

NMD105-2026s Assignment 1

Drawing shapes

Submission guidelines

- [Instructions](#) for when to submit, where to submit, how to submit, and how assignments are graded.
- [Rubric](#) for this specific assignment.

Challenges

Challenge 1: Gridded initial with squares

This exercise is meant to teach you how to draw squares in P5js, but using rote numbers. So no variables or loops allowed!

Using a 400px by 400px canvas, create a grid of squares 80px by 80px wide. The squares should be a color of your choice (eg, 'yellow') and separated with a 1px white grid line. You'll end up with five columns and five rows.

Then change the color of some of the squares to draw the first letter of your name, eg "J" for "Jane." You may increase the number of squares in the grid if you want to draw a more accurate letter.

On the line in your code before each set of `rect()` calls that make your initials, add a comment like *// Special tile* so you can tell the foreground tiles from the background tiles.

 Sample result: <https://editor.p5js.org/jonippolito/full/lu4hE5cdt>

How you can use AI for this challenge

You must not use AI to write your code directly, but you may use a chatbot to:

- **Ask general questions**
 - Type your question manually into the chat, eg "What are the parameters that go into the P5js `rect()` function?"
- **Interpret an error**
 - Ask the chatbot "What does the following P5js error mean and how do I fix it?"
 - Then paste the error you see in the console (make sure you get the whole error) into the chat.
- **Debug your sketch**
 - Ask the chatbot a *specific* question about your code, eg "In the following P5js sketch, why is the red circle not lining up vertically with the orange circle?"

- Paste part or all of your code into the chat separated by triple backticks, eg:

```
```\n\n  // Draw eggs in a horizontal row.\n  fill("red");\n  ellipse(62, 36, 100);\n  fill("orange");\n  ellipse(157, 54, 100);\n  ```
```

💡 Adding descriptive comments will help the chatbot understand your intent.

⚠️ Chatbots can get things wrong, especially without enough context.

ℹ️ See the [course submission guidelines](#) for more information.

## Challenge 2: Gridded initial with circles (and a variable)

Duplicate your sketch of a gridded letter from Question 1, but this time replace the squares with circles using the *ellipse()* method.

Give your sketch a colored background. Notice that some of the circles are off the page.

You could fix this by tweaking each of the numbers by hand, but for this assignment make your life easier with variables.

In the first line of your code, define a variable with *let* to equal the number of pixels that each circle will need to be offset by to fit neatly inside the background. Then, inside the *ellipse* function calls that appear later in your code, add your new variable as necessary to the x and y values leftover from your code for Question 1. To do this, you'll type + and your variable name next to the appropriate parameters inside the parentheses.

(If you're clever, you can do this with Edit > Find and Replace. Otherwise it's easy enough to paste the offset repeatedly.)

👤 Sample result: <https://editor.p5js.org/jonippolito/full/CoVldYZPO>

## Challenge 3: Gridded initial with gradient

Duplicate your sketch from Question 1 (squares) or Question 2 (circles), and decide on a gradient—that is, a gradual transition from one color to another—that you'd like to apply to your initial. Choose one or more of the following features to vary:

- Hue (red, blue, yellow, or...?)
- Saturation (bold, subdued, or gray-ish)
- Lightness (light, medium, or dark)
- Alpha (1 for completely opaque, .5 for translucent, 0 for transparent)

Using the *hsla* color function, create one gradient from top to bottom for all the tiles that aren't part of your initial; for example, each new row of squares or circles could be a little darker than the row above it, or more blue, or more saturated.

You can make your life easier by using variables. Set one variable to however much you want the color to change in each row, and then add it to a parameter for each row. For example, you might create a variable equal to 10 or 20 at the top of your sketch (above the *setup()* function), and then add multiples of that variable to the *hue* parameter in your *hsla* functions.

When you're done, each row of tiles in the background should have a different color, progressing row by row from one shade to another. Do not change the tiles that represent your initial (eg, "J"), so we can still read that letter.

 Advanced option A: Change the foreground tiles representing your initial to have a different gradient from the background gradient.

 Advanced option B: Create a 2-dimensional gradient that starts in one corner and progresses to the opposite corner.

 Sample result if you chose squares: <https://editor.p5js.org/jonippolito/full/WBI9Lko5V>

 Sample result if you chose circles: <https://editor.p5js.org/jonippolito/full/tRKucJyNb>

 Sample result for Advanced option A: <https://editor.p5js.org/jonippolito/full/6KQyHNODA>

 Sample result for Advanced option B: <https://editor.p5js.org/jonippolito/full/JBLz26o1a>

## Challenge 4: Gridded initial with gradient using variables

Duplicate your gradient sketch from Question 3 and refactor (re-write) the color definitions using variables.

You'll want at least two variables:

- a "current" variable that stores the color attribute value.
- an "increment" variable that tells how much you want a color attribute to change in each row.

You can set the "current" variable to any number in the range of the color attribute. Remember that hue goes from 0-360, saturation from 0-100%, lightness from 0-100%, and alpha (opacity) from 0.0-1.0.

As examples of the increment variable, you might want the hue to increase by 20 each row. Or you might want the lightness to decrease, in which case you could set its increment variable to a negative number such as -10. (This can be the same variable you created for Question 3.)

Set the “current” variable to an initial value at the top of your code, then increment it before each row.

 Although the *hsla* method requires commas and percent signs, your variables are just numbers (integers). Fortunately, if you surround everything with backticks (`), you can inject variables or expressions into a string with \$ signs and ``, eg:`

```
fill(`hsla( ${ 300 + currentSaturation }, 50%, 50%, 1 )`);
```

 If you set your increment too high (or too negative), the color attribute may go out of bounds of the attribute range. For example, if you have one row that has lightness of 100 and another of lightness 120, they will both be white 😞.

So when you’re done rewriting your code, try adjusting the values of your “current” and “increment” variables until you like the resulting color gradients.

 Sample result if you chose squares: <https://editor.p5js.org/jonippolito/full/WBI9Lko5V>

 Sample result if you chose circles: <https://editor.p5js.org/jonippolito/full/tRKucJyNb>

 Sample result for Advanced option A: <https://editor.p5js.org/jonippolito/full/6KQyHNODA>

 Sample result for Advanced option B: <https://editor.p5js.org/jonippolito/full/JBLz26o1a>