1. What exactly is []?

[] defines a list placing elements inside the square brackets

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.)

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)]?

‘d’

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

‘d’

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

[‘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')?

1

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

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

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

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

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

Concatenation: +, Replication: \*

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

append() add the data at the end, but insert() add the data at the specific position

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

pop(), remove()

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

String values are immutable but list values are mutable

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

Tuples are immutable and static where lists are mutable and dynamic

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

(42,)

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

tuple(list), list(tuple)

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

references

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

copy() copies the values not the references so, both the two list can behave independently. But deepcopy() copies with the inner references so, if one list variable changes it affects on both the variables as both the list are referencing the same list.