

MODULE 02 - 048: Python - Data Structures

So far in this course, we've discussed a few different data types.

We've discussed strings and numbers and both of those data types are relatively straightforward because there is a one to one kind of analogy between numbers and strings with what you use on a day in and day out basis.

You use numbers when you count, you use strings when you're using letters and paragraphs and sentences.

This next section might be a little bit more abstract because we're going to talk about **collections**.

Now collections as a whole it's not a specific data type but **it's a grouping or a category of data types**.

And so in this section, we're going to talk about **three different data types** in Python:

- Lists
- Dictionaries
- Tuples

If you've never heard of those data types do not worry, I want to take a few steps back and look at some real-world analogies between these types of collections.

And then we're going to get into **the syntax** and this is going to be a pretty long comprehensive section because this is **one of the most critical and fundamental tools** that you're going to learn in this entire course.

Pretty much **every kind of program you're ever going to use in Python is going to use one of these collection-based data types**.

And so **it's very important to get a low-level understanding** and then you can use that and take it and use more advanced kind of concepts once you understand it.

Lists

So, first I want to look at a list if you're coming from another programming language such as say Ruby or Javascript or some language like that **a list is very similar to an array**.

So a list is pretty descriptive. It's simply **a list of items** and so I have a few examples right here on the table.

Imagine the list being like this set of cups.

So we have this set. It is a collection you can see we have a number of them.

And so if we had a counter we could have a length account on how many are here.

We have this one which is in the first possession then we have another one which is in the second and so on and so forth we could sort them we could look at them and see their attributes and it's a way that we can use a full collection of items.

Now it's not a perfect analogy because in a Python list you can actually put any kind of collection of data there.

So that means that even though this looks like a list where all of the items are the same these are all very similar cups in Python you technically could have the first element be a number in the second one could be a string and the third one could be another list.

However, **it's usually a good practice to keep the list of data types as uniform as possible**.

And we're going to walk through all of that in the guides dedicated just to lists and we're going to walk through all kinds of functions you can use on them.

But right now I want to give you just a very high-level idea of what they are it's a **collection of data** and it is simply like it says **it's a list of items**.

Dictionaries

The second type of data structure we're going to talk about is the dictionary data type now a dictionary is a collection just like a list is however it behaves much differently.

And for that, I brought an actual dictionary here so if you've ever used a dictionary before then you would know as you look through the dictionary you're going to find different keys.

So you're going to say right now I look at this page and look up Caterina and Caterina is right there.

And if I were to read it it would describe the value that Caterina has in the dictionary and if I flipped to another spot there is one called Parker and then if I read the full description that is the value that Parker has.

A **dictionary** in python works exactly the same way a dictionary has **a set of keys and then it has associated values for each one of those keys**.

So you can think of it the same way that you think about a real-world dictionary.

So whereas a list is simply a collection of items imagine a collection of usernames that you just want to loop through a dictionary represents more of a key-value kind of set of attributes.

So imagine an actual user where they have a full name and email address any metadata associated with that one user as each one has a key and then a value that is associated with that.

I know this is still a very high level. I'm simply wanting to give you an overview and a real-world analogy for what each one of these data types represents.

Tuples

The third one we're going to discuss is called a tuple.

Now, **a tuple is very similar to lists**.

Once again our stack of cups right here a tuple is simply a collection of items and a **tuple has a few differentiating characteristics** that will walkthrough.

But when it comes to understanding the way **how it's structured** you can think of it as a list.

A similar set of items that you are going to be able to loop through and work with.

Very different than the key-value pair dictionary that you are going to have whenever you're working with the dictionary data type.

So now that you have that high-level overview and understanding.

Now we're going to get into the fun stuff and we're going to start with the list data structure.