# Assignment 4

### **Question 1:-**

**What exactly is []?**

#### **Ans:-**

[ ] is a empty list.

The empty list value is a list value that contains no items. This is similar to how “ is the empty string value.

Index brackets ([ ]) have many uses in Python. First, they are used to define “list literals”, allowing you to declare a list and its contents in your program. Index brackets are also used to write expressions that evaluate a single item within a list or a single character in a string.

We can create an empty list using an empty pair of square brackets []. Square brackets [] are commonly used in Python to create empty lists because it is faster and more concise.

### **Question 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.)**

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

#### **Ans:-**

spam[2] = ‘hello’ (Notice that the third value in a list is at index 2 because the first index is 0).

### **Question 3:-**

**What is the value of spam[int(int(‘3’ \* 2) / 11)]?**

#### **Ans:-**

‘d’ (Note that ‘3’ \* 2 is the string ‘33’, which is passed to int() before being divided by 11. This eventually evaluates to 3. Expressions can be used wherever values are used).

### **Question 4:-**

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

#### **Ans:-**

‘d’ (Negative indexes count from the end).

### **Question 5:-**

**What is the value of spam[:2]?**

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

#### **Ans:-**

[‘a’, ‘b’]

### **Question 6:-**

**What is the value of bacon.index(‘cat’)?**

#### **Ans:-**

1

### **Question 7:-**

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

#### **Ans:-**

[3.14, ‘cat’, 11, ‘cat’, True, 99]

### **Question 8:-**

**How does bacon.remove(‘cat’) and change the look of the list in bacon?**

#### **Ans:-**

[3.14, 11, ‘cat’, True]

### **Question 9:-**

**What are the list concatenation and list replication operators?**

#### **Ans:-**

The operator for list concatenation is +, while the operator for replication is \*. (This is the same as for strings).

### **Question 10:-**

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

#### **Ans:-**

| Append | Extend |
| --- | --- |
| The append method will add an item to the end of the list. | The extend method will extend the list by appending all the items from the iterable. |
| The length of the list will be increased by 1. | The length of the list will be increased depending on the length of the iterable. |
| It will update the original list itself. | It will update the original list itself. |
| The return type is None. | The return type is None. |
| Example:-  a = [1, 2, 3]  a.append([4, 5])  Output :-  a = [1, 2, 3, [4, 5] ] | Example :-  a = [1, 2, 3]  a.extend([4, 5])  Output :-  a = [1, 2, 3, 4, 5] |

### 

### **Question 11:-**

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

#### **Ans:-**

The methods are remove(), pop(), and clear(). The remove() helps remove the first given element matching from the list. The pop() method removes an element from the list based on the index given. The clear() method will remove all the elements in the list.

### **Question 12:-**

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

#### **Ans:-**

The values that make up a list are called its elements. Lists are similar to strings, which are ordered collections of characters, except that the elements of a list can have any type and for any one list, the items can be of different types.

The similarity between Lists and Strings in Python is that both are sequences. The difference between them is that firstly, Lists are mutable but Strings are immutable Secondly, elements of a list can be of different types whereas a String only contains characters that are all of the String type.

### **Question 13:-**

**What’s the difference between tuples and lists?**

#### **Ans:-**

| Tuple | List |
| --- | --- |
| list() is a collection of ordered and changeable data. | A tuple is a collection of data that is ordered and unchangeable. |
| The items are surrounded in parentheses (). | The items are surrounded in square brackets [ ]. |
| Tuple are immutable in nature. | Lists are mutable in nature. |
| There are 33 available methods on tuples. | There are 46 available methods on the list. |
| In a dictionary, we can create keys using tuples. | In dictionary, we can’t use lists as keys. |
| List iteration is slower and is time-consuming. | Tuple iteration is faster. |
| List consumes more memory. | Tuple consumes less memory. |
| List operations are more error-prone. | Tuples operations are safe. |
| List provides many in-built methods. | Tuples have less in-built methods. |
| List is useful for insertion and deletion operations. | Tuple is useful for readonly operations like accessing elements. |

### **Question 14:-**

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

#### **Ans:-**

tuple = (42,)

tuple

Output:-

(42,)

### **Question 15:-**

**How do you get a list value’s tuple form? How do you get a tuple value’s list form?**

#### **Ans:-**

list = [2, 3]

l = tuple(list)

l

Output :-

(2, 3)

t1 = (3, 4)

t = list(t1)

t

Output :-

[3, 4]

### **Question 16:-**

**Variables that “contain” list values are not necessarily lists themselves. Instead, what do they contain?**

#### **Ans:-**

They contain references to list values

### **Question 17:-**

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

#### **Ans:-**

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.