

Lists

Let's learn about Lists and their usage in Python

WE'LL COVER THE FOLLOWING ^

- Creating a list
- Combining two lists
- Sort a list
- Slice a list

Creating a list

A Python list is similar to an array in other languages. In Python, an empty list can be created in the following ways.

```
my_list = []  
my_list = list()
```

As you can see, you can create the list using square brackets or by using the Python built-in, **list**. A list contains a list of elements, such as strings, integers, objects or a mixture of types. Let's take a look at some examples:

```
my_list = [1, 2, 3]  
my_list2 = ["a", "b", "c"]  
my_list3 = ["a", 1, "Python", 5]
```

The first list has 3 integers, the second has 3 strings and the third has a mixture. You can also create lists of lists like this:

```
my_list = [1, 2, 3]  
my_list2 = ["a", "b", "c"]  
  
my_nested_list = [my_list, my_list2]  
print(my_nested_list) # [[1, 2, 3], ['a', 'b', 'c']]
```





Combining two lists

Occasionally, you'll want to combine two lists together. The first way is to use the extend method:

```
combo_list = []  
one_list = [4, 5]  
combo_list.extend(one_list)  
print(combo_list) # [4, 5]
```



A slightly easier way is to just add two lists together (*yes, it really is that easy*).

```
my_list = [1, 2, 3]  
my_list2 = ["a", "b", "c"]  
  
combo_list = my_list + my_list2  
print(combo_list) # [1, 2, 3, 'a', 'b', 'c']
```



Sort a list

You can also sort a list. Let's spend a moment to see how to do that:

```
alpha_list = [34, 23, 67, 100, 88, 2]  
alpha_list.sort()  
print(alpha_list) # [2, 23, 34, 67, 88, 100]
```



Now there is a got-cha above. Can you see it? Let's do one more example to make it obvious:

```
alpha_list = [34, 23, 67, 100, 88, 2]  
sorted_list = alpha_list.sort()  
print(sorted_list)
```





In this example, we try to assign the sorted list to a variable. However, when you call the **sort()** method on a list, it sorts the list in-place. So if you try to assign the result to another variable, then you'll find out that you'll get a None object, which is like a Null in other languages. Thus when you want to sort something, just remember that you sort them in-place and you cannot assign it to a different variable.

Slice a list

You can slice a list just like you do with a string:

```
alpha_list = [34, 23, 67, 100, 88, 2]
alpha_list.sort()
print(alpha_list[0:3])
# [2, 23, 34]
```



This code returns a list of just the first 3 elements.