Brief lecture for classtime:

Power of repetition. The ability to do many operations is the key to computing. But as a programmer, need to be careful. If you give it correct instructions, it repeats those. If you give it incorrect instructions, then it repeatedly does the wrong thing.

Syntax: while loop and for loop

Strategy: Forget about the loop at first and think about what series of commands you want executed.

Example: lines1

A series of vertical lines

```
line(0, 0, 0, height);
line(10, 0, 10, height);
line(20, 0, 20, height);
line(30, 0, 30, height);
...
```

Example: lines2

Express this with a while loop

```
int x = 0;
while (x < width) {
  line(x, 0, x, height);
  x += 10;
}
```

Example: lines3

Same approach express with a for loop syntax

```
for (int x = 0; x < width; x += 10) {
  line(x, 0, x, height);
}
```

Example: lines4

Approach based on desired number of lines (rather than desired gap)

```
int numLines = 50;
float gap = width/numLines;

for (int k=0; k < numLines; k++) {
  line(k * gap, 0, k * gap, height);
}
```

Example: lines5

Can use loop variables as the basis for many properties

```
int numLines = 20;
float gap = width/numLines;

for (int k=0; k < numLines; k++) {
  strokeWeight(1+k);
  stroke(255 * k / numLines);
  line(k * gap, 0, k * gap, height);
}
```

Example: staircase1

```
size(500,500);

rect(0,   0, 100, 100);
rect(0, 100, 200, 100);
rect(0, 200, 300, 100);
rect(0, 300, 400, 100);
rect(0, 400, 500, 100);
```

Example: staircase2

```
  int numLevels = 5;
  float gap = height/numLevels;

  for (int k=0; k < numLevels; k++) {
    rect(0, gap * k, gap * (k+1), gap);
  }
```
