

## Module 2 Summary: Python Data Structures

Congratulations! You have completed this module. At this point, you know that:

- In Python, we often use tuples to group related data together. Tuples refer to ordered and immutable collections of elements.
- Tuples are usually written as comma-separated elements in parentheses "()".
- You can include strings, integers, and floats in tuples and access them using both positive and negative indices.
- You can perform operations such as combining, concatenating, and slicing on tuples.
- Tuples are immutable, so you need to create a new tuple to manipulate it.
- Tuples, termed nesting, can include other tuples of complex data types.
- You can access elements in a nested tuple through indexing.
- Lists in Python contain ordered collections of items that can hold elements of different types and are mutable, allowing for versatile data storage and manipulation.
- A list is an ordered sequence, represented with square brackets "["].
- Lists possess mutability, rendering them akin to tuples.
- A list can contain strings, integers, and floats; you can nest lists within it.
- You can access each element in a list using both positive and negative indexing.
- Concatenating or appending a list will result in the modification of the same list.
- You can perform operations such as adding, deleting, splitting, and so forth on a list.
- You can separate elements in a list using delimiters.
- Aliasing occurs when multiple names refer to the same object.
- You can also clone a list to create another list.
- Dictionaries in Python are key-value pairs that provide a flexible way to store and retrieve data based on unique keys.
- Dictionaries consist of keys and values, both composed of string elements.
- You denote dictionaries using curly brackets.

- The keys necessitate immutability and uniqueness.
- The values may be either immutable or mutable, and they allow duplicates.
- You separate each key-value pair with a comma, and you can use color highlighting to make the key more visible.
- You can assign dictionaries to a variable.
- You use the key as an argument to retrieve the corresponding value.
- You can make additions and deletions to dictionaries.
- You can perform an operation on a dictionary to check the key, which results in a true or false output.
- You can apply methods to obtain a list of keys and values in a dictionary.
- Sets in Python are collections of unique elements, useful for tasks such as removing duplicates and performing set operations like union and intersection. Sets lack order.
- Curly brackets "{}" are helpful for defining elements of a set.
- Sets do not contain duplicate items.
- A list passed through the set function generates a set containing unique elements.
- You use “Set Operations” to perform actions such as adding, removing, and verifying elements in a set.
- You can combine sets using the ampersand "&" operator to obtain the common elements from both sets.
- You can use the Union function to combine two sets, including both the common and unique elements from both sets.
- The sub-set method is used to determine if two or more sets are subsets.

Mark as completed

Like

Dislike

Report an issue