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 more user-friendly, and the code is more readable, since iPython uses contrasting fonts and colors and automatic indentation for nesting statements.

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
Lists	Type of ordered sequence like tuples, except they are mutable.	Non-Scalar
Strings	Immutable array of characters, that can be composed of alphanumeric characters as well as symbols and are surrounded by "".	Non-Scalar
Dictionaries	Stores values and objects within itself indexed by keys. It's an unordered set of items, each of them a key-value pair, where each key is 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 main difference between lists and tuples is that lists are mutable meaning you can modify, add, or remove items, but tuples are immutable meaning once they are created, they can't be modified.

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 choose the dictionaries because they allow me to define the data items in key value pairs, so I can define words, definitions and categories as keys. Dictionaries are also mutable so it's easy to modify the data inside the flashcards.