

# CodeHS Lovelace Digital Information

# Helpful Information



# Lesson 1: Getting Started

## Encodings Everywhere

Start Screen

What You Need to Do

Give an example of encodings in your life.

```
4.1.4: Encodings Everywhere
```

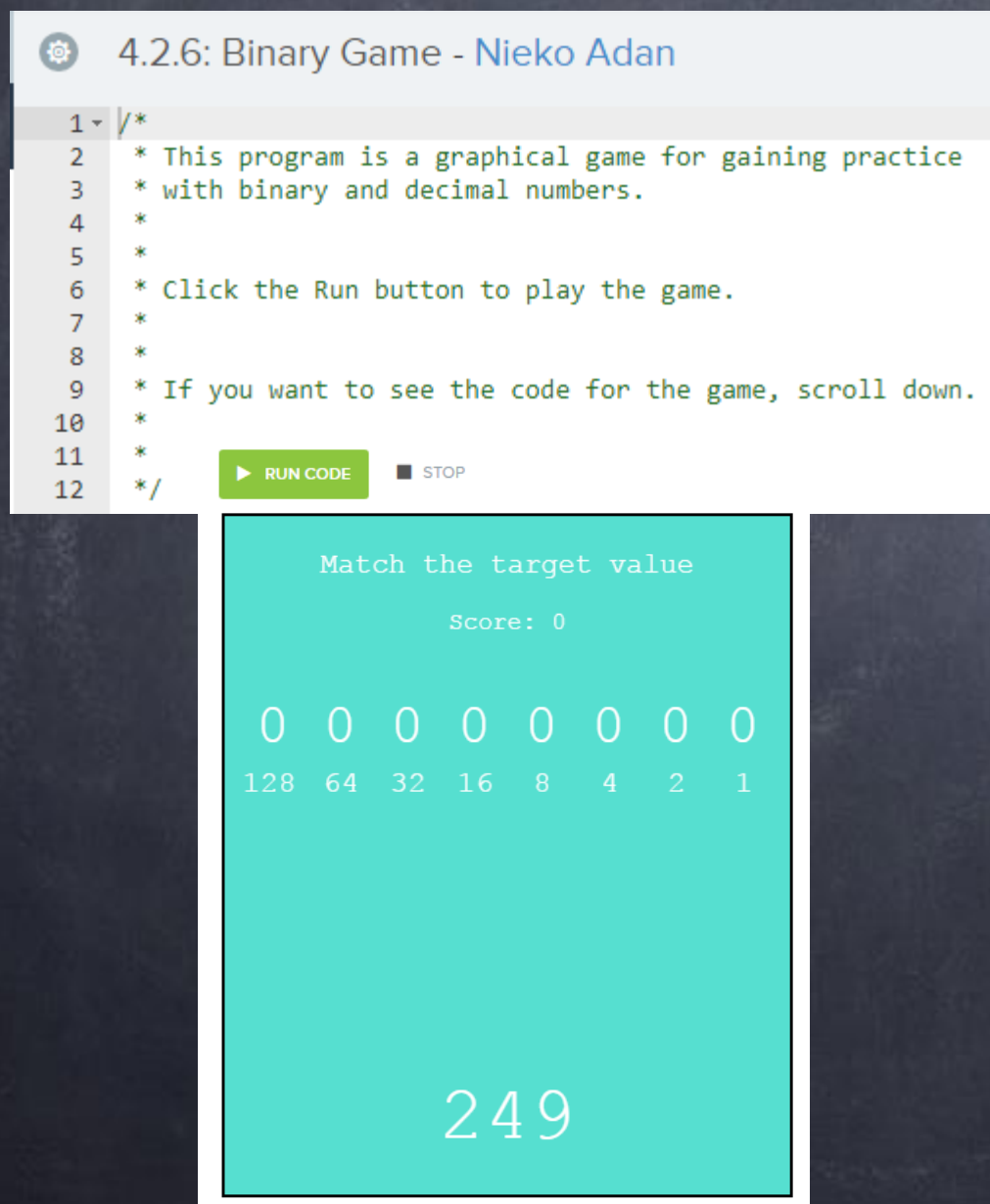
```
1 My basketball team encodes the plays that we run, so I can call out 4 or show
2 the number 4 on my fingers, and everyone on my team is able to interpret that
3 number as a play that they need to execute. The numbers encode full plays, and
4 since everyone on my team has learned what each number corresponds to, we can
5 use this encoding scheme in a live game. We are all following the same protocol.
```



# Lesson 2: Number Systems

## Binary Game

### Start Screen



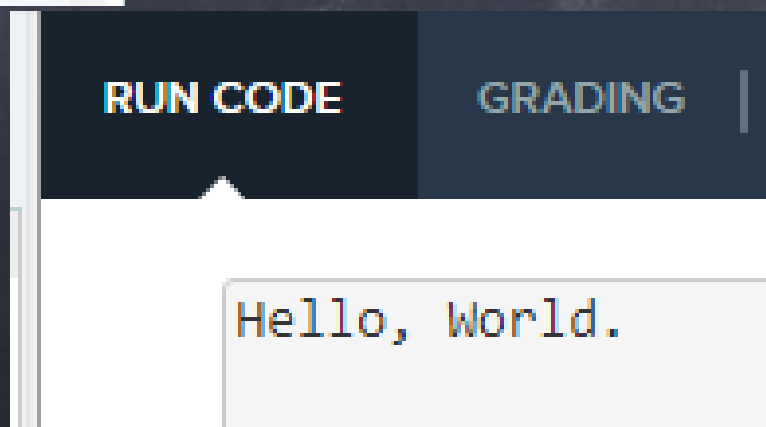
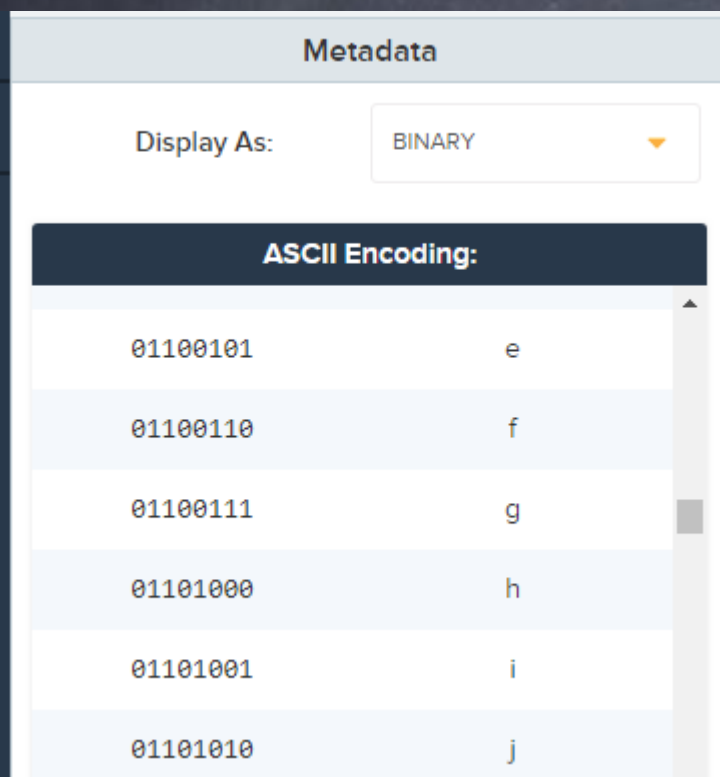
### What You Need to Do

- This exercise is a visual game to practice making numbers in binary.
- Clicking on each bit toggles the value from a 0 to a 1, or vice versa.
- The decimal target value is shown in the bottom of the window.
- Flip the bits of the binary number until you successfully create the binary value that represents the given decimal value.
- Try to get at least 500 points before moving on!

# Lesson 3: Encoding Text with Binary

## Hello World in Bits

### Start Screen



### What You Need to Do

Using the given ASCII Encoding Table for reference, type out the binary encoding for the string Hello, World.

Before you type out the bits, consider the following questions:

How many total bits do you think it will take to encode Hello, World.

How many bytes is that?

### Where is the ASCII Encoding Table?

On the left-hand side of the screen and you can scroll to look for the binary encodings of different letters



# Lesson 3: Encoding Text with Binary

## Create Your Own Encoding

### Start Screen

### What You Need to Do

In this exercise you'll [develop your own text encoding scheme](#) with a partner. If you and your partner agree on the same encoding scheme, then you will be able to transmit binary messages to each other, and only you will be able to correctly interpret them!

### Requirements

Your encoding scheme needs to be able to represent:

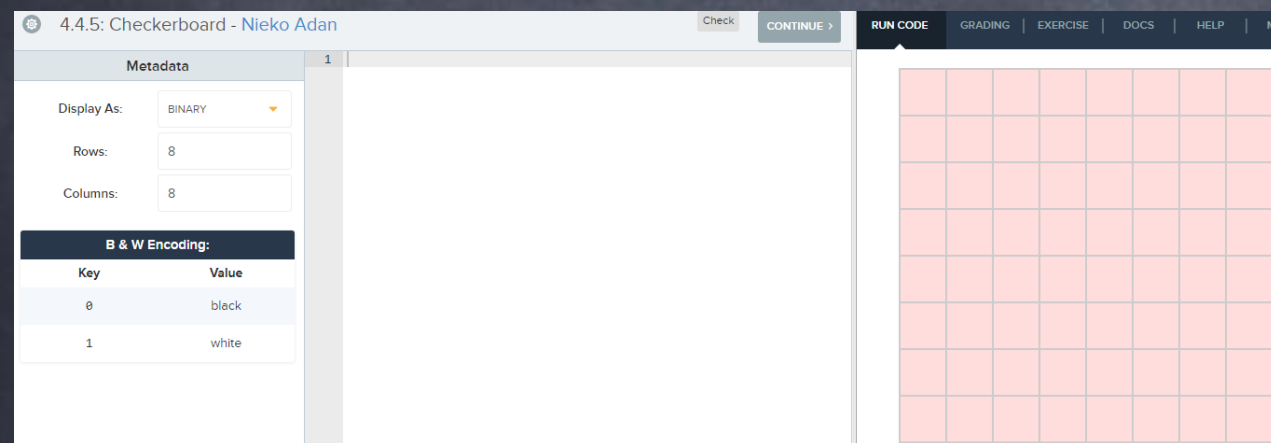
Every capital letter A-Z

The space character

# Lesson 4: Pixel Images

## Checkerboard

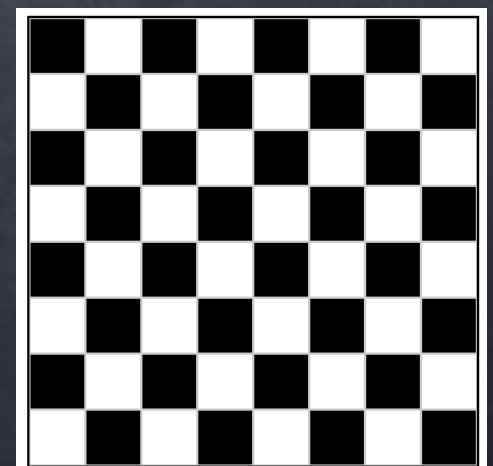
### Start Screen



### What You Need to Do

Using the pixel tool, type out the binary encoding for a black and white checkerboard.

The result should look like this:

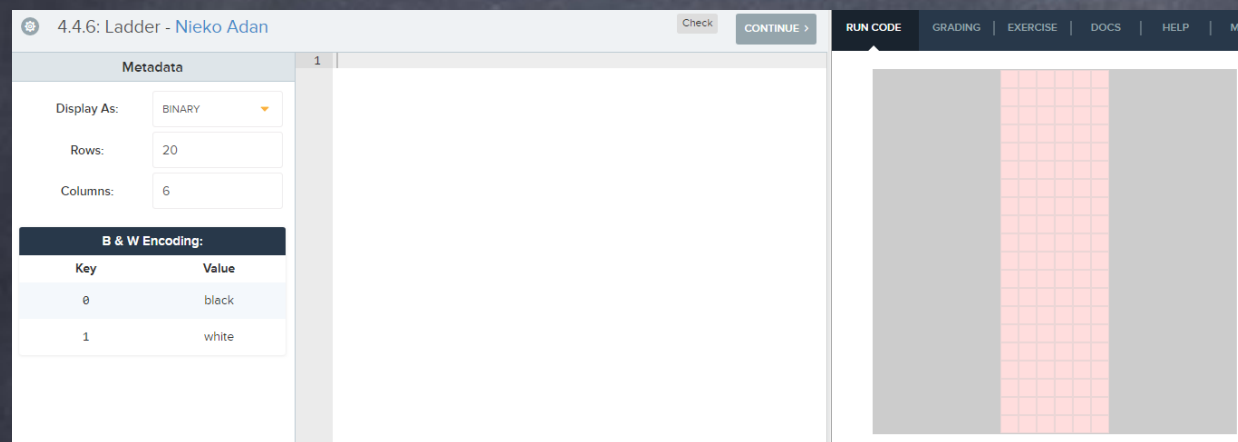




# Lesson 4: Pixel Images

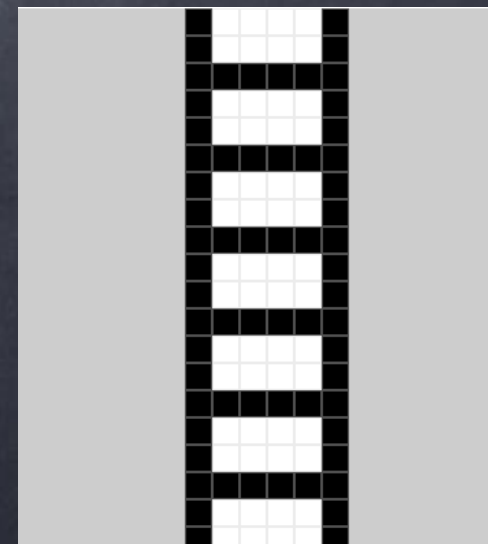
## Ladder

### Start Screen



### What You Need to Do

- In this exercise, create an image of a ladder that has a height of 20 and a width of 6.
- Students don't know where to start: Students can start typing binary digits (0 or 1) into the editor. The display will change on the right depending on the digit.
- The resulting image should look similar to this:



# Lesson 4: Pixel Images

## Create an Image

### Start Screen

4.4.7: Create an Image! - Nieko Adan

Check CONTINUE >

**Metadata**

Display As: BINARY

Rows: 8

Columns: 8

**B & W Encoding:**

Key	Value
0	black
1	white

1

RUN CODE GRADING EXERCISE DOCS HELP MC

### What You Need to Do

You have a blank canvas, create whatever image you'd like!

Play around with the width and height as well, not all images need to be square!

Have fun!



# Lesson 5: Hexadecimal Binary to Hex Game

**Start Screen**

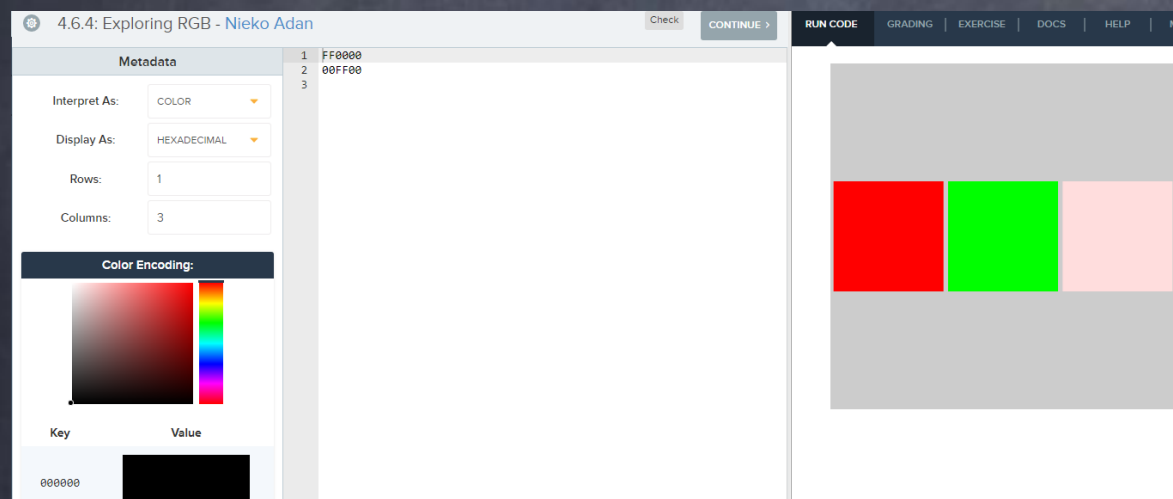
**What You Need to Do**

This exercise is a visual game to practice making hexadecimal numbers from bits. Clicking on each bit toggles the value from a 0 to a 1, or vice versa. The hexadecimal target value is shown in the bottom of the window. Flip the bits of the binary number until you successfully create the binary value that represents the given hexadecimal value. Try to get at least 100 points.

# Lesson 6: Pixel Colors

## Exploring RGT

### Start Screen



### What You Need to Do

Colors have a red, blue, and green component between 0 and 255. When representing a color, you choose a value for each of red, green, and blue.

For example, to display a red pixel, you would use this color:

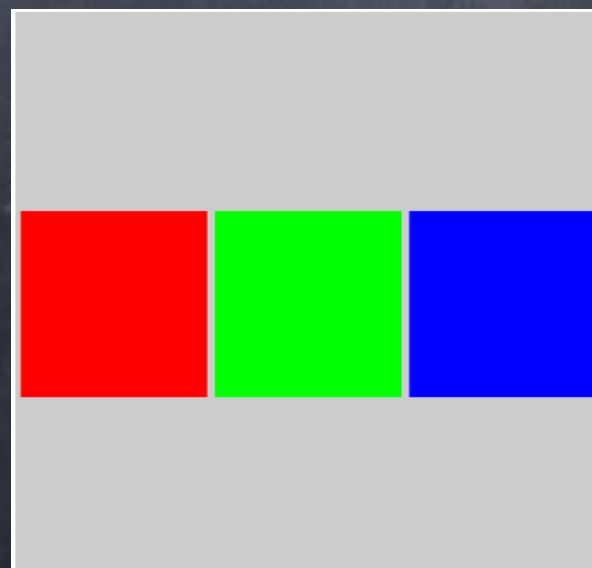
255, 0, 0

This is represented in hexadecimal as  
FF0000

Green is represented as 00FF00.

**Your task:**

Try adding a blue pixel to the screen



### The key isn't changing???

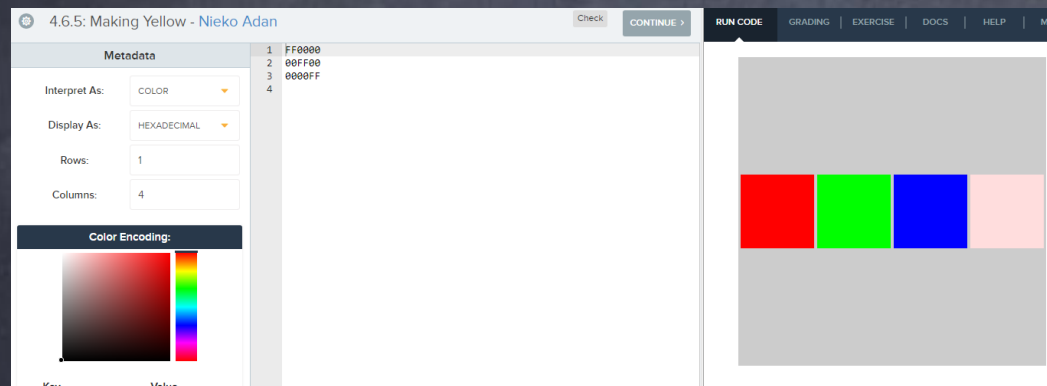
Besides dragging the bar on the side to change the colors in the square, you must also drag the little black dot on the square so that the key displays the RGB value.



# Lesson 6: Pixel Colors

## Making Yellow

### Start Screen



### What You Need to Do

So we've seen how to represent the 3 standard colors:

Red:

FF0000 (hexadecimal)

11111111 00000000 00000000 (binary)

Green:

00FF00 (hexadecimal)

00000000 11111111 00000000 (binary)

Blue:

0000FF (hexadecimal)

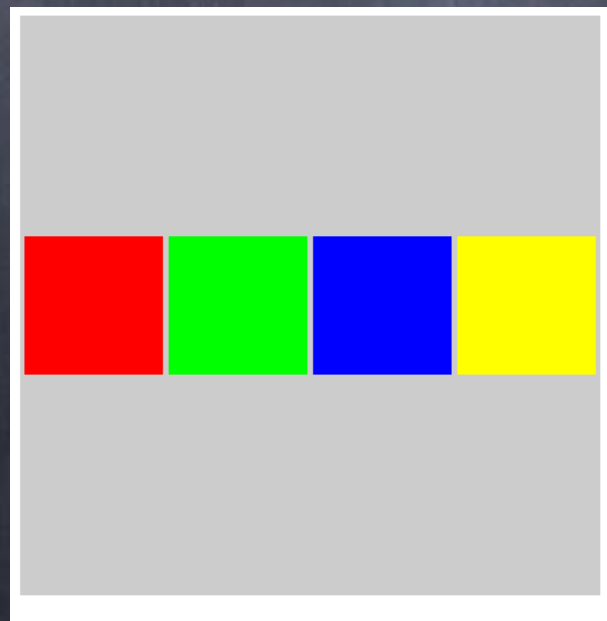
00000000 00000000 11111111 (binary)

**Your task:**

Your challenge in this problem is to make the color yellow.

Which color channels are turned "on" and which are turned "off" to represent yellow?

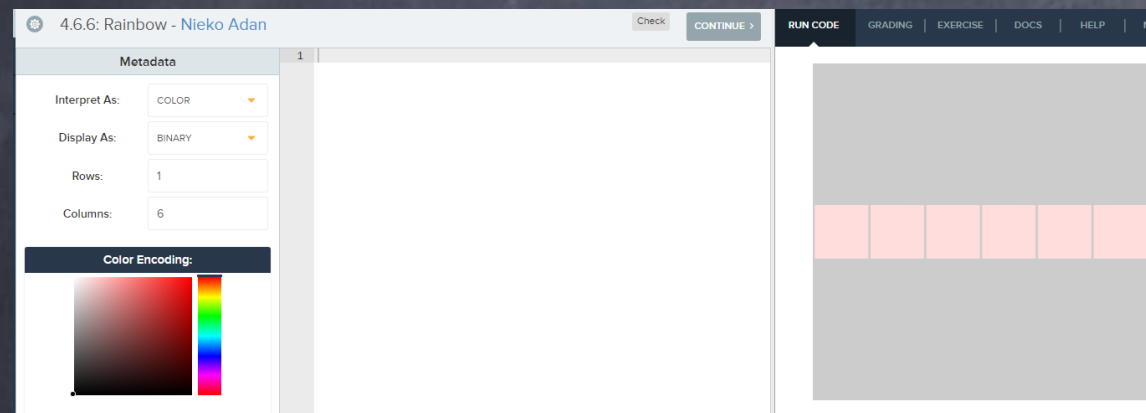
Why do you think that's the case?



# Lesson 6: Pixel Colors

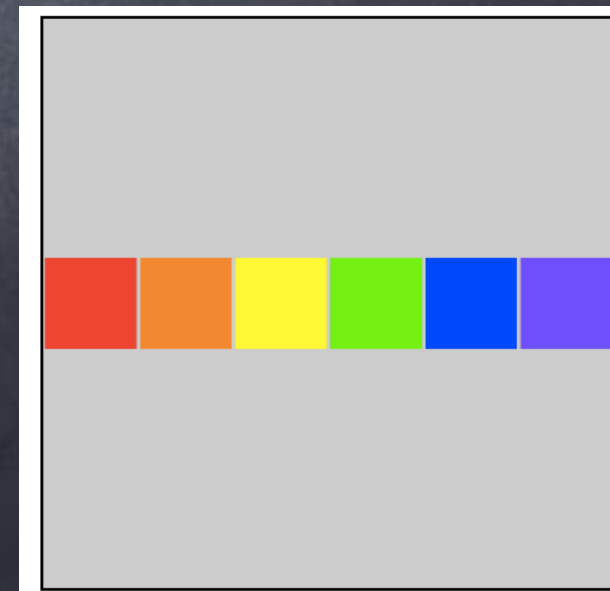
## Rainbow

### Start Screen



### What You Need to Do

The goal of this exercise is to make a pixel rainbow. See if you can determine the RGB values to make a rainbow strip like this:

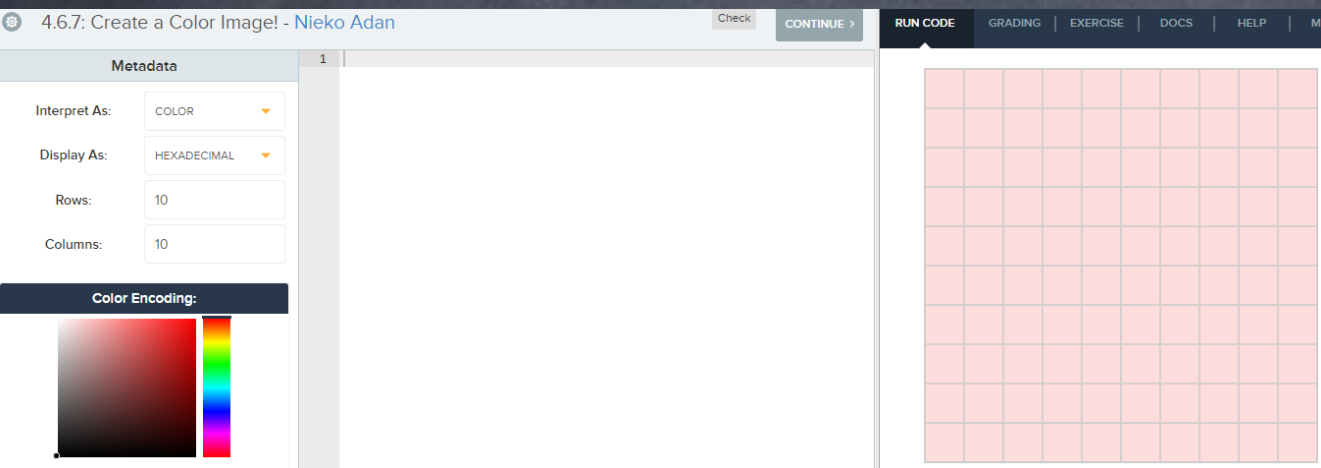




# Lesson 6: Pixel Colors

## Create a Color Image

### Start Screen



### What You Need to Do

Now that you can encode color images with binary and/or hexadecimal, type out the pixel encoding for any image you wish! You have a blank canvas to create whatever you'd like.  
You are now a true digital artist.