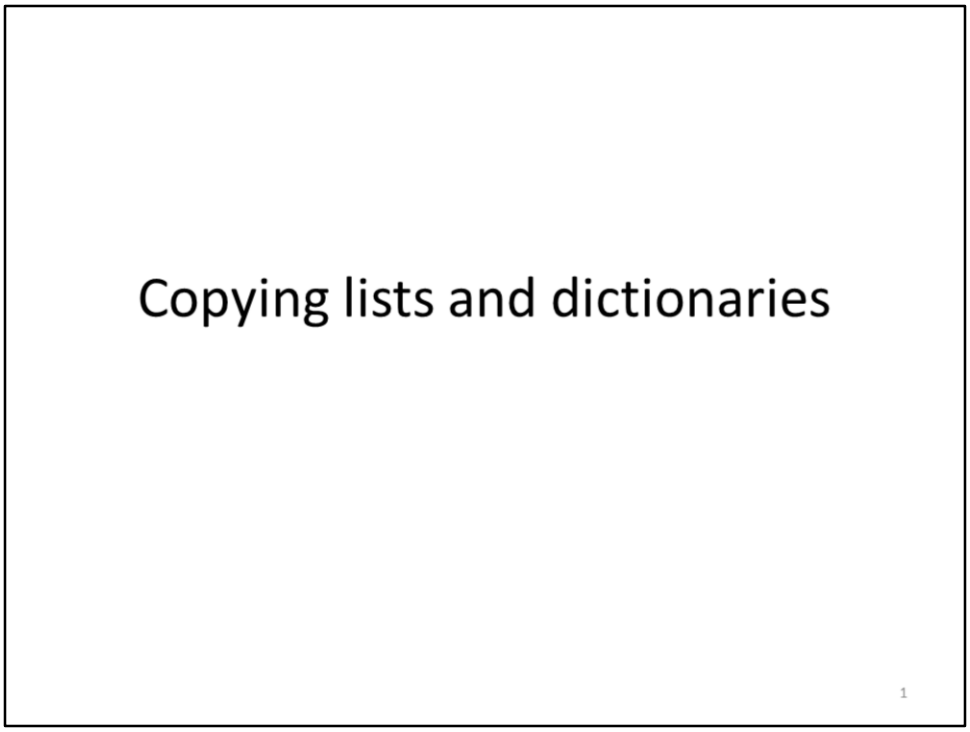


This video will discuss how to copy lists and dictionaries.

Copying lists / dictionaries

- It is occasionally useful to copy a list or a dictionary
- Assigning a new variable does not create a copy, it just assigns another variable to the same list or dictionary.

```
>>> List1 = [1,2,3,4]
>>> List2 = List1
>>> del List2[0]
>>> List1
[2,3,4]
```

modifying List2 also modifies List1 – List1 and List2 refer to the same list

2

Copying lists or dictionaries can occasionally be useful; however, the correct method for copying a list is not particularly obvious. For example, assigning a new variable to an existing list does not copy the list – the new variable just refers to the original list.

In this example, both the List1 and List2 variable refer to the same list. Modifying List2 also modifies List1.

Copying lists / dictionaries – "superficial" copy

- To copy a list or dictionary that has 1-level (i.e. contains no sub-lists or sub-dictionaries)…

```
>>> import copy
>>> List1 = [1,2,3,4]
>>> List2 = copy.copy(List1)
>>> del List2[0]
>>> List1
[1,2,3,4]
```

modifying List2 does not affect List1 – List2 is an independent list

- Note: this method is not suitable for a list that contains sublists or a dictionary that contains nested lists or dictionaries.

3

The copy module can be used to create a superficial copy of a list or a dictionary. This method will only create a true copy for a 1-level list.

In the example, the copy method creates a true copy of List1. Modifying List2 does not affect List1.

If a list or dictionary contains multiple levels, then the sub-lists and sub-dictionaries will not be true copies – modifying the "copies" will also modify the originals.

Copying lists / dictionaries – "deep" copy

- To copy a list or dictionaries that has 1 or more levels (i.e. contains sub-lists or sub-dictionaries)…

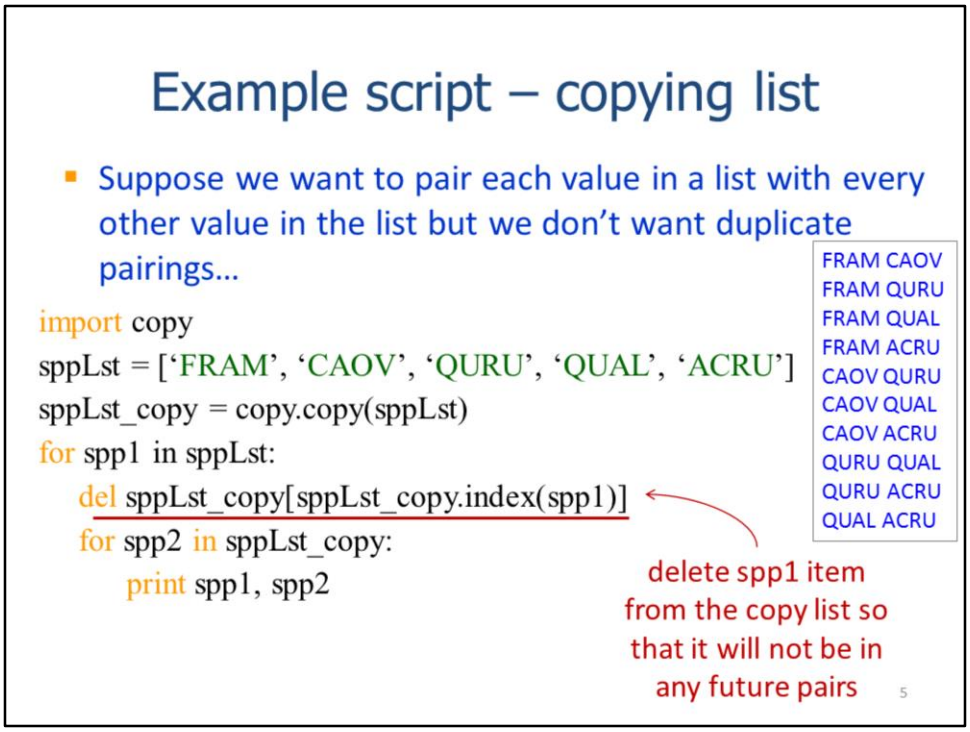
```
>>> import copy
>>> List1 = [[1,2], [6,7]]
>>> List2 = copy.deepcopy(List1)
>>> del List2[0][-1]
>>> List1
[[1,2] ,[6, 7]]
```

modifying sublist of List2 does not affect List1 – List2 and all sublists are independent of List1

4

The **deepcopy** method will create a true copy for a multi-level list or dictionary.

In this example, all elements of List1 and List2 are independent – modifying a sub-list in List2 does not affect the sub-list in List1.



This slide will show an example of a script that uses a copied list. In this example, we want to pair each item in the list with every other item but we don't want to allow any duplicate pairings.

The list is a single level list so we'll use the **copy** method to copy it.

We'll create a 2-level for loop – the top-level iterates through the original list.

At the beginning of the top-level loop we'll delete spp1 from the copied list. This eliminate redundant pairings.

The 2nd level of the loop iterates through the copy of the list.

The result of the script has all possible unique combinations of the list items.