

Arduino Workshop

— The very basics of electricity
and how to make a nightlight —

Author: Ammon Shepherd

License: [Arduino Workshop](#) by Ammon Shepherd is licensed under [CC BY-NC-SA 4.0](#)

Date: October 26, 2020

Workshop Slides:

<https://tinyurl.com/arduino-basics-workshop>

Workshop Files:

<https://github.com/ammonshepherd/arduino-tuts/workshops/arduino-basics>

Workshop Objectives

- Understand purpose and use of some basic electronics components: resistor, LED, photoresistor, button
- Know what an Arduino is and why they can be useful
- Build a circuit and program the Arduino
- Be empowered to create your own electronics projects that will improve your environment

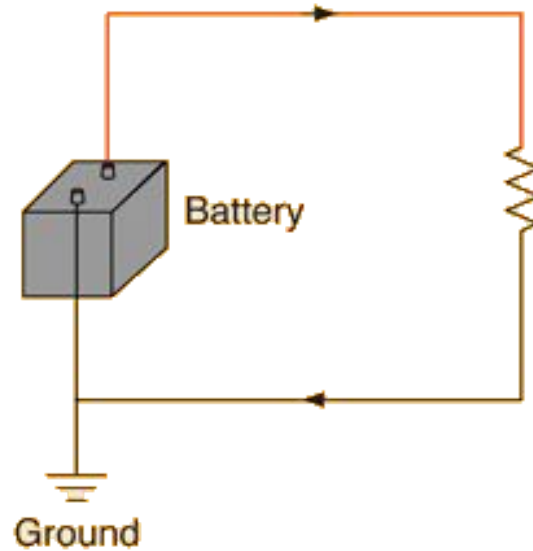
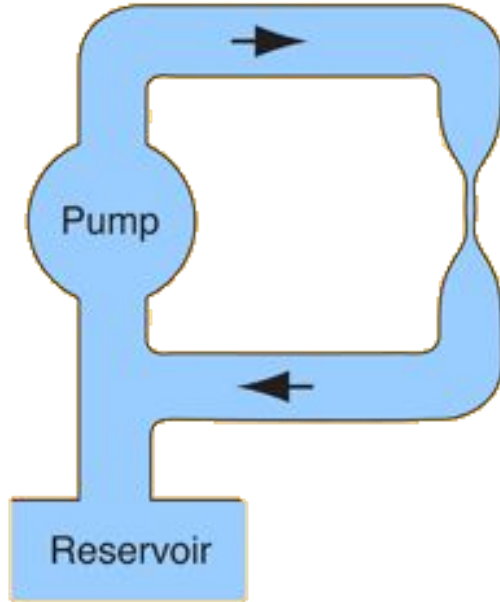
Introductions

Tell us:

- Your name
- What you do
- Why interested in Arduino

Basics of Electricity

Electrical current



Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

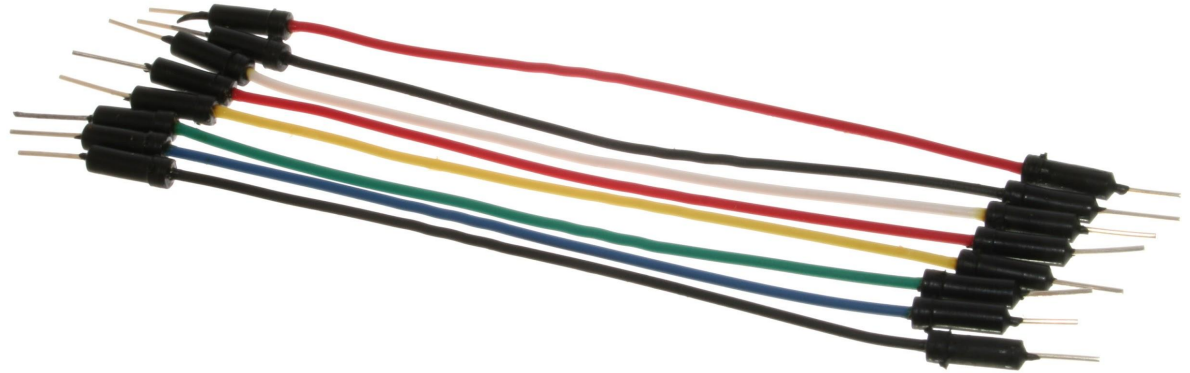
Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



220 Ohm or 330 Ohm

Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

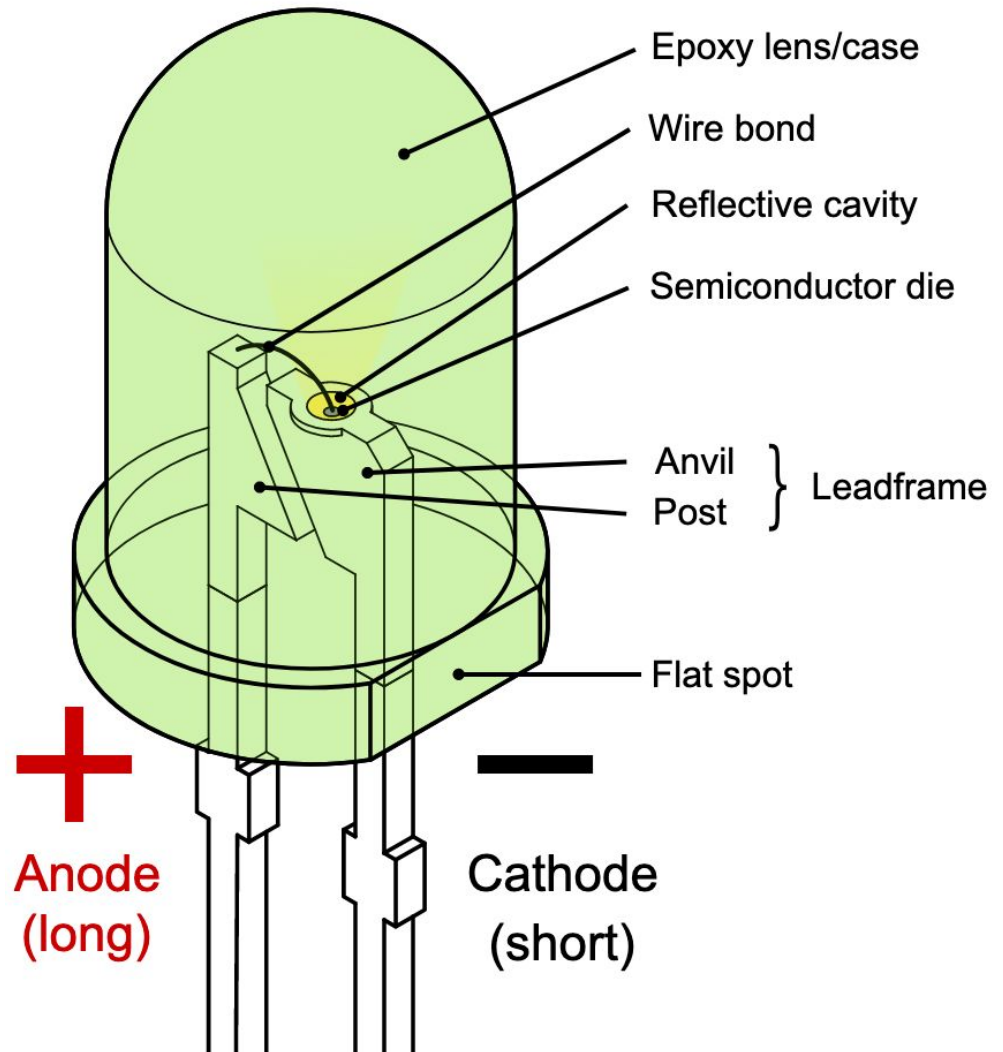
Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

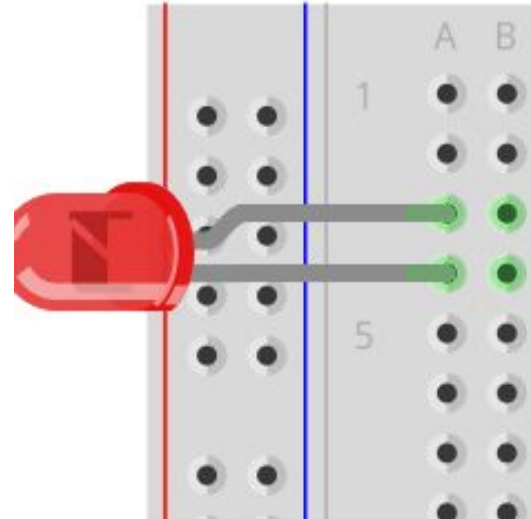
Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

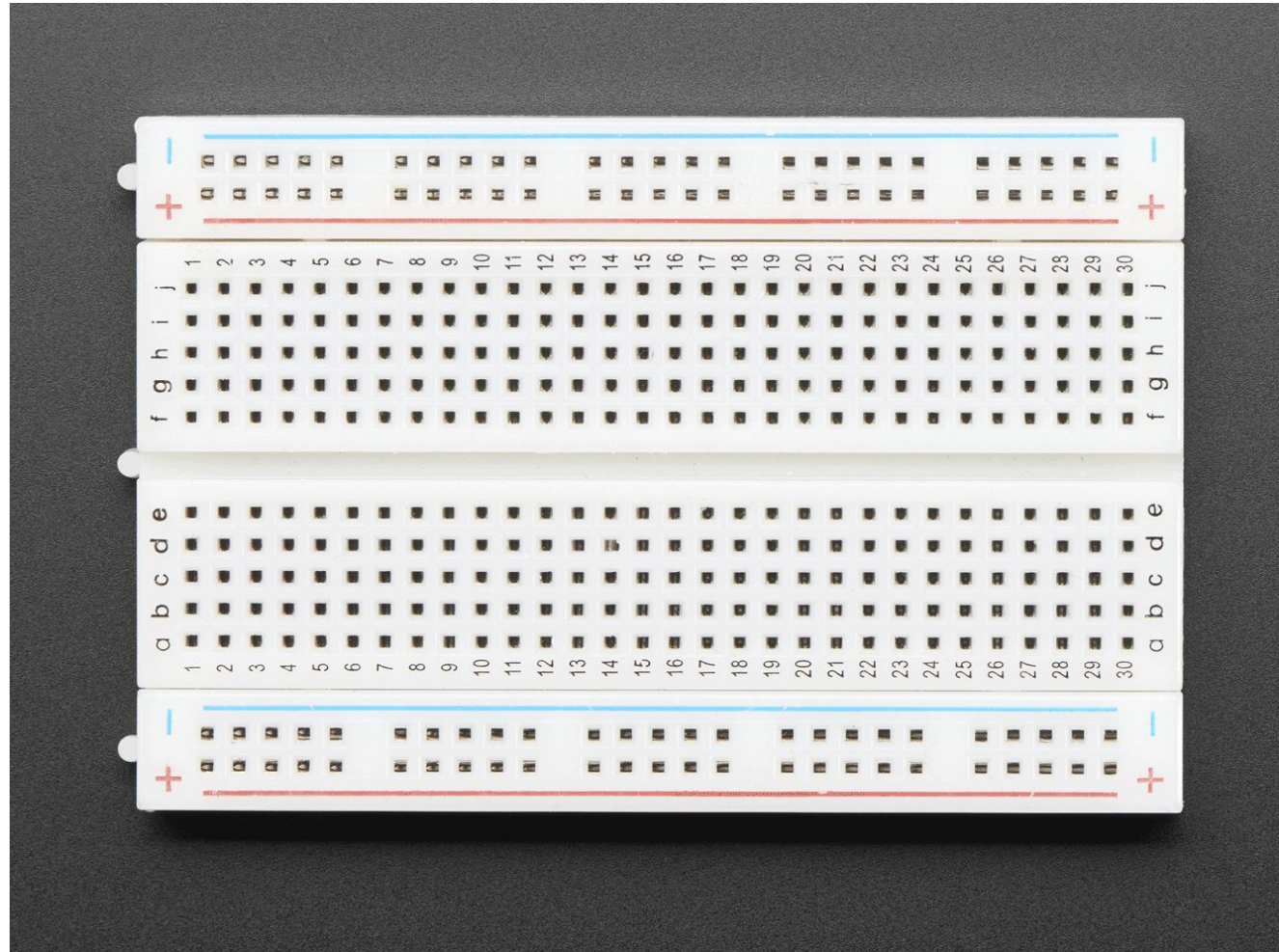
Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Electrical Components

Power source

Wires/conductors

Resistors

Photoresistor

LED (light emitting diode)

Breadboard

Arduino



Arduino IDE

Download from <https://www.arduino.cc/en/Main/Software>

SOFTWARE ENGLISH



ARDUINO WEB EDITOR

Start coding online with the **Arduino Web Editor**, save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards.

[GETTING STARTED](#)

[CODE ONLINE](#)

Download the Arduino IDE



ARDUINO 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows 7 and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10 [Get](#)

Mac OS X 10.10 or newer

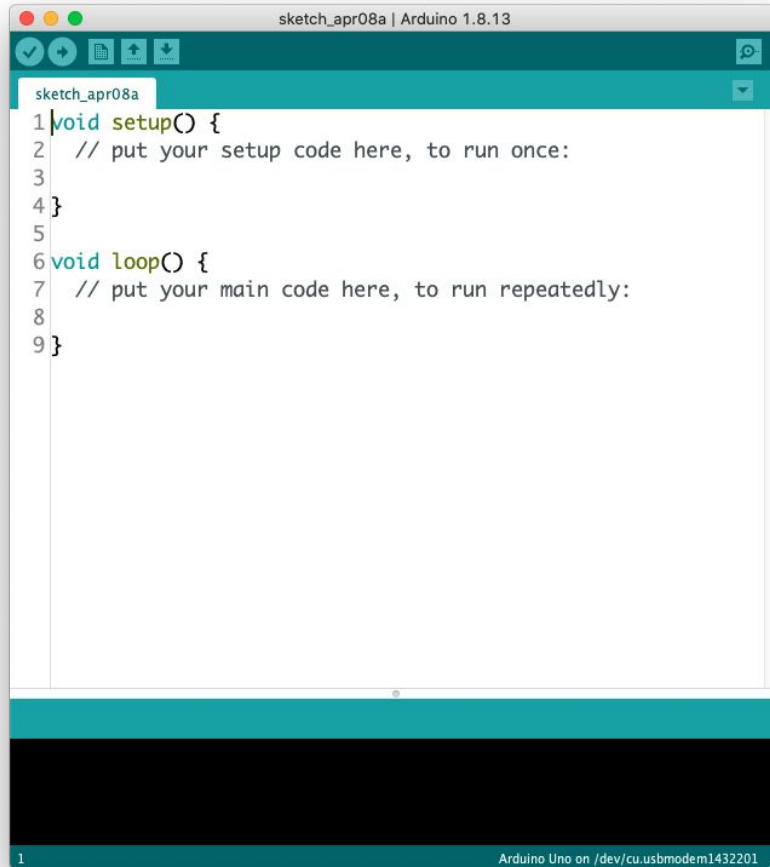
Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

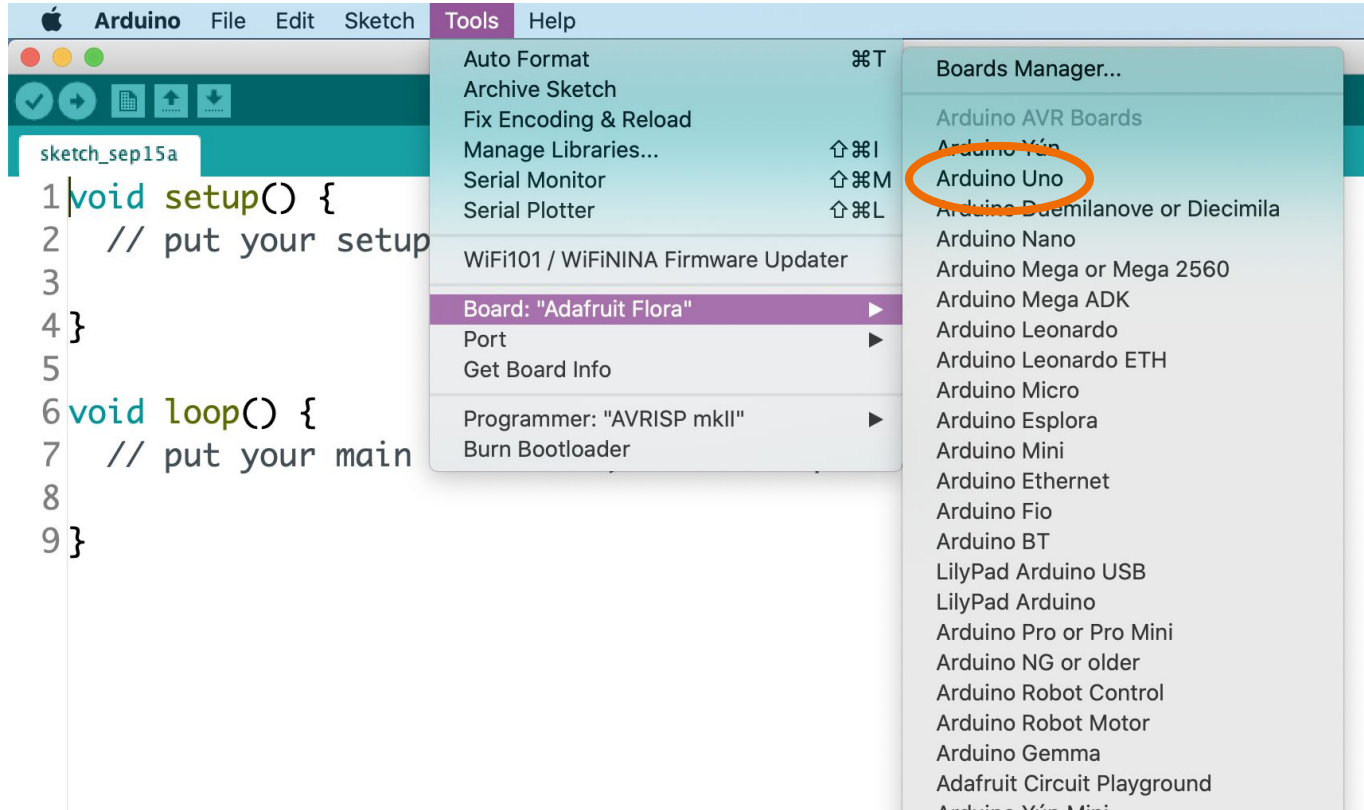


The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_apr08a | Arduino 1.8.13". The main editor area contains the following code:

```
sketch_apr08a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

At the bottom of the IDE, the status bar shows "1" on the left and "Arduino Uno on /dev/cu.usbmodem1432201" on the right.

Check the board



Boards Manager...

Arduino AVR Boards

Arduino Yún

Arduino Uno

Arduino Duemilanove or Diecimila

Arduino Nano

Arduino Mega or Mega 2560

Arduino Mega ADK

Arduino Leonardo

Arduino Leonardo ETH

Arduino Micro

Arduino Esplora

Arduino Mini

Arduino Ethernet

Arduino Fio

Arduino BT

Type

All

101

Arduino SAMD Boards (32-bits ARM Cortex-M0+)by **Arduino**

Boards included in this package:

Arduino MKR WiFi 1010, Arduino Zero, Arduino MKR1000, Arduino MKRZERO, Arduino MKR FOX 1200, Arduino MKR WAN 1300, Arduino MKR WAN 1310, Arduino MKR GSM 1400, Arduino MKR NB 1500, Arduino MKR Vidor 4000, Arduino Nano 33 IoT, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.

[Online Help](#)[More Info](#)

1.8.9

Install

Intel Curie Boardsby **Intel** version **2.0.4** **INSTALLED**

Boards included in this package:

Arduino 101.

[More Info](#)

Close

Check port

The screenshot shows the Arduino IDE interface. The menu bar includes Apple, Arduino, File, Edit, Sketch, Tools, and Help. The Tools menu is open, displaying options such as Auto Format, Archive Sketch, Fix Encoding & Reload, Manage Libraries..., Serial Monitor, Serial Plotter, WiFi101 / WiFININA Firmware Updater, Board: "Arduino Uno", Port: "/dev/cu.usbmodem1432201 (Arduino Uno)", Get Board Info, Programmer: "AVRISP mkII", and Burn Bootloader. The Port option is selected, and its submenu is open, showing Serial ports: /dev/cu.Bluetooth-Incoming-Port and /dev/cu.usbmodem1432201 (Arduino Uno), which is checked. The code editor shows the following code:

```
1 void setup() {  
2   // put your setup  
3  
4 }  
5  
6 void loop() {  
7   // put your main  
8  
9 }
```

Check your code for errors

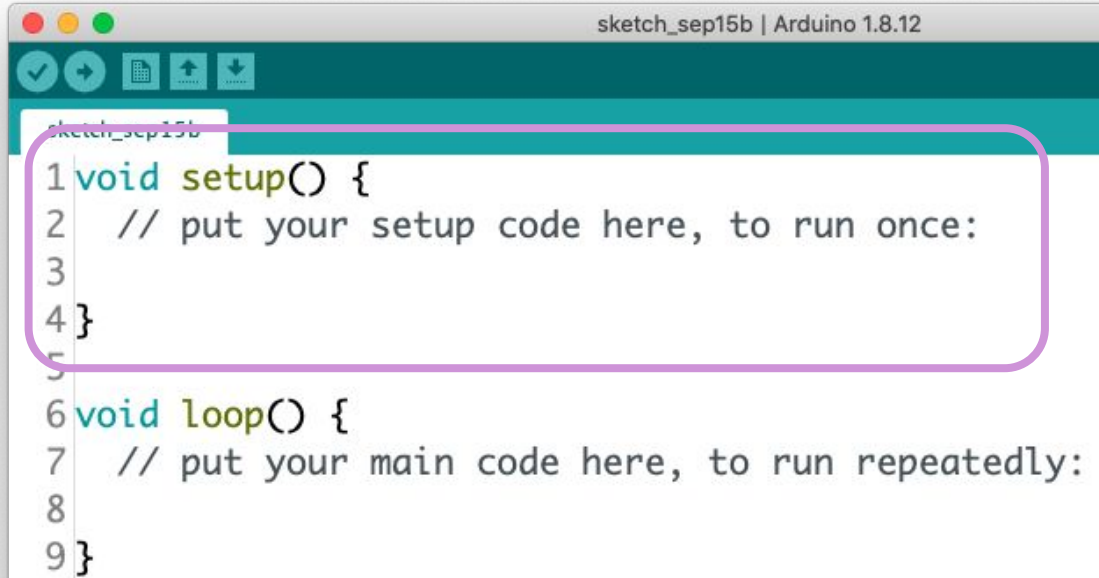


Check your code for errors

Check code and send to the
Arduino



This code is run on initial start up of the Arduino. It's like the pre-flight checklist. It will run once.



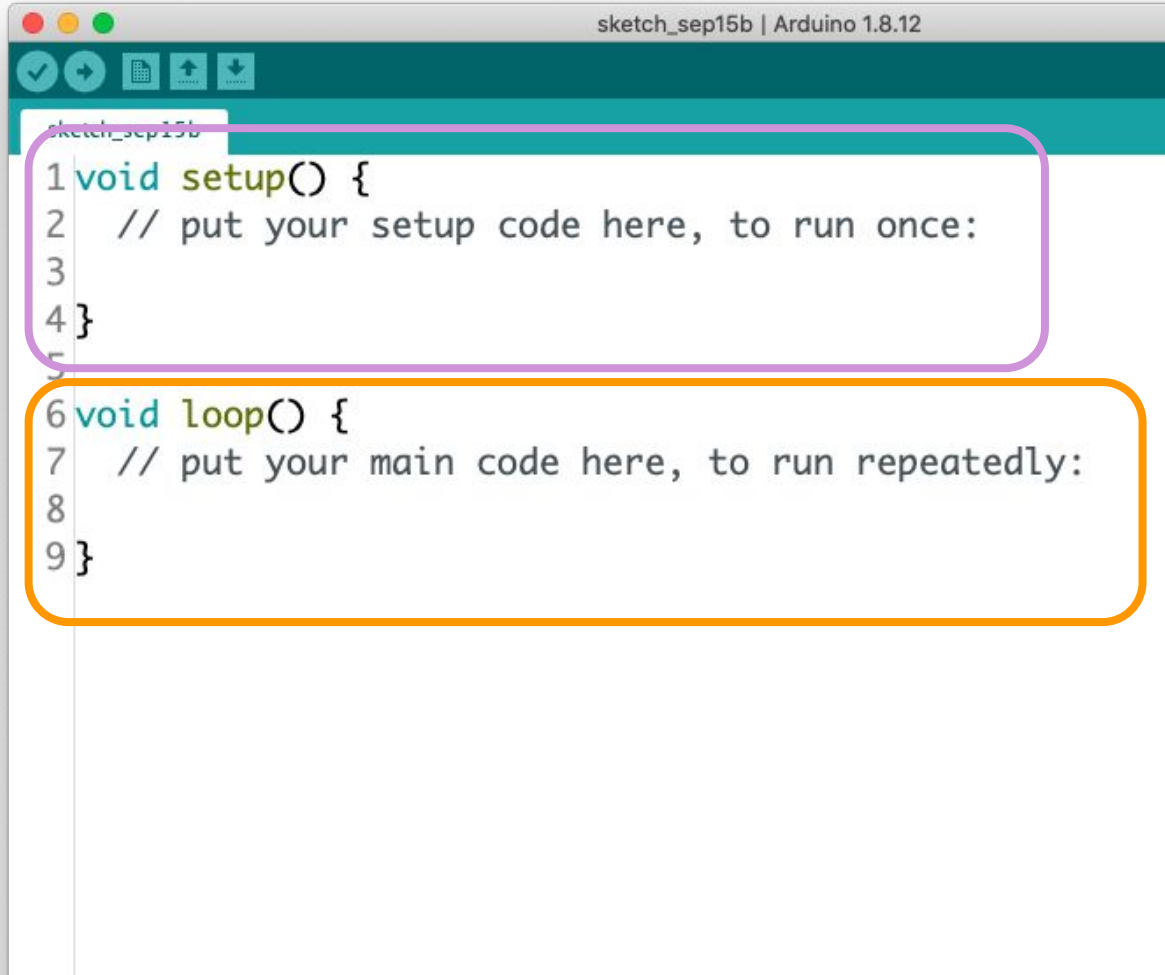
The screenshot shows the Arduino IDE interface. The title bar reads "sketch_sep15b | Arduino 1.8.12". The code editor displays the following code:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

A purple rounded rectangle highlights the `void setup()` function and its comment.

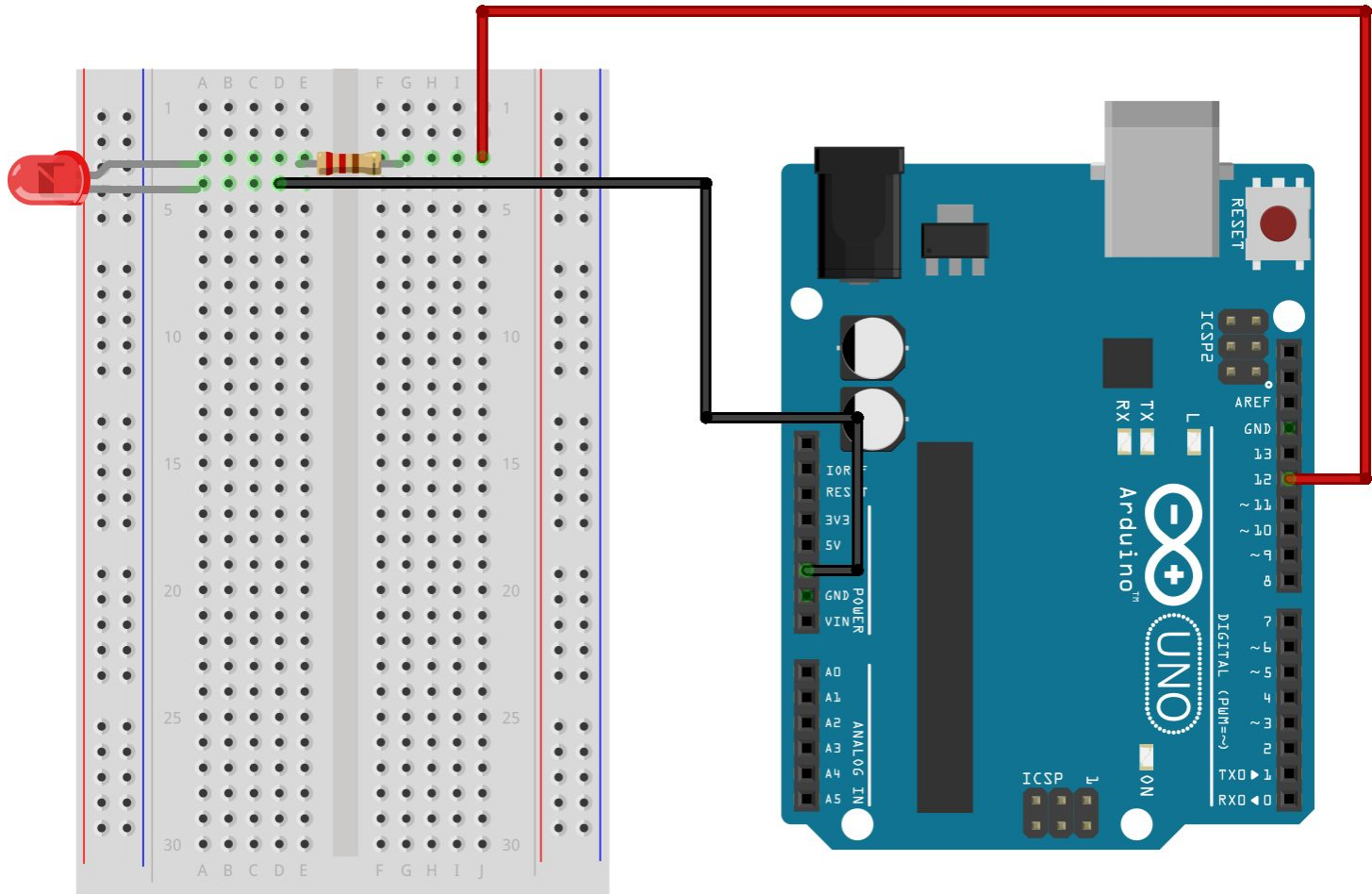
This code is run on initial start up of the Arduino. It's like the pre-flight checklist. It will run once.

This code will be run over and over again, until the Arduino loses power, or you send it new code.

A screenshot of the Arduino IDE interface. The window title is "sketch_sep15b | Arduino 1.8.12". The code editor shows two functions: `void setup()` and `void loop()`. The `setup()` function is highlighted with a purple box and contains a comment: `// put your setup code here, to run once:`. The `loop()` function is highlighted with an orange box and contains a comment: `// put your main code here, to run repeatedly:`. The code is as follows:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

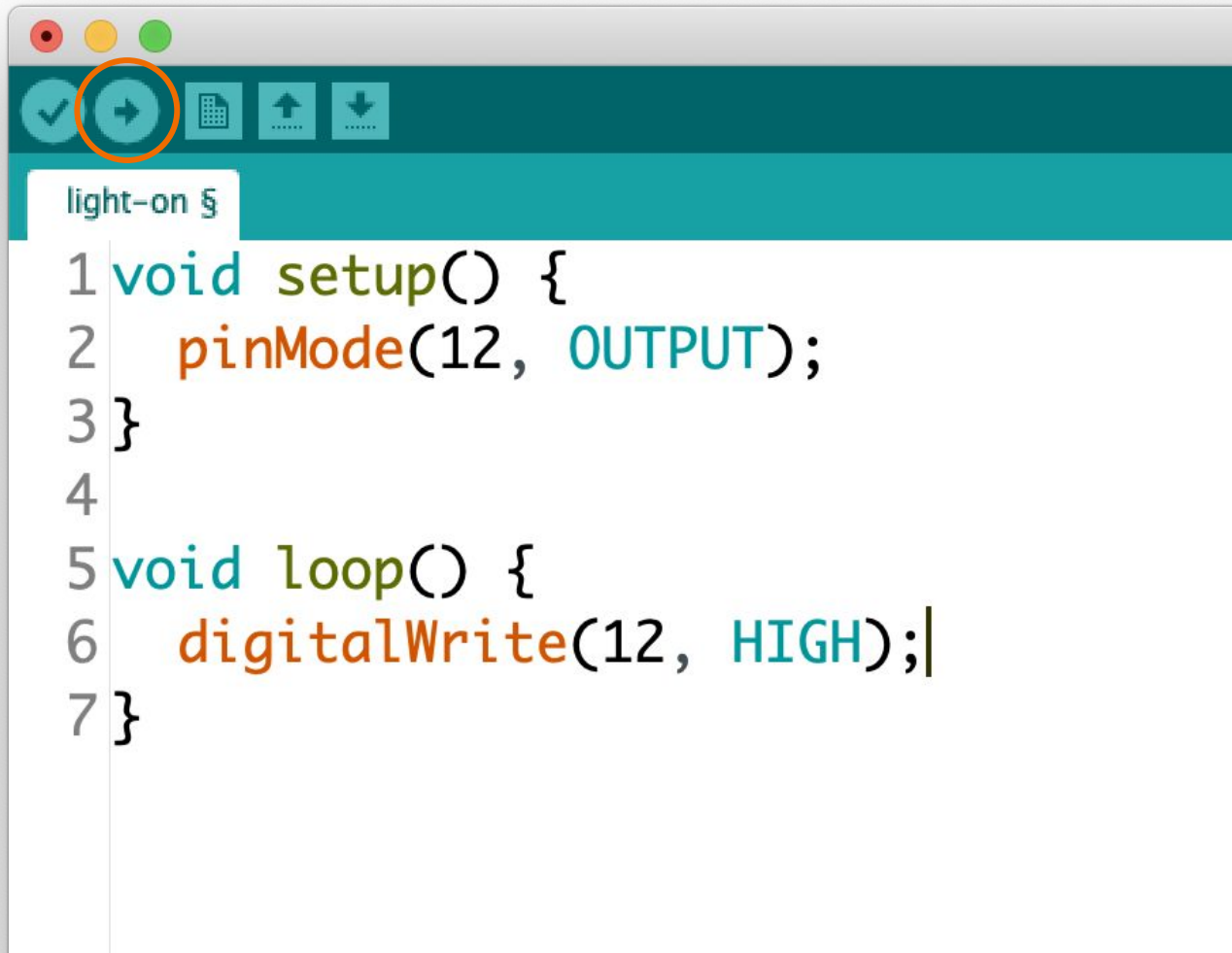
Circuit Assembly



Light on!

```
// <pin #> = a number on the Arduino  
// <type> = OUTPUT or INPUT  
// format = pinMode(<pin #>, <type>);  
pinMode(12, OUTPUT);
```

```
// <pin #> = a pin/hole on the Arduino board  
// <type> = HIGH or LOW  
// digitalWrite(<pin #>, <type>);  
digitalWrite(12, HIGH);
```



The image shows a screenshot of an IDE window with a teal header bar. The header bar contains several icons: a checkmark, a right-pointing arrow (circled in orange), a document icon, an up arrow, and a down arrow. Below the header bar, the text "light-on §" is visible. The main area of the window contains the following C++ code:

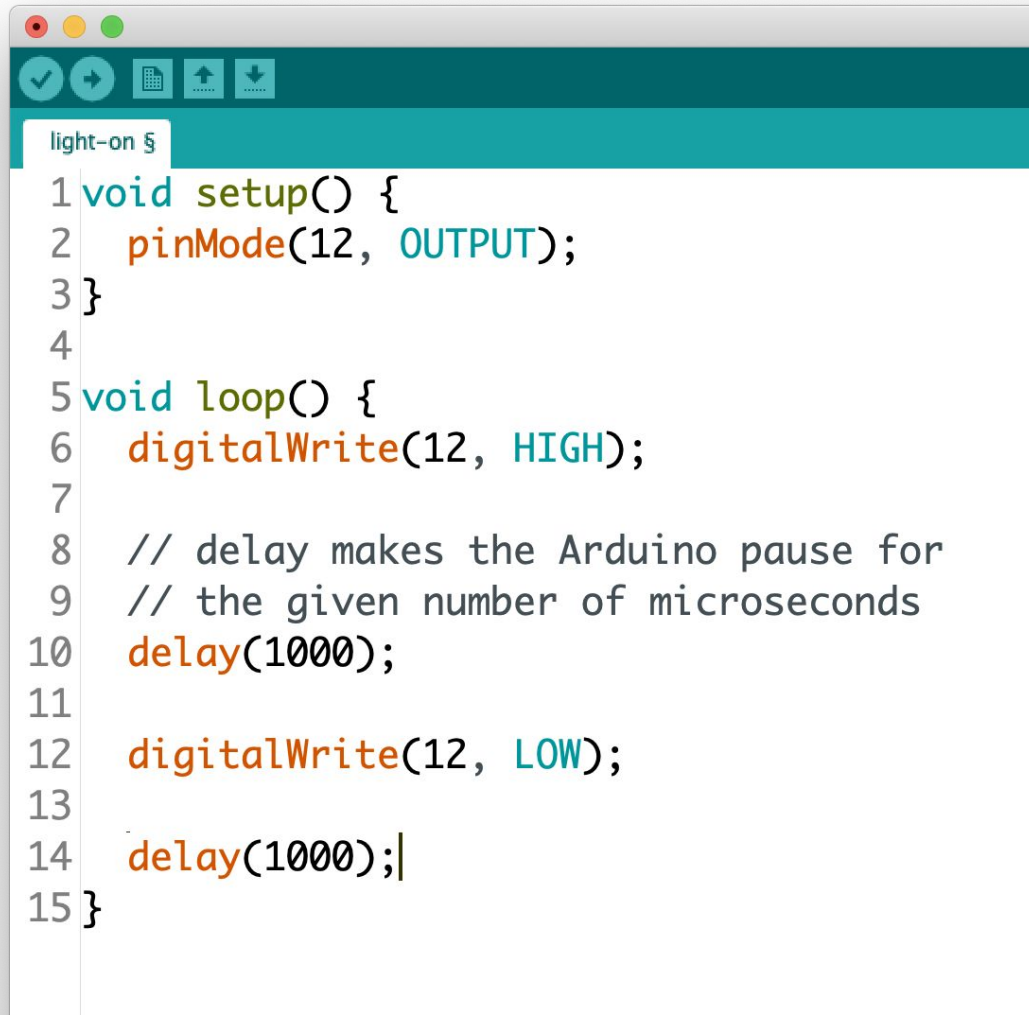
```
1 void setup() {  
2   pinMode(12, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(12, HIGH);  
7 }
```

Why HIGH and LOW instead of ON and OFF?

	HIGH	LOW
INPUT	<p><code>pinMode(12, INPUT)</code> and <code>digitalRead(12, HIGH)</code></p> <ul style="list-style-type: none">• a voltage greater than 3.0V is present at the pin (5V boards)• a voltage greater than 2.0V volts is present at the pin (3.3V boards)	<p><code>pinMode(12, INPUT)</code> and <code>digitalRead(12, LOW)</code></p> <ul style="list-style-type: none">• a voltage less than 1.5V is present at the pin (5V boards)• a voltage less than 1.0V (Approx) is present at the pin (3.3V boards)
OUTPUT	<p><code>pinMode(12, OUTPUT)</code> and <code>digitalWrite(12, HIGH)</code></p> <ul style="list-style-type: none">• 5 volts (5V boards)• 3.3 volts (3.3V boards)• enable the internal 20K pullup resistors	<p><code>pinMode(12, OUTPUT)</code> and <code>digitalWrite(12, LOW)</code></p> <ul style="list-style-type: none">• the pin is at 0 volts (both 5V and 3.3V boards)

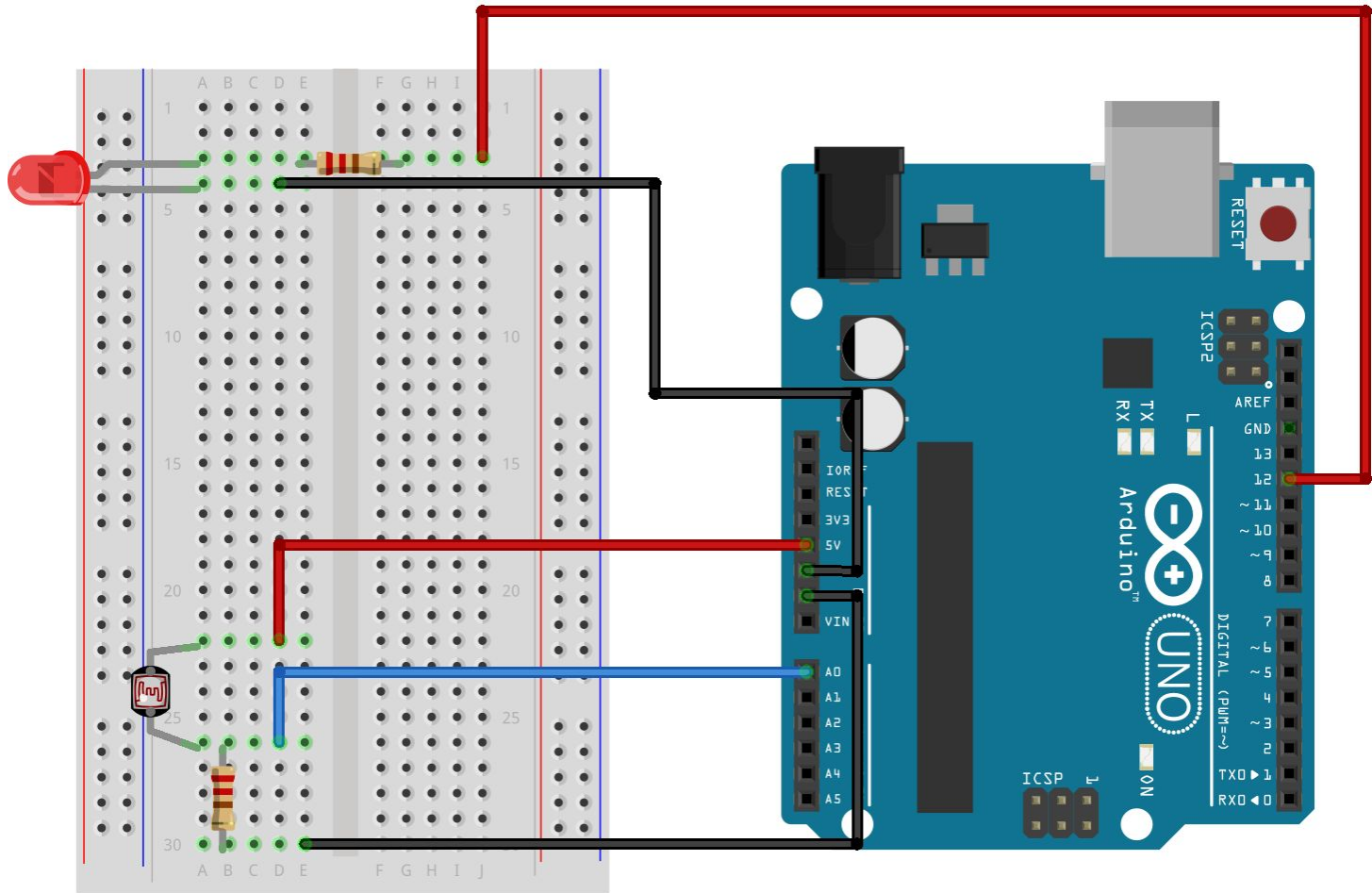
Blinking light

```
// Pause the script for # of milliseconds  
// 1000 milliseconds = 1 second  
// delay(#);  
delay(1000);|
```



The image shows a screenshot of an IDE window with a teal header bar. The header bar contains a tab labeled "light-on §" and a toolbar with icons for a checkmark, a right arrow, a grid, an up arrow, and a down arrow. Below the header bar, the code is displayed with line numbers 1 through 15 on the left margin. The code is as follows:

```
1 void setup() {
2   pinMode(12, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(12, HIGH);
7
8   // delay makes the Arduino pause for
9   // the given number of microseconds
10  delay(1000);
11
12  digitalWrite(12, LOW);
13
14  delay(1000);|
15 }
```



Nightlight

nightlight 5

```
1 // Set up three variable to use later.
2
3 int threshold = 100;
4 int resistance;
5 int initialPhotoValue;
6
7 void setup() {
8   pinMode(12, OUTPUT);
9   initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial| value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }
```

Nightlight

int = a number

float = a number with a decimal

char = letters, numbers and other characters

nightlight 5

```
1 // Set up three variable to use later.
2
3 int threshold = 100;
4 int resistance;
5 int initialPhotoValue;
6
7 void setup() {
8   pinMode(12, OUTPUT);
9   initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial| value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }
```

Nightlight

int = a number

float = a number with a decimal

char = letters, numbers and other characters

Variables start with a letter and can only have letters and numbers, no spaces.

nightlight 5

```
1 // Set up three variable to use later.
2
3 int threshold = 100;
4 int resistance;
5 int initialPhotoValue;
6
7 void setup() {
8   pinMode(12, OUTPUT);
9   initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial| value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }
```

Nightlight

int = a number

float = a number with a decimal

char = letters, numbers and other characters

Variables start with a letter and can only have letters and numbers, no spaces.

nightlight 5

```
1 // Set up three variable to use later.
2
3 int threshold = 100;
4 int resistance;
5 int initialPhotoValue;
6
7 void setup() {
8   pinMode(12, OUTPUT);
9   initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial| value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }
```


Nightlight

int = a number

float = a number with a decimal

char = letters, numbers and other characters

Variables start with a letter and can only have letters and numbers, no spaces.

nightlight 5

```
1 // Set up three variable to use later.
2
3 int threshold = 100;
4 int resistance;
5 int initialPhotoValue;
6
7 void setup() {
8   pinMode(12, OUTPUT);
9   initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial| value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }
```


You are now a coder and an electrical engineer!

Resources

- Arduino Guides and Projects
 - <https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-the-arduino-101genuino-101-board/all#experiment-1-blinking-an-led>
 - <https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/all#project-1-light>
 - <https://create.arduino.cc/projecthub/projects/tags/arduino>
 - <https://www.instructables.com/Arduino-Projects/>
 - <https://maker.pro/arduino>
 - <https://www.hackster.io/arduino>
- Arduino Software IDE
 - <https://www.arduino.cc/en/Main/Software>
- ELEGOO Arduino Kit
 - <https://www.amazon.com/ELEGOO-Project-Tutorial-Controller-Projects/dp/B01D8KOZF4>
- Online Arduino Simulator
 - <https://www.tinkercad.com/learn/circuits>
- Dig deeper into how electricity works
 - <https://sites.google.com/a/acsbr.org/mr-leong-chuen-kit-physics-resources-site/17-current-of-electricity>
 - <https://theengineeringmindset.com/how-electricity-works/>
 - <https://www.youtube.com/watch?v=mc979OhitAg>