Loops

Remapping

Nested Loops

**Textbook Readings:**
Chapter 4, Examples 4-5, 4-6, 4-7, 4-8, 4-9, 4-10, 4-11, 4-12, 4-13
Chapter 5, Examples 5-22, 5-23

**Coding Train Videos:**
4.1: while and for Loops  [https://youtu.be/cnRD9o6odjk](https://youtu.be/cnRD9o6odjk)
4.2: Nested Loops  [https://youtu.be/1c1_TMdf8b8](https://youtu.be/1c1_TMdf8b8)
Recap: Create a grayscale gradient

create a vertical grayscale gradient in a 100 x 300 canvas.

1. What do I want to repeat?
   - a line

2. What do I want to change each time?
   - the y position and the stroke shade

3. Where do they start, how do they change?
   - start y at 0, increment by 1
   - start gradient at 0, increment by 1

4. How long should it repeat?
   - as long as y is less than the canvas height

https://editor.p5js.org/sanghosuh/sketches/DvA4ytkk
Recap: Creating a rainbow

create a vertical color gradient in a 100 x 300 canvas.
Recap: Creating a rainbow

create a vertical color gradient in a 100 x 300 canvas.

• **Step 1.** change color mode from RGB to HSB

```javascript
colorMode(HSB);
```

- Why?

  Color type for RGB: 3 values (RGB)
  Color type for HSB: 1 value (Hue)
Recap: Creating a rainbow

create a vertical color gradient in a 100 x 300 canvas.

• **Step 1.** change color mode from RGB to HSB
  
  ```javascript
  colorMode(HSB);
  ```

• **Step 2.** change stroke
  
  ```javascript
  stroke(shade); => stroke(hue, 100, 100);
  ```


https://editor.p5js.org/sanghosuh/sketches/lvW7dzUS
But...

Range for hue is 0 ~ 360

- that is, there are 360 different colors
- but we are showing only 300 out of 360 colors

How can we show all 360 colors?
How can we show all 360 colors?

[ideas]

#1 change canvas size

https://editor.p5js.org/sanghosuh/sketches/clloKIKb
How can we show all 360 colors?

[ideas]

#2 change how much we increment hue value each time so that hue = 360 when y = 300

```javascript
// loop variables
y = 0;
hue = 0;

while (y <= height) {
    // shade
    stroke(hue, 100, 100);
    line(0, y, width, y);
    y++;
    hue += 1;
}
fill(255);
text("y: 0, hue: 0", 2, 10);
```

Starter code: [https://editor.p5js.org/sanghosuh/sketches/pZCVhapx](https://editor.p5js.org/sanghosuh/sketches/pZCVhapx)
[https://editor.p5js.org/sanghosuh/sketches/0chOCVNg](https://editor.p5js.org/sanghosuh/sketches/0chOCVNg)
Intuition: By how much should we increment hue?

- You have to reach a pole at 300 m
- You must jump only 300 times to get there
- How wide should each jump be?
Intuition: By how much should we increment hue?

- You have to reach a pole at 360 m
- You must jump only 300 times to get there
- How wide should each jump be?

\[ x = 1.2 \text{ m per jump} \]
How can we show all 360 colors?

[ideas]

#2 change how much we increment hue value each time so that hue = 360 when y = 300

Starter code: [https://editor.p5js.org/sanghosuh.sketches/pZCVhapx](https://editor.p5js.org/sanghosuh.sketches/pZCVhapx)
[https://editor.p5js.org/sanghosuh.sketches/0chOCVNg](https://editor.p5js.org/sanghosuh.sketches/0chOCVNg)
How can we show all 360 colors?

[ideas]

#2 change how much we increment hue value each time so that hue = 360 when y = 300

Starter code: https://editor.p5js.org/sanghosuh/sketches/pZCVhapx
https://editor.p5js.org/sanghosuh/sketches/0chOCVNg
How can we show all 360 colors?

[ideas]

#3 figure out how y value maps to hue value

\[
\begin{align*}
y &= 0, \text{ hue } &= 0 \\
y &= 76, \text{ hue } &= ? \\
y &= 173, \text{ hue } &= ? \\
y &= 300, \text{ hue } &= 360
\end{align*}
\]
How can we show all 360 colors?

[ideas]

#3 figure out how y value maps to hue value

// when y = 0,
stroke(0, 100, 100);
line(0, y, width, y);

// when y = 76,
stroke(?, 100, 100);
line(0, y, width, y);

// when y = 173,
stroke(?, 100, 100);
line(0, y, width, y);

. . .
How can we map $y$ to hue?

$y$ coordinate

line hue.
Intuition: mapping

- A small example to explain the idea
Intuition: mapping

- How would 2 map to the line on the right? (“what would 2 in the scale of 10 correspond to in the scale of 20? 17? 33?”)
Intuition: mapping

- what is the distance from the origin for 2?
- this distance is 2/10 of the scale
- should also be 2/10 of its scale from the origin
Intuition: mapping

Thus, ? is 4
Intuition: mapping

\[ h_1 = \left( \frac{y_1}{10} \right) \times 20 \]

\[ h_2 = \left( \frac{y_2}{10} \right) \times 20 \]

\[ h = \left( \frac{y}{10} \right) \times 20 \]

ratio
Applying this intuition

- let’s apply this intuition to map y to hue
Applying this intuition

- when \( y = 76 \), hue = ?
- when \( y = 173 \), hue = ?
Applying this intuition

\[ h_1 = \left( \frac{y_1}{300} \right) \times 360 \]

\[ h_2 = \left( \frac{y_2}{300} \right) \times 360 \]

\[ h = \left( \frac{y}{300} \right) \times 360 \]

\[ \text{hue} = \left( \frac{y}{\text{height}} \right) \times 360 \]

\[ \text{target} = \text{ratio} \times \text{targetRange} \]
How can we show all 360 colors?

[ideas]

#3 figure out how y value maps to hue value

target = ratio x targetRange

```javascript
// initialize loop variable
let y = 0;

// loop until we hit the bottom of the canvas
while (y < height) {
   // calculate the colour for the line to be drawn
   let hue = (y / height) * 360;
   stroke(hue, 100, 100);
   line(0, y, width, y);
   y++;
}
```

https://editor.p5js.org/sanghosuh.sketches/vrXDXH7v
Summary: How can we show all 360 colors?

[ideas]

#1 change canvas size

#2 change how much we increment hue value each time so that hue = 360 when y = 300

#3 figure out how y value maps to hue value

\[
\text{target} = \text{ratio} \times \text{targetRange}
\]
But what if...(1)

we want to draw rainbow with top and bottom margin in a 100 x 400 canvas?
Which means...

Our yStart and yStop are no longer 0 and height.
rainbow-margin

draw rainbow with top and bottom margin in a 100 x 400 canvas

https://editor.p5js.org/cs105/sketches/6JBffsBzM
But what if...(2)

We want to draw a color gradient from hue 100° to 200° between yStart and yStop?
Which means...

The source range and target range both changed.
How do we calculate this one?
The point is...

- Remembering how to calculate this & calculating it are all very cumbersome

- Should you memorize how to calculate these? => NO, just understand the idea

- However, you should know which function to use and how => That function we should use is...
\[ \text{value2} = \text{map(value1, start1, stop1, start2, stop2)} \]

- **value1**: the incoming value to be converted
- **start1**: lower bound of the value's **current** range
- **stop1**: upper bound of the value's **current** range
- **start2**: lower bound of the value's **target** range
- **stop2**: upper bound of the value's **target** range
remap variable using `map()`

```javascript
let hue = map(y, yStart, yStop, hueStart, hueStop);
```

https://editor.p5js.org/cs105/sketches/kWxRVPKz9
map() works for any remapping (case 1)

\[ \text{value2} = \text{map(} \text{value1, start1, stop1, start2, stop2) \} \]
map() works for any remapping (case 2)

\[ \text{value2} = \text{map}(\text{value1}, \text{start1}, \text{stop1}, \text{start2}, \text{stop2}) \]
map() works for any remapping (case 3)

\[
\text{value2} = \text{map(value1, start1, stop1, start2, stop2)}
\]
So the bottom line is...

- use `map()` for mapping values from one scale to another
Nested Loop

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```
Nested Loop

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```

Starter: [https://editor.p5js.org/sanghosuh/sketches/82RZDHIF][1]
[https://editor.p5js.org/sanghosuh/sketches/BcyipXq0][2]
How many times does this inner loop run?

```javascript
for (let y = 0; y < height; y += 10) {
  for (let x = 0; x < width; x += 10) {
    rect(x, y, 10, 10);
  }
}
```

canvas: 100 x 100
How many times does this **inner** loop run?

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```

canvas: 100 x 100
How many times does this **outer** loop run?

```javascript
for (let y = 0; y < height; y += 10) {
  for (let x = 0; x < width; x += 10) {
    rect(x, y, 10, 10);
  }
}
```

canvas: 100 x 100
How many times does this **outer** loop run?

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```

canvas: 100 x 100
How many times does nested loop run?

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```

This nested loop runs _____ times
How to understand how nested loop behaves

**Step 1.** Look at the **innermost** code block

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```
How to understand how nested loop behaves

**Step 2.** Check which variable is fixed vs changing in the inner loop

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```
How to understand how nested loop behaves

Step 3. Check the output of the innermost code block

```
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```

outer loop #1 (y = 0)
inner loop #1 ~ 10
rect(0, 0, 10, 10);  rect(70, 0, 10, 10);
rect(10, 0, 10, 10);  rect(80, 0, 10, 10);
rect(20, 0, 10, 10);  rect(90, 0, 10, 10);
rect(30, 0, 10, 10);  rect(100, 0, 10, 10);
rect(40, 0, 10, 10);
rect(50, 0, 10, 10);
rect(60, 0, 10, 10);
How to understand how nested loop behaves

**Step 4.** Move to the next iteration of the outer loop and check the output of the innermost code block

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```
How to understand how nested loop behaves

outer loop #1 (y = 0)
inner loop #1 ~ 10
rect(0, 0, 10, 10);
rect(10, 0, 10, 10);
rect(20, 0, 10, 10);
rect(30, 0, 10, 10);
rect(40, 0, 10, 10);
rect(50, 0, 10, 10);
rect(60, 0, 10, 10);
rect(70, 0, 10, 10);
rect(80, 0, 10, 10);
rect(90, 0, 10, 10);
rect(100, 0, 10, 10);

outer loop #2 (y = 10)
inner loop #1 ~ 10
rect(0, 10, 10, 10);
rect(10, 10, 10, 10);
rect(20, 10, 10, 10);
rect(30, 10, 10, 10);
rect(40, 10, 10, 10);
rect(50, 10, 10, 10);
rect(60, 10, 10, 10);
rect(70, 10, 10, 10);
rect(80, 10, 10, 10);
rect(90, 10, 10, 10);
rect(100, 10, 10, 10);
How to understand how nested loop behaves

**Step 5.** Repeat until there is no more outer loop to run

```javascript
for (let y = 0; y < height; y += 10) {
    for (let x = 0; x < width; x += 10) {
        rect(x, y, 10, 10);
    }
}
```
How to understand how nested loop behaves

outer loop #1 (y = 0)
inner loop #1 ~ 10
rect(0, 0, 10, 10);
rect(10, 0, 10, 10);
rect(20, 0, 10, 10);
rect(30, 0, 10, 10);
rect(40, 0, 10, 10);
rect(50, 0, 10, 10);
rect(60, 0, 10, 10);
rect(70, 0, 10, 10);
rect(80, 0, 10, 10);
rect(90, 0, 10, 10);
rect(100, 0, 10, 10);

outer loop #2 (y = 10)
inner loop #1 ~ 10
rect(0, 10, 10, 10);
rect(10, 10, 10, 10);
rect(20, 10, 10, 10);
rect(30, 10, 10, 10);
rect(40, 10, 10, 10);
rect(50, 10, 10, 10);
rect(60, 10, 10, 10);
rect(70, 10, 10, 10);
rect(80, 10, 10, 10);
rect(90, 10, 10, 10);
rect(100, 10, 10, 10);

outer loop #9 (y = 80)
inner loop #1 ~ 10
rect(0, 80, 10, 10);
rect(10, 80, 10, 10);
rect(20, 80, 10, 10);
rect(30, 80, 10, 10);
rect(40, 80, 10, 10);
rect(50, 80, 10, 10);
rect(60, 80, 10, 10);
rect(70, 80, 10, 10);
rect(80, 80, 10, 10);
rect(90, 80, 10, 10);
rect(100, 80, 10, 10);

outer loop #10 (y = 90)
inner loop #1 ~ 10
rect(0, 90, 10, 10);
rect(10, 90, 10, 10);
rect(20, 90, 10, 10);
rect(30, 90, 10, 10);
rect(40, 90, 10, 10);
rect(50, 90, 10, 10);
rect(60, 90, 10, 10);
rect(70, 90, 10, 10);
rect(80, 90, 10, 10);
rect(90, 90, 10, 10);
rect(100, 90, 10, 10);
Nested Loop Clock Analogy

```javascript
for (let m = 0; m < 60; m++) {
    for (let s = 0; s < 60; s++) {
        print(s, m);
    }
}
```
Nested Loop Clock Analogy

```javascript
for (let h = 0; h < 24; h++) {
    for (let m = 0; m < 60; m++) {
        for (let s = 0; s < 60; s++) {
            print(s, m);
        }
    }
}
```
for (let y = 0; y <= height; y += 40) {

    // pick a hue for the row
    let hue = map(y, 0, width, 360, 0);
    fill(hue, 80, 80);

    // all cells in this row
    for (let x = 0; x < width; x += 20) {
        rect(x, y, 20, 40);
    }
}

https://editor.p5js.org/cs105/sketches/chDptlowO
for (let x = 0; x < width; x += 20) {

  // pick a hue for the column
  let hue = map(x, 0, width, 360, 0);
  fill(hue, 80, 80);

  // all cells in this column
  for (let y = 0; y <= height; y += 40) {
    rect(x, y, 20, 40);
  }
}

https://editor.p5js.org/cs105/sketches/yyohyZRac
let s = 25; // size of each cell
let a = 0; // count the cells

// column
let x = 0;
for (let i = 0; i < 4; i++) {
    // row
    let y = 0;
    for (let j = 0; j < 3; j++) {
        // draw text at position x, y
        text(a, x, y);
        a = a + 1;
        y = y + s;
    }
    x = x + s;
}
... let x = 0;
while (x < width) {
  if (mouseX >= x && mouseX <= x + size &&
      mouseY >= y && mouseY <= y + size) {
    fill("#FF0000"); // red
  } else {
    fill("#FFFFFF");
  }
  rect(x, y, size, size);
  x += size;
}
...
Creating an infinite loop:

Typical causes:

- forgetting to change the counter variable
- changing the counter variable incorrectly
- using wrong test

An infinite loop is a logical error.

Note: This can almost always be avoided by using for loops!

https://editor.p5js.org/sanghosuh/sketches/kZIjpYuk
infinite loop due to float precision

```javascript
let a = 0;

function setup() {
    frameRate(2);
}

function draw() {
    // infinite loop
    while (a !== 5) {
        // watch the console output!
        print(a);
        a += 0.1;
    }
}
```

https://editor.p5js.org/cs105/sketches/3owVNGdti
What does this code print to the console?

A. 0  
B. 1  
C. 2  
D. 3  
E. 4

function setup() { 
let i = 0; 
for (let y = 0; y < 2; y++) { 
i = i + 1; 
} 
print(i); 
}
What does this code print to the console?

A. 0
B. 2
C. 3
D. 5
E. 7

```javascript
function setup() {
  let i = 0;
  for (let y = 0; y < 2; y++) {
    i = i + 1;
  }
  for (let x = 0; x < 3; x++) {
    i = i + 1;
  }
  print(i);
}
```
What does this code output?

```javascript
function setup() {
    let i = 0;
    for (let y = 0; y < 2; y++) {
        for (let x = 0; x < 3; x++) {
            i++;
        }
    }
    print(i);
}
```