DAY 1 Assignment 4

21WU0102040

1. What exactly is []?

**Answer:**

[] is an empty list in Python.

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.)

**Answer:**

spam[2] = 'hello' (Lists are zero-indexed, so the third value is at index 2).

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)]?

**Answer:**

spam[int(int('33') / 11)] → spam[3], which is 'd' (since spam = ['a', 'b', 'c', 'd']).

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

**Answer:**

spam[-1] gives the last element in the list, which is 'd'.

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

**Answer:**

spam[:2] gives the first two elements, which 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')?

**Answer:**

bacon.index('cat') returns the index of the first occurrence of 'cat', which is 1.

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

**Answer:**

It adds 99 to the end of the list, so the list becomes [3.14, 'cat', 11, 'cat', True, 99].

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

**Answer:**

It removes the first occurrence of 'cat', so the list becomes [3.14, 11, 'cat', True].

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

**Answer:**

Concatenation: + (e.g., [1, 2] + [3, 4] → [1, 2, 3, 4])

Replication: \* (e.g., [1, 2] \* 2 → [1, 2, 1, 2])

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

**Answer:**

append() adds an element to the end of the list.

insert() adds an element at a specified index.

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

**Answer:**

remove() (removes the first occurrence of a value).

pop() (removes an element at a specified index or the last element if no index is given)

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

**Answer:**

Both lists and strings are sequences that can be indexed, sliced, and iterated over.

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

**Answer:**

Lists are mutable (can be changed), while tuples are immutable (cannot be changed after creation).

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

**Answer:**

(42,) (A tuple needs a trailing comma to distinguish it from a regular expression in parentheses).

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

**Answer:**

List to tuple: tuple(list)

Tuple to list: list(tuple)

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

**Answer:**

They contain references to the list objects.

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

**Answer:**

copy.copy() creates a shallow copy (references nested objects), while copy.deepcopy() creates a deep copy (copies all objects recursively).