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 is more practical than the default shell, because it provides more guidance (code text is visibly clearer thanks to syntax highlighting, text for nested statements is indented automatically) and lets you test out small chunks of code quickly and easily.

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?
Boolean (bool)	This data type represents one of two values: True or False. It's useful to evaluate an expression in Python (*e.g.* compare two values, or run conditions in if statements).	Scalar
None Type (NoneType)	This data type is a special data type in Python that carries one value: None.	Scalar
Tuple	This data type stores multiple items in a single variable. A tuple is a collection that is ordered and unchangeable, and allows duplicates. Tuples are written with round brackets.	Non-Scalar
Dictionary	This data type stores values in key:value pairs. A dictionary is a collection that is ordered (as of Python version 3.7), changeable, and doesn't allow duplicates.	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 data types that store multiple items in a single variable. Lists are ordered, changeable, and allow duplicates; tuples are ordered, unchangeable, and allow duplicates. This means that, unlike the items in a list, we cannot change, add or remove items to a tuple after it has been 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.**
   A data structure (list) with all the vocabulary words (all_vocabulary). Once again I need an ordered sequence in which the internal elements can be modified or deleted.
   A data structure (dictionary) for each word that contains the following keys:
   - word (str): contains the word;
   - definition (list): contains the definitions (str);
   - category (str): contains the category (str).

   For each word I need to associate values with keys to look for them efficiently by key (*e.g.* search by category).