Python Assignment 4

1. What exactly is []?

Answer:

[] represents empty list with no elements.

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.)

Answer:

Code for assigning ‘hello’ as third value.

spam = [2,4,6,8,10]

spam[2] = ‘hello’

print(spam)

It displays 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)]?

Answer:

‘3’*2= 33, int(33) converts string into integer

33/11 = 3

spam[int(int('3' * 2) / 11)] reduces to spam[3] and as indexing starts from 0 in python so spam[3] is ‘d’.

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

Answer:

The value of spam[-1] is ‘d’.

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

Answer:

The value of spam[:2] is a list containing elements from index 0 to 2-1 that is [“a”,”b”].

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

Answer:

The value of bacon.index(“cat”) is 1 as indexing start from 0 and the first occurrence is at 1 index.

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

Answer:

Append function adds the value at the end of the list therefore the value of bacon.append(99)

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

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

Answer:

Remove method in python removes the first occurrence of the value in the given list therefore the value of bacon.remove(‘cat ’) is: [3.14, 11, 'cat', True]

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

Answer:

List concatenation “ + “ combines two or more list to generate a new list, it does not modify the original list.

Example: list1 = [‘1’, ’2’, ’3’]

list2 = [‘a’, ‘b’, ‘c’]

list3 = list1 + list2

list3 is now ['1', '2', '3', 'a', 'b', 'c']

List Replication “ * ” is replicating a list or multiplying the list specified number of times. It does not modify the original list.

Example: list1 = [‘1’, ’2’, ’3’]

```
list2 = list1 *3
```

list2 is now: ['1', '2', '3', '1', '2', '3', '1', '2', '3']

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

Answer:

Append() and insert() are methods to add values in an list but append() adds element in end of the list and it takes only one argument that is the value that is to be added in the list whereas insert is used to add element in the list at an specific index it takes two inputs, one value and other the index where we want to add value.

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

Answer:

Methods to remove items in python that commonly used are remove() and pop()

remove() removes the first occurrence of the value in the string, it only takes one parameter that is the value to be removed.

Example list1 = ['1','2','3','4','5']

```
list1.remove('3')

print(list1)
```

list1() becomes ['1', '2', '4', '5']

pop() removes the value at the index, it takes only one parameter that the index at which we want to remove the value.

Example: list1 = ['1','2','3','4','5']

```
list1.pop(3)

print(list1)
```

list1 becomes ['1', '2', '3', '5'].

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

Answer:

Both list and string have len() function to calculate the length have indexes, both can be used in for loops, both can be concatenated and replicated.

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

Answer:

Key difference between tuples and lists is that tuples are immutable which means they cannot be changed whereas list are mutable means they can be modified.

Tuples are represented in () brackets whereas list are represented in [] brackets.

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

Answer:

To create a tuple containing only the integer 42 we write in this format:

tuple(42,)

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

Answer:

We use method tuple() to convert list into tuple and vice versa use list() method to convert tuple into list.

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

Answer:

They contain references/ pointers to list objects.

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

Answer:

Both copy.copy() and copy.deepcopy() are modules that are used to create copies in python. Copy.copy() creates a shallow copy whereas copy.deepcopy() creates a deep copy. Changes made using copy.copy() reflect in original list whereas copy.deepcopy() does not change the original list.