Lesson Plan (2-3 periods):
30 min: Students complete #1-4 below (5-7 optional)
Share Code

Observations?
-Very repetitive
-DRY coding principal: Don't Repeat Yourself

How to reuse the same code over and over again? Use a function! ← pick up here next day

Day 2:
Do Now Optional Challenge: What does this Morse Code message say? Play in TinkerCad. (Answer: HEY)

We wrote Morse Code that looks something like this:
Annotated basic version: https://www.tinkercad.com/things/57XQ7hDPsCD-copy-of-morse-code-basic/editel?tenant=circuits (Copied below)

Super repetitive. Can get around using functions.

Show how to write function for dot

```
void dot(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Note: void b/c it is not returning (outputting) anything to the computer program
Dot = name of function

This is a function definition. Teaches the computer what dot() means. To actually use the function we have to call it by typing dot(); inside of loop.
Show 1 replacement.

Student task: Replace all instances where we want a dot with calls to the dot function

Students each write method for dash. Go over as a class.

How could we use functions to improve this code even more?
-Letterbreak and space functions
-Functions for individual letters

Students complete challenges (each pair must complete at least 1) and submit on GC.

Updated code: https://www.tinkercad.com/things/eK9RF4rptA3-morse-code-dot-and-dash/editel (copied below)

ANNOTATED BASIC VERSION:

```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  //H
  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //Letter break
  delay(3000);

  //E
  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //Letter Break
  delay(3000);

  //Y
  //dash
```

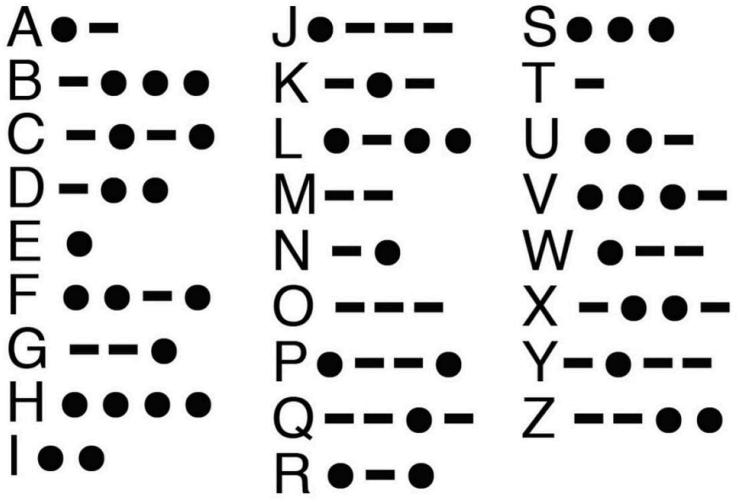
```
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dot
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dash
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //dash
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)

  //Long delay before the program repeats
  delay(10000);
}

UPDATED VERSION:
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{

  //H
  dot();
  dot();
  dot();
  dot();

  letterBreak();


  //E
  dot();

  letterBreak();
```

```
  //Y
  dash();
  dot();
  dash();
  dash();

  space();
}

void dot(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}

void dash(){
  digitalWrite(LED_BUILTIN, HIGH);
  delay(3000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}

void letterBreak(){
  delay(3000);
}

void space(){
  delay(10000);
}
```

Names: _____ and _____
*Submit one assignment per pair
Electrical Engineering

Morse Code Part 2: Arduino

Morse code uses a combination of shorter signals (called dots) and longer signals (called dashes) to symbolize characters. One dash lasts for approximately three times as long as one dot. A length of time roughly equal to one dash separates different letters, with a longer pause between different words.



Directions:

1. Build a circuit with a resistor and LED on a breadboard.
2. Connect a red jumper wire from pin 13 in the digital side to the breadboard.
3. Using a black jumper wire, connect the breadboard to the "GND" (ground) pin on the "Power" section of the arduino.
4. Use the blue cord to connect your microcontroller to the USB port of your computer.
5. Go to https://create.arduino.cc/ and log in
6. On the left-hand sidebar, click "Examples" → "Built In" → "01.Basics (6)" → "Blink"
7. Click the right arrow to upload the code to the arduino and verify that it is connected properly
8. Write down a message that is at least 2 words and has 6-15 letters.
9. Convert your message to Morse code.
10. Modify the code so that the LED blinks out your message in Morse code. Take a video and upload to Google Classroom.

1. My message (English)

2. My message (Morse code)

3. My code

4. Video of my Morse code message (you can embed the video, upload the video into the GC assignment, or upload it to Youtube and include a link here)

If you finish the required portion of the activity, share your message with another group

5. Optional: Message I received from other group (Morse code)

6. Optional: Message I received from other group (English)

7. Optional: Message other group sent, according to other group (English)

Challenges (must answer Q8 and complete at least one challenge):
1. Define a function for dot(), dash(), letterBreak(), and space().
2. Define a function for each letter used
3. Add a second LED and blink it in some useful way (such as when the message resets or whenever there is a vowel)
3. Define a morseCode("hello world") function that takes in an argument of a message and causes the LED to blink out the message in Morse code.
4. Add a buzzer that rings every time the LED is on. Program the buzzer to play a different note when there is a dot vs dash.
5. Program the serial monitor to ask the user for a message, then blink that message in Morse code.

8. What is the advantage of using functions in this program?

Final code: