Exercise 1.2: Data Types in Python

Learning Goals

- Explain variables and data types in Python
- Summarize the use of objects in Python
- Create a data structure for your Recipe app

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?
    - iPython is considered a more user-friendly shell, especially when it comes to reading code. Python's default shell is one color of text, which can make it difficult for some to follow and see individual lines of code. Additionally, iPython has some auto features like indentation instead of indenting each line as you're working through it. Lastly, iPython lets you test out small chunks of code as each command is executed immediately after you type it in and expected responses are printed right away.

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?
Tuples	Linear arrays that can store multiple values of any type.	Non-Scalar
Strings	Immutable array of characters surrounded by single or double quotations and generally composed of alphanumeric characters and symbols.	Scalar
Lists	Similar to Tuples, but different in that they are mutable. Meaning any of the internal elements of a list can be modified or deleted.	Non-Scalar
Dictionaries	Stores values and objects within itself and indexed by identifiers or keys.	Non-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.

    - Lists and tuples are both used to store collections of items, but their primary difference lies in their mutability. Lists are mutable and can be modified by such actions as adding, removing, or changing elements after their creation. Tuples, on the other hand, are immutable and their elements cannot be changed once they're created.

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.

    - I believe for a language learning app where users create inputs for vocabulary words, categories, and definitions, dictionaries would be the best data structure. Dictionaries would allow for key-value pairing where words would be the, while definitions and categories be the values. As one of the user requirements is to be able to create, update, and delete words, dictionaries would also serve as a good choice since they offer flexibility to perform these tasks. If this app were to develop further in the future and offer new features such as audio, sentence structure, or social functions, it may be beneficial to use a combination of lists, dictionaries, and tuples depending on the apps use and needs.