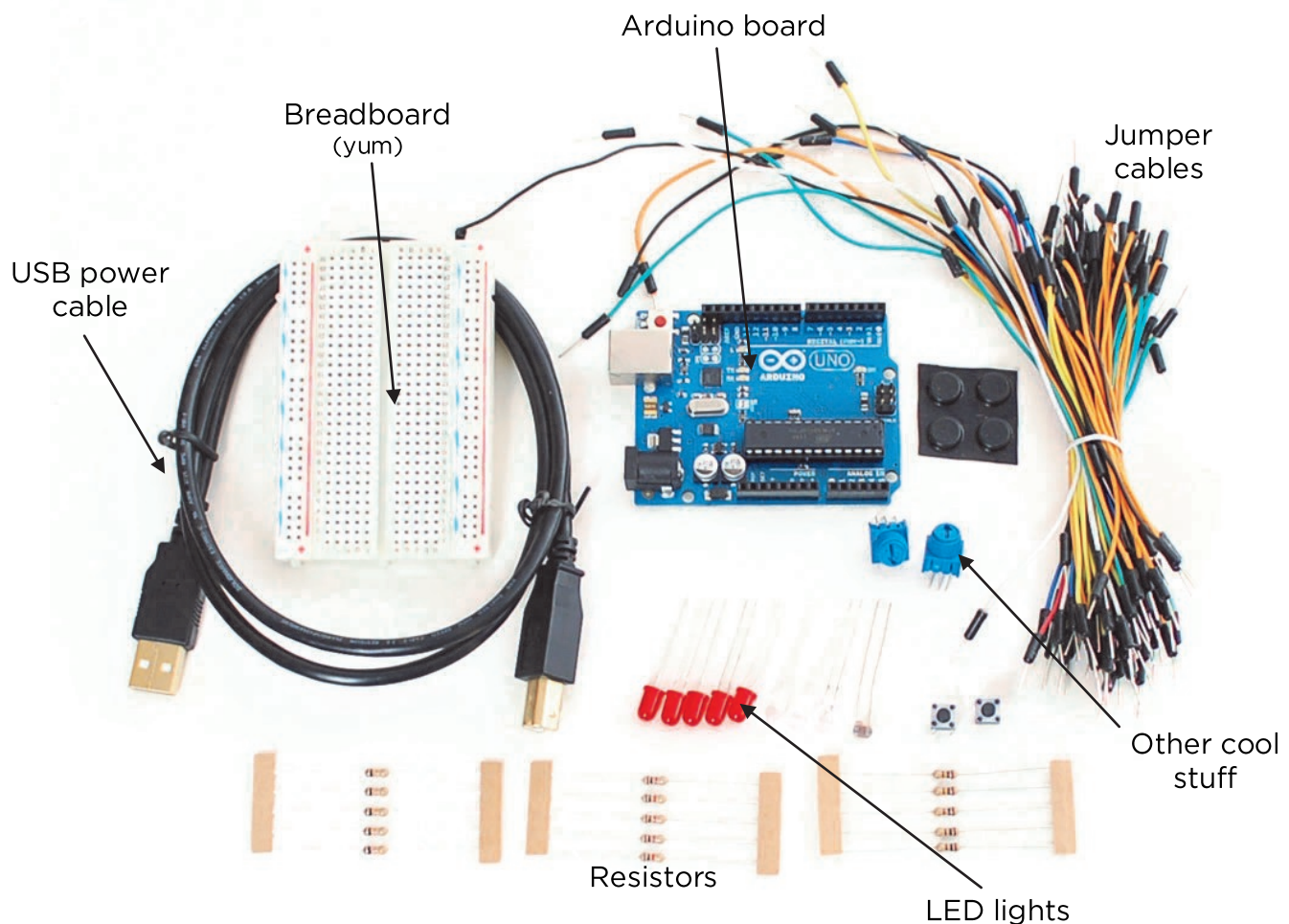


## What is Arduino?

Arduino is a combination of simple hardware (stuff you can find at an electronics store) and open software (stuff you can download for free). Arduino doesn't do anything by itself. You have to connect it to other components (the hardware) and then tell it what to do (the software).

Its strength is in *rapid prototyping*. That means if you have an idea for motion activated light, or a light-sensitive buzzer, you can get working on it right away without making any permanent changes to your stuff!

## The Hardware



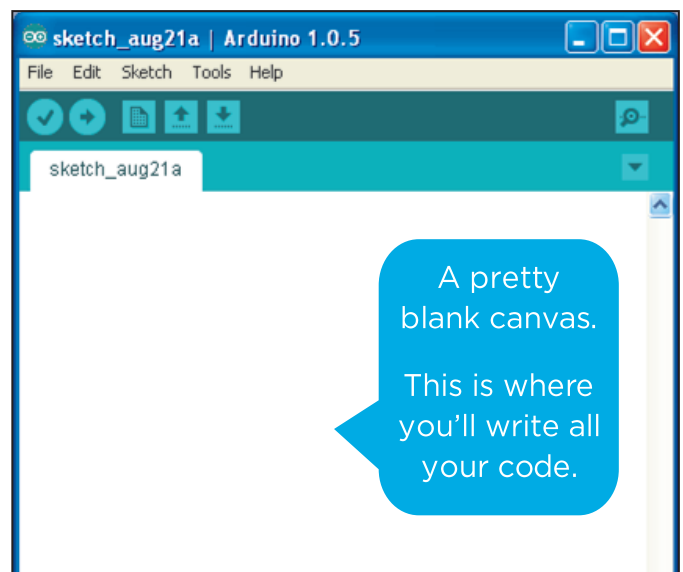
That's a lot of stuff! But don't worry, we're not going to get into all of that yet. For now, just grab a board and a USB power cable.

## The software

Before we even look at the code, let's spend a moment getting to know the program environment.

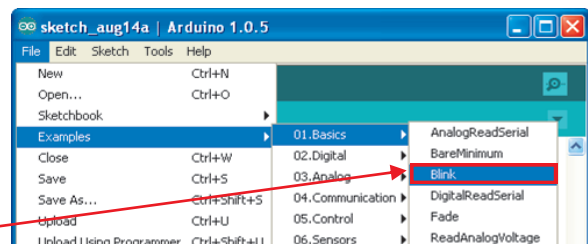
First, the toolbar.

	<b>Verify:</b> Checks your code for errors without uploading to your board.
	<b>Upload:</b> Compile and upload your code to your board.
	<b>New:</b> Start a new sketch.
	<b>Open:</b> Open a previous sketch.
	<b>Save:</b> Save your sketch. Save early, save often!
	<b>Serial Monitor:</b> Opens up the serial monitor. More on this later.



You'll use **Verify** and **Upload** a lot. You'll also hopefully use **Open** a lot to look at all the wonderful examples included with the environment.

Let's check out a fun example. Go ahead and click on **Open**, or on **File** in the menubar, and look at the examples. Select *Blink* from the Basics folder.



## Sketches

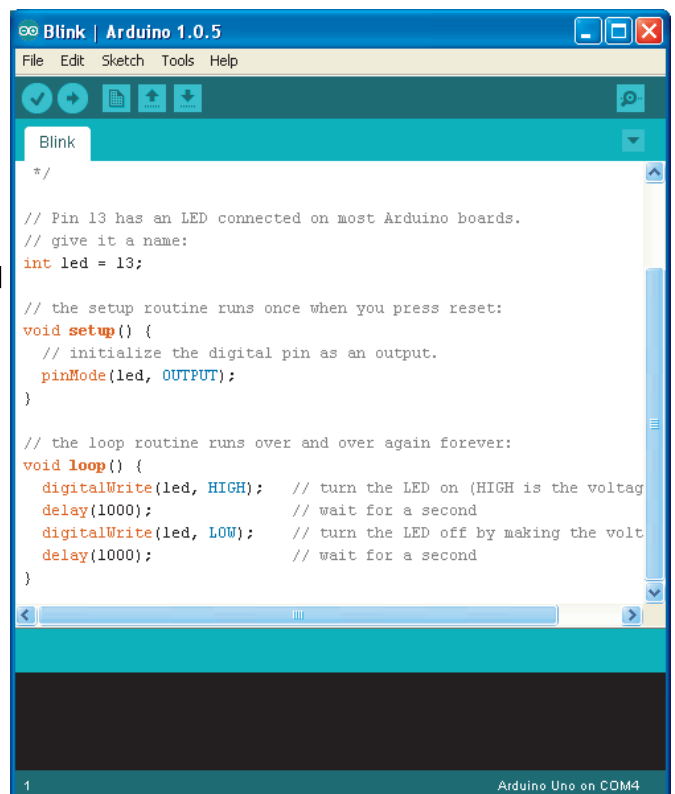
Programs in Arduino are called "sketches." All sketches have at least two sections. One is called **setup()**, and the other is called **loop()**.

**Setup()** will run once, after you power up your board. After that, **Loop()** will run over and over, and will control your board. Most of your code and logic will go in **loop**.


The code can look like a foreign language, but don't worry. There's a built in dictionary. Try this: highlight the word "**digitalWrite**" (or some other orange word), right-click on it, and select "Find in reference."

Finally, get in the habit of visiting the Arduino website. There's a lot of awesome stuff there.

- <http://playground.arduino.cc/>
- <http://arduino.cc/en/Reference/>
- <http://arduino.cc/en/Tutorial>



## Putting it all together

“Blink” is a sketch that you can run with NO additional components. Plug in your Arduino and try  **uploading** it.

If it works, you’ll see a tiny, orange light start to blink on and off.

If you look closely, next to that blinking light, you’ll see some numbers printed on the board. These label the pins, which are the black holes along the sides of the board. Connecting the pins to other circuitry components with cables and a breadboard is what the Arduino is all about.

Check out pin 13. It’s special because it connects to the on-board LED, the one that’s blinking on and off.

Now go back to your computer and check out your code. You might be able to understand a lot of what’s going on now.

See where “13” gets assigned to “*led*”? That’s telling the program which pin to manipulate.

See where we set the **pinMode** of *led* to **OUTPUT**? That’s because almost all the pins can read in and read out, and we

Upload error? The most common culprits are **DRIVERS** and **PORTS**. Don’t know what that means? Don’t panic. Ask for help. We’ll figure this out.



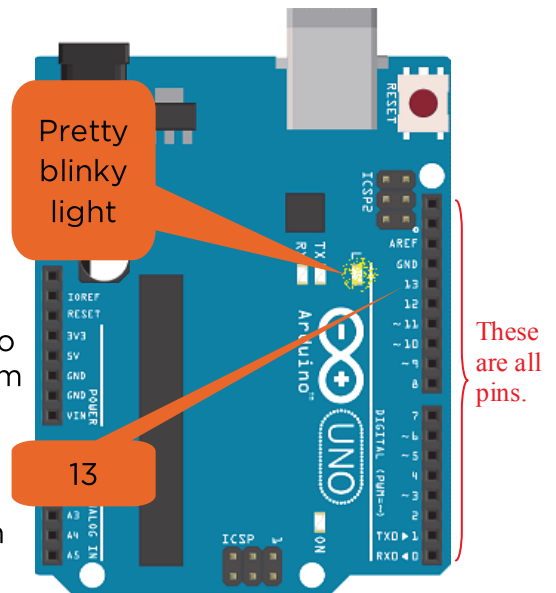
```
Arduino 1.0.5
File Edit Sketch Tools Help

Blink

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

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

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```



have to tell Arduino which we want to do.

Now see in **loop** where we turn *led* on by writing **HIGH** to **digitalWrite**? And then after a **delay**, we turn it back off by using **digitalWrite LOW**? These are the basics of programming your Arduino. Try manipulating the delays to make the light blink faster. Slower. Add a few more **digitalWrites** to the **loop** to make a blinking pattern.

Make it blink out a message in Morse Code.



### Hungry for more?

With what you’ve now learned, it’s time to break out and build your first actual circuit. Grab the **Breadboard 101** project card to get started!

### WHERE TO GET MORE HELP:

- Ask for the SIK Book for more examples

**For more ideas and help visit:**

- <http://arduino.cc/en/tutorial/>
- <http://learn.sparkfun.com/tutorials>