1. What exactly is []?

Soloution: The empty list value, which is a list value that contains no items. This is similar to how '' is the empty string value.

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

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

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

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

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

```
'd'
```

```
## Q4. What is the value of spam[-1]?
spam[-1]
# Solution:d
```

```
'd'
```

```
## 5. What is the value of spam[:2]?
spam[:2]
# Solution: ['a', 'b']
```

```
['a', 'b']
```

```
bacon = [3.14, 'cat', 11, 'cat', True]
```

```
## Q. 6. What is the value of bacon.index('cat')?
bacon.index('cat')
# Solution: 1
```

```
     1
```

```
## Q 7. How does bacon.append(99) change the look of the list value in bacon?
bacon.append(99)
bacon
## Solution: [3.14, 'cat', 11, 'cat', True, 99]
```

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

```
## Q 8. How does bacon.remove('cat') change the look of the list in bacon?
bacon.remove('cat')
bacon
## Solution: [3.14, 11, True, 99]
```

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

```
## Q.9 What are the list concatenation and list replication operators?
## Solution : The operator for list concatenation is +, while the operator for replica
print([23,3,4,5,6,7]+[33,4,5,6,7,89])
[23,4,5,6,7,8]*3
```

```
     [23, 3, 4, 5, 6, 7, 33, 4, 5, 6, 7, 89]
     [23, 4, 5, 6, 7, 8, 23, 4, 5, 6, 7, 8, 23, 4, 5, 6, 7, 8]
```

```
## Q 10. What is difference between the list methods append() and insert()?
## Solution: While append() will add values only to the end of a list, insert() can ad
list = [23,4,5,7,8,90]
list.append("Narender")
print(list)
list.insert(3,"beniwal")
print(list)
```

```
     [23, 4, 5, 7, 8, 90, 'Narender']
     [23, 4, 5, 'beniwal', 7, 8, 90, 'Narender']
```

```
## Q.11 What are the two methods for removing items from a list?
## Solution: The methods are remove(), pop() and clear(). It helps to remove the very
# The pop() method removes an element from the list based on the index given. The clea
list.pop(2)
list
```

```
     [23, 4, 'beniwal', 7, 8, 90, 'Narender']
```

```
## Q 12. Describe how list values and string values are identical.
## Solution : Both lists and strings can be passed to len(), have indexes and slices,
len(list)
len("Narendeer")
```

```
9
```

```
## Q. 13. What's the difference between tuples and lists?
## Solution: Lists are mutable; they can have values added, removed, or changed.
#. Tuples are immutable; they cannot be changed at all.
#  Also, tuples are written using parentheses, ( and ), while lists use the square bra
```

```
## Q. 14. How do you type a tuple value that only contains the integer 42?
# Solution: (42,) (The trailing comma is mandatory.)
```

```
## Q. 15. How do you get a list value's tuple form? How do you get a tuple value's lis
# Solution: The tuple() and list() functions, respectively
print(tuple(list))
```

```
(23, 4, 'beniwal', 7, 8, 90, 'Narender')
```

```
## Q 16. Variables that "contain" list values are not necessarily lists themselves. In
## Solution: They contain references to list values.
```

```
## Q 17. How do you distinguish between copy.copy() and copy.deepcopy()?
## The copy.copy() function will do a shallow copy of a list, while the copy.deepcopy(
# That is, only copy.deepcopy() will duplicate any lists inside the list.
```

✓ 0s completed at 9:21 AM ● ✕