6.7. Concatenation and Repetition

Again, as with strings, the `+` operator concatenates lists. Similarly, the `*` operator repeats the items in a list a given number of times.

Save & Run 4/30/2021, 7:32:24 PM - 2 of 2 Show in CodeLens

```
fruit = ["apple","orange","banana","cherry"]
print([1,2] + [3,4])
print(fruit+[6,7,8,9])

print([0] * 4)

```

```
[1, 2, 3, 4]
['apple', 'orange', 'banana', 'cherry', 6, 7, 8, 9]
[0, 0, 0, 0]
```

Activity: 1 -- ActiveCode (ac5_7_1)

It is important to see that these operators create new lists from the elements of the operand lists. If you concatenate a list with 2 items and a list with 4 items, you will get a new list with 6 items (not a list with two sublists). Similarly, repetition of a list of 2 items 4 times will give a list with 8 items.

One way for us to make this more clear is to run a part of this example in codelens. As you step through the code, you will see the variables being created and the lists that they refer to. Pay particular attention to the fact that when `newlist` is created by the statement `newlist = fruit + numlist`, it refers to a completely new list formed by making copies of the items from `fruit` and `numlist`. You can see this very clearly in the codelens object diagram. The objects are different.

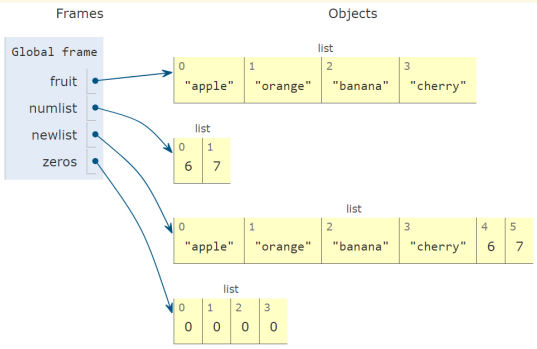
Python 3.3

```
1 fruit = ["apple","orange","banana","cherry"]
2 numlist = [6,7]
3
4 newlist = fruit + numlist
5
6 zeros = [0] * 4
```

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

line that has just executed
next line to execute

Visualized using Online Python Tutor by Philip Guo



Program output:

Activity: 2 -- CodeLens: (clens5_7_1)

Note

WP: Adding types together

Beware when adding different types together! Python doesn't understand how to concatenate different types together. Thus, if we try to add a string to a list with `['first'] + "second"` then the interpreter will return an error. To do this you'll need to make the two objects the same type. In this case, it means putting the string into its own list and then adding the two together like so: `['first'] + ["second"]`. This process will look different for other types though. Remember that there are functions to convert types!

Check your understanding

sequences-7-1: What is printed by the following statements?

```
alist = [1,3,5]
blist = [2,4,6]
print(alist + blist)
```

A. 6
B. [1,2,3,4,5,6]
C. [1,3,5,2,4,6]
D. [3,7,11]

Check me | Compare me

✔ Yes, a new list with all the items of the first list followed by all those from the second.

Activity: 3 -- Multiple Choice (question5_7_1)

sequences-7-2: What is printed by the following statements?

```
alist = [1,3,5]
print(alist * 3)
```

A. 9
B. [1,1,1,3,3,3,5,5,5]
C. [1,3,5,1,3,5,1,3,5]
D. [3,9,15]

Check me | Compare me

✔ Yes, the items of the list are repeated 3 times, one after another.

Activity: 4 -- Multiple Choice (question5_7_2)

You have attempted 5 of 4 activities on this page

6.6. The Slice Operator">
Slice Operator">
6.8. Count and Index">

✔ Completed. Well Done!

| Back to top