**Exercise 8: Functions**

**Time: 60 minutes**

Please work through as many of the following exercises as you’re able to in the allotted time and upload the results of the last task you were able to complete**.**

1. Write a function called display\_message() that prints one sentence telling everyone what you are learning about in this module. Call the function, and make sure the message displays correctly
2. Write a function called favorite\_book() that accepts one parameter, title. The function should print a message, such as, “One of my favorite books is Alice in Wonderland.” Call the function, making sure to include a book title as an argument in the function call.
3. Write a function called make\_shirt() that accepts a size and the text of a message that should be printed on the shirt. The function should print a sentence summarizing the size of the shirt and the message printed on it. Call the function once using positional arguments to make a shirt. Call the function a second time using keyword arguments.

* **Stretch and Challenge:** Modify the make\_shirt() function so that shirts are large by default with a message that reads, “I love Python.” Make a large shirt and a medium shirt with the default message, and a shirt of any size with a different message.

1. Write a function called describe\_city() that accepts the name of a city and its country. The function should print a simple sentence, such as, “Reykjavik is in Iceland.” Give the parameter for the country a default value. Call your function for three different cities, at least one of which is not in the default country.
2. Write a function called make\_album() that builds a dictionary describing a music album. The function should take in an artist name and an album title, and it should return a dictionary containing these two pieces of information. Use the function to make three dictionaries representing different albums. Print each return value to show that the dictionaries are storing the album information correctly. Add an optional parameter to make\_album() that allows you to store the number of tracks on an album. If the calling line includes a value for the number of tracks, add that value to the album’s dictionary. Make at least one new function call that includes the number of tracks on an album.

* **Stretch and Challenge:** Write a while loop that allows users to enter an album’s artist and title. Once you have that information, call make\_album() with the user’s input and print the dictionary that’s created. Be sure to include a quit value in the while loop.

**Stretch & Challenge**

If you complete the previous steps within the allotted time please move on to the following.

1. Make a list of magician’s names. Pass the list to a function called show\_magicians(), which prints the name of each magician in the list.

**Further Stretch and Challenge:**

* Write a function called make\_great() that modifies the list of magicians by adding the phrase, “the Great” to each magician’s name.
* Call show\_magicians() to see that the list has actually been modified.
* Call the function make\_great() with a copy of the list of magicians’ names. Because the original list will be unchanged, return the new list and store it in a separate list. Call show\_magicians() with each list to show that you have one list of the original names and one list with the Great added to each magician’s name

1. Write a function that accepts a list of items a person wants on a sandwich. The function should have one parameter that collects as many items as the function call provides, and it should print a summary of the sandwich that is being ordered. Call the function three times, using a different number of arguments each time.
2. Using a program you wrote that has one function in it, store that function in a separate file. Import the function into your main program file, and call the function using each of these approaches:

- import module\_name from module\_name

- import function\_name from module\_name i

- import function\_name as fn import module\_name as mn

- from module\_name import \*