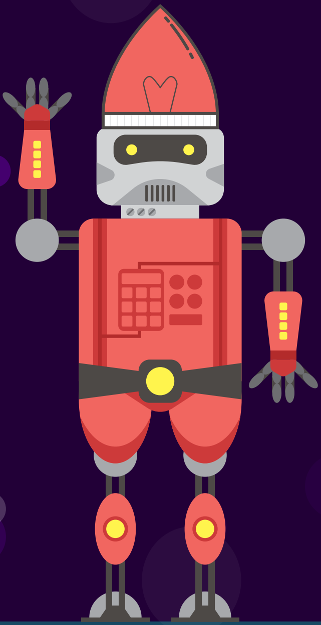
Holiday Project



Creation Crate

BUILDING THE MAKERS OF TOMORROW

INDEX

INTRODUCTION

Welcome to the Creation Crate! Delivering a simple and fun way to learn about electrical engineering and coding. You've taken your first step in learning how to build and program electronics.

WHAT ARE WE CREATING?

Today, we're making a special holiday project of your choice!

You'll choose from three holiday-themed cards and animate them with LEDs.

WHAT WILL WE LEARN IN THIS PROJECT?

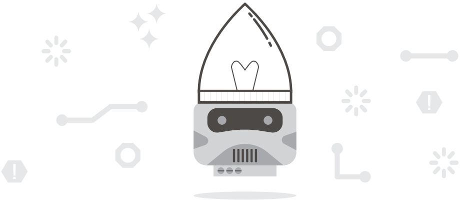
WE WILL LEARN ABOUT ELECTRONICS:

- What a microcontroller is
- What a breadboard is
- What resistors, LEDs, and jumper wires are

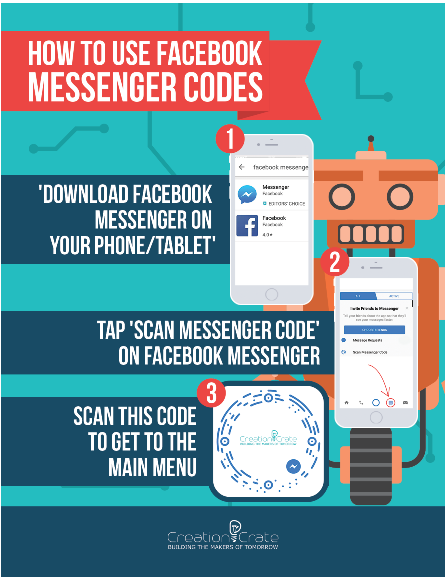
PROGRAMMING

- How to get started writing code
- The Arduino programming language
- About these terms: variables,
- About these commands: Void Setup,

1. This is the first of our lessons that includes Facebook Messenger Codes. You can learn more about many components by accessing short video descriptions through the Codes pictured throughout this project plan. Here's how it works:



INTRODUCTION



HOW TO USE FACEBOOK MESSENGER CODES

1 'DOWNLOAD FACEBOOK MESSENGER ON YOUR PHONE/TABLET'

2 TAP 'SCAN MESSENGER CODE' ON FACEBOOK MESSENGER

3 SCAN THIS CODE TO GET TO THE MAIN MENU

CreationCrate
BUILDING THE MAKERS OF TOMORROW

PROJECT SUPPORT PAGE:

LINK

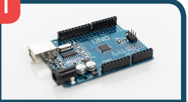
www.creationcrate.com/happy-holidays

IRMV93

Still Need Help?
Send us a note **www.creationcrate.com/contact-us**

PART LIST

X1

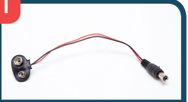


UNO R3 (ARDUINO -COMPATIBLE) MICROCONTROLLER

X1

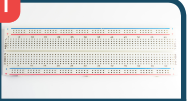
9V BATTERY

X1



9V POWER CABLE

X1



BREADBOARD

X8 EA

(WHITE & YELLOW) LEDS

X2 EA

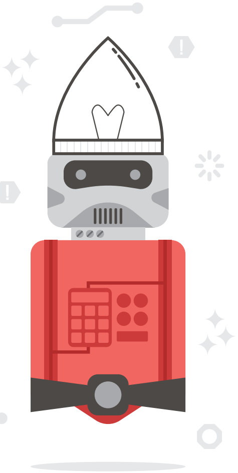
(RED, GREEN, BLUE) LEDS

X8

220OHM RESISTOR

X2

PIECE DOUBLE-SIDED TAPE



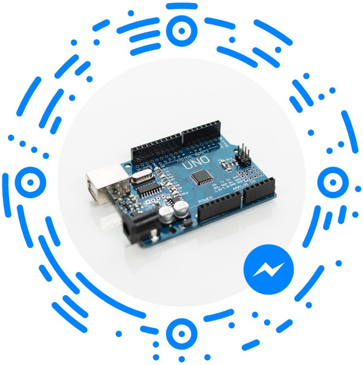
**** You will need a couple household items that are not included in the box.**

- A couple pieces of clear tape to hold the decorative card to the breadboard
- A sharp pencil (or something else to poke some holes in the card)

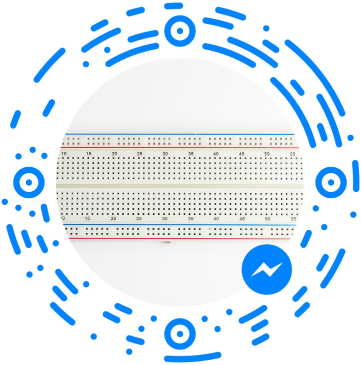
Want to learn more about what these components do? Visit **www.creationcrate.com/classroom/component-library** for a more detailed description!

WHAT IS A MICROCONTROLLER?

In this project, we will be programming the UNO R3 microcontroller to turn LEDs on and off using a randomized sequence. A microcontroller is a tiny computer that runs programming code. Instead of a keyboard and screen, there are multiple input and output "pins" that can read the value of buttons and sensors or turn lights and motors on and off. In this project, you'll learn how to write the code to control these pins.

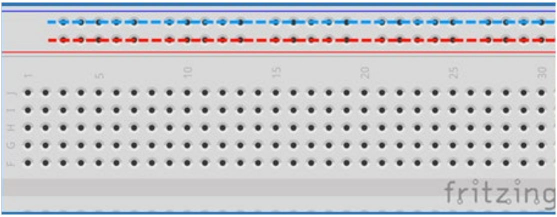


WHAT IS A BREADBOARD?



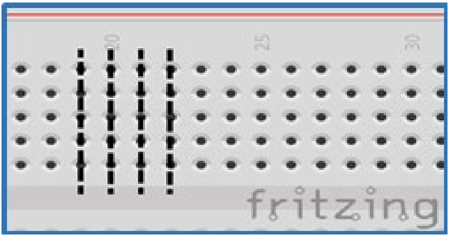
We will be using what's known as a *breadboard* to connect all the wires and components together. A breadboard is used for prototyping a circuit. This way the wires don't have to be twisted together or soldered. Instead the breadboard has a series of columns and rows that are connected internally.

The two rows on the top and bottom are connected inside the breadboard. For example, all of the holes in the blue row are connected and are used for the negative side of the current. Red is used for the positive current.



WHAT IS A BREADBOARD?

The middle of the breadboard is made up of columns that are connected in the same way. There is a separation in the middle of the board creating two distinct halves of the board, which are not connected. For example, the five holes on the top portion of column 20 are all connected, as are the five on the bottom half.

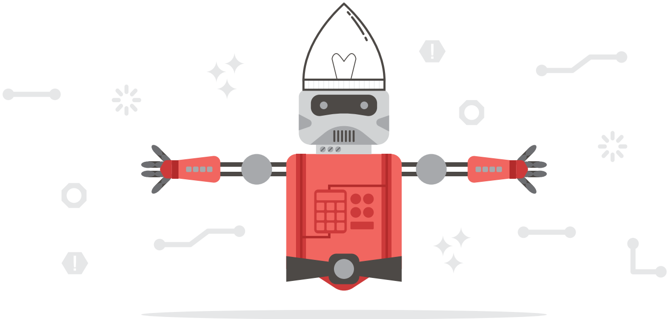


Wires in any of the five holes in the same column of the same half of the board are connected.

WHAT IS A LIGHT EMITTING DIODE (LED)?

First, a **diode** is a two-terminal electronic component that conducts electricity primarily in one direction.

A **light-emitting diode** is a special type of diode that, when a suitable voltage is applied to the leads, forces electrons to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called **electroluminescence**.



WHAT IS A RESISTOR?

A resistor does exactly what it sounds like: it resists the flow of electricity. You can think of electricity running through wires as water running through a hose. If you form a kink in the hose, the water will run slower.

Resistors have different values, measured in Ohms. The higher the Ohm rating, the less electricity that can pass through it. To tell the difference between the values, resistors have color bands around them. In this project, we'll use a 220 Ohm resistor, which has a color band code of **Red Red Brown**. **www.creationcrate.com/classroom/resistor-chart**

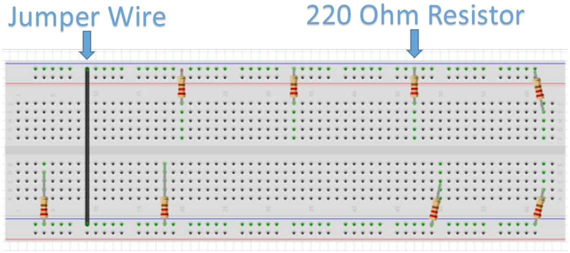
Want to learn more about microcontrollers, resistors, and breadboards? Visit **creationcrate.com/classroom/component-library** for a more detailed description!

HARDWARE

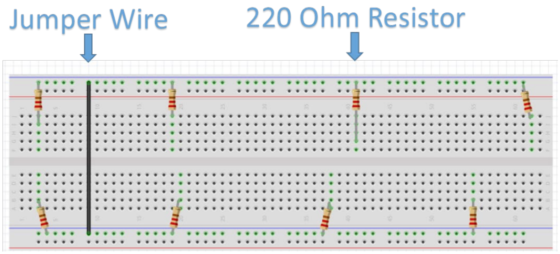
1. First, let’s connect the Resistors and a jumper wire for the negative current. Use the 220 Ohm resistors with the Red – Red - Brown banding.

   Since the LEDs are in different locations, we will need to place the resistors in different locations based on the card you choose.

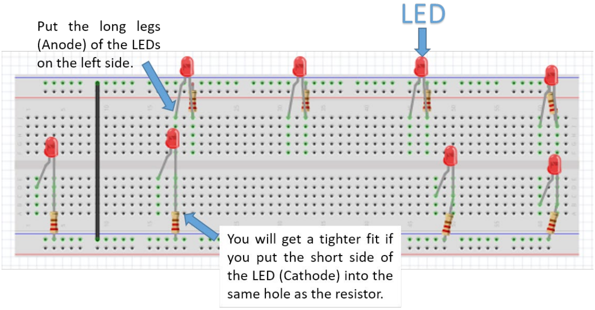
Configuration 1 is for both the Christmas and Hanukkah Cards.



Configuration 2 is for the Holiday Card.



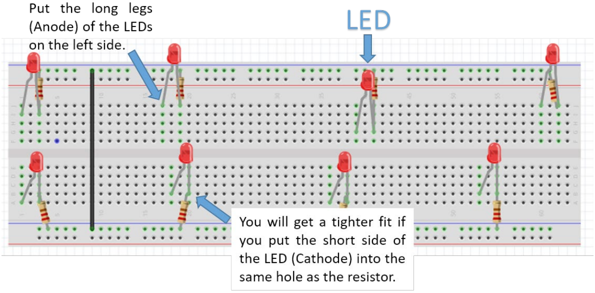
2. Next, place all of the LEDs in the locations below so they will line up with the holiday cards. Again, **Configuration 1** is for the Christmas and Hanukkah Cards, **2** is for the Holiday Card.



Put the long legs (Anode) of the LEDs on the left side.

You will get a tighter fit if you put the short side of the LED (Cathode) into the same hole as the resistor.

Configuration 2 is for the Holiday Card.

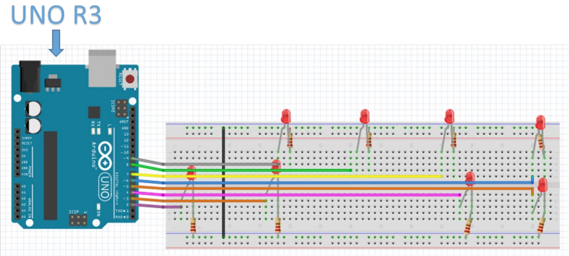


Put the long legs (Anode) of the LEDs on the left side.

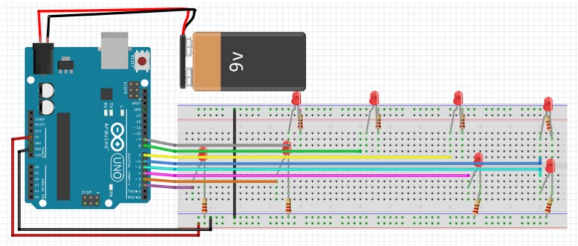
You will get a tighter fit if you put the short side of the LED (Cathode) into the same hole as the resistor.

3. Now, let’s attach the UNO R3 to the breadboard. First, orient the UNO R3 like it is in the image below, with the micro USB port on the top. Start on the bottom with D2, connecting it to the first LED on the bottom half of the UNO R3. Even though the LEDs are in a different location for the Holiday card, the wires will still be connected in the same order. **Just ensure that the wire is in the same row as the positive lead of the LED.**

(Tip: The color of the jumper wire does <u>not</u> matter. They all do the same thing!)



4. Connect 9V battery to the barrel plug of the UNO R3. This will power the microcontroller when it is not plugged in to the computer.



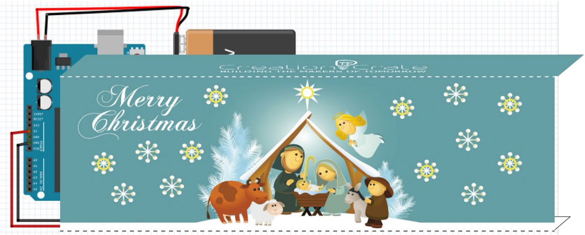
5. Time to set up the decorative card. Regardless of the theme you choose you will start by taking a sharp pencil and carefully poke a hole in each of the locations with a circle. (Tip: fold up a dish towel and place the card on top. This should provide enough cushion for the pencil tip to penetrate the card and make the holes. You don't need to poke it all the way through, just enough for the light from the LED to shine through.)

6. Fold the card back along the dotted lines, creating a 90 degree angle.

Creation Crate
BUILDING THE MAKERS OF TOMORROW
Merry Christmas

7. Place the card over the breadboard. You may need to move some of the LED slightly so they line up with the holes in the card. (Tip: they do NOT need to be pushed through the card.) You may also need to move the wires a little but so they don't push the card up too much.

Once the card is in place, use your clear tape to connect it to the breadboard on both sides.



8. Place the project on its side so the breadboard is in the back and the card is facing outward. You can then tuck the battery behind the breadboard and use the mounting squares to tape them to the back.

After you are done programming the UNO R3 below, you can also mount it to the back of the breadboard in the same way.

THAT'S IT FOR THE HARDWARE!

NOW WHAT?

AWESOME!

You built the hardware and it looks great!...but it doesn't do anything yet.

In order to animate it, you need to program the UNO R3! Microcontrollers are a combination of hardware and coding. You can't have a working project with just one half.

HOW DO WE START PROGRAMMING?

1. Let's start by going to **https://www.arduino.cc/en/Main/Software**
2. Download the Arduino Software (IDE)
3. Open the Arduino program and navigate to "File > New"
4. Delete all of the existing text/code. We'll add this back later.
5. Navigate to "File > Save" and save the program as "Holiday Project"
6. Turn the page and start programming!

PROGRAMMING BASICS:

WHAT IS CODE?

Code is a set of instructions that the microcontroller will do. We have to tell the microcontroller when an instruction is done, and the next one should be started. For the most part, every line of code is a different instruction. We use semicolons ";" to separate these lines of code, and almost every line ends with one.

WHAT IS A VARIABLE?

A **variable** is a placeholder for a number or other value.   You can also think of variables like containers - they're used to store information. For example, say we make a **variable** called "x". To make a variable that holds an **int (short for integer)**, we write:

```
int x;
```

If we set x equal to 6 and tell the microcontroller to read x, it will tell us the value of x is 6.

To set x equal to 6 we write:

```
X = 6;
```

We can do math on these variables, so for instance we can set Y equal to X plus 3;

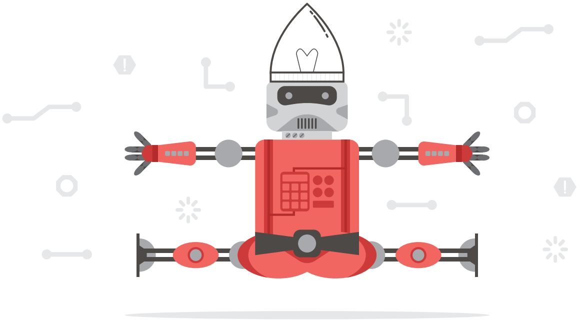
```
Y = X + 3;
```

The value of Y is now 9.

WHAT ARE THE SETUP() AND LOOP() COMMANDS?

When you start the Arduino IDE, your code will automatically have two “functions” called **“void setup()”** and **“void loop().”** Anything inside the curly brackets { } after the line ***void setup()*** will run once when you turn on your UNO R3. Here you will establish the starting points for the functions you want the UNO R3 to perform. For example, we want to start with all of the LEDs turned off.

Anything inside the curly brackets { } after the line ***void loop()*** will run again and again until you turn off your Uno R3. In this case, we are going to write some commands that will randomly turn the LEDs on and off to give a twinkling effect.

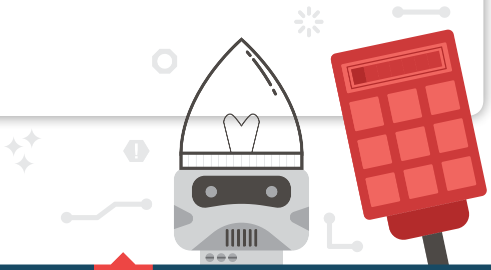


PROGRAM

Note: Each line of text here is a line of code. They run one after another. Each command ends with a semicolon. (Usually this is after every line, although some commands can be multiple lines long.)
Lines that start with "//" or /* are called comments which are just explanations and will not execute. The /* is for a comment that spans multiple lines and will need to end with the */ .You don't need to type these lines below out if you don't want to. In fact, it's encouraged to write your own comments if it helps you understand the code better when you're reading it.

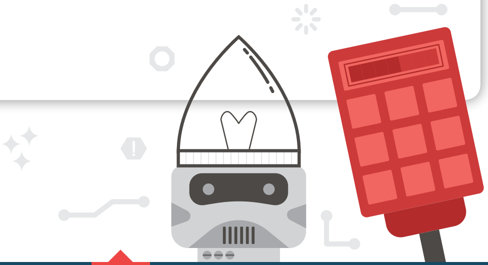
Notice that the first few lines explain something about the project and the code and start with the /* and end with the */ . Let's begin!

```
/* Creation Crate Christmas | Hanukkah | Holiday project
Celebrate the season with a fun project that lights up a holiday card
with randomly activated LEDs.

Step 1: Set User Defined Variables
In our circuit we placed a wire
in the Arduino's digital pins 2-8.  Below we are giving each of these a
name so the code will be easier
to understand later.  The [int] command tells the program we are
assigning and INTEGER, or whole number, to the named variable.
For example, we are hooking PIN 2 of the Arduino to LED 1 on the
breadboard*/

int led1 = 2; //This means that when we want to use PIN2 on the board,
we can use "led1" instead, which helps remember what's connected to that
pin.  In the first project we will not need these variables, however, you
will need to use them for the monthly challenge at the end.
int led2 = 3;
int led3 = 4;
int led4 = 5;
int led5 = 6;
int led6 = 7;
int led7 = 8;
int led8 = 9;
```



PROGRAM (CONT'D)

```
int randLed; //Here we are setting a name for a variable that we will
assign a value to later.
long randDelay; //Here we are doing the same thing as above, except we
can use a longer value.

/* Step 2: Create Setup Loop
This 'loop' is not really a loop. It runs once in the beginning to
create the default values for our LEDs.
Below we will tell the program how each PIN should be used, either INPUT
or OUTPUT of data [pinMode].  In this case we are telling the UNO R3 to
send data OUT to the LEDs.*/

void setup () {
pinMode (led1,OUTPUT); //This means that PIN 2 (which is assigned to the
led1 variable) will send data to what its connected to.
pinMode (led2,OUTPUT);
pinMode (led3,OUTPUT);
pinMode (led4,OUTPUT);
pinMode (led5,OUTPUT);
pinMode (led6,OUTPUT);
pinMode (led7,OUTPUT);
pinMode (led8,OUTPUT);

/* Now, we need a value to start with.  Sending a value [digitalWrite]
of LOW sends no current and turns the LED off, a value of HIGH sends a
current and turns the LED on.*/

digitalWrite (led1,LOW);
digitalWrite (led2,LOW);
digitalWrite (led3,LOW);
digitalWrite (led4,LOW);
digitalWrite (led5,LOW);
digitalWrite (led6,LOW);
digitalWrite (led7,LOW);
digitalWrite (led8,LOW);

}
```



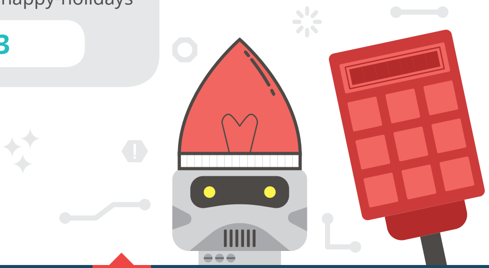
PROGRAM (CONT'D)

```
/* Step 3: Create Main Loop
The previous sections are where we set up the variables. This section
is where we put them to work! This part of the program is a 'loop'. It
repeats itself over and over again.*/

void loop () {

randLed = random(2,10);
digitalWrite (randLed,HIGH);
randLed = random(2,10);
digitalWrite (randLed,HIGH);
randLed = random(2,10);
digitalWrite (randLed,HIGH);
delay(500);

randLed = random(2,10);
digitalWrite (randLed,LOW);
randLed = random(2,10);
digitalWrite (randLed,LOW);
randLed = random(2,10);
digitalWrite (randLed,LOW);
delay(500);
}

//After the code in loop() finishes, it starts again from the first line
in loop()
```

DOWNLOAD THE CODE AT:

www.creationcrate.com/happy-holidays

🔒 **IRMV93**



ALMOST THERE!

Now that you've entered the code, it's time to see if it works!

MAC USERS

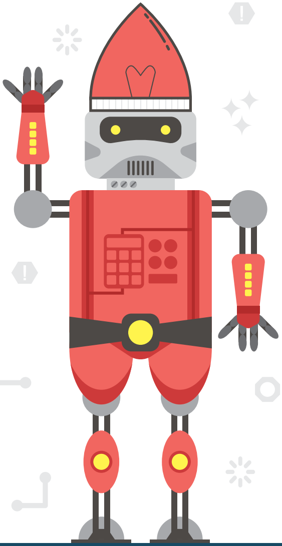
Go to **https://creationcrate.com/happy-holidays** (password: **IRMV93**) and download the Mac OS driver before continuing. If you are not on a Mac, or have done this in a previous project, you can skip this step.

Password:

1. Connect your Uno R3 board to your computer with the USB cable provided.
2. Navigate to 'Tools > Port' and select the port connected to your Uno R3. (If you have trouble finding the right port, try disconnecting other USB devices).
3. Navigate to 'Tools > Board' and select "Arduino/Genuino Uno"
4. Navigate to 'Sketch > Upload'. You should see this at the bottom:

(If the program doesn't upload, there are errors in your code. Don't worry! Errors are very common - I'd be more surprised if you didn't have any! See the next page for possible solutions).

5. Your project should now be working!
6. Solve the exercises on pages 22-23 to test your understanding of the project.

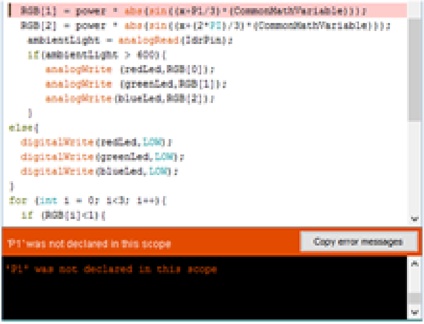


COMMON ERRORS

Note: If your project is working, you can skip this section!

PROGRAMMING ERRORS

If a line is highlighted red like bellow, there is an error on that line! Fix this error first and try again.



```
 RGB[1] = power * abs(sin((x+P1/3)*(CommonMathVariable)));
 RGB[2] = power * abs(sin((x+(2*PI)/3)*(CommonMathVariable)));
  ambientLight = analogRead(ldrPin);
  if(ambientLight > 600){
     analogWrite (redLed,RGB[0]);
     analogWrite (greenLed,RGB[1]);
     analogWrite(blueLed,RGB[2]);
  }
else{
 digitalWrite(redLed,LOW);
 digitalWrite(greenled,LOW);
 digitalWrite(blueled,LOW);
}
for (int i = 0; i<3; i++){
 if (RGB[i]<1){
```

'P1' was not declared in this scope

```
'P1' was not declared in this scope
```

These are the most common programming errors:

1. ***Typos*** - The words have to be exact! Make sure your code matches the code in this booklet. Note: Letters are case-sensitive.

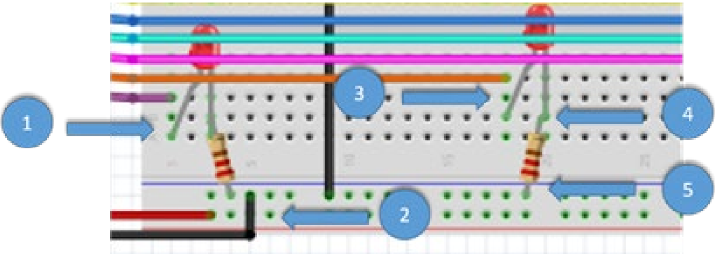
2. ***Missing ';'*** - Any time you write a line of code, it has to end with ';'. The only exceptions in this code are lines that start with 'void', 'if', 'else', or 'for'.

3. ***Missing/extra brackets*** - Any time you use an opening bracket '(' or '{', you must have a closing bracket to match it ')' or '}'. Having too many brackets will also create errors. Lastly, make sure these are placed correctly!

HARDWARE ERRORS

If your code checks out and your project still isn't working, there may be something wrong with your wiring.

Here are some common hardware wiring issues:



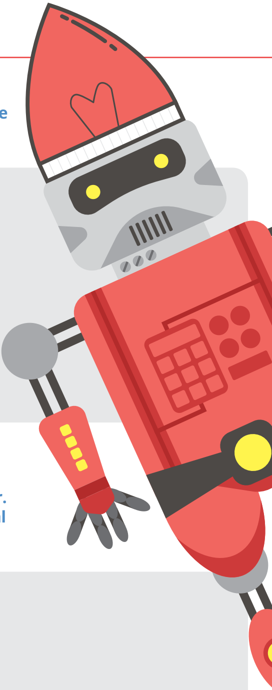
1. Positive (Anode) side of the LED should be on the left. This is the longer of the two leads.
2. Make sure the positive and negative leads are in the correct rails. The Positive is the red rail, the Negative is the blue.
3. The jumper wires from the UNO R3 are in the same row as the positive lead of the LED
4. One side of the resistor is in the same row as the negative lead of the LED. Because these leads are so thin, sometimes it is helpful to put both of these leads in the same hole on the breadboard.
5. The other side of the resistor is in one of the holes of the negative (blue) rail of the breadboard.

STILL NEED HELP?

Visit **www.creationcrate.com** and use our contact form!

EXERCISES

SOLVE THESE PROBLEMS AND WRITE THE ANSWERS BELOW.



1. Right now the delay is set to 500 milliseconds. Use the randDelay variable we declared and change this to generate and use a random number from 200-2000 milliseconds.

Answer:

2. We set up the variables for the individual LEDs. However, you may have noticed that we did not use them in the initial version of the code. Instead we just used a random generator. Your challenge is to modify the code to control each individual LED deliberately.

Answer:

HINT: You will need to turn the LED on and off with a delay in-between. You can do this individually or in groups.

MONTHLY CHALLENGE

There are many different approaches to solving problems. With this project, we chose to control each LED individually. Your challenge is to find a way to create two groups of LEDs by changing the wiring on the breadboard and then change the code to control these two groups. One group can be the even numbered LEDs and the other group the odd ones.

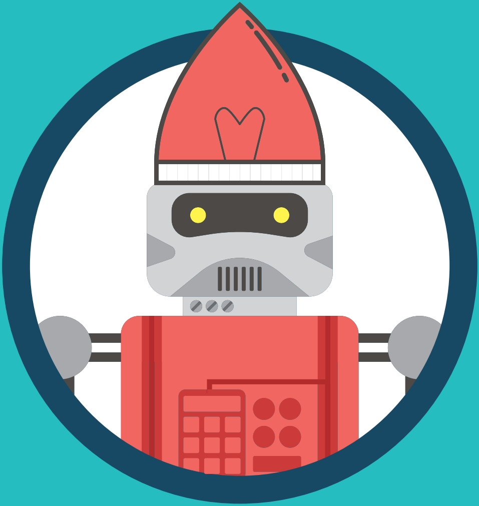
HINT: You can use jumper wires to connect them to each other or use the positive rails on the top and bottom.

SUPPORT PAGE:

www.creationcrate.com/happy-holidays

🔒 **IRMV93**

Still need help?
Go to **www.creationcrate.com** and use our contact page!



Creation Crate

BUILDING THE MAKERS OF TOMORROW