

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?
 - a. The iPython shell is extremely good at speeding up the production of usable code and databases by creating an enhanced REPL(read-eval-print-loop) environment. It's a powerful interface designed for creating a comprehensive environment for interactive and exploratory computing. It's ability to support interactive data visualization and the use of GUI toolkits means that data is easier for everyone to parse, which means fewer mistakes and a better understanding across the team.
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.
 - a. Python hosts a wide variety of data types, such as numerical data types: int & float, sequence data types: list, tuple or range, and types like str & bool, strings and booleans respectively. Sequence data types allow users to create sequences, collections of data that are placed in order, that can be modified, appended and more. Sequence objects like lists are non-scalar, which means they are mutable, or changeable. Numerical data types, as listed above, strings or str and bool or boolean values are scalar. They are immutable and cannot be modified once initialized, they can only be deleted or reassigned a new value.

Data type	Definition	Scalar or Non-Scalar?
Int	This data type represents integers, including both negative and non-negative numbers (from zero to infinity): -5, 0, 7, 25, 1005, etc. (In Python, the int data type has no theoretical limit, apart from the memory available on your system.)	Scalar
Bool	This data type stores either of two values—True or False—and is useful for storing the output of any condition that may be checked. For example, the expression <code>1 < 2</code> would return a boolean value of True, while <code>1 > 2</code> would give you False.	Scalar
String	Strings in Python are an immutable array of characters, generally denoted in Python by str. They can be composed of alphanumeric characters as well as symbols and are surrounded by either single or double quotes	Scalar
Lists	<p>Lists are used to store multiple items in a single variable.</p> <p>Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.</p>	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.
 - a. The primary difference between lists and tuples is that lists are mutable whereas tuples are immutable. This means it is possible to change a list but not a tuple. A tuple cannot be changed once it has been created due to its immutability.

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. I think the best structure for this would be similar to the way I built my recipes. By using a collection of dictionaries on a list, you're able to store the data you want for each individual flashcard, i.e. the type of word, the translation and an explanation and the word itself, then you would be able to add them to a list as you work through them. You could have a check function that allows you to mark them as learned, which could move them to a new list of learned words that you could separate from your learning list.