**1. What exactly is []?**

It is an empty 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.)**

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

spam.insert( 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)]?**

**The answer is ‘d’**

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

**The answer is ‘d’**

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

Answer is ['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')?**

**The answer is 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?**

The operator for list **concatenation** is **+**, while the operator for **replication** is **\***.

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

|  |  |
| --- | --- |
| append() | Insert() |
| Append() always add the given value into the list at the end | Insert() includes the given value at the required index and insert would take 2 parameters while passing , one for index and other the value |
| Ex : spam =[2, 4, 6, 8, 10]  spam.append( "hello")  Output is [2, 4, 6, 8, 10,”hello”] | Ex : spam =[2, 4, 6, 8, 10]  spam.insert(2, "hello")  Output is [2, 4, ,”hello”, 6, 8, 10] |

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

remove() and pop() are the two methods to remove items from the list

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

List values and Strings are identical in a way both are sequential collections of characters where each value is identified by an index

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

|  |  |
| --- | --- |
| Tuples | Lists |
| Tuples are immutable and cannot be modified | Tuples are mutable and can be modified |

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

tuple = (42,)

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

|  |  |
| --- | --- |
| Tuple to List | List to tuple |
| aTuple = (True, 28, 'Tiger')  aList = list(aTuple)  print(aList)  Output 🡪 [True, 28, 'Tiger']  print(type(aList))  output - list | aList = [True, 28, 'Tiger']  aTuple =tuple(aList)  print(aTuple)  Output 🡪 (True, 28, 'Tiger')  print(type(aTuple))  output - tuple |

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

Variables **contain references** to list values rather than list values themselves. But for strings and integer values, variables simply contain the string or integer value.

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

The difference between shallow and deep copying is only relevant for compound objects (objects that contain other objects, like lists or class instances):

* A shallow copy constructs a new compound object and then (to the extent possible) inserts references into it to the objects found in the original.
* A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original.