

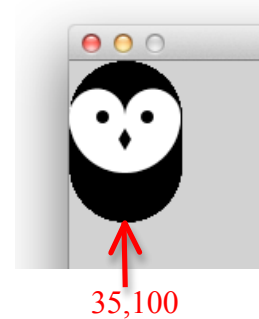


Homework Exercise 6: Functions

Goal: To get practice writing functions, especially using parameters.

In Chapter 8 of *R & F*, the authors use a cute little owl to illustrate several programming ideas, functions being the main one. Find the file `owl.txt` with this assignment. It is the (slightly simplified) owl program. The function for the owl is as follows:

```
void owl (int myx, int myy) {
  stroke(0);
  strokeWeight(70);
  line(myx, -35+myy, myx, -65+myy); // bodmyy
  noStroke();
  fill(255);
  ellipse(-17.5+myx, -65+myy, 35, 35); // left eye dome
  ellipse( 17.5+myx, -65+myy, 35, 35); // right eye dome
  arc(0+myx, -65+myy, 70, 70, 0, PI);
  fill(0);
  ellipse(-14+myx, -65+myy, 8, 8); // left eye
  ellipse( 14+myx, -65+myy, 8, 8); // right eye
  quad(0+myx, -58+myy, 4+myx, -51+myy, myx, -44+myy, -4+myx, -51+myy);
}
```



Notice that the owl is a 70 x 100 figure.

Part 1: Add Parameter. Change the `owl()` function so that it accepts one more input parameter, namely a variable of type `color`. Call the parameter `c`, for color. A legitimate colored input could be, say, `color(0,0,0)` which continues the black theme, as in `owl(35, 100, color(0,0,0))`. To make a brown owl, you might write `owl(105,100, color(160, 106, 6))`.



In any case the owl should have black eyes and black beak. Comment the additions to the program.

Part 2: Add Function. Write a new function with the following specification:

Name: `pick`

Parameters: *none*

Returned type: `color`

Returned value: `color(random(255), random(255), random(255))`

The returned value will create a random color. Here's how it works. The Processing function `random()` returns an unpredictable value every time it is called. The number

comes from the range (0-arg), and since the arguments here are 255, the random number is a random RGB value. Do it three times, and you have a random color! The function is only one line long.

Part 3: Make A Flock. Make a new function to draw a row of 8 owls each of a random color. To replace the brown color with a random color using your new pick() functions, write owl(105,100, pick()). The owls should be in a row, and they should be tight together, as shown above. Call your function owl8(). It should have two parameters:

x, y are integers, which are the starting position of the first owl (left most) of the sequence,

Notice that the argument for the color parameter of owl() calls will be a call to the pick() function. That is, it's OK to call a function as an argument. That's what we did in color() above. Be sure to comment your program.

Exciting Development! Comment out the noLoop() function in the setup() function, by placing two slashes (//) at the start of the line. Wow! What's happening? Remember, the draw() function runs over and over, and each time it runs, it picks a new random color!



Part 4: Make Extra Flocks. Make a function to draw six rows, each containing 8 owls. The function should have two parameters, x, y for the position of the initial row. Comment your program.

Part 5: Finish Up: Add a branch and “claws” to make the owl look like it's sitting on a green the branch. (They're just 4x4 black boxes!) All copies of the owl should have the branch, and it should look like one continuous branch if more than one owl is drawn. See figure at right. Comment your program.



Wrap Up: You have practiced writing functions, calling functions and including parameters. These skills will be used throughout your computational thinking.

Turn in your .pde code to the class dropbox.