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?

The iPython shell has really beneficial features including automatic indentation, syntax colorization, and code completion which increases readability and speed of use.

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?**
Tuple	Linear arrays storing multiple values of any type	Non-scalar
List	Mutable sequence of characters that must be wrapped in [ ]	Non-scalar
String	Immutable sequence of characters wrapped in " "	Non-scalar
Dictionary	Set of items, unordered, created and stored in a key-value pair, each key being unique.	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.

The primary difference between tuples and lists is that lists are mutable – can be changed - and tuples are immutable – remain the same.

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 would use a dictionary in this case. The key-value pairing lends itself well to the type of data that needs storing, the ability to modify the data in the future is advantageous and the fact that varying types of data can be stored opens up the potential for broader language learning as the users' skills develop.