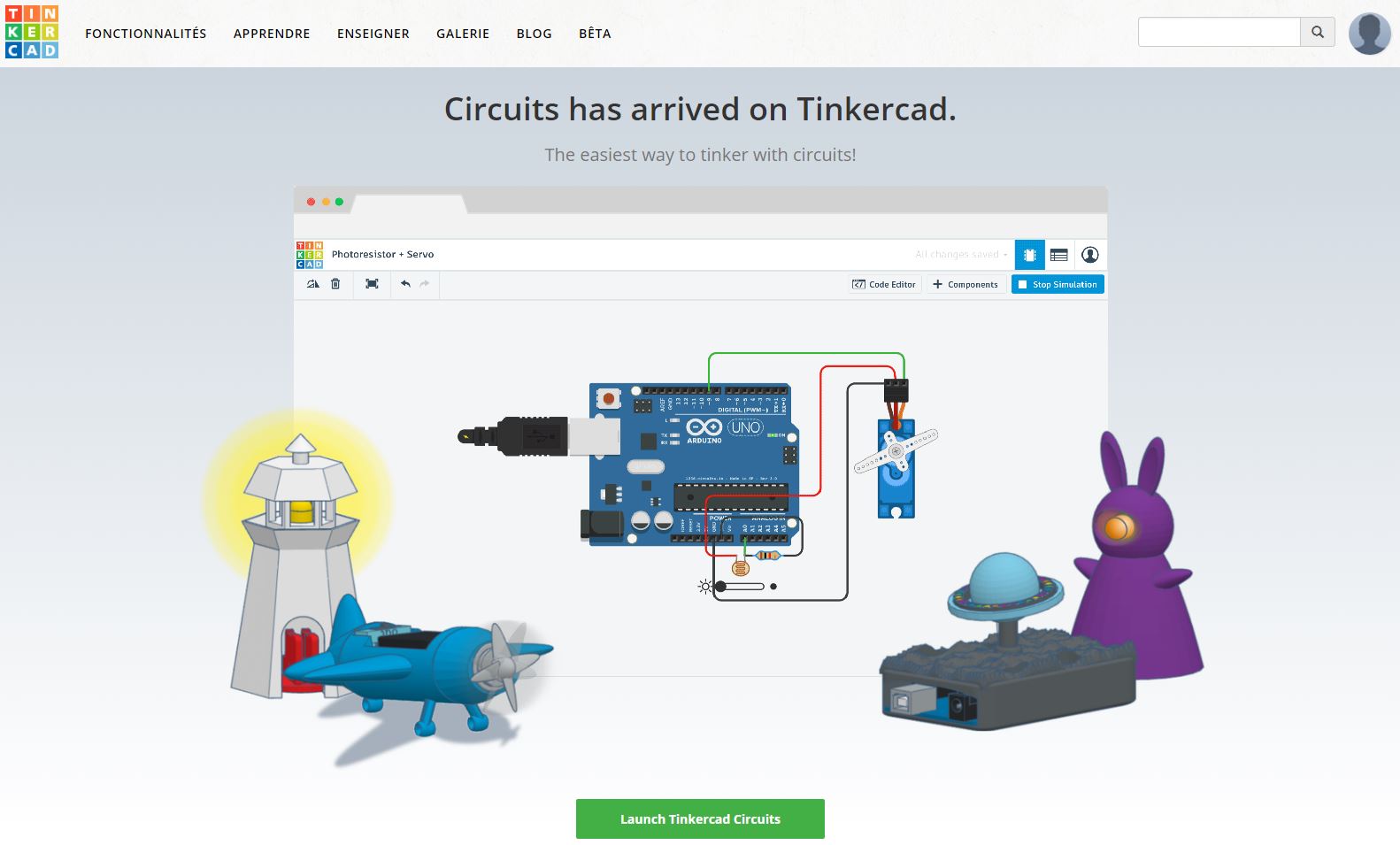
Lab 4: **Using the Arduino simulator**

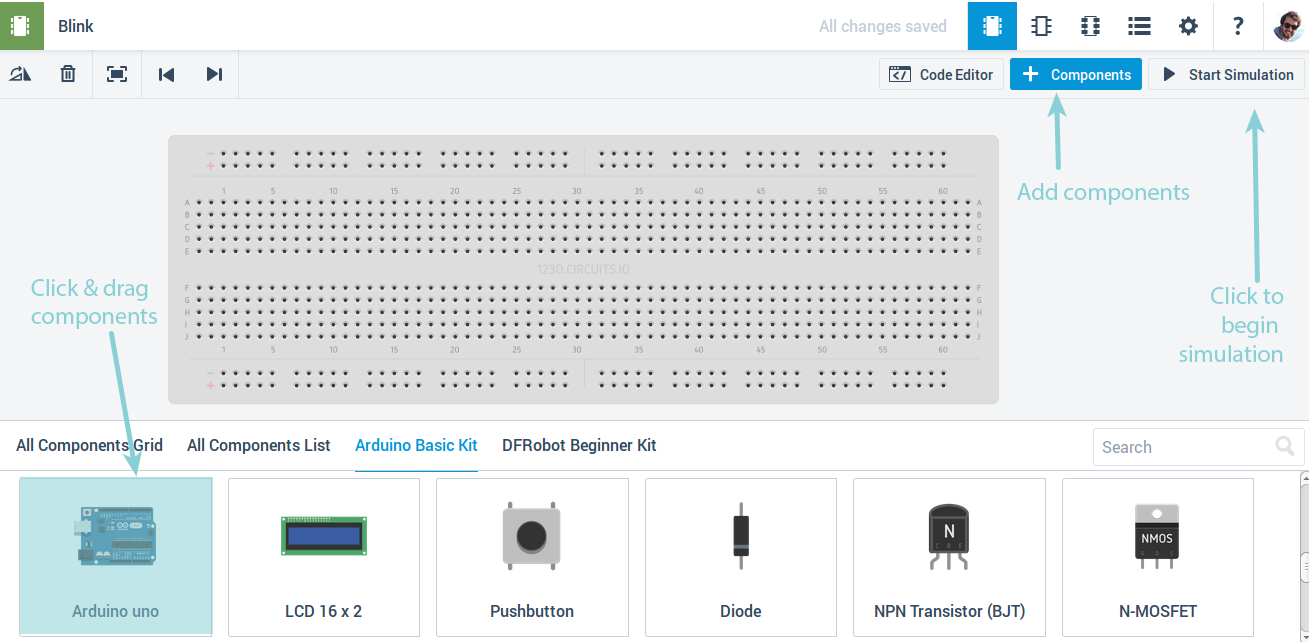
Before you buy an Arduino, it is possible to make basic Arduino assemblies thanks to an online simulator, which is free to use (but proprietary, i.e. not open source).

This simulator, called  [TinkerCad Circuit](https://www.tinkercad.com/circuits" \t "[object Object]), is provided by the company Autodesk. If you do not want to use a real Arduino for now, follow the tutorial below:

**Tutorial**

1. First create an account on the [Autodesk](https://accounts.autodesk.com/register) website (this account will in no way be linked with this MOOC).  We recommend the [Chrome](https://www.google.com/intl/fr_fr/chrome/browser/) browser, which can provide the full function of the simulator.
2. Go to [TinkerCad Circuit](https://www.tinkercad.com/circuits" \t "[object Object]). Once you’ve logged in with your Autodesk details, you should have access to this page:



1. create a new timeline. Fill in the necessary information in the page that opens, and then click "*Create New!*"
2. Once the simulator loads, here are the different possible actions in the simulator:
3. To send instructions to Arduino, click on it and then on*Code editor*.
4. Copy the following code and paste into the editor:

/\*  Blink

  Light the LED for 1 second,

  then turn it off for 1 second.

\*/

// Number of the pin connected to the LED:  
int led = 13;

// the function runs once when you press reset or power the board

void **setup**() {

// initialize digital pin 'led' as an output.

  pinMode(led, OUTPUT);

}

// this code runs over and over again as long as there is power

void **loop**() {

  digitalWrite(led, HIGH); // light LED (send 5V to the pin)

  delay(1000); // wait 1000ms = 1s

  digitalWrite(led, LOW); // turn off LED (0V to the pin)

  delay(1000); // wait another second

}

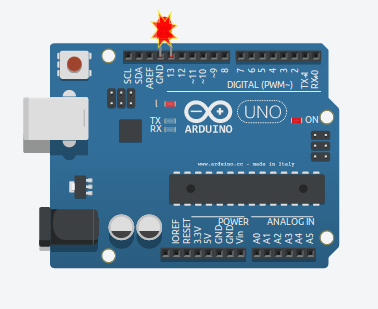
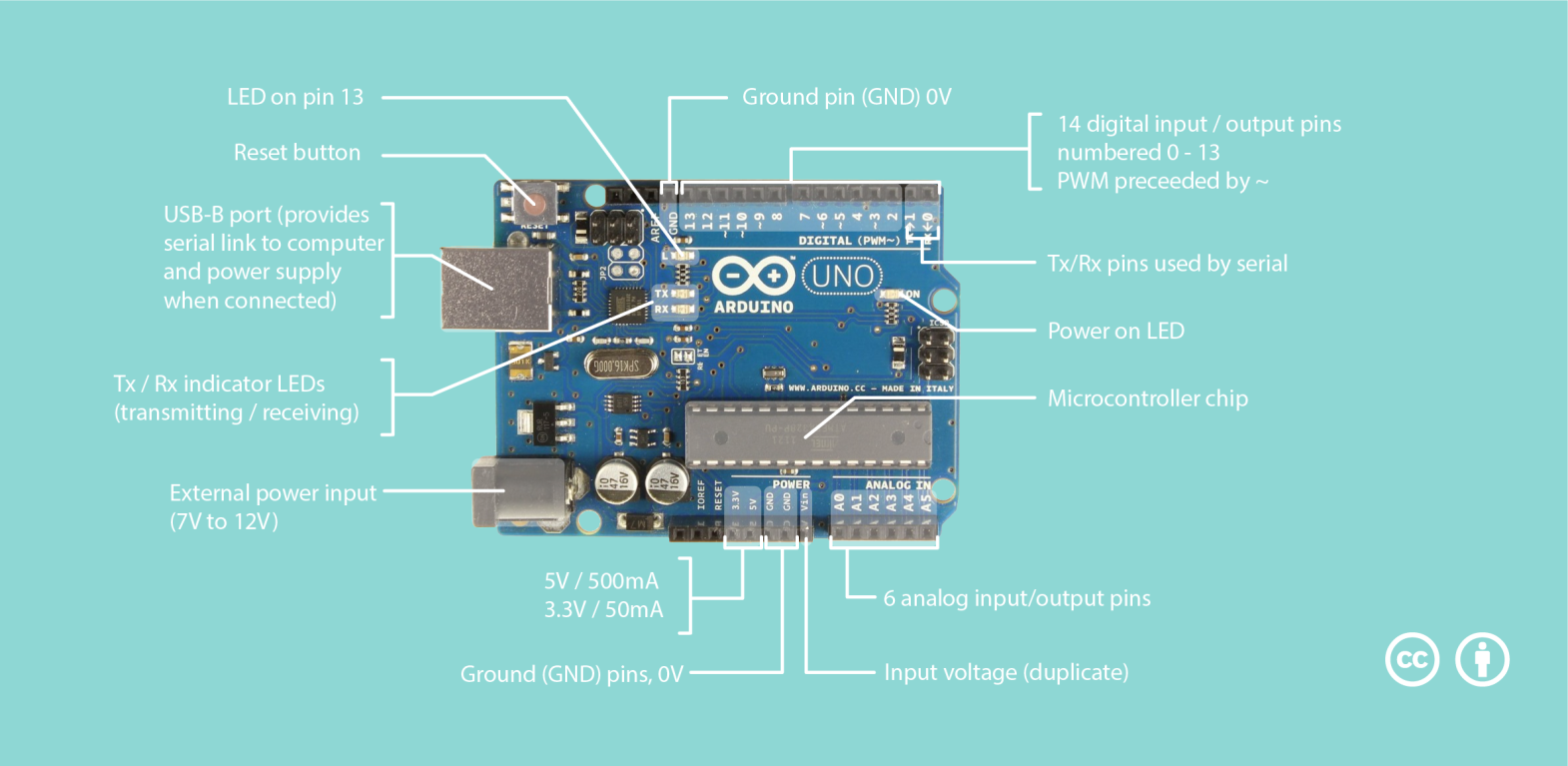
1. Finally, add a LED on pin 13, by dragging it from the right directly onto the:

Figure 1: pins 13 and GND on the Arduino

1. Click “*Start Simulation*” in the top right. Joy and jubilation - our Virtual LED flashes! We’re going to cover how the code and electronics works in the following sections.

**Arduino**

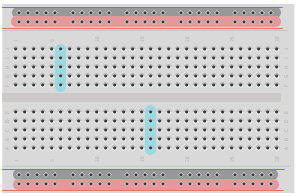
Here is a map that explains the connector pins found on an Arduino board:



To avoid plugging directly into the Arduino, we will use the prototyping board or **breadboard**.

**The principle of a breadboard**

Of course, the board is full of holes, but many of them are connected electrically. Here is a quick diagram that will help us understand.



As you can see from the picture, I have drawn various areas. The red and black areas correspond to the power lines. Often there are two vertical lines like these that allow you to connect your components to the necessary power supplies. By convention, the Black Represents ‘ground’ at 0V, and **red**,  the power supply (in this case, +5V). Usually, all the holes in the same horizontal **ligne** are connected to this area. So, you have a power line running the length of the board.

Then you can see areas in blue. These areas are linked together by connectors underneath the plastic in a vertical column – all the holes in the same column are connected to each other. However, each column is separate from the others. By overlapping components across multiple columns, you can connect them to each other.

Finally, you may notice a space that cuts the board in half symmetrically. This space also cuts the connection of the columns. Thus, in the above drawing, you can see that each column has 5 holes that are connected. The space in the middle is standardized and is the width of the standard integrated circuits (IC’s). By placing an integrated circuit in the middle, across the space in the middle, each leg lines up with its own section of a column, which you can then connect other things to.

If you want to see an example of this, I recommend you try the software [Fritzing](http://fritzing.org/home/" \t "_blank" \o "Fritzing), which allows to create circuits in a fairly simple and intuitive way. You will see how the columns are separated from each other.

As we will see in the coming weeks, the pins **0** and **1** (also called **Rx** and **Tx**) can be used to communicate with the computer. We therefore recommend that you do not use these pins to connect your LEDs, but instead focus on the numbered pins from 2 to 13!