today

Due: Ex 2
Loops
Introduce Ex 3: eQuilt
Student Presentations
Reading: Ch 6

Wednesday, Jan 27

Due: Paper sketches for Ex 3
revisit: Change the World
more on counting + code: eQuilt
Student Presentations

conditions

Make sure to format your code!

```
if (Tuesday) {
  eat tuna;
```

```
if (Tuesday) {
  eat tuna;
} else {
  eat tofu;
}
```

```
if (Tuesday) {
  eat tuna;
} else if (Thursday) {
  eat turkey;
}
```

```
if (Tuesday) {
  eat tuna;
} else if (Thursday) {
  eat turkey;
} else {
  eat tofu;
}
```

conditions

What's wrong with this?

```
if (Tuesday) {
  eat tuna;
} else {
  eat tofu;
} else if (Thursday) {
  eat turkey;
}
```

conditions

What’s wrong with this?

```
if (Tuesday) {
  eat tuna;
} else {
  eat tofu;
} else if (Thursday) {
  eat turkey;
}
```

logic error: Thursday is a part of the else statement, then do I eat tofu or turkey? According to order of operation, else if is ignored.

syntax error: **`else`** has to come at the end, if Thursday is an exception.

Loops

Loops are like the industrial revolution!

Repetitive tasks are done by the machine.

two types of loops

while loop

for loop

while loop

The idea of taking a single concept and repeat it many many times.

A while loop controls a sequence of repetitions. Ok, that’s beautiful, because it saves so much time!

However, loops MUST have an exist condition, or it will lock out! STUCK AND NEVER GET OUT!

compare if to while

In an if statement, if the evaluation inside the () is true, execute the statement **ONCE**; if the evaluation is false, don't execute the statement.

```
if ( boolean expression ){
   statement;
}
```

In a while statement, as long as the evaluation is true, the statement is executed **INFINITE TIMES**.

syntax:

```
while ( boolean expression ){
   statement;
}
```

The code will run, while the condition inside the () is true. Once the condition is no longer met, it jumps out of the while loop.

using if

```
float x=0;

void setup() {
  size (600, 400);
}

void draw() {
  background (0);
  fill(255);
  noStroke();

  if (x<width) {
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

using while

```
float x=0;

void setup() {
  size (600, 400);
}

void draw() {
  background (0);
  fill(255);
  noStroke();

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

check out - sketch_6_0_while_loop.pde - //REVIEW EXAMPLE

when does it come out of the while loop?

```
float x=0;

void setup() {
  size (600, 400);
}

void draw() {
  background (0);
  fill(255);
  noStroke();

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

```
float x=0;

void setup() {
  size (600, 400);
  background (0);
}

void draw() {
  ~~background (0);~~
  fill(255);
  noStroke();

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

check out - sketch_6_0_while_loop.pde - //EXAMPLE 1

how many circles did it produce?

```
float x=0;

void setup() {
  size (600, 400);
  background (0);
}

void draw() {
 fill(255);
  noStroke();

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+1;     //600 circle
  }
}
```

```
float x=0;

void setup() {
  size (600, 400);
  background (0);
}

void draw() {
 fill(255);
  noStroke();

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+100; //6 circles, where?
  }
}
```

check out - sketch_6_0_while_loop.pde

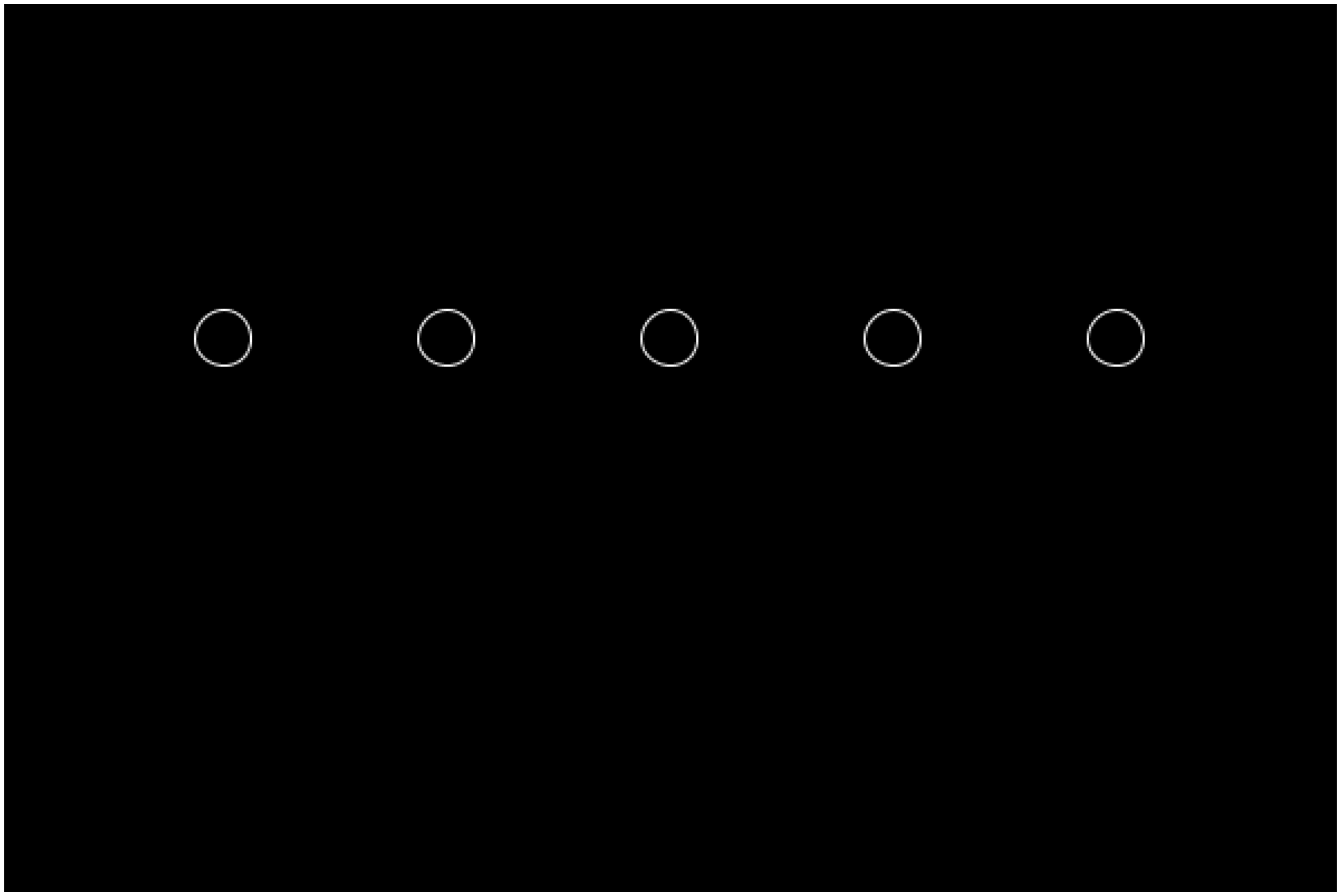
use println(x); to help you debug

```
float x=0;

void setup() {
  size (600, 400);
  background (0);
}

void draw() {
  fill(255);
  noStroke();
  println(x);

  while (x<width {
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

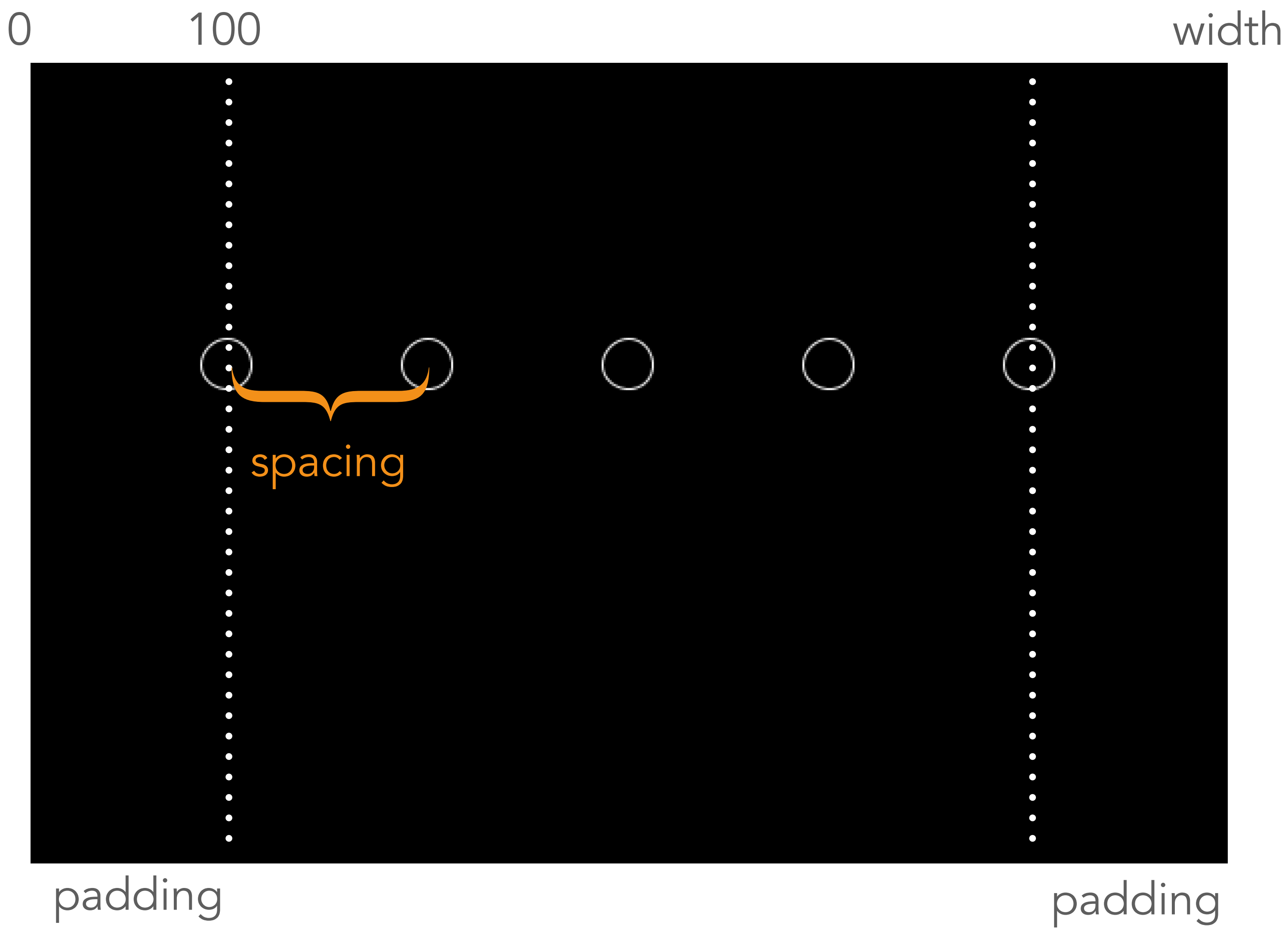
```
float x=0;

void setup() {
  size (600, 400);
  background (0);
}

void draw() {
   fill(255);
   noStroke();

  while (x<width {
    println(x);
    ellipse (x, 100, 25, 25);
    x=x+1;
  }
}
```

check out - sketch_6_0_while_loop.pde - //EXAMPLE 1

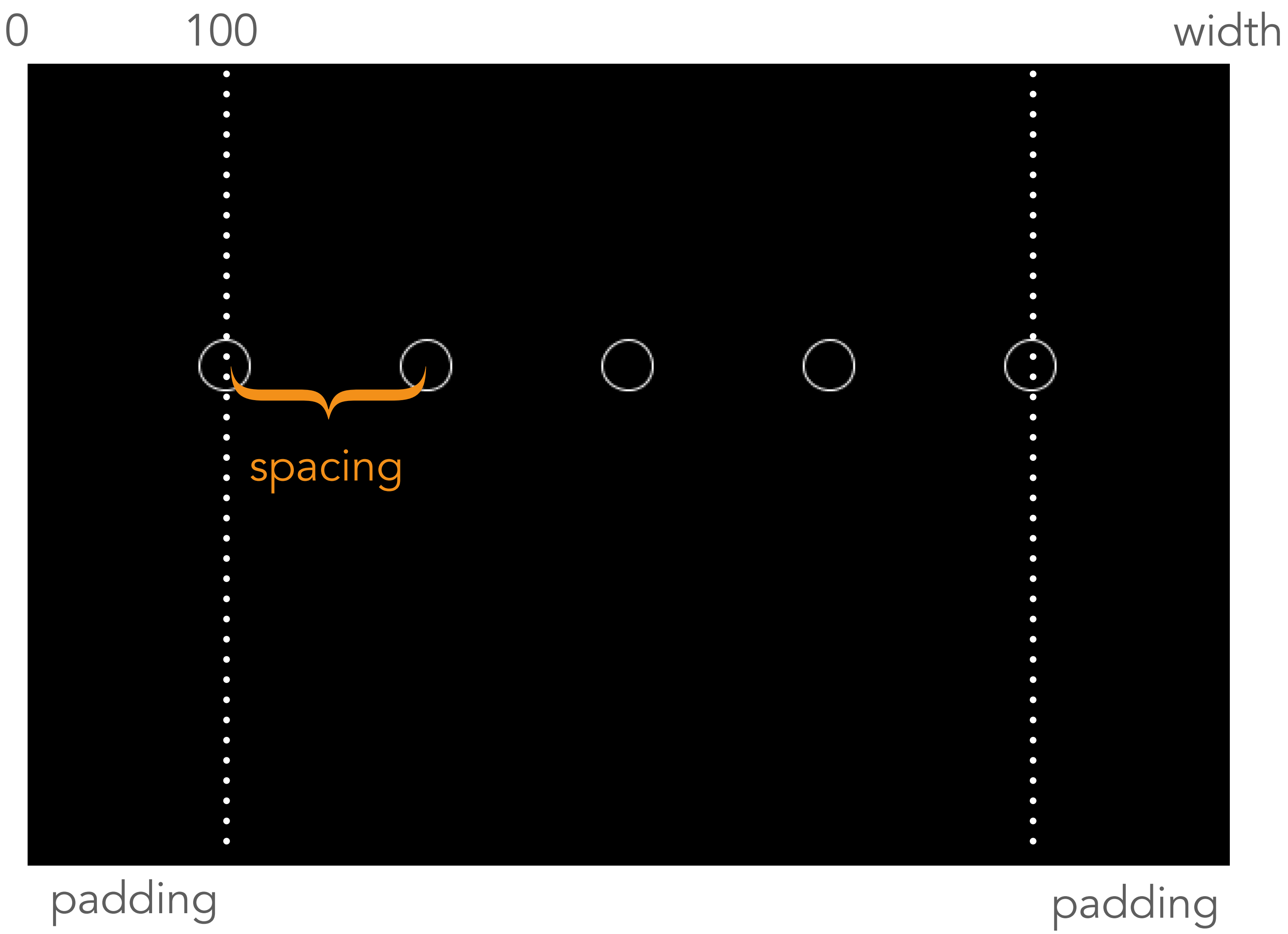
//EXAMPLE 2:
How to symmetrically fit x amount of evenly spaced circles on a canvas?
What is the spacing between the circles?
x+=spacing;



//EXAMPLE 2:
Assuming a particular amount of padding on the left and right, the math is as follows: spacing = (width - 2*padding) / (number of circles - 1)



math is as follows: (width - 2*padding) / (number of circles - 1)
(600-2*100)/(5-1)=100
code is: x+=100;



check out - sketch_6_0_while_loop.pde - //EXAMPLE 2

loop dangers!

Loops MUST have an exist condition, or it will lock out!
STUCK AND NEVER GET OUT!

```
/*DANGER!! LOOP STUCK, even though all syntax is correct!
 x will forever go to the left and never get out of the loop!
 Loops must have an exist plan!
 */

x=0;

while (x<width){
x=x-1;
ellipse(x, 100, 20, 20);
}
```

two loops

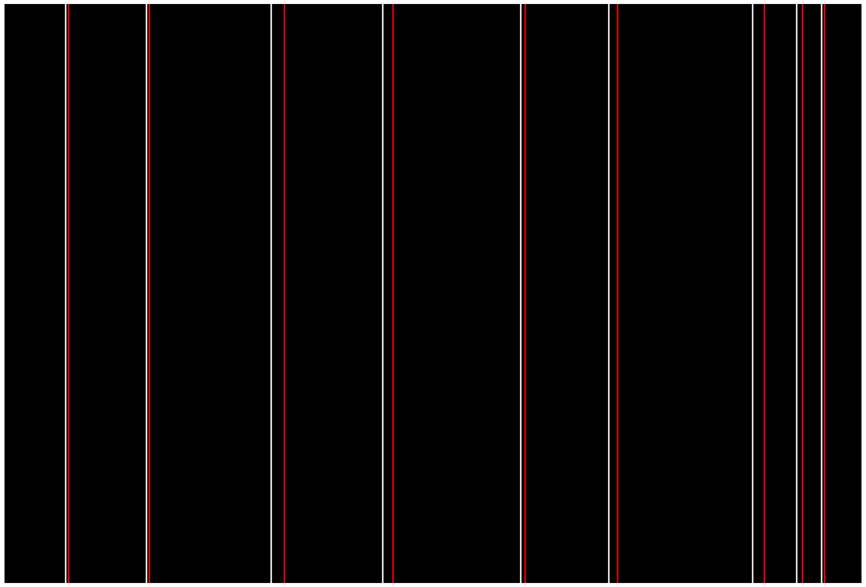
check out - sketch_6_1_two_loop_grid.pde - //EXAMPLE 1, 2

```
//EXAMPLE 1: DRAW VERTICAL LINES.

float x=50;

void setup() {
  size(600, 400);
  background (0);
}

void draw() {
  while (x<width) {
    stroke(255);
    line (x, 0, x, height);
    x=x+50;
  }
}
```

```
//EXAMPLE 2: TWO LOOPS - X AND Y. DRAW A GRID.
float x=50;
float y=50;

void setup() {
  size(600, 400);
  background (0);
}

void draw() {
  while (x<width) {
    stroke(255);
    line (x, 0, x, height);
    x=x+50;
  }
  while (y<height) {
    stroke(255);
    line (0, y, width, y );
    y=y+50;
  }
}
```

while loops can draw infinitely!

check out - sketch_6_1_two_loop_grid.pde - //EXAMPLE 3



```
//EXAMPLE 3: USE LOOPS TO CREATE ANIMATION.
float x=0;

void setup() {
  size(600, 400);
}

void draw() {
  background (0);

  x=0;
  while (x<width && x>=0) {
    stroke(255);
    line (x, 0, x, height);
    x+=random(0, 10);

    stroke(255, 0, 0); //red line.
    line (x, 0, x, height);
    x+= random(10, 100);
  }
  println(x);
}
```

for loop

When you don't need it to be infinite and forever. Great for controlling repetition. Can be identical to the while loop, except shorter.

```
for(init; test; update) {
  statements
}
```

semicolon!

the following blocks of codes are identical.

check out - sketch_6_2_for_loop.pde

code uses the while loop:

```
int x=0;
while (x<width) {
        line (x, 0, x, height);
        x=x+20;
 }
```

code uses the for loop:

```
for (int x=0; x<width; x=x+20) {
 line (x, 0, x, height);
}
```

Variable Scope AGAIN!

global variable: we declare at the very top.

for example, at the very top of the code, we write:
float x=0;

this means, it’s true for all blocks of codes.

Rule of thumb for Variable Scope: try to make variable as local in scope as possible.

global variables don’t work well with for loops! as you can imagine by its very set up, it can get very confused.

Here is a good example:

```
void setup() {
  size (600, 400);
}

void draw() {
  background (0);
  stroke(255);

  for (int x=0; x<width; x=x+20) {
    line(x, 0, x, height);
  }
}
```

x here is a local variable; declare it right when you need to use it.

check out - sketch_6_3_Variable_Scope.pde

for loop: how many times to loop?

There is a counting mechanism embedded in for loop.

The most common variables used for generic counting:

`i`, `j` and `k`

for loop

(desire 10 iterations) start at 0 and count up to 9

```
for (int i=0; i<10; i=i+1) //i++
```

(desire 10 iterations) start at 0 and count up to 100 by 10

```
for (int i=0; i<100; i=i+10) //i+=10
```

(desire 20 iterations) start at 100 and count down to 0 by 5

```
for (int i=100; i>=0; i=i-5) //i-=5
```

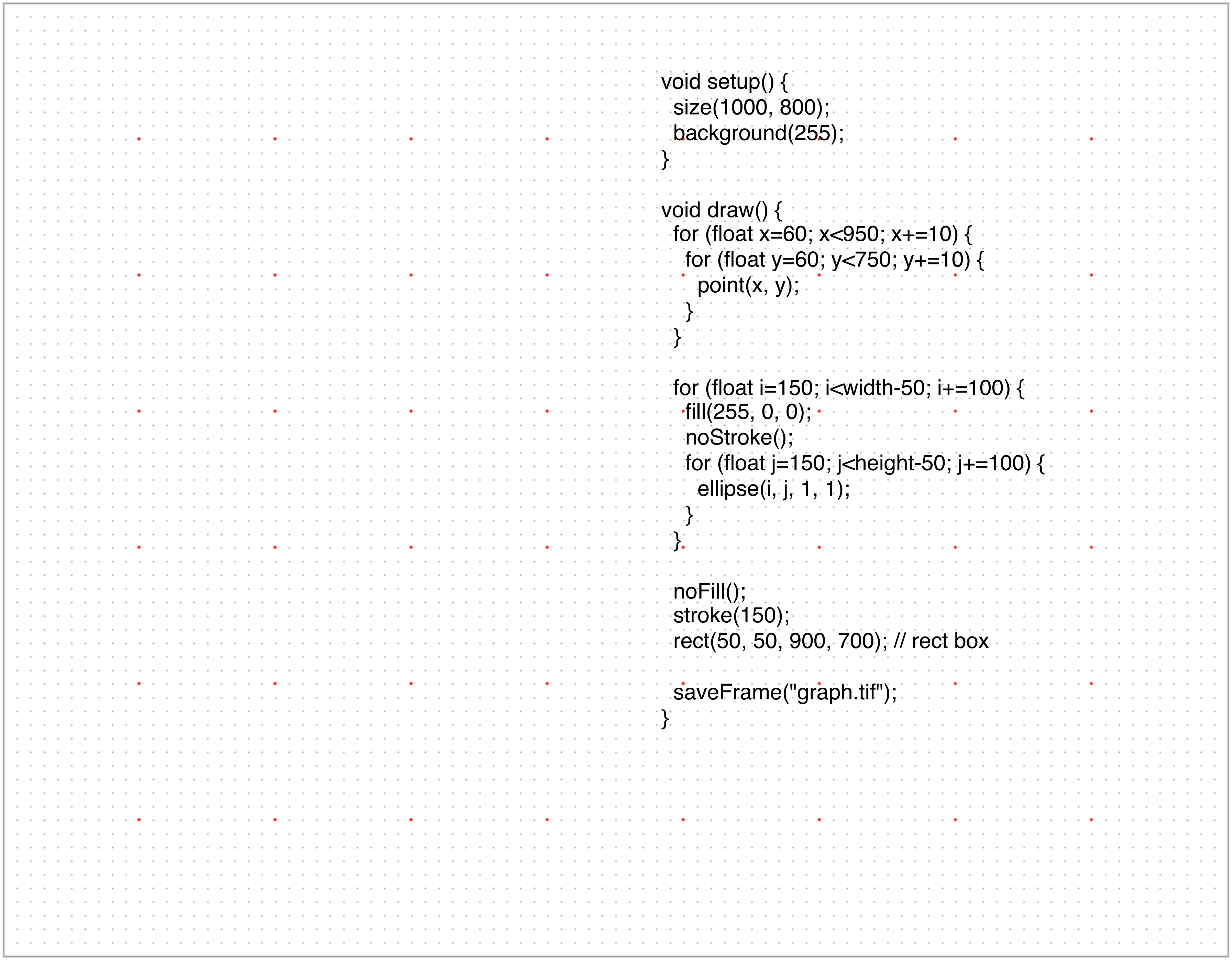
for loop: nested loops

Used in a situation where two or more variables needed to be evaluated.

DIY graph paper!

two sets of two for loops

check out - graph_paper_10_px.pde



```
void setup() {
  size(1000, 800);
  background(255);
}

void draw() {
  for (float x=60; x<950; x+=10) {
    for (float y=60; y<750; y+=10) {
      point(x, y);
    }
  }

  for (float i=150; i<width-50; i+=100) {
    fill(255, 0, 0);
    noStroke();
    for (float j=150; j<height-50; j+=100) {
      ellipse(i, j, 1, 1);
    }
  }

  noFill();
  stroke(150);
  rect(50, 50, 900, 700); // rect box

  saveFrame("graph.tif");
}
```

i'm feeling loopy and iffy!...which one to use?

Use a **`while`** loop if you don't know how many times you want the loop to execute (or based on an existing variable)

Use a **`for`** loop if you know how many times you want to repeat

Statement inside **`if`** only gets executed once

loop vs. draw

LOOP INSIDE OF DRAW

inner loop (while or for) produces one frame of an animation.

outer loop (draw):
example:
    step 1 - refresh background
    step 2 - draw a frame according to what the while loop produces.

———

This is how draw produces animation.

check out - sketch_6_4_loop_vs_draw.pde

```
void setup () {
  size (600, 400);
}

void draw() {
  background(0);
  stroke(255);

  int x=0;

  while (x<width) {          // everything the loop does all go inside one frame.
    line(x, 0, x, height);    // in this case, all these vertical lines will show up at once,
    x=x+20;                   // when the loop is done looping.
  }                           // what if i want the lines to show up one by one?
                             // this type of animation needs to be achieved in draw.
}
```

check out - sketch_6_4_loop_vs_draw.pde

```
//Example 2: draw a vertical line one by one to the right.

int x=50;

void setup() {
size(600, 400);
background (0);
stroke(255);
}

void draw() {
line(x, 0, x, height);
x=x+50;
}
//this one goes by too fast. how to control the speed of animation? See
sketch_6_4_loop_animation.
```

check out - sketch_6_4_loop_animation.pde

```
float frX=0; //flexible animation frame width, so that the while loop can meet exit condition.
int aSpeed=1; //control advancing speed

void setup () {
  size (600, 400);
}

void draw() {
  background(0);
  stroke(255);

  int x=0; //each time it draws, x needs to go back to 0 and start all over from left,
  //where frX continues to get bigger and bigger.
  while (x<frX) {
    line(x, 0, x, height);
    x=x+20;    //x can only be 0, 20, 40, 60, etc.
  }

  frX=frX+aSpeed;
}

/*
step 1 draw a background.
step 2 program goes inside the while loop, and draws a number of lines on a frame,
or it doesn't draw and comes out the loop and continue to increase frX until it hits the next increment of x.

The animations goes like this: frame 1 - one line, frame 2 - two lines, frame 3 - three lines, etc. refresh background in
between. each frame.
*/
```

today

Due: Ex 2

Loops

Introduce Ex 3: eQuilt

Student Presentations

Reading: Ch 6

Wednesday, Jan 27

Due: Paper sketches for Ex 3

revisit: Change the World

more on counting + code: eQuilt

Student Presentations