17.6. Deep and Shallow Copies

Earlier when we discussed cloning and aliasing lists we had mentioned that simply cloning a list using [:] would take care of any issues with having two lists unintentionally connected to each other. That was definitely true for making shallow copies (copying a list at the highest level), but as we get into nested data, and nested lists in particular, the rules become a bit more complicated. We can have second-level aliasing in these cases, which means we need to make deep copies.

When you copy a nested list, you do not also get copies of the internal lists. This means that if you perform a mutation operation on one of the original sublists, the copied version will also change. We can see this happen in the following nested list, which only has two levels.

Save & Run   Original - 1 of 1   Show in CodeLens

```
original = [['dogs', 'puppies'], ['cats', "kittens"]]
copied_version = original[:]
print(copied_version)
print(copied_version is original)
print(copied_version == original)
original[0].append(["canines"])
print(original)
print("-------- Now look at the copied version -----------")
print(copied_version)

```

```
[['dogs', 'puppies'], ['cats', 'kittens']]
False
True
[['dogs', 'puppies', ['canines']], ['cats', 'kittens']]
-------- Now look at the copied version -----------
[['dogs', 'puppies', ['canines']], ['cats', 'kittens']]
```

17.5. Structuring Nested Data"> (WPStructuringNestedData.html)
17.7. Extracting from Nested Data"> (WPExtractFromNestedData.html)
Structuring Nested Data"> (WPStructuringNestedData.html) (ac17_100_1)
17.7. Extracting from

Assuming that you don’t want to have aliased lists inside of your nested list, then you’ll need to perform nested iteration.

Save & Run

Original - 1 of 1

Show in CodeLens

```
original = [['dogs', 'puppies'], ['cats', "kittens"]]
copied_outer_list = []
for inner_list in original:
    copied_inner_list = []
    for item in inner_list:
        copied_inner_list.append(item)
    copied_outer_list.append(copied_inner_list)
print(copied_outer_list)
original[0].append(["canines"])
print(original)
print("-------- Now look at the copied version -----------")
print(copied_outer_list)

```

```
[['dogs', 'puppies'], ['cats', 'kittens']]
[['dogs', 'puppies', ['canines']], ['cats', 'kittens']]
-------- Now look at the copied version -----------
[['dogs', 'puppies'], ['cats', 'kittens']]
```

Activity: 2 -- ActiveCode (ac17_100_2)

Or, equivalently, you could take advantage of the slice operator to do the copying of the inner list.

Save & Run

Original - 1 of 1

Show in CodeLens

```
original = [['dogs', 'puppies'], ['cats', "kittens"]]
copied_outer_list = []
for inner_list in original:
    copied_inner_list = inner_list[:]
    copied_outer_list.append(copied_inner_list)
print(copied_outer_list)
```

17.5. 🧑‍💻 Structuring Nested Data">
(WPStructuringNestedData.html)
17.7. 🧑‍💻 Extracting from Nested Data">
(WPExtractFromNestedData.html)
Structuring Nested Data">(WPStructuringNestedData.html)
17.7. 🧑‍💻 Extractin from

```
original[0].append(["canines"])
print(original)
print("-------- Now look at the copied version -----------")
print(copied_outer_list)

```

```
[['dogs', 'puppies'], ['cats', 'kittens']]
[['dogs', 'puppies', ['canines']], ['cats', 'kittens']]
-------- Now look at the copied version -----------
[['dogs', 'puppies'], ['cats', 'kittens']]
```

Activity: 3 -- ActiveCode (ac17_100_2a)

This process above works fine when there are only two layers or levels in a nested list. However, if we want to make a copy of a nested list that has *more* than two levels, then we recommend using the `copy` module. In the `copy` module there is a method called `deepcopy` that will take care of the operation for you.

Save & Run   Original - 1 of 1   Show in CodeLens

```
import copy
original = [['canines', ['dogs', 'puppies']], ['felines', ['cats',
shallow_copy_version = original[:]
deeply_copied_version = copy.deepcopy(original)
original.append("Hi there")
original[0].append(["marsupials"])
print("-------- Original -----------")
print(original)
print("-------- deep copy -----------")
print(deeply_copied_version)
print("-------- shallow copy -----------")
print(shallow_copy_version)

```

17.5. Structuring Nested Data">
(WPStructuringNestedData.html)

17.7. Extracting from Nested Data">
(WPExtractFromNestedData.html)

Structuring Nested Data"> (WPStructuringNestedData.html)

17.7. Extracting from

```
-------- Original -----------
[['canines', ['dogs', 'puppies'], ['marsupials']], ['felines', ['cats', 'kitten
s']], 'Hi there']
-------- deep copy -----------
[['canines', ['dogs', 'puppies']], ['felines', ['cats', 'kittens']]]
-------- shallow copy -----------
[['canines', ['dogs', 'puppies'], ['marsupials']], ['felines', ['cats', 'kitten
s']]]
```

Activity: 4 -- ActiveCode (ac17_100_3)

You have attempted 5 of 4 activities on this page

✔ Completed. Well Done!

| Back to top

17.5. Structuring Nested Data">

‹

(WPStructuringNestedData.html)

17.7. Extracting from Nested Data">

›

(WPExtractFromNestedData.html)

Structuring Nested Data"> (WPStructuringNestedData.html)

17.7. Extracting from