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

**Ans**: 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.)**

**Ans:** 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)]?**

**Ans: '**a**'**

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

**Ans: '**d**'**

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

**Ans: '**c**'**

**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,99]

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

**Ans:**

* List concatenation: +
* List replication: \*

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

**Ans:**

* append() 🡪 will add element at end of list
* insert(index,elem) 🡪 will add element at a particular index

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

**Ans:**

1. List.remove(elem)
2. List.pop()

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

**Ans:** Both can be accessed through indexing

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

**Ans:** tuple are immutable, lists are mutable

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

**Ans:** (42,)

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

**Ans:** Let a list l = [1,2,3] and a tuple t = ('a', 'b', 'c')

* List from tuple: List(t) >> ['a', 'b', 'c']
* Tuple from list: tuple(l) >> (1,2,3)

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

**Ans:** It contains memory address of the list

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

**Ans:**

* copy.copy() copies the refrence of object and assign it to new variable any change in original will change the duplicate object also
* copy.deepcopy() creates a new object and assign it to new variable any change in original will have no change in the duplicate object.