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 all the functionality of the default shell, plus several key advantages. These advantages include code syntax highlighting, automatic indentation of nested statements, and immediate code execution. These make Python code easier to write, read, and execute.

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?
int	Whole number (integer)	Scalar
tuple	Linear, immutable array storing multiple values of any type	Non-scalar
dictionary	Non-linear, mutable, unordered set of items organized as key-value pairs	Non-scalar
float	Decimal number	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.

While there are many similarities between lists and tuples, there are a few important differences. Lists are wrapped by brackets, while tuples are wrapped by parentheses. Lists are mutable, meaning their contents can be modified, but tuples are immutable, so replacing their contents requires reassignment of values or deleting and replacing the values.

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 structure each flash card as a dictionary, storing data in key-value pairs (i.e. {'word': 'bailar', 'definition': 'to dance', 'category': 'verb'}). Dictionaries will keep the data well-organized and modifiable going forward. A user's flashcards would be stored in lists so that they can be modified whenever the user needs to, and flashcards can be added or deleted easily.