

ASSIGNMENT-4

1. What exactly is []?

Ans: [] is a empty list. There are no values in the 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: The spam list is given as spam=[2,4,6,8,10], so to assign the value 'hello' as third value we need to do spam[2]='hello' now if we print spam we will get the third value as 'hello'

```
spam=[2,4,6,10]
spam[2]='hello'
print(spam)
```

output:

[2, 4, 'hello', 10]

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

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

output:

d

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

```
spam= ['a', 'b', 'c', 'd']
print(spam[-1])
```

output:

d

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

```
spam= ['a', 'b', 'c', 'd']  
print(spam[:2])
```

output:

['a', 'b']

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

```
bacon=[3.14, 'cat', 11, 'cat', True]  
print(bacon.index('cat'))
```

output:

1

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

```
bacon=[3.14, 'cat', 11, 'cat', True]  
bacon.append(99)  
print(bacon)
```

output:

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

This statement add the value 99 at the end of the list.

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

```
bacon=[3.14, 'cat', 11, 'cat', True]  
bacon.remove('cat') # removes the first occurrence of the 'cat' from the list  
print(bacon)
```

output:

[3.14, 11, 'cat', True]

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

Ans: The concatenation operator in the list is '+'

The replication operator in the list is '*'

```
list1=[1,2,3,4,5]
list2=[6,7,8,9]
print(list1+list2)#the concatenation operator
```

output:

[1, 2, 3, 4, 5, 6, 7, 8, 9]

```
list1=[1,2,3,4,5]
print(list1*2)#the replication operator
```

output:

[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]

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

Ans: The append() method adds the given value at the end of the list where as the insert() adds the value in the location that we have specified index.

```
list1=[1,2,3,4,5]
list1.append(9)
print(list1) # appends the value at the end of the list
```

output:

[1, 2, 3, 4, 5, 9]

```
list1=[1,2,3,4,5]
list1.insert(2,'hello')
print(list1) # appends the value at the specified index here it is 2
```

output:

[1, 2, 'hello', 3, 4, 5]

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

Ans: The two methods to remove the elements from the list are pop and remove

```
list1=[1,2,3,4,5]
list1.remove(3) # remove the specified element from the list
print(list1)
```

output:

[1, 2, 4, 5]

```
list1=[1,2,3,4,5]
list1.pop() # this method removes the element from the end of the list
print(list1)
```

output:

[1, 2, 3, 4]

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

Ans: list and strings have some common operations like indexing and slicing and they both can be passed to the common function len() which gives the length of the list and length of the string. Both the list and string can be concatenated and replicated. List and String have some common operators like in and not.

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

Lists	Tuples
*lists are mutable i.e the values can added and removed from the list.	*Tuples are Immutable i.e the values can not be added and removed from the tuples.
*Lists are represented by using [].	*tuples are represented by using ().
*lists consume more memory .	* Tuples consume less memory as compared to the lists.

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

```
t=(42,)
print(t)
```

output:

(42,)

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

Ans: Lists Values in tuple form

```
list1=[1,2,3,4,5,6,7]
tuple1=tuple(list1)
print(tuple1)
```

output:

(1, 2, 3, 4, 5, 6, 7)

Tuple values in the form of lists

```
tuple1=(1,2,3,4,5,6)
list1=list(tuple1)
print(list1)
```

output:

[1, 2, 3, 4, 5, 6]

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

Ans.They contain reference variable to the list value.

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

Ans:

`Copy.copy()` means this function will do shallow copy of the given list.

`Copy.deepcopy()` means this function will do deep copy of the list.The `copy.copy()` this function will duplicate any list inside the list.

