1. What exactly is []?

Ans : [] Represent List

[] is the empty list or if you can add an element you can add it

1. 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, 4, 6, 8, 10]

spam[2]='hello'

print(spam)

output: [2, 4, 'hello', 8, 10]

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 : spam=['a', 'b', 'c', 'd']

print(spam[int(int('3' \* 2) / 11)])

output : 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]

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

Ans :

Concatenation :-

It is done between the same data type only, and its is done by + keyword

Here we are concatenating only same(string) data types

s1="Welcome"

s2="to"

s3="ineuron"

s4=s1+s2+s3

print(s4)

output: Welcometoineuron

Replication:-

Make multiple copies of values or objects. Use \* keyword for replicating

s1="ineuron"

print (s1\*3)

output: ineuronineuronineuron

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

Ans : append() add element at the end of the list

spam=[3.14, 'cat', 11, 'cat' ,True]

spam.append(99)

print(spam)

insert():- add the element at the specified position by passing index and element

spam=[3.14, 'cat', 11, 'cat' ,True]

spam.insert(4,99)

print(spam)

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

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

spam.remove(‘cat’)

spam.pop(‘cat’)

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

def check(list):

    return all(i == list[0] for i in list)

print(check(['a', 'b', 'c']))

print(check([1, 1, 1]))

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

|  |  |
| --- | --- |
| LIST | TUPLE |
| List are mutable | Tuple is immutable |
| Consume more memory | Consume less memory |
| Slow as compared to tuple | Fast as compared to a list |
| Has several built-in values | Few built-in value as compared to a list |

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

spam=(42)

print(spam)

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

We just use “tuple” keyword to convert list to tuple

spam=[3.14, 'cat', 11, 'cat' ,True]

print(tuple(spam))

we just use “list” keyword to convert to list from tuple

spam=[3.14, 'cat', 11, 'cat' ,True]

print(list(spam))

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

copy.copy(): -

* A shallow copy creates a new object which stores the reference of the original elements.
* The changes made in the copied object also reflect the original object.
* Shallow copy is faster than Deep copy.

Copy.deepcopy():-

* A deep copy creates a new object and recursively adds the copies of nested objects present in the original elements.
* There is no reflection on the original object when the changes are made in the copied object.
* Deep copy is slower than Shallow copy.