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?
   I would say that iPython has certain advantages such as it has similar to the regular Python REPL, but with additional features such as syntax highlighting, auto-indentation and robust auto-complete features. It has magic commands that simplify common tasks. It also has a rich history of commands that can be reused and can save session environment across different runs.

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?
Integer	Whole number	Scalar
Float	Decimal number	Scalar
List	Ordered, mutable collection of items. Can contain elements of different types	Non-Scalar
Dictionary	Collection of key-value pairs. Keys must be immutable types	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 difference between a list and a truple is a list is mutable while a truple is immutable. The list syntax is [ ] while a truple is ( ). Also, lists tend to have slower performance than truples. Lists have more methods and methods for modification, while truples have fewer methods and they are not for modification
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.

For each individual card I would still stick with dictionary because keys-value pair would work well with adding definitions and word types for each word. I could put each flashcard into a truple because it runs faster and the cards would probably not need to be updated after being put in the truple.