Project: Python Programming – Choose Your Own Adventure

A guide for Data Maverick Premium members

Written by David Venturi (davidanalyst.com)

Congratulations on completing the most important chapters of Dr. Chuck's Python for Everybody course! You have gained a solid foundation in Python programming. Now, it's time to apply your newly acquired skills. This guide is for members that want to create their own project rather than follow the standard project (which you can find and use as inspiration here).

Project Specification

Here's how you can create your own project:

- Choose a problem to solve. Think about a problem that you can solve using Python programming. It can be something related to your personal interests, a task that you do regularly, or a common problem that you encounter. For example, you could create a program that calculates the average of a list of numbers, a program that generates random passwords, or a program that converts units of measurement. If you are having trouble thinking of a project idea, you can use ChatGPT to generate ideas for you. For example, ask it: *"Create a project scope for a student that just finished the first 11 chapters of Dr. Chuck's Introduction to Python programming. Use the skills taught in those chapters only."*
- Define the inputs and outputs of your program. Think about what inputs your program will need and what outputs it will produce. For example, if you are creating a program that calculates the average of a list of numbers, the input would be a list of numbers, and the output would be the average.
- Write the code for your program. Start by defining any necessary variables and functions, and then write the code to perform the desired task. You can use the programming concepts that you learned in the course, such as variables, control structures, and functions.
- Test your program thoroughly to make sure it is working correctly. Try different input values and check that the output is what you expect.
- Document your program. Write comments to explain what each part of your code does, and include a description of the inputs and outputs of your program.

Note: Here is my first ever Python project, which you can use as inspiration as well.

Project Rubric

1. **Functionality (80 points)**
    - The project makes use of 8 of the 9 main concepts taught in the course, and uses them according to the best practices taught in the course. (10 points each x 8)
        - Variables, expressions, and statements (chapter 2)
        - Conditional execution (chapter 3)
        - Functions (chapter 4)
        - Loops and iterations (chapter 5)
        - Strings (chapter 6)
        - Files (chapter 7)
        - Lists (chapter 8)
        - Dictionaries (chapter 9)
        - Tuples (chapter 10)
2. **Code Quality (10 points)**
    - The code is well-structured, organized, and readable with appropriate naming conventions. (5 points)
    - Proper indentation and consistent coding style are maintained. (3 points)
    - Comments are used to explain complex logic or enhance code understanding. (2 points)
3. **Documentation (10 points)**
    - The project includes a clear and concise project specification explaining the requirements. (5 points)
    - The code includes comments or docstrings to explain the purpose and functionality of each function. (5 points)

Total: 100 points