APCSP - Part02

Arrays and Iteration

Kai kai@42.us.org

Summary: Use arrays in your program and write functions that loop over every item in the array.

Contents

Chapter I

Tips for using this Class

I.1 Creativity

First and foremost, remember: Your goal is to learn how to write code from scratch. Copying code from others is risky simply because you will not learn how it is to struggle and invent solutions when you are trying to do something that no one has done before. Be playful with the code, use trial and error, and think about how you would give step by step instructions to a little kid. Then, put those instructions into the language of code.

I.2 Formatting

You should always type code examples from the PDFs, rather than copying and pasting them. As you type, make sure that all the code you write has good indentation. Copying and pasting from the PDF messes up the formatting and also you will understand the code better if you type it yourself.

```
Increase your keyboard speed to type more efficiently!  You can
adjust it under System Preferences -> Keyboard.  I like to turn
up both Key Repeat and Delay Until Repeat to the fastest/shortest
settings.
```

I.3 Stick to the Documentation

Don't worry about learning the Absolute Truth of how Things Are Done in Javascript. The truth is, Javascript and other programming languages change their syntax all the time! In this class we are practicing how to read coding **documentation** by becoming experts in the p5js library. Not all Javascript code shown online will work here, and Processing has some nice parts that don't exist in other Javascript versions. Always reference the p5js Reference Docs and the Mozilla Developer Docs for Javascript.

I.4 Error Messages

Error messages are your friend! They usually provide helpful information about where and what the error is. Be grateful that you have an error message, rather than the code failing with no explanation. Try to read and understand what the error message has to say.

I.5 Debugging

On that note, you can create your own "error messages" or just feedback from the code by adding print statements which tell you what the value of a variable is, or which part of the code executes in which order. This habit will improve your programming skill exceptionally. Don't stare at code wondering what is happening; Add prints and comments!

I.6 Commenting

Don't forget that you can temporarily remove code by commenting out those lines. In most code editors, you can highlight the lines that you want to comment, and then press the "Command" and "/" keyboard buttons at the same time.

I.7 Peer to Peer!

Be as helpful as you can to your classmates and know that it is your responsibility to help teach each other! The more you explain a concept, the better you understand it and the better your communication skills will be.

Chapter II

The Building Blocks of Algorithms

An essential concept to remember for the APCSP curriculum is that `algorithms` are made up of `sequencing`, `selection` and `iteration`.

II.1 Sequencing

refers to the fact that when we write a program, each line of code will execute its action in the order that we write them.

II.2 Selection

is about if/else statements and Booleans. We can use logical statements in our programs to make the computer choose different options depending on where you click, what you type, or what input is randomly generated.

II.3 Iteration

refers to repeating something over and over again. This week we will talk about how to use `for loops` for iteration. `While loops` are another tool you can use for a very similar purpose. The point of iteration is that we often need a block of to repeat many times, but we control how many times and when it stops.

Chapter III

Arrays

III.1 Read

Remember the basic data types we have described so far: `strings`, `numbers` and `Booleans`?

`Arrays` are a data type that allows you to create a list or collection of any other data type. The use square brackets and you should put a comma between each item in the array.

Skim through the Mozilla Developer Docs on this topic: (MDN: Javascript Arrays). They have lots of helpful examples, and you can run those in the Processing window and remix them.

Arrays are `indexed`, which means, each item in the array is labelled by a consecutive number starting at zero. You can grab a specific item from the array by asking for the item at that index.

III.2 Watch

Watch the video The Coding Train 7.1: What is an Array?

III.3 Homework Part 1: SuperKitty Name Generator

Cats Protection is the UK's leading feline welfare charity. They have provided this helpful chart for new cat ~~owners~~ servants who have been chosen by a fluffy one:

Imagine that you work at a pet shelter and want to make sure that as many cats get adopted as possible, by giving them awesome names. For this purpose you will build a kittyhero name generator!

Your program should have two arrays, one holding first names (like "Doctor" and "Super"), and one holding last names (like "Destroyer" and "Scratcher"). You can choose your favorite 10+ names from the list for each array, or make up your own.

Every time the user clicks the mouse on the campus, you should pick a random item from both arrays. Concatenate the first and last name together into one name, and print it out to the screen.

Use a variable that gets bigger when mouse is clicked to print each new name to a different row on the screen.

Make sure that the previous name is erased before you print out another new name.

Chapter IV

For Loops

When we are coding, we are giving exact step-by-step instructions to the computer. The computer will do exactly what we tell it to do, in the order that we gave the instructions, as long as it understand what we are saying.

However, what if you have something repetitive that you want the computer to do? For example, "look up the superpower of each superhero", or, "draw the 42 logo sixteen times in a line across the screen." There's no way you want to repeat the 42 logo assignment 16 times, with different coordinates. You'll want to use a for loop.

IV.1 Watch

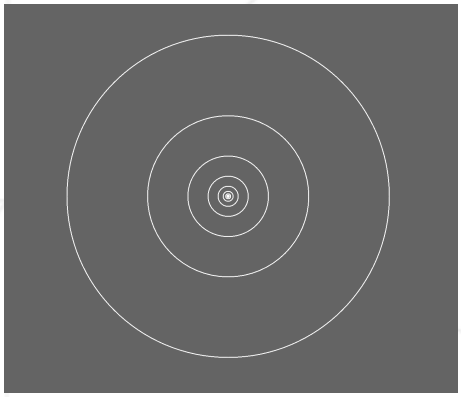
Watch The Coding Train 7.2: Arrays and Loops.

IV.2 Read

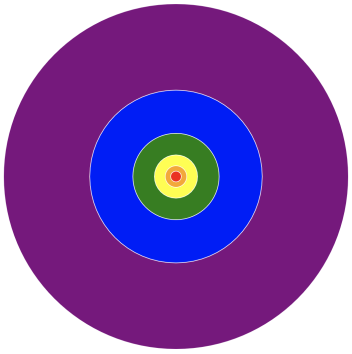
Skim a few sections of the MDN docs: JS Building Blocks: Looping Code, Javascript Guide: Loops and Iteration, and Javascript Reference: for.

IV.3 Homework Part 2: Bullseye

Use a for loop to draw a bullseye where each circle has a diameter that is twice the diameter of the next smallest one, from 1 up until 512. For each one you will draw only the outline without filling it in (use noFill()), so that we can see all the circles at once. I recommend also using noLoop() so that the draw() function only runs once.



Bonus: Make a multicolor bullseye with the circles filled in. You will need to start at the outside and move inwards to draw first the largest one, and then the next-smallest, and so on.



Chapter V

Arrays of Arrays

V.1 Watch

Watch and follow along with The Coding Train: 2D Arrays in Javascript.

V.2 Read

Just as an array can contain strings, numbers or Booleans, it can also contain other arrays.

You can initialize an array of arrays like this:

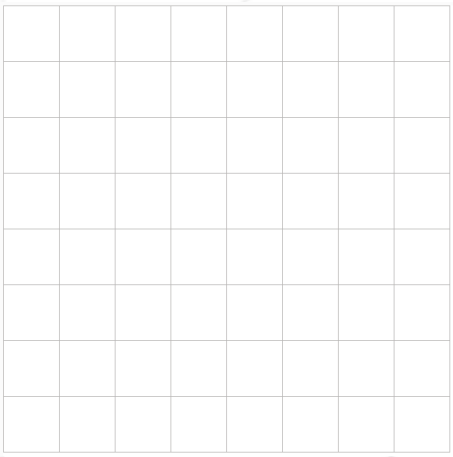
Or like this:

Notice that the board[i][j] notation allows you to read or change the value of an array inside and array. board[i] gets you to the i'th position in the board array. board[i][j] gets you to the j'th position in the board[i] arrray.

Computer programmers often use "i" and "j" as variables when they are counting along the x and y axes. It's one of those things, like having (0,0) in the top left-hand corner. Why are variables i and j used for counters?

V.3 Homework Part 3: Simple Grid

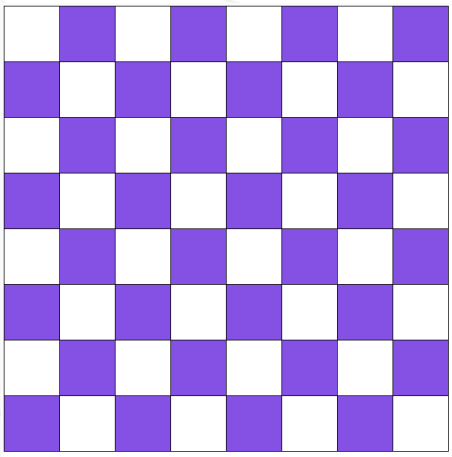
Use a double for loop to draw an 8x8 grid on the canvas, like a chess board. I want you to draw each square of the grid as an individual rectangle (don't just draw the grid lines.) Hint: You will need a variable to hold the height or width of each square on the grid.



V.3.1 Bonus

Bonus: Use if statements and the modulo operator (%) to color in alternating squares of the grid like a chessboard.

Figure this one out yourself! Use that brain!



V.4 SuperBonus: Whack-a-Mole

Create a game on the blank grid:

- Your grid (more fun if it's larger than 8x8) has some squares that are colored in. (You can use an array of arrays to store x and y coordinates for the target squares, or an array of objects.)
- Use frameCount in the p5js reference to control time. Periodically, another random square should turn into a colored square.
- When the user clicks on one of the colored squares, their score goes up and the square goes back to default color.

- Display their score on the board!

Chapter VI

Turning in Your Code

VI.1 Style Guidelines

Part of being a good coder is making sure that your code is organized and readable to other people - like having good handwriting.

You should follow the indendation shown in the examples and use understandable names for your variables in order to make sure that your corrector gives you a "good style" mark!

Indentation rules:

- All code starts with no indentation, by default.
- When a block of code starts, the code inside the block is indented one more than the outside. Functions, If statements, Else statements, and For loops are all code blocks where the inside should be indented.
- When the block of code ends, go back to the previous level of indentation.

VI.2 Push to Vogsphere

You should turn your work into Vogsphere like so:

1. After closing your team for the project and opening your grading slots for the end of the day, clone your repository from Vogsphere.
2. Create a text document inside your project folder, and paste the links to your Processing sketches for this week into the text document.
3. Push your project folder to Vogsphere before setting the project as finished.
4. Sign up for corrections on intra.

Additional detail about how to use Vogsphere is on the 'Hello 42! Hello Terminal!' PDF.

