LESSON 5: CREATING FUNCTIONS

Adapted from Code.org curriculum

Objectives: Students will be able too…

- Recognize functions in programs as a form of abstraction
- Write a program that solves a turtle drawing problem using multiple levels of abstraction (i.e. functions that call other functions within your code)
- Explain why and how functions can make code easier to read and maintain
- Define and call simple functions that solve turtle drawing tasks

Getting Started: Simplifying the 3x3 Challenge

- In the previous lesson we created simple turtle drawings using only four commands.
- As some point you probably wished that more commands were available to you.
- Tell your neighbor you wanted to be able to use and explain in which way you wanted to use it.

Notes on Program Style

- Camelcase: Function names are written in camelcase.
  - Example: thisFunctionNameIsCamelcase()
- Function Definitions at Bottom: Function definitions appear at the bottom of the program. This is done so that the first line of the program is the first line of code actually run. The opposite format is also very common.
- Functions Names Can’t Start with Numbers: In most programming languages, including JavaScript, function names are not allowed to begin with numbers or most symbols. Underscore, however, is allowed.

Activity: Code Studio Stage 5

- Video about JavaScript
- Go through the 7 challenges

Vocabulary

- Abstraction – pulling out specific differences to make one solution work for multiple problems
- Function – a piece of code that you can easily call over and over again

Wrap-up: Reflection on benefits of functions – write short response

- List the benefits of being able to define and call functions in a program. Who specifically gets to enjoy those benefits?
- How is the use of a function an example of abstraction?