# Lab 3.04 - Aliasing & Scope ## In Your Notebook ### Aliasing 1. Will updating b affect a? Explain why or why not? ```python a = [1, 2, 4] b = a ``` 2. Predict what `my_list` list will print out when this code is run. If you are not sure check the code by copying and running it. ```python # input: a list of ints # output: an int def update_list(a_list): a_list[3] = "yo" b = a_list[4] b = 100 my_list = [1, 2, 3, 4, 5] update_list(my_list) ``` ### Scope 1. Draw a stack diagram for the following: ```python var_1 = "kittens" var_2 = "cookies" # input: a string # output: a string def my_function(my_favorite_things): song_lyrics = "rain drops on roses," combined_song = song_lyrics + my_favorite_things return combined_song # input: a string # output: a string def my_function_2(item, item2): full_lyrics = item + "on " + item2 full_song = my_function(full_lyrics) return full_song my_song = my_function_2(var_1, var_2) ``` ## Complete the following on your own 1. Write down what (if anything) is wrong with the following code. 2. If there was an issue write out how to fix it. 3. If you are unsure copy and run the code and fix it ### Problem 1 ```python var_1 = 'cat' var_2 = 'dog' def print_out_my_favorite(favorite_pet): if favorite_pet == var_1: print("My favorite pet is the cat.") if favorite_pet == var_2: print("My favorite pet is the dog.") var_2 = "cat" print_out_my_favorite(var_1) print(var_2) ``` ### Problem 2 ```python var_1 = 'cat' var_2 = 'dog' def print_out_my_favorite(favorite_pet): var_1 = 'dog' var_2 = 'cat' if favorite_pet == var_1: print("My favorite pet is the cat.") if favorite_pet == var_2: print("My favorite pet is the dog.") print_out_my_favorite(var_1) print(var_1 + " " + var_2) ``` ### Problem 3 ```python var_1 = 'cat' var_2 = 'dog' def print_out_my_favorite(favorite_pet): if favorite_pet == var_1: print("My favorite pet is the cat.") if favorite_pet == var_2: print("My favorite pet is the dog.") print_out_my_favorite(var_1) print(var_2) ``` ## In your your console ### Write a program using the following specifications 1. That has a global variable, `my_num`. 2. Create three functions that update `my_num` 3. `add2`: this function adds 2 to `my_num` 4. `multiply_num`: this function takes in a parameter, `multiplier`, and multiplies `my_num` by that parameter 5. `add2_and_multiply`: this function takes in a parameter, `multiplier`, and calls `add2`, then calls `multiply_num`. ### Complete the program Write the following code in the main part of the program. 1. sets `my_num` to some initial value you choose 2. prints `my_num` 3. calls `add2_and_multiply()` with some argument you choose 4. prints the final value of `my_num` 5. Confirm that the printed values match what you expected.