

HW 2:

Pro tips: Think about the problems before writing code. Take your time and test every little piece of code before you add more. Google and help() are your friends.

Reading:

1. Review chapters 2-3 (optional read 4)
2. Read chapters 5-12 (We covered a TON this lecture - about 6 weeks of material. These chapters go through the concepts slowly and with more examples.)

Basic review:

3. Lists, loops, and strings practice.
 - a. Create (define) a list with the numbers: 1,4,5,2,6,10,14. Remember that to create an object like a list, you need to make a variable to store it. Call it
 - b. loop through the list with a for loop, and display each integer in the list.
 - c. Create a counter variable called "counter", and initialize it to 0 before looping through the list.
 - d. Increment the counter by 1 in every iteration of the loop.
 - e. Use string formatting operator, to define a string of the form: "The value of list entry ___ is ___." Replace the blanks with "counter" and the current list value.
 - f. Show the string from (e) Inside the loop.
4. Functions
 - a. Define a function called "verbosePrint" that takes one argument, a list of integers.
 - b. Have the function do steps of part 4, but with a list that is passed to the function. Hints: Copy paste your code from problem 4. Remember that when you define a function, the code does not execute until you call it. So call the function to test it.
 - c. After defining your function, test it like "verbosePrint([1,7,2,2,3])".
5. While loops
 - a. Do problem 4, but instead of using a "for" loop use a "while" loop. The conditional of the while loop should be true for every iteration of the list, and false when you have displayed all entries of the list. A straightforward way is to have a conditional like "iteration < numInList".
 - b. Hint: "iteration" is essentially the "counter" variable you already have.
 - c. Hint: Get the number of items in a list with "len" function.

Exercises:

6. Write a sort function that sorts an arbitrary list of integers. Do not use the "sort" or "find" functions to implement this. Be careful copying lists and passing lists to functions. Lists are passed to functions by their address (reference), so whatever you do to the list inside the function will be reflected in the main program scope.
7. Do exercise 3 from Lecture 2 (Write a function that draws an ascii tree having any number of rows). Remember to take your time and test little bits of code as you build the function. Hints:
 - a. Create a string of 10 zeros: `x = ['0'] * 10`
 - b. Concatenate strings: `x = '000' + 'XYZ' + '000'`