Exercise 1.2: Data Types in Python

Reflection Questions

1. Imagine you're having a conversation with a future colleague about whether to use the iPython Shell instead of Python's default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

The default shell is not as user friendly. The lines are the same color and they can quickly become muddled together as they are not numbered. iPython provides a substantial user-friendly improvement. Lines are color coded for inputs and outputs, and they are numbered to allow for quicker access and tracking of your work. When it is suggesting code using autocompletion, the suggestions are much easier to read and are clearly separate from any code-meaning they won't get confused with the actual code.

2. Python has a host of different data types that allow you to store and organize information. List 4 examples of data types that Python recognizes, briefly define them, and indicate whether they are scalar or non-scalar.

Data type	Definition	Scalar or Non-Scalar?
int	integer	scalar
float	decimal number	scalar
string	string of characters	non-scalar
bool	true/false statement	scalar

3. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.

The main two differences between tuples and lists- tuples are quicker to access especially as data size scales up, and lists are able to be modified (mutable) unlike tuples meaning certain data operations can be performed quickly and easily compared to with tuples.

4. In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can

input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language-learning app beyond vocabulary memorization.

Tuples seem like a good choice since definitions and words are static, unless there is a spelling error or the user wanted to edit a card there should generally be no changes made. A mutable data structure like a list would not be advantageous because of this. But if the user is looking for the UX ability to make edits to flashcards- then perhaps lists would be the better choice. Another thing to factor in is how much definition needs to be stored in each object. If the syntax desired is to have key-value pairs for each value then perhaps a dictionary could be better suited for the task. Otherwise, a class could be used to structure words, definitions, and categories in an ordered index manner (word name in [0], definitions in [1] and categories in [2], etc...)