CS 1132 lecture 6

I. Subfunctions
    a. Motivation
        i. Example: modular pi approximation
    b. May define multiple functions in same ".m" file, but only first one (whose name must match file's) can be seen from outside (called from scripts, other functions, or command window)
    c. May end functions with "end" keyword, but if used, must be used consistently
        i. Required for subfunctions in scripts
    d. Useful for modularizing tasks (makes high-level logic easier to see), or for helper functions that are too specific to be reused in other contexts

II. Vectorized code
    a. Motivation
        i. Example: vector geometry
        ii. Example: Computing weighted grades for whole class
        iii. For-loops can be slow in MATLAB
    b. "scalar" variable/expression: stores a single value
    c. Can use for-loops to perform elementwise scalar operations on vectors
    d. Vectorized operations, dot syntax
        i. Simple rule: always use dot for multiplication, division, exponentiation
    e. Vectorized functions

        i. Most built-in math functions are vectorized
        ii. Aim to make your own functions vector-compatible (preemptively use dot-operators everywhere)

III. Slicing
    a. Syntax: range expressions as indices
    b. Subarray can be used as if it were its own array
    c. When used on LHS, assignments will change original array

IV. Stencil operations
    a. Motivation: image processing, PDEs
    b. Approach: define a function to process a whole array, then slice subarrays corresponding to neighborhoods and pass those to the function

V. Boundary conditions
    a. If ignored, could get an index-out-of-bounds error
    b. Option 1: pad matrix with border
        i. Need a “valid” border value
        ii. Construction options
            1. Concatenate boundary rows, columns using brackets
            2. Copy data over fill by assigning to slice
        iii. Need to adjust indices specified by user to account for new top, left border
    c. Option 2: use variables for subarray extents, clamped with min() and max()