Day 3:

Note that many of the examples used are the same as in Day 2. This is intentional! The focus of this class should be turning pseudocode into actual code. Syntax is required to make sure a program is well formed and readable by the computer.

* What is javascript?
  + Javascript is a language
    - What’s a language again?
  + A language is something that we use to communicate with the computer
  + A language has special rules (syntax) that we must follow
  + We can turn our pseudocode programs into actual programs by applying the syntax rules!
* Variables
  + Types matter!
  + Introduction to types
    - Numeric
    - Boolean
    - Strings
  + Must be careful to use types appropriately
    - What happens when we add strings?
    - Explain concatenation
  + Tell a story using variables:
    - Variables.js script
    - Looks very much the same as our pseudocode!
    - Show the pseudocode side by side
    - Some minor syntax changes (var, console.log)
  + Ask them to tell a story using variables (10 minutes to work on this)
    - One variable per person
    - Be careful with order of operations!
    - Walk through the solution step by step
* Choice
  + Computers are intelligent enough to make decisions
  + We must provide them with enough information to make these decisions
  + Example: MaxWithEquals.js
    - Three possible outcomes (x > y, x < y, x == y)
    - Pay attention to syntax!
  + Example: MaxNested.js
    - Again notice that it looks much like our pseudocode!
  + Exercise: swap (10 minutes to work on this)
    - Refer back to your pseudcode solution and apply syntax rules
* Iteration
  + Example: Loops.js
    - The required pieces are all there
    - Syntax required:
      * Parentheses
      * Curly Brackets
  + Exercise: loop writing (5 minutes)
    - These problems all involve changing the pieces of a loop definition
  + Exercise: Fibonacci (10 minutes)
    - Must define necessary variables, before the loop!
    - Must set variable values up for next iteration
  + Exercise: Fizzbuzz (10 minutes)
    - If statements and loops can be used together!
* Functions
  + Again, requires only some syntactical changes
  + Example: mathFunction.js
    - Highlight pieces required for a function definition
      * Output type
      * Name
      * Input type
    - This is the same example as shown on the slide
    - Nice example of a mathematical function
  + Example: maxFunction.js
    - The stuff we have discussed so far (loops, ifs, variables) are fair game for functions!
  + Exercise: Fibonacci function (5 minutes)
    - Turn Fibonacci into a function
    - Code can remain mostly the same, just need to add the function definition!
  + Exercise: Mult (10 minutes)
    - A bit more difficult
    - Uses a loop to perform the multiplication with repeated addition