



17.4. Nested Iteration

When you have nested data structures, especially lists and/or dictionaries, you will frequently need nested for loops to traverse them.

Save & Run

Original - 1 of 1

Show in CodeLens

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

```
level1:
  level2: a
  level2: b
  level2: c
level1:
  level2: d
  level2: e
level1:
  level2: f
  level2: g
  level2: h
```

Activity: 1 -- ActiveCode (ac17_4_1)

Line 3 executes once for each top-level list, three times in all. With each sub-list, line 5 executes once for each item in the sub-list. Try stepping through it in CodeLens to make sure you understand what the nested iteration does.

Python 3.3

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

<< First < Back Program terminated Forward > Last >>

⇒ line that has just executed
→ next line to execute

Visualized using Online Python Tutor by Philip Guo

Frames

Objects

```
graph LR
    subgraph Frames
        GF[Global frame]
        GF -- nested1 --> L1
        GF -- x --> L2
        GF -- y --> L3
    end
    subgraph Objects
        L1["list: ['a', 'b', 'c']"]
        L2["list: ['d', 'e']"]
        L3["list: ['f', 'g', 'h']"]
    end
    L1 -- 0 --> a["a"]
    L1 -- 1 --> b["b"]
    L1 -- 2 --> c["c"]
    L2 -- 0 --> d["d"]
    L2 -- 1 --> e["e"]
    L3 -- 0 --> f["f"]
    L3 -- 1 --> g["g"]
    L3 -- 2 --> h["h"]
```

Program output:

```
level2: a
level2: b
level2: c
level1:
level2: d
level2: e
level1:
level2: f
level2: g
level2: h
```

Activity: 2 -- CodeLens: (clens17_4_1)

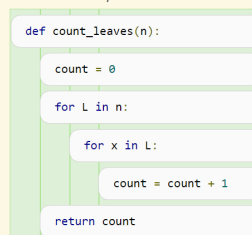
Check Your Understanding

nested-3-1: Now try rearranging these code fragments to make a function that counts all the "leaf" items in a nested list like nested1 above, the items at the lowest level of nesting (8 of them in nested1).

Drag from here



Drop blocks here



Check Reset

Perfect! It took you 5 tries to solve this. Click Reset to try to solve it in one attempt.

Activity: 3 -- Parsons (pp17_4_1)

2. Below, we have provided a list of lists that contain information about people. Write code to create a new list that contains every person's last name, and save that list as `last_names`.

Save & Run

5/14/2021, 9:34:32 PM - 2 of 2

Show in CodeLens

```
1
2 info = [['Tina', 'Turner', 1939, 'singer'], ['Matt', 'Damon', 1970, 'actor'], ['Kri
3 last_names = []
4 for entertainer in info:
5     last_names.append(entertainer[1])
6     print(last_names)
7
```

```
['Turner']
['Turner', 'Damon']
['Turner', 'Damon', 'Wiig']
['Turner', 'Damon', 'Wiig', 'Phelps']
['Turner', 'Damon', 'Wiig', 'Phelps', 'Obama']
```

Activity: 4 -- ActiveCode (ac17_4_2)

Result	Actual Value	Expected Value	Notes
Pass	['Tur...ama']	['Tur...ama']	Testing that last_names was created correctly.

Expand Differences

You passed: 100.0% of the tests

3. Below, we have provided a list of lists named `L`. Use nested iteration to save every string containing "b" into a new list named `b_strings`.

Save & Run

5/14/2021, 9:35:26 PM - 2 of 2

Show in CodeLens

```
1
2 L = [['apples', 'bananas', 'oranges', 'blueberries', 'lemons'], ['carrots', 'peas',
3 b_strings = []
4 for lst in L:
5     for word in lst:
6         if 'b' in word:
7             b_strings.append(word)
8 print(b_strings)
9
```

```
['bananas', 'blueberries', 'cucumbers', 'green beans', 'root beer', 'cranberry juice']
```

Activity: 5 -- ActiveCode (ac17_4_3)

Result	Actual Value	Expected Value	Notes
Pass	['ban...ice']	['ban...ice']	Testing that b_strings was created correctly.

Expand Differences

You passed: 100.0% of the tests

You have attempted 6 of 5 activities on this page

17.3. Processing JSON results">

17.5. Structuring Nested Data">

Processing JSON results">

Completed. Well Done!

17.5. Structuring Nested Data">Next Section - 17.5. Structuring Nested Data