1. What exactly is []?

It used create a list. If nothing is there in it. It is called as empty list.

In [2]:
```
# Ex:
a = []
type(a)
```

Out[2]:
```
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.)

In [4]:
```
spam = [2, 4, 6, 8, 10]
for i in range(len(spam)):
    if i == 2:
        spam[i] = "hello"

spam
```

Out[4]:
```
[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)]?

In [5]:
```
spam = ['a', 'b', 'c', 'd']
spam[int(int('3' * 2)/11)]
```

Out[5]:
```
'd'
```

In [8]:
```
int(int('3' * 2) / 11)
```

Out[8]:
```
3
```

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

In [9]:
```
spam[-1] # Gives the last value in the list.
```

Out[9]:
```
'd'
```

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

In [10]:
```
spam[:2]
```

```
Out[10]: ['a', 'b']
```

Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.

```
In [12]: bacon = [3.14, 'cat', 11, 'cat', True]
```

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

```
In [13]: bacon.index('cat')#prints the index of the value in the list.

Out[13]: 1
```

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

```
In [14]: bacon.append(99) #Adds the value ending of the list.

In [15]: bacon

Out[15]: [3.14, 'cat', 11, 'cat', True, 99]
```

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

```
In [16]: bacon.remove('cat') #it remove first value only not all value.

In [17]: bacon

Out[17]: [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()?

The difference between the two methods is that .append() adds an item to the end of a list, whereas .insert() inserts and item in a specified position in the list.

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

The methods are remove() and pop(). It helps to remove the very first given element matching from the list. The pop() method removes an element from the list based on the index given.

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

The values that make up a list are called its elements, or its items. We will use the term element or item to mean the same thing. Lists are similar to strings, which are ordered collections of characters, except that the elements of a list can be of any type.

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

Tuples are immutable and lists are mutable

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

We can type it aa (42,)

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

tuple() function to pass the full list as an argument, and it will return the tuple data type as an output and call list() builtin function and pass the tuple as argument to the function. list() returns a new list generated from the items of the given tuple.

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

contain?

Variables will 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()?

copy() create reference to original object. If you change copied object - you change the original object. . deepcopy() creates new object and does real copying of original object to new one. Changing new deepcopied object doesn't affect original object.

```
In [ ]:
```