1. What exactly is []? Index Brackets

Ans : - [ ] is a empty list, like a =[ ]

1. 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 : input*

*# solution by changing the value in index 3*

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

spam[2] = 'hello'

spam

Output

[2, 4, 'hello', 8, 10]

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

1. What is the value of spam[int(int('3' \* 2) / 11)]?

Input

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

spam[int(int('3' \* 2) / 11)] *# spam[int(33/11)] = spam[3]*

Output

'd'

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

Input

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

spam[-1] *# negative index # d*

Output

'd'

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

Input

spam[:2] *# c*

*Output*

['a', 'b']

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

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

Input

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

bacon.index('cat') *# it returns the index of first occurrence of 'cat'*

Output

1

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

Input

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

bacon.append(99) *# append adds the item at the end of the list*

bacon

out put

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

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

Input

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

bacon.remove('cat') *# remove first occurrence of item*

bacon

outut

[3.14, 11, 'cat', True]

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

( \* ) is list replication operator ( + ) is list concatination operator

input

l1 = [1,4]

l2 = [2,5]

*# list concatination*

1. l1+l2

out put

[1, 4, 2, 5]

Input

l1 = [7,4]

*# list replication*

l1\*3

output

[7, 4, 7, 4, 7, 4]

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

append() Appends object to the end of the list  
insert() Insert object before index

input

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

bacon.append(99) *# append adds the item at the end of the list*

bacon

output

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

Input

*# solution by inserting value in 3rd index*

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

spam.insert(2,'hello')

spam

output

[2, 4, 'hello', 6, 8, 10]

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

Input

*#remove(item) - removeds first occurence of a item*

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

bacon.remove('cat')

bacon

output

[3.14, 11, 'cat', True]

Input

*#pop() - Remove and returns item at index (default last).*

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

bacon.pop()

bacon

output

[3.14, 'cat', 11, 'cat']

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

Both lists and strings can be passed to len()

Have indexes and slices

Can be used in for loops

Can be concatenated or replicated

Can be used with the in and not in operators

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

Lists : are mutable - they can have values added, removed, or changed. lists use the square brackets, [ and ]  
Tuples : are immutable; they cannot be changed at all. Tuples are written using parentheses, ( and )

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

Input

tuple = (42,)

tuple

output

(42,)

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

Input

l1 = [2,3]

l = tuple(l1)

l

output

(2, 3)

Input

t1 = (3,4)

t = list(t1)

t

output

[3, 4]

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

They contain references to list values

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

The copy.copy() function will do a shallow copy of a list,  
The copy.deepcopy() function will do a deep copy of a list. only copy.deepcopy() will duplicate any lists inside the list