To download the Arduino IDE go to https://www.arduino.cc/en/Main/Software

This will allow you to be able to write software that can be programmed into your Arduino.

To compile your program, click on the checkmark in the top left. To write to your Arduino, click on the right facing arrow to the right of the checkmark:

There are two important functions of any program that runs on an Arduino: The setup and the loop:

```
void setup() {
  // put your setup code here, to run once:

}
```

The setup function only runs at the very beginning when the program is initialized.

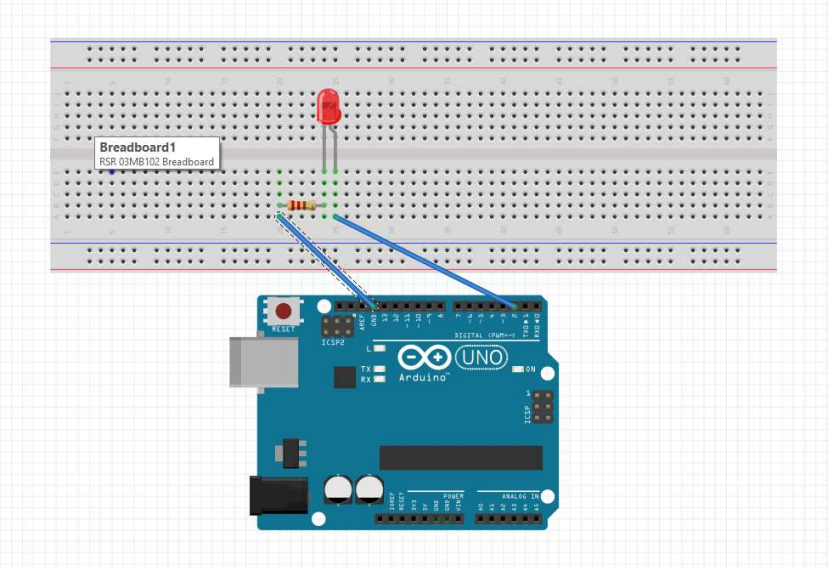
```
void loop() {
  // put your main code here, to run repeatedly:

}
```

The loop function will run infinitely and will be the crux of the actual functionality.

Let's set up so that your Arduino makes an LED blink!

First you will need to wire the LED to the Arduino like so:



Note that the positive end of the LED is the longer stem and that the negative end is the shorter stem. Also note that the LED is in series with a resistor. That resistor essentially throttles the current to make sure we don't short circuit the LED.

Now we need to set up the code in the Arduino IDE. First stage is to set the specific pin we're using as an output. There is a built in function called pinMode(). Its arguments ask for a digital pin as well as the mode to set it to. We currently have the LED connected to digital pin 2 and we want to set it to output.

```
void setup() {
  // put your setup code here, to run once:
  pinMode(2, OUTPUT);
}
```

Now we want to make the LED turn on. We can simply just use the function digitalWrite(). Its arguments ask for a digital pin (obviously) as well as the mode for it to write out as. We want it to turn the LED on, so we want to write a HIGH. To turn it off we want to write a LOW.

```
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(2, HIGH);
  digitalWrite(2, LOW);
}
```

The above turns the LED on and then off immediately. You may notice though that if you run the above code, it looks like the LED is only on though. This is because the loop() function runs so fast that once the LOW is written to the LED, a HIGH is immediately, turning the LED back on immediately. We can insert some delays to hold the LED in a specific state.

We need to include the following library before the setup function to make this work:

```
#include <util/delay.h>
```

We can use the function _delay_ms(). This pauses the program for the number of milliseconds written as the argument.

```
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(2, HIGH);
  _delay_ms(
  digitalWrite(2, LOW);
}
```

Now we can write this onto our Arduino and now we have a blinking LED!