

ASSIGNMENT-4 LIST

1. What exactly is []?

Ans: It is list.

2. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)

Ans: `spam[2] = "hello"`

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

3. What is the value of `spam[int(int('3' * 2) / 11)]`?

Ans: 'd'

4. What is the value of `spam[-1]`?

Ans: 'd'

5. What is the value of `spam[:2]`?

Ans: ['a', 'b']

Let's pretend bacon has the list [3.14, 'cat', 11, 'cat', True] for the next three questions.

6. What is the value of `bacon.index('cat')`?

Ans: 1

7. How does `bacon.append(99)` change the look of the list value in bacon?

Ans: [3.14, 'cat', 11, 'cat', True, 99]

8. How does `bacon.remove('cat')` change the look of the list in bacon?

Ans: [3.14, 11, 'cat', True, 99]

9. What are the list concatenation and list replication operators?

list concatenation operator: +

list replication operator: *

10. What is difference between the list methods `append()` and `insert()`?

Ans: With `insert(position, new_entry)` we can create a new entry exactly in the position we want. The `append` method adds a new item to the end of a list.

11. What are the two methods for removing items from a list?

Ans: `remove()` and `pop()`

12. Describe how list values and string values are identical.

Ans: Both are Ordered Sequential Collection. Slicing method works similar in list and string.

13. What's the difference between tuples and lists?

Ans: Tuples are unchangeable.

We can't insert new elements in tuple, after its object creation. Where as in list, we can insert new elements after list object creation.

14. How do you type a tuple value that only contains the integer 42?

Ans: mytuple = (42,)

15. How do you get a list value's tuple form? How do you get a tuple value's list form?

Ans: Suppose, t is tuple.

tuple value's list form: a=list(t)

list value's tuple form: b=tuple(a)

16. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?

Ans: String

17. How do you distinguish between copy.copy() and copy.deepcopy()?

Ans: When we use **copy.copy()** change in any nested objects in old_list, than that changes appears in new_list. While, in **copy.deepcopy()**, if we make changes to any nested objects in original object old_list, we can see no changes to the copy new_list.