Loops

Variable Scope
Remapping

**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)
What could you see after 2 frames?

It never gets to frame 2 because it's an infinite loop.
From Lecture 10: Assignment Operator “Short Forms”

These all add 1 to x (they are all equivalent):

\[
x = x + 1;
\]

\[
x += 1;
\]

\[
x++;
\]

These both add 5 to x (they are both equivalent):

\[
x = x + 5;
\]

\[
x += 5;
\]

Other examples

// same as x = x + 10 * y;
\[
x += 10 * y;
\]

// same as x = x + random(-2, 2);
\[
x += random(-2, 2);
\]
From Lecture 10: More Assignment Operator “Short Forms”

- `x--;`  
  decrement operator

- `x -= 10; // subtract 10 from x`

- `x *= 10; // multiply x by 10`

- `x /= 10; // divide x by 10`
What is scope?

- A device used to see something in distance

CREDIT: https://upload.wikimedia.org/wikipedia/commons/2/2b/Edit_4x_rifle_scope.jpg
What is scope?

- A device used to see something in distance

CREDIT: https://upload.wikimedia.org/wikipedia/commons/2/2b/Edit_4x_rifle_scope.jpg
What is scope?

- An area in which something acts or operates
What is variable scope?

- An area in which variable operates
What is *variable* scope?

- An **area** in which variable operates
- This area can be either global or local
  - We say that “variables have **global** or **local** scope”
What is global scope?

- We have been declaring variables “outside” of functions
- These are called **global variables**
What is global scope?

- We have been declaring variables “outside” of functions
  - These are called **global variables**
- Global variables can be *used* anywhere in your program after they're declared

```javascript
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    
    
}
```
What is global scope?

- We have been declaring variables “outside” of functions
  - These are called **global variables**
- Global variables can be *used* anywhere in your program *after* they're declared
  - Since the area in which they operate is the entire program, we say that they have **global scope**
What is global scope?

- We have been declaring variables “outside” of functions
  - These are called **global variables**
- Global variables can be *used* anywhere in your program after they're declared
  - Since the area in which they operate is the entire program, we say that they have **global scope**
- Built-in Processing variables *are* global
  - e.g. width, frameCount, mouseX, key, mouseIsPressed

```javascript
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  .
  .
  .
```
What is local scope?

- We can also declare variables “inside” functions
  - These are called **local variables**

```javascript
function setup() {
  let w = 100;
  let y = w;
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {

}
```
What is local scope?

- We can also declare variables “inside” functions
  - These are called **local variables**

- Local variables can be *used* anywhere in *that* code block after they're declared

```javascript
function setup() {
  let w = 100;
  let y = w;
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  .
  .
  .
```

**local variables** `w, y` can be used anywhere in this code block
What is local scope?

- We can also declare variables “inside” functions
  - These are called **local variables**
- Local variables can be *used* anywhere in *that* code block *after* they're declared
  - Recall, a code block is anything in between `{ and `}
  - loops have code blocks
  - conditionals have code blocks
  - functions have code blocks
What is local scope?

- We can also declare variables “inside” functions
  - These are called **local variables**

- Local variables can be *used* anywhere in *that* code block *after* they're declared

```javascript
function setup() {
  let w = 100;
  let y = w;

  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  print(w);
  print(y);
}
```

Can **local variables w, y** be used in the draw() function?
What is local scope?

- We can also declare variables “inside” functions
- These are called **local variables**
- Local variables can be *used* anywhere in *that* code block *after* they're declared and **ONLY in that block**

```javascript
function setup() {
  let w = 100;
  let y = w;
  let h = w * 3;
  createCanvas(w, h);
}
```

```javascript
function draw() {
  // local variables w, y cannot be used in the draw() function
  .
  .
  .
}
```
What is local scope?

- We can also declare variables “inside” functions
  - These are called **local variables**
- Local variables can be *used* anywhere in *that* code block *after* they're declared and **ONLY in that block**
- Local variables *cease to exist after* the code block is exited
  - they have **local scope**

```javascript
function setup() {
  let w = 100;
  let y = w;
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  .
  .
  .
}
```
function draw() {
  background(200);

  // x is a local variable only visible to draw()!
  let x = 0;

  while (x < width) {
    line(x, 10, x, height - 10);
    x += spacing;
  }
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let n = int(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let $w = 100$;
let $y = w$;

function setup() {
  let $h = w \times 3$;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, $y$, w, $y$);

  let $n = \text{int}(\text{random}(0, 500))$;
  for(let $i = 0; i < n; i++$) {
    let $px = \text{random}(0, w)$;
    let $py = \text{random}(0, y)$;
    point($px$, $py$);
  }
  $y -= 1$;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);
    let n = int(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let n = int(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);

  let n = int(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
  }
  y--;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);
    let n = int(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let n = int(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

```javascript
function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);
  let n = int(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
    y--;
  }
}
```
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);

  let n = let(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
  }
  y--;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let n = let(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);
  let n = let(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
  }
  y--;
}
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);
  let n = let(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
  }
  y--;
}
let w = 100;
let y = w;

function setup() {
  let h = w * 3;
  createCanvas(w, h);
}

function draw() {
  background(240);
  line(0, y, w, y);

  let n = let(random(0, 500));
  for(let i = 0; i < n; i++) {
    let px = random(0, w);
    let py = random(0, y);
    point(px, py);
  }
  y--;
}
let \( w = 100 \);
let \( y = w \);

function setup() {
    let \( h = w \times 3 \);
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let \( n = \) let(random(0, 500));
    for(let \( i = 0 \); i < n; i++) {
        let \( px = \) random(0, w);
        let \( py = \) random(0, y);
        point(px, py);
    }
    y--;
}
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);

    let n = let(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }

    y--;
}

Memory and Variables
Memory and Variables

```javascript
let w = 100;
let y = w;

function setup() {
    let h = w * 3;
    createCanvas(w, h);
}

function draw() {
    background(240);
    line(0, y, w, y);
    let n = let(random(0, 500));
    for(let i = 0; i < n; i++) {
        let px = random(0, w);
        let py = random(0, y);
        point(px, py);
    }
    y--;
}
```
Declare global or local variable?

**local**
(use it then throw it out)

**global**
(remember it)

---

**Bill:**
555-1150

**IOU:**
$2 \$5
$15 \$25

- Bill
When to use local variables

- When you don't need to "remember" the value after the code block completes
  - a loop variable
  - an intermediate calculation
  - to re-use a common value in a function call
function draw() {
    
    // calculate start and end angles
    let start = radians(dir - mouthOpening/2);
    let end = radians(dir + mouthOpening/2);

    // use those angles to draw an arc
    arc(width/2, height/2,
        mouseX, mouseX,
        start, end);

    ...
function draw() {

    ...

    } else if (shape == 3) {
        // half size
        let hs = size / 2;
        triangle(mouseX - hs, mouseY + hs,
                mouseX, mouseY - hs,
                mouseX + hs, mouseY + hs);

    }

    ...

}
Potential errors when using local variables

- “out of scope” syntax error
- logic error
function draw() {
    ...

    } else if (shape == 3) {

        triangle(mouseX - hs, mouseY + hs,
                mouseX, mouseY - hs,
                mouseX + hs, mouseY + hs);

        // half size
        let hs = size / 2;
    }
    ...

    }

    }  // end of function draw

當地的變數 hs 在使用後才被定義。

找不到變數 "hs" 這個錯誤。
function setup() {
    // local variable
    // only accessible to setup()
    let speed = 2;
}

function draw() {
    ellipse(x, 50, 20, 20);
    // this will cause an error: there is no
    // variable called 'speed' accessible to draw()
    x += speed;
}
function draw() {
    let x = 0;
    ellipse(x, 50, 20, 20);
    x += 2;
}

x is declared as a local variable, it will have a value of 0 every frame.

the local variable x will be incremented, but then x will be thrown away as soon as draw finishes.
(logic errors with local variables)

```javascript
let x = 0;

function draw() {
    let x = 0;
    ellipse(x, 50, 20, 20);
    x += 2;
}
```

- The local variable `x` will be incremented, but then `x` will be thrown away as soon as `draw` finishes.

- `x` is declared as a local variable and it will "shadow" (hide) the global variable `x`.

- The local variable `x` will be assigned a value of 0 each frame.
gradient

create a vertical grayscale gradient

eexample of varying more than one variable

1. What do I want to repeat?
2. What do I want to change each time?
3. Where do they start, how do they change?
4. How long should it repeat?

https://editor.p5js.org/cs105/sketches/U_fvzbRqx
Changing Multiple “Things” in One Loop

Make a rainbow.

1. What do I want to repeat?
2. What do I want to change each time?
3. Where do they start, how do they change?
4. How long should it repeat?

https://editor.p5js.org/sanghosuh/sketches/DvA4ytkk
Recap: p5 Colour Tutorial (from Lecture 04 Attributes)

- https://p5js.org/learn/color.html
  - Note the section on “Custom Color Ranges” and changing colorModel to HSB

Color

In the digital world, when we want to talk about a color, precision is required. Saying "Hey, can you make that circle bluish-green?" will not do. Color, rather, is defined as a range of numbers. Let’s start with the simplest case: black & white or grayscale. 0 means black, 255 means white. In between, every other number—50, 87, 162, 209, and so on—is a shade of gray ranging from black to white.
Recap: HSB (Lecture 04 Attributes)

- **Hue**
  - color type

- **Saturation**
  - vibrancy of color

- **Brightness**
  - brightness of color

\[H \quad S \quad B\]
(Hue, Saturation, Brightness)
Recap: p5 Colour Tutorial (from Lecture 04 Attributes)

// switch color mode to HSB

colorMode(HSB);

colorMode(HSB, 100, 100);

Changing Multiple “Things” in One Loop

Make a rainbow in a 100 by 400 canvas.
HINT: Use colorMode to switch to HSB colour model.

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

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

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

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

Make a rainbow in a 100 by 400 canvas.

HINT: Use colorMode to switch to HSB colour model.

ideas:
1. [HACK] createCanvas(360, 100);

2. [GOOD] start hue at 0,
   increment so hue = 360 when y = height

3. [BEST] express hue as a “function of y”

Starter: https://editor.p5js.org/cs105/sketches/DB5GxfifW

https://editor.p5js.org/cs105/sketches/xgRuIisdFP
Remapping Height to Hue

$y$ coordinate

line hue.

height

hue

360°
rainbow-margin

draw rainbow with top and bottom margin

https://editor.p5js.org/cs105/sketches/6JBffsBzM
Remapping with Height to Hue with Margin

\[ y_{\text{start}} = m \]
\[ y_{\text{stop}} = \text{height} - m \]
Remapping Different Scales

- draw a gradient from hue 100° to 200°
value2 = 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
rainbow-map

remap variable using map()

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

https://editor.p5js.org/cs105/sketches/kWxRVPKz9
What does this loop output?

```
let a = 2;
function setup() {
    for (let i = 0; i < a; i++) {
        print("duck");
    }
    print("goose");
}
```

A

B

C

D

- duck
- goose
- duck
- goose
- duck
- goose
- duck
- goose
What does this loop output?

```javascript
let a = 2;
function setup() {
    let a = 3;
    for (let i = 0; i < a; i++) {
        print("duck");
    }
    print("goose");
}
```

```
duck
duck
duck
goose
duck
goose
duck
goose
duck
goose
duck
goose
duck
goose
```
Announcement

- Survey FAQ
  https://docs.google.com/document/d/1nWs0RuQ2hZblR5mhVja35TaqETQo_B9uhNAW53n1wcg/edit?usp=sharing
- I will hold office hours on Wednesdays after the class as well, but you **must** sign up