# CS 101 -- Section 5 -- *Nested for-loops*

We've done quite a lot of work with for-loops the past few weeks. We've seen them in class, you had section problems on them, and you have been using them in the recent homework assignments.

Today's in section, we are going to talk about **nested for-loops**. "Nesting" for loops just means we are putting one for-loop inside of the other. It is often useful to "nest" for loops. For example, whenever you want to create a Processing programs that draws things in a two-dimensional grid pattern, for-loops are great. Let's practice...

(1)   Open up processing, and write a program that draws the below pattern using a single for-loop.
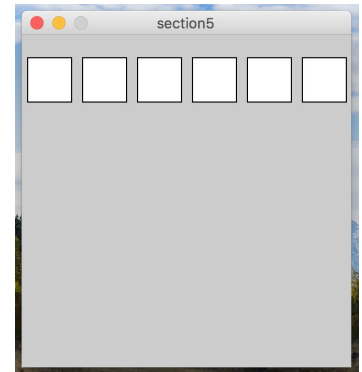
(2)   Next, modify the code you wrote above to generate this canvas instead. Do so using three separate for-loops, one after the other. You can just copy-paste the first for loop, and change the **Y** coordinate of the rectangles in each.
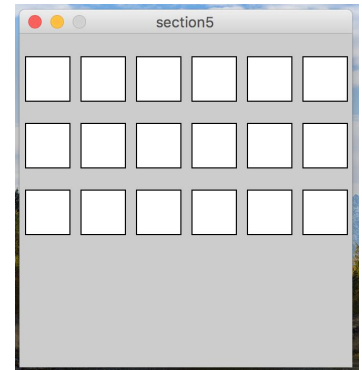
(3)   Now, take a step back and consider the solution for 2. You duplicated nearly the exact same for loop, one after the other. *The only thing that changed between each one is the Y coordinate of the rectangles*. Whenever we see repetition with minor changes, we can typically use a for-loop to reduce the redundancy! Try drawing the same canvas again, but instead use only 2 for-loops, one inside of the other.
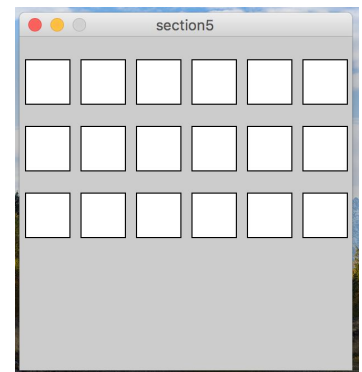
(4)   Now that you are using a nested for-loop, change the program so that is displays one more "row" of squares, pictured to the right. Most likely, you can do this by changing only the for-loop **test**.
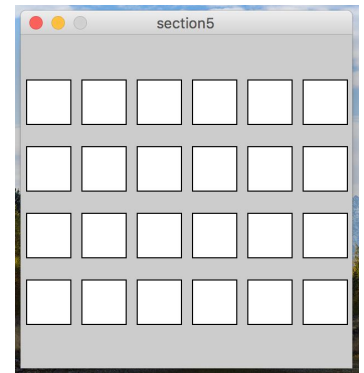
(5)   Make a few more changes to your program so that the canvas looks like the one picture to the right. Notice that **(A)** the canvas is wider, **(B)** more squares are drawn in each row, **(C)** The fill of the circles goes from dark-red to light red as the rows progress.
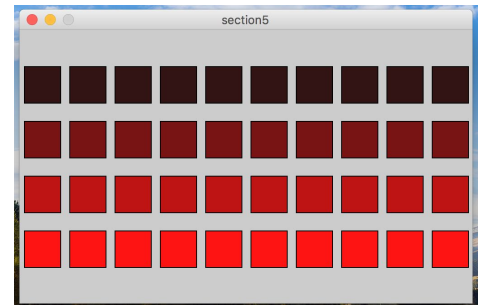
(6)   Again, make some additional changes so that the canvas looks like the picture. Make sure to get the colors and spacing to match.