



17.5. 🧱 Structuring Nested Data

When constructing your own nested data, it is a good idea to keep the structure consistent across each level. For example, if you have a list of dictionaries, then each dictionary should have the same structure, meaning the same keys and the same type of value associated with a particular key in all the dictionaries. The reason for this is because any deviation in the structure that is used will require extra code to handle those special cases. The more the structure deviates, the more you will have to use special cases.

For example, let's reconsider this nested iteration, but suppose not all the items in the outer list are lists.

Save & Run

Original - 1 of 1

Show in CodeLens

```
1 nested1 = [1, 2, ['a', 'b', 'c'], ['d', 'e'], ['f', 'g', 'h']]
2 for x in nested1:
3     print("level1: ")
4     for y in x:
5         print("    level2: {}".format(y))
6
```

level1:

Activity: 1 -- ActiveCode (ac17_50_1)

Error

TypeError: 'int' object is not iterable on line 4

Description

Type errors most often occur when an expression tries to combine two objects with types that should not be combined. Like raising a string to a power

To Fix

To fix a type error you will most likely need to trace through your code and make sure the variables have the types you expect them to have. It may be helpful to print out each variable along the way to be sure its value is what you think it should be.

Now the nested iteration fails.

We can solve this with special casing, a conditional that checks the type.

Save & Run

Original - 1 of 1

Show in CodeLens

```
1 nested1 = [1, 2, ['a', 'b', 'c'], ['d', 'e'], ['f', 'g', 'h']]
2 for x in nested1:
3     print("level1: ")
4     if type(x) is list:
5         for y in x:
6             print("    level2: {}".format(y))
7     else:
8         print(x)
9
```

level1:
1
level1:
2
level1:
 level2: a
 level2: b
 level2: c
level1:
 level2: d
 level2: e
level1:
 level2: f
 level2: g
 level2: h

Activity: 2 -- ActiveCode (ac17_50_2)

You can imagine how many special case if-thens we'd need, and how complicated the code would get, if we had many layers of nesting but not always a consistent structure.

You have attempted 3 of 2 activities on this page



17.4. Nested Iteration">

17.6. Deep and Shallow Copies">

17.4. Nested Iteration">

✓ Completed. Well Done! ➡

17.6. Deep and Shallow Copies">Next Section - 17.6. Deep and Shallow Copies