***Assignement-4***

1. It represents empty list.
2. spam = [2, 4, 6, 8, 10]

spam[2] = 'hello'

print(spam) # Output: [2, 4, 'hello', 8, 10]

1. ‘3’\*2 means ‘33’, then ‘33’/11 means internal diviiosn  
   and in the end we get spam[3]  
   which means if spam[‘a’,’b’,’c’,’d’]=spam[3]=’d’
2. Spam[-2]=’d’
3. Spam[:2]=[‘a’,’d’]
4. ‘ 1 ‘
5. bacon.append(99)

print(bacon) # Output: [3.14, 'cat', 11, 'cat', True, 99]

1. [3.14, 'cat', 11, True, 99]
2. List Concatenation Operator +: The + operator is used to concatenate two or more lists, creating a new list that contains all the elements from the original lists in the order they appear. It does not modify the original lists but instead creates a new list with the combined elements.

List Replication Operator \*: The \* operator, when used with a list, replicates the elements of the list a specified number of times, creating a new list with the repeated elements.

1. Append has the ability to change.

Insert can only insert new values.

1. We can use pop or del functions.
2. Indexing: Both lists and strings are indexed sequences in Python. This means that you can access individual elements in a list or a string using square brackets and an index value. The index starts from 0 for the first element.

Slicing: Both lists and strings support slicing, allowing you to extract a portion of the sequence using a start index, an end index, and an optional step value. Iteration: You can iterate over both lists and strings using loops like for loops to access each element or character one by

one.

1. The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. Both lists and tuples can store any data such as integer, float, string, and dictionary.
2. Also, tuples are written using parentheses, ( and ), while lists use the square brackets, [ and ]. How do you type the tuple value that has just the integer value 42 in it? (42,) (The trailing comma is mandatory.)
3. An iterable can be passed as an input to the tuple () function, which will convert it to a tuple object. If you want to convert a Python list to a tuple, you can use the tuple() function to pass the full list as an argument, and it will return the tuple data type as an output.
4. Variables that "contain" list values in Python do not actually store the list directly; instead, they store a reference to the list. In Python, all data structures, including lists, are objects. When you create a list and assign it to a variable, the variable holds a reference to the memory location where the list is stored.
5. copy — Shallow and deep copy operations

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.