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 has many benefits over the default shell that Python has, such as syntax highlighting and automatically indenting text. The syntax highlighting increases readability of the shell thanks to the contrasting colors that the standard shell that does not offer .

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 be used to store different values of any type; Immutable; Defined by ( )	Non-Scalar
Lists	Another ordered sequence in Python; Mutable; Defined by [ ]	Non-Scalar
Strings	Composed of alphanumeric characters; Immutable; Enclosed with either ' ' or " "	Non-Scalar
Dictionaries	Unordered set of items that use key-value pairs, where each key is unique; Mutable; Defined by { }	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.**

   Tuples and lists are both sequential data structures, but they differ in how you can interact with them as a developer. Tuples on one hand are immutable, so they cannot be altered once they're defined so you would need to write a new version of the tuple to change anything. Lists, however, are mutable so, any element inside can be modified or deleted.

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 a language-learning app such as the one described in the question, I feel as though using a dictionary data structure would make the most sense. You can use the word as the key while the definition and category can be stored as a nested dictionary value. This makes it easy to update the dictionary of words or add any extra component to each word since it’ll be stored in its own dictionary. The structure I am picturing for this would be along the lines of:

```
{

‘Word’: {‘Definition’: str, ‘Category’: str},

‘Word’: {‘Definition’: str, ‘Category’: str},

 … }
```