Quick reference to find each project required technique:

1. Container type,
   * plentiful in project. Easy reference budget\_constructor.py line 31, questionnaire dict. Example of one use is in the intake\_questionnaire function, line 291.
2. Iteration type:
   * Also plentiful, see also line 296 of budget\_constructory.py for quick reference.
3. Conditional:
   * See line 277 budget\_constructor.py for quick reference.
4. Input file (write to file) ex:
   * Budget\_constructor.py within CompleteBudget class methods line 193
5. Reading file:
   * Within fetch\_survey function in budget\_constructor.py beginning line 222.
6. There are four user defined classes total in this project, three in budget\_constructor
   * CompleteSurvey has private attribute self.\_\_survey line 109
   * Additionally complete survey has the private method \_\_generate\_title line 126
   * Public methods and attributes are common
     1. CompleteSurvey’s edit\_response is a quick reference to a public method that takes args.
   * CompleteSurvey also has both a \_\_repr\_\_ and \_\_str\_\_.
7. Tests are with unittest.TestCase classes in tests.py and thusly not included in an if \_\_name\_\_==’\_\_main\_\_’: block.

***Project concept:***

This app is designed to automate the process of generating an estimate range given certain fluctuating parameters for the cost of travel of The U.S. Army Blues Jazz Ensemble, “Pershing’s Own”. The app runs from run\_app.py where the user will either be editing an existing budget or creating a new budget. This file controls the flow of the user’s session within the app based on the purpose for running. In either case prompts are displayed on the console. This will eventually need to be refactored into either a Flask or Django app for an improved user experience.

The file budget\_constructor.py handles all the user input, handles cataloging survey responses for later easy editing via writing the responses to a .txt in the intake\_questionnaires subdirectory (and handles creating said sub-directory if not already there). Similarly, budget\_constructor.py calls on helper functions from functions.py to make required calculations in order to write those to a <title>\_<year>\_budget\_draft.txt file in the budgets subdirectory.

The functions.py file contains functions for necessary calculations and additionally functions for fetching data from the three API’s queried (tolls, meals and lodging, and commercial flight rates).

The person.py file is contains the definition for the Person class and the groups.py file both instantiates the pool of actual people and groups them by job.

It is important to note: these budgets are designed to output both low ball and high ball estimates to present these figures up the chain to individuals who will allocate funding typically within the following fiscal year. Thusly, the constant in functions.py (line 40) GOVT\_FLIGHT\_RATE is used as a baseline high ball to pad a figure if flight is required on the trip. Commercial rates are dramatically lower as fees for late cancellations and changes are not included in these figures as they are with government flight rates. Additionally, in the current climate I am required to present a figure for all individual rooms in the case it is deemed safe enough to make a trip but not spend such prolonged periods in a room with another. See get\_total\_rooms on line 290 in functions.py for details on how the rooms breakdown under “normal” circumstances.

***Required Packages:***

* requests and python-decouple

The build was done in a pipenv virtual environment and I thusly installed these packages to my working environment with the “pipenv install <package\_name>” command in the terminal.

***Known Issues:***

There is no error checking for input in edit mode yet. I do intend to add this feature by factoring out the checking from the intake\_survey function to another function, perhaps clean\_input, in budget\_constructor.py, then plugging that function both into the intake\_survey and run\_app.py for edit mode.