

# CSE423: Embedded System Summer-2020

## Introduction to TinkerCAD (Online Simulator-1)

---



# Today's Lecture

---



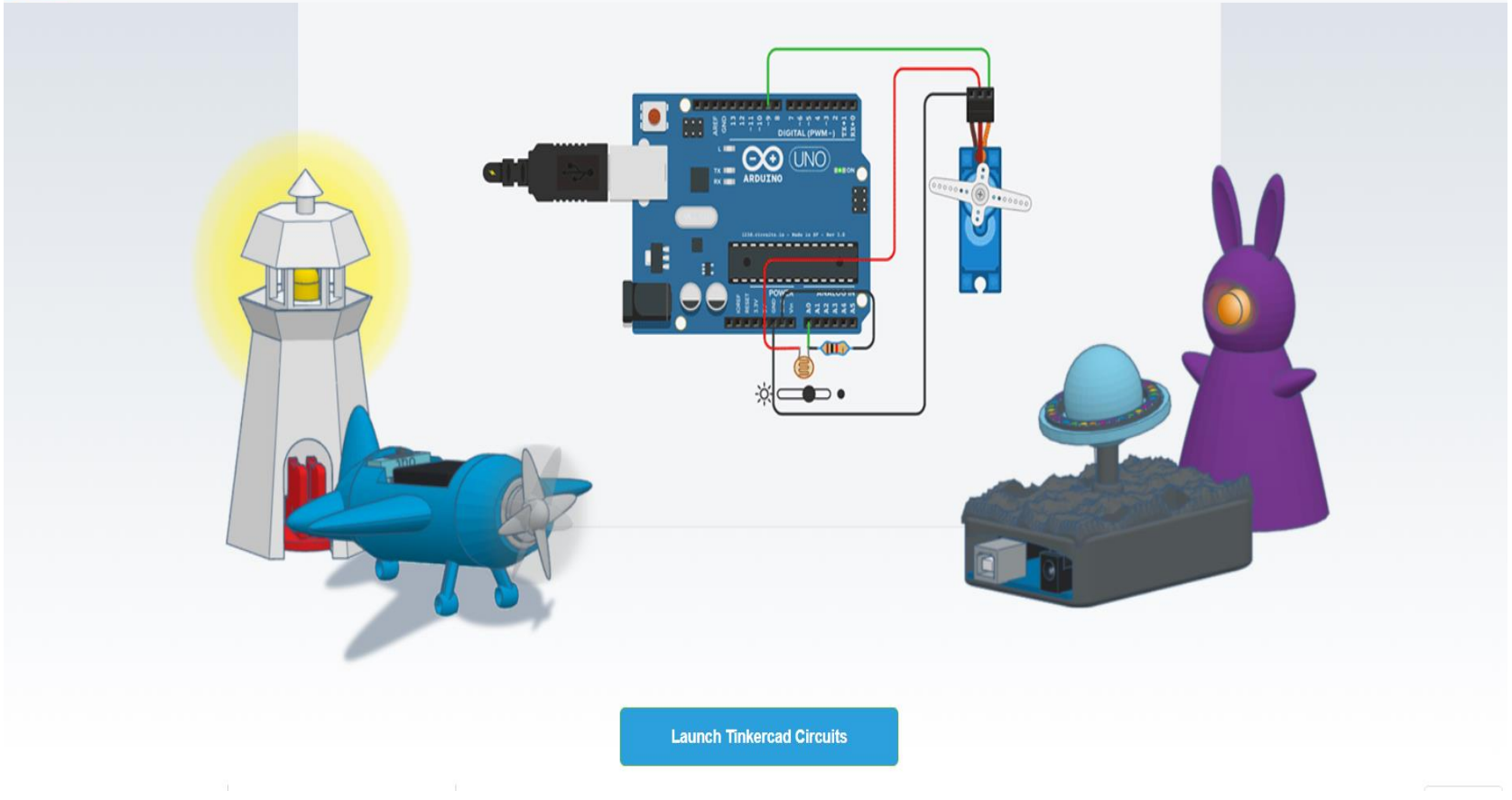
- *Introduction to TinkerCAD*
- *How to use TinkerCAD*
- *Tutorial available at:*

<https://youtu.be/yyG0koj9nNY>

# TinkerCAD Website <https://www.tinkercad.com/circuits>



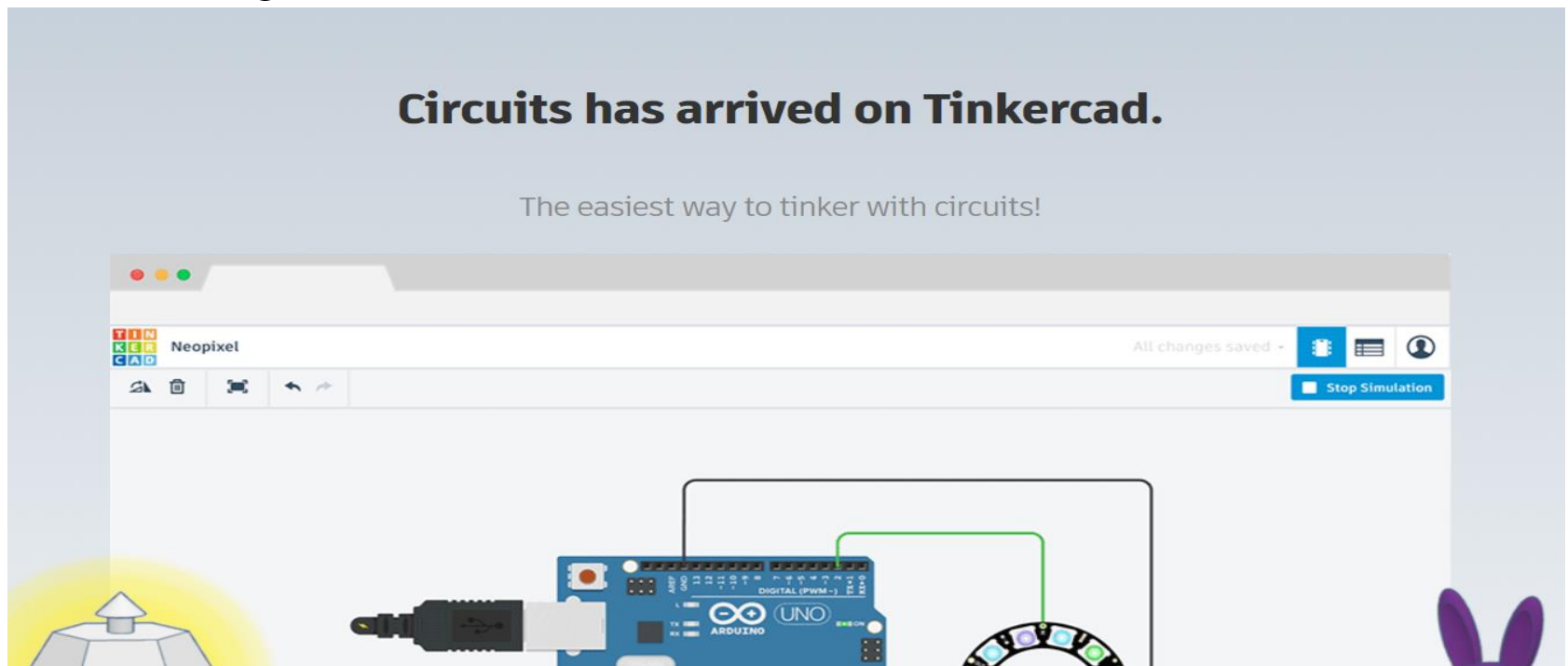
Gallery Blog Learn Teach 🔍 Sign in JOIN NOW



# What is TinkerCAD?

---

**TinkerCAD** is a free online service for creating basic 3D shapes and developing digital prototypes of electronic components. These prototypes include basic circuits with LED lights, buzzers, switches, and even light sensors.



# Why TinkerCAD Circuits ?

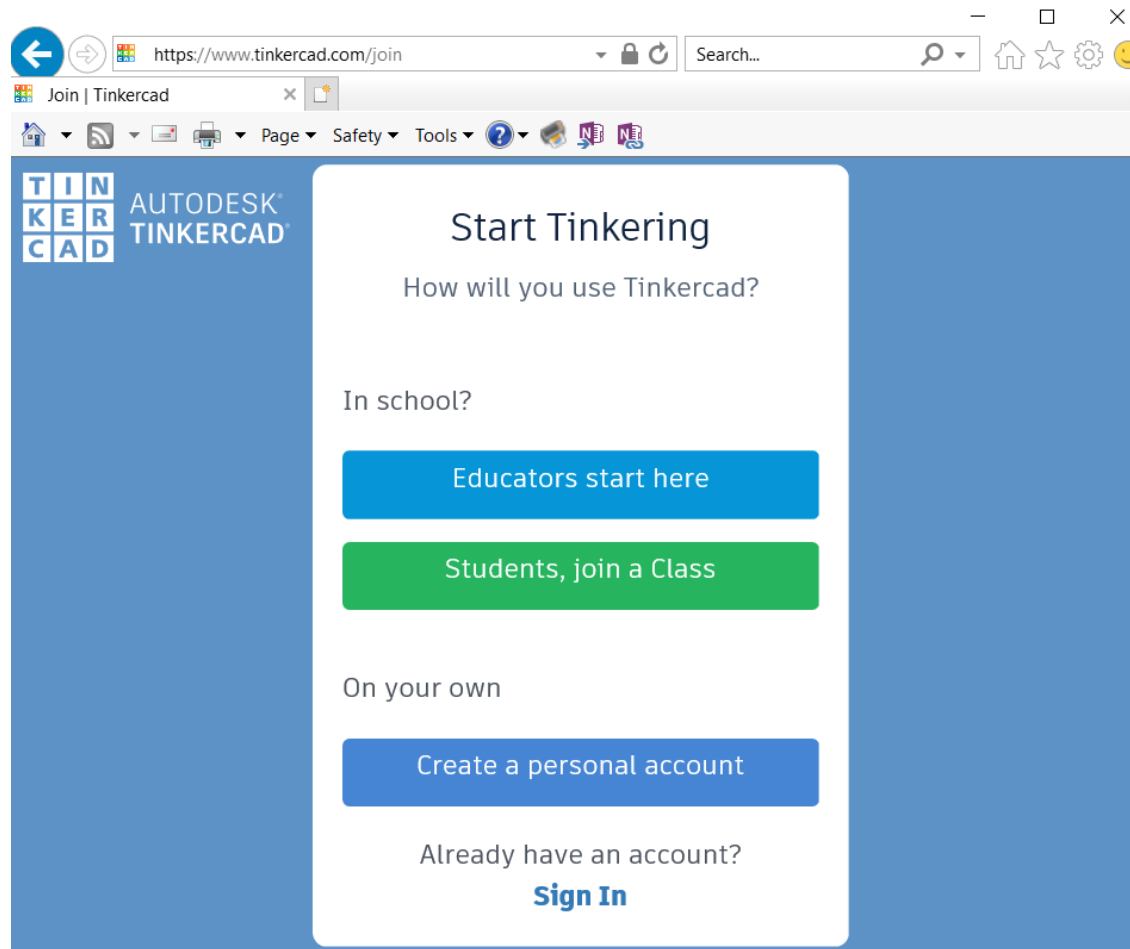
---



TinkerCad offers many benefits :

- 1- Online : You do not need to install any thing in your PC.
- 2- OpenSoure : Free, no license needed, for everyone.

# Creating an Account



# TinkerCAD Circuits



- ❑ Click on Circuits to switch from **3D Designs** to **Circuits** mode, and then click on **Create new Circuit**. If everything is all right let's start working and discover this platform.

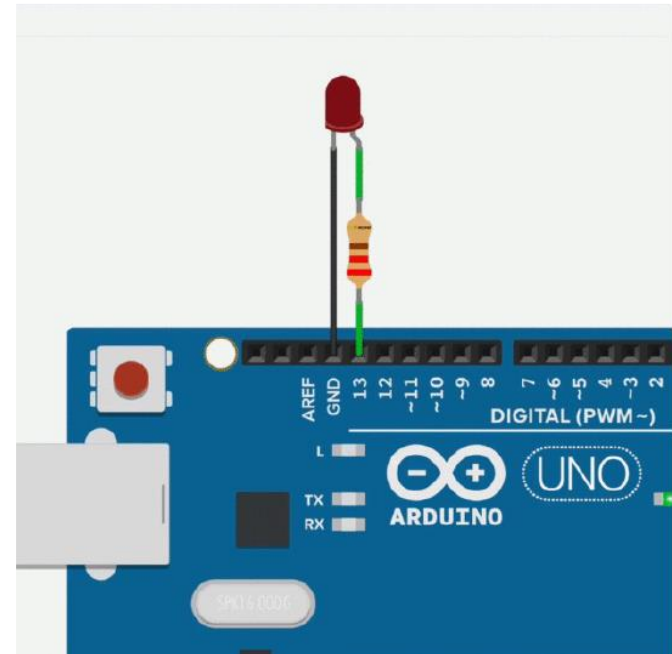
The screenshot shows the TinkerCAD web interface. At the top left is the TinkerCAD logo. To its right is the text 'TINKERCAD FOR...' followed by navigation links: 'GALLERY', 'COMMUNITY', 'LEARN', and 'TEACH'. A search bar and a user profile icon are on the top right. On the left side, there is a sidebar with a user profile for 'ahmnouira', a search bar for designs, and a list of categories: '3D Designs', 'Circuits' (which is highlighted in blue), and 'Lessons'. Below these are 'Projects' for 'Project 2' and 'Project 1', and a 'Create project' button. The main area is titled 'Circuits' and features a 'Create new Circuit' button. Below this is a grid of project thumbnails. The first row contains four projects: 'Smooth Lappi' (20 days ago, Private), 'Funky Blorr' (2 months ago, Private), 'Super Migelo' (2 months ago, Private), and 'Funky Bojo' (2 months ago, Private). The second row shows the beginning of another set of project thumbnails, including one with a breadboard and another with a blue PCB.

# Let's Blink a LED in TinkerCAD!



## □ Step 1: LED Resistor Circuit

The LED's legs are connected to two pins on the Arduino: **ground and pin 13**. The component between the LED and pin 13 is a resistor, which helps limit the current to prevent the LED from burning itself out. The colored stripes identify the resistor's value, and for this circuit, anywhere from 100 ohms to 1000 ohms will work great.



The LED, on the other hand, is polarized, which means it only works when the legs are connected a certain way. The positive leg, called the anode, usually has a longer leg, and gets wired to power, in this case coming from your Arduino's output pin. The negative leg, called the cathode, with its shorter leg, connects to ground.

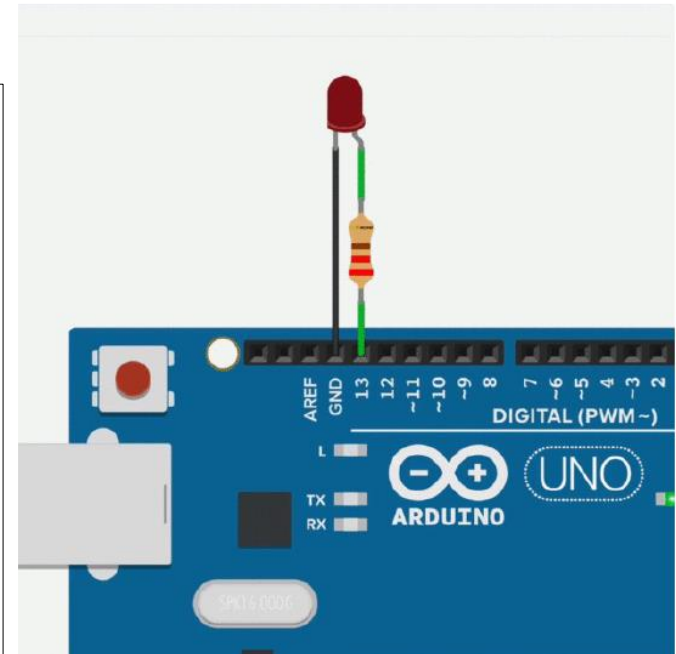


# Let's Blink a LED in TinkerCAD!



## □ Step 1: LED Resistor Circuit

- In the Tinkercad Circuits components panel, drag a resistor and LED onto the workplane.
- Edit the resistor's value by adjusting it to 220 ohms in the component inspector which appears when the resistor is selected.
- Back in the components panel, find and bring over an Arduino Uno board. Click once to connect a wire to a component or pin, and click again to connect the other end.
- Connect your resistor to either side of the LED. If you connected your resistor to the LED's anode (positive, longer), connect the resistor's other leg to Arduino's digital pin 13.
- If you connected your resistor to the LED's cathode (negative, shorter leg), connect the resistor's other leg to Arduino's ground pin (GND).
- Create another wire between the unconnected LED leg and pin 13 or ground, whichever is still not connected.



# Let's Blink a LED in TinkerCAD!



## □ Step 2: Simple Code With Blocks

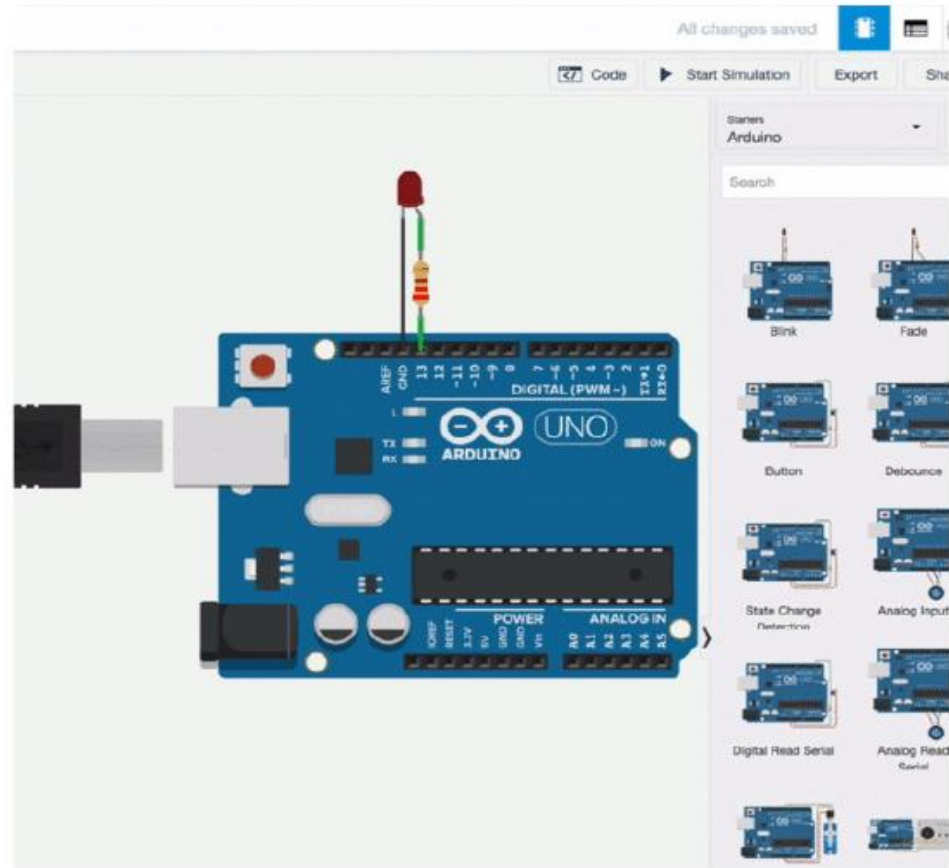
- Let's go through the simple code controlling the **blink** by opening the code editor (button labeled "**Code**"). You can resize the code editor by clicking and dragging the left edge.
- The code starts out with **two gray** comment blocks, which are just notes for us humans to read. The first blue output block sets the built-in LED HIGH, which is Arduino's way of describing "on."
- This output command will activate a 5V signal to anything connected to the specified pin. Next up is a **yellow** command block that waits for one second, simple enough.



# Let's Blink a LED in TinkerCAD!



- Step 2: Simple Code With Blocks



# Let's Blink a LED in TinkerCAD!

---



## □ Step 2: Simple Code With Blocks (Complete Code)

*/\* This program blinks pin 13 of the Arduino (the built-in LED) \*/*

**void setup()**

```
{  
  pinMode(13, OUTPUT);  
}
```

**void loop()**

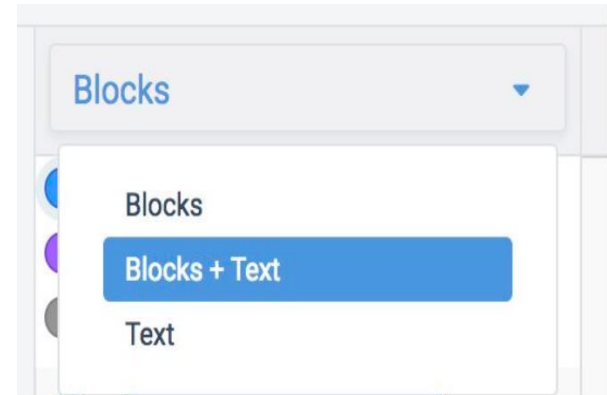
```
{  
  // turn the LED on (HIGH is the voltage level)  
  digitalWrite(13, HIGH);  
  delay(1000); // Wait for 1000 millisecond(s)  
  // turn the LED off by making the voltage LOW  
  digitalWrite(13, LOW);  
  delay(1000); // Wait for 1000 millisecond(s)  
}
```

# Let's Blink a LED in TinkerCAD!



## □ Step 3: Blink Arduino Code Explained

When the code editor is open, you can click the dropdown menu on the left and select "Blocks + Text" to reveal the Arduino code generated by the code blocks. All the extra symbols are part of Arduino's syntax, but don't be intimidated! It takes time to learn to write proper code from scratch. We'll go through each piece here, and you can always use the blocks for comparison as you level up.



This first section is title block comment, describing what the program does. Block comments are bookended by an opening `/*` and closing `*/`.

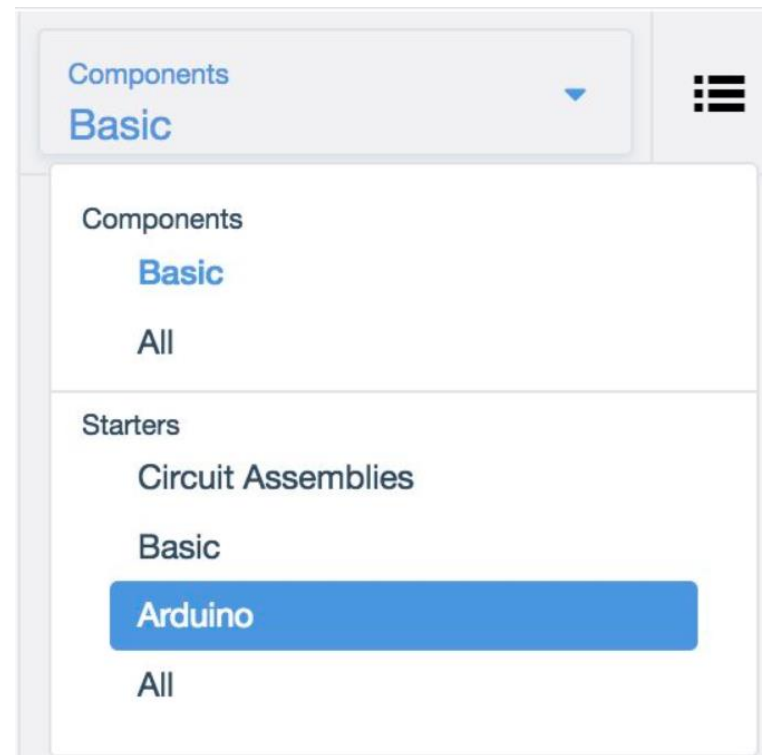
Next is the code's setup, which helps set up things your program will need later. It runs once when the program starts up, and contains everything within its curly braces `{ }`. Our blink sketch's setup configures pin 13 as an output, which prepares the board to send signals to it, rather than listen.

# Let's Blink a LED in TinkerCAD!



- Step 4: Use the Blink Circuit Starter

Grab this circuit and code combo any time using the starter available in the components panel ([dropdown menu -> Starters -> Arduino](#)).



# Task

---

- Try the same LED Blink for Multiple LEDs

