

Exercise 1.2: Data Types in Python

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 syntax styling that displays various elements in different colors is reason enough, due to the endless number of silly mistakes you make and correct with it, that would otherwise cause you to start over on a given function or line of code. It also has some auto complete suggestions, that it pulls from your personal history in that REPL, that can be very helpful when repeating similar lines of code multiple times.
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.

Data type	Definition	Scalar or Non-Scalar?
int (integers)	Positive and negative whole numbers	Scalar
Tuples	Linear arrays that can store multiple values of any type	Non-Scalar
bool (boolean)	Value defining an expression as either 'True' or 'False'	Scalar
Lists	Ordered sequence with elements that can be modified or deleted	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. Tuples are simple linear arrays that store multiple values of a given type, making them useful for saving many values to consistent sequenced variables, where naming each variable would be unnecessary tedium. While lists are similar to tuples in that they are sequenced, they are mutable, and the internal elements within can be modified, deleted, rearranged, and even insert new elements at will, making them especially useful for scenarios that may eventually change.
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. Dictionaries. I would like to offer the user to be able to learn in both "directions". They could have the flashcards display words, and answer for what they mean (definitions) or for their grammatical category (noun, verb, etc.). They could also have the flashcards display definitions (word meanings), and answer for what word I used for the meaning. As well, due to an app being more dynamic than physical flashcards, they could even have the app's flashcards display a grammatical category (noun, verb, etc.), and provide a word (or even a list of words) that match that category. It is only with dictionaries that I could make these functions work properly due to the key-values that can be attached to each element, as well as the need to store different data types to the varying three element types.