

# Programming a Blinking Light

---



## Background/Context

This is an initial lesson in computer programming.

## At a Glance

- **Age Level:** 12-14
- **Subject:** Computer Science
- **Time Needed:** 2 sessions, 45 minutes to an hour each

## Learning Objectives

Basic programming knowledge including output, commands, and loops.

## Standards Alignment

Computer Science Teachers Association: Level 2 Standards CT 3, CT 4, CT14, CL 4, CPP 4, CPP 5

## Inquiry Process

### Ask

There are flashing lights such as red lights on top of tall buildings that warn airplanes, caution lights, or stop signs. Ask students how they think these work. Clearly there is not somebody standing there and flipping a switch on and off. Students should come to the conclusion that they must be controlled in some way by computers.

### Investigate

## Programming a Blinking Light

Students will gain hands-on experience with an Arduino 101 board and get a chance to make an LED blink in one second intervals. Then they will figure out how to change the lines of code for blinking at other intervals. Students will also investigate what a loop is and learn how to terminate it.

### **Create**

Once students are familiar with blinking, loops, and terminating loops, they will be asked to create their own patterns with long on/short off times, or a 1 second/3 second/5 second blink.

### **Discuss**

Have students modify the original blink sketch with an interval of ## milliseconds on then off. Ask what happens. (It will appear to be a solid light.) Ask them why this is so. (Because it is blinking so fast that our eyes do not sense any change in the output. We cannot pick up the blink.)

### **Reflect**

Have students reflect on what was learned, and think about how this can be used in everyday life. Perhaps guide them on next steps and how they could extend this activity.

## **Prerequisite Skills**

None needed

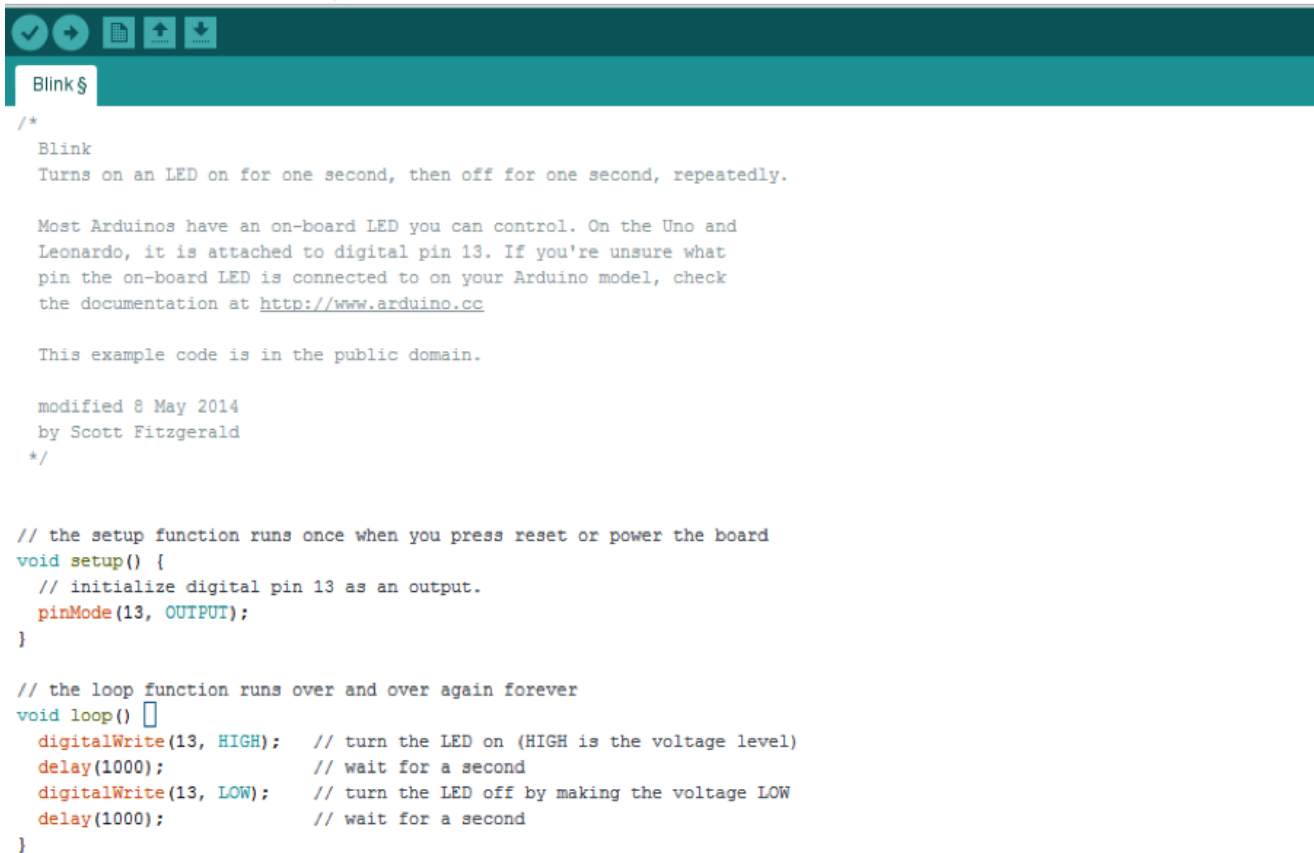
## **Technology and Resources**

- Arduino 101 Board
- Computer with Arduino 101 Software
- LED Light (Recommended): one per every 1-3 children

## Activity

 Blink | Arduino 1.6.9

File Edit Sketch Tools Help



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```


The blink program is an example Arduino 101 program available to any user. It is used to turn an LED light on for one second and off for one second. It continually repeats this process and will not stop until the Arduino 101 is powered off. Explain the program to your students and have them identify the output, the blinking light.

In the blink program DigitalWrite and Delay are command lines, an instruction given by a user telling a **computer** to do something. They produce a specific output, something that is produced, delivered, or supplied (data) using a computer or other device. In the blink program "digitalWrite (13, HIGH)" sets the output for pin 13 (the LED light) to high, turning on the light. "digitalWrite (13, LOW)" does the opposite, by setting the output for pin 13 to low it turns off the light. The delay command creates a 1,000 millisecond (one second) delay in issuing other commands, creating a one second pause between turning the light on and off. Students will work on creating their own command lines. Have your students add in command lines that will turn the light on for 2 seconds and off for 2 seconds after the existing blink. The program should look similar to the following.

## Programming a Blinking Light

 Blink | Arduino 1.6.9

File Edit Sketch Tools Help

```
  
Blink $  
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  Most Arduinos have an on-board LED you can control. On the Uno and  
  Leonardo, it is attached to digital pin 13. If you're unsure what  
  pin the on-board LED is connected to on your Arduino model, check  
  the documentation at http://www.arduino.cc  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(2000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(2000);           // wait for a second  
}
```

Next have your students use different delays until they find a blink that happens to quickly for human eyes to see. They will instead see a continuous light. Have your students discuss the speed computers are capable of sending commands and why they cannot see the blinking.

Next your students will learn about loops and how to terminate them in the Arduino 101 Program.

A loop is a sequence of instructions that is continually repeated until a certain condition is reached. In the basic blink program the void loop() function creates a loop, but has no terminating condition. This creates a program that will only end when the Arduino 101 is turned off. To terminate the loop a user must add a conditional statement, a set of rules performed if a certain condition is met, a variable to keep track of the number of times the loop has run, and a form of increasing the value of the variable. Have your students create a program that will terminate after 7 occurrences, it should look like the following.

## Programming a Blinking Light

 Blink | Arduino 1.6.9

File Edit Sketch Tools Help

```

Blink$
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

//create a variable to count the number of loops
int time=0;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
  time++;                //increase the variable for time by 1 (add one occurrence to loop)
  if(time==7)            //conditional statement to check the number of times the loop has occurred
  {
    exit(0);              //terminate loop if condition has been met
  }
}

```

Lastly, have your students use what they know about commands and loops to create their own pattern for their clown nose. Students should be able to change the number of occurrences, the time of the blinks, and the time of the delays. The following is an example of a program that will blink for one second, then two seconds. It will repeat the pattern 7 times.

## Assessment

To assess your students give them a code and have them determine what the output of the code should be. Your code should include different time delays and terminating conditions. You may also assess your students by asking them to produce a specific outcome. Have them create a specific pattern using commands that will repeat a specific amount of times.

## Additional Tips and Information

There is an LED light included in the Arduino 101 board. It is suggested that an LED is plugged into pin 13 to create a more noticeable effect, but it is not required.

## Source

Kristen Messer [kmesser](#), Teacher, Manteca Unified School District, California, United States

Students will understand that a computer controls flashing lights and then create code and build a blinking light with an Arduino 101 board in one second intervals.