Module 8: Python Data Types

List: List is group of elements which can be of same or different data types.

- List is created by placing the elements inside square bracket [ ] separated by comma.
- Lists are mutable meaning we can append, insert , remove etc.

List Operations:

- **+** ➔ to perform concatenation between lists else we get TypeError.
- ***** ➔ To repeat elements in the list int number of times. If one of them is not int then we get TypeError:
- **in** to check the membership in List.

Changing & Adding Elements in List:

- Changing the element via =

Note: if index is not in range then we get IndexError.

- **append(element):** Adds element to the end of the List.

Note: if we supply more then one element then we get TypeError: append() takes as input only one argument

- **extend([e]/(e)/{e}):** Appends all the element of the List/Tuple/Set to the main List

Note: if we supply more then one element then we get TypeError: append() takes as input only one argument

- **insert():** Inserts an item at the defined index

Note: if index is too high then element is inserted at last position and if index is too low then element is inserted at first position.

Removing Elements from List:

- **remove(ele):** Removes an item specified from the List.

Note: If element is not present in the List then we get ValueError.

- **del ln[pos]:** Deletes the specified index element from the List

Note: if index is not in range then we get IndexError.

- **pop():** Removes the last element from the List.

Note: if List is empty then we get IndexError: pop from empty list.

- **pop(index):** Removes the element present at the index position.

Note: if index is not in range then we get Ind-exError: pop index out of range.

- **clear():** Removes all item from the List.

Occurrence of items & Sorting Items in List:

- **index(ele):**

Returns the index of first matched item.

Note: If item is not present in the List then we get ValueError.

- **count(ele):**

Returns the count of the item passed as argument in the List.

- **sort():**

Sorts item in a list in ascending order.

Note: if the items are of mixed types then we get TypeError < not supported.

- **reverse():**

It reverses the order of items in the List

Tuple:

- Tuple is similar to a List with the main difference of Tuples are immutable i.e. we cannot change the elements of a tuple once it is assigned whereas Lists are mutable.
- Advantages:
  - It guarantees that the data will remain write-protected.
  - Tuples can be used in Dictionary (while List cannot)
  - Iterating becomes faster.
- Tuples are created by placing the elements in ().

Occurrence of items in Tuple:

- **index(ele):** Returns the index of first matched item.

Note: If item is not present in the Tuple then we get ValueError.

- **count(ele):** Returns the count of the item passed as argument in the Tuple.

Set:

- Set is an unordered collection of elements in which every element is unique(no duplicates).
- Sets can be created by placing elements in { }.
- To access the elements we should not use indexing or slicing.
- Set is mutable meaning we can add or remove items.

Set Operations:

- **| performs Union➔** set of all elements from both sets.
- **& performs Intersection ➔** elements common in both sets.
- **- performs Difference ➔** elements in set1 not in set2
- **^ performs Symmetric Diff ➔** elements in both sets except the common

Adding to Set:

- **add(ele):** Adds element to the set if its not present.
- **update([e]/(e)/{e}):** Updates the set with List or Tuple or Set.

Removing from set:

- **discard(ele):** Removes element from the set if it is a member else nothing.
- **remove(ele):** Removes element from the set else KeyError
- **pop():** Removes and returns an arbitrary set element. If the set is empty we get KeyError: pop from an empty set.
- **clear():** Removes all elements from set.

Dictionary:

- Python dictionary is an ordered collection of items where elements are in the key:value pairs.
- Keys should be unique and immutable(number, string or tuple).
- If we enter same key again then old key will be overwritten.
- All key value pairs are inserted in curly braces { }.

Accessing Dictionary:

- **dn['key']:** it will return the value of the key if its present else KeyError.
- **get('key') / get('key', dv):** It returns value for the specified key if key is in dictionary. None if the key is not found. We can specify default value also.

Modifying Dictionary:

- **dn['key'] = value:** it will add element if it is not present and if present then update the value of the key.
- **update(dict):** updates the dictionary with the elements from the another dictionary object

Removing from Dictionary:

- **pop('key'):** removes and returns an element from a dictionary having the given key. If element is not present then we get KeyError.
- **popitem():** returns and removes an arbitrary element (key, value) pair from the dictionary.
- **clear():** removes all items from the dictionary.

Module 9: Python Data Types

Q1) What datatype is the object below ? **L = [1, 23, 'hello', 1]**

Options:

a) list
b) dictionary
c) array
d) tuple

Solution:

Q2) Suppose **list1 is [1, 3, 2]**, What is **list1 * 2** ?

Options:

a) [2, 6, 4].
b) [1, 3, 2, 1, 3].
c) [1, 3, 2, 1, 3, 2] .
d) [1, 3, 2, 3, 2, 1].

Solution:

Q3) You write the following code:

```
list_1 = [1, 2]
list_2 = [3, 4]
list_3 = list_1 + list_2
list_4 = list_3 * 3
print(list_4)
```

You run the code. What is the output value?

Options:

A. [3, 6, 9, 12]
B. [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
C. [[1, 2], [3, 4], [1, 2], [3, 4], [1, 2], [3, 4]]
D. [[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]

Solution:

Q4) Suppose **list1 is [2, 33, 222, 14, 25]**,
What is **list1[:-1]** ?

Options:

a) [2, 33, 222, 14].
b) Error
c) 25
d) [25, 14, 222, 33, 2].

Solution:

Q5) You develop a Python application for your company. A list named employees contain 200 employee names, the last five being company management. You need to slice the list to display all employees excluding management. Which two code segments should you use?

Options:

A. employees[1:-4]
B. employees[:-5]
C. employees[1:-5]
D. employees[0:-4]
E. employees [0:-5]

Solution:

Q6) You develop a Python Practice Test application for your school. A list named colors contains 200 colors. You need to slice the list to display every other color starting with the second color. Which code should you use?

Options:

A. colors[1:2]
B. colors [::2]
C. colors[2:2]
D. colors [1::2]

Solution:

Q7) You evaluate the following code:

```
numList = [0,1,2,3,4]
print(5 in numList)
```

What is the output of the print statement?

Options:

A. 4
B. False
C. True
D. 5

Solution:

Q8) What is the result of the following code:

```
numList = [1,2,3,4,5]
alphaList = ["a", "b", "c", "d", "e"]
print(numList is alphaList)
print(numList == alphaList)
numList = alphaList
print(numList is alphaList)
print(numList == alphaList)
```

Options:

A. True True True True
B. False False True True
C. False False False False
D. True False True False

Solution:

Q9) You develop a Python Practice Test application for your company.
How should you complete the code so that the print statements are accurate?
To answer, select the appropriate code segments in the answer area.

```
numList = [1,2,3,4,5]
alphaList = ["a","b",'c',"d","e"]
[1] _____________________________
      print("The values in numList are equal to alphaList")
[2] _____________________________
      print("The values in numList are not equal to
alphaList")
```

Option1:

A. if numList == alphaList: B. if numList == alphaList
C. else: D. else

Option2:

A. if numList == alphaList: B. if numList == alphaList
C. else: D. else

Solution:

Q10) What is the output of the following?

```
num = [10, 30, 20]
num.append(40)
num.extend([60, 50])
num.insert(3, 90)
print(num)
```

Options:

a) [10, 30, 20, 90, 40, 50, 60]
b) [10, 30, 90, 20, 40, 60, 50]
c) [10, 30, 20, 90, 40, 60, 50]
d) Error

Solution:

Q11) What is the output of the following?

```
veggies = ['carrot', 'broccoli', 'potato', 'celery']
veggies.insert(veggies.index('broccoli'), 'celery')
veggies.pop(1)
veggies.pop()
veggies.remove('celery')
print(veggies)
```

Options:

a) Error
b) ['carrot', 'potato']
c) ['potato', 'carrot']
d) ['potato', 'carrot', 'celery']

Solution:

Q12) you find errors while evaluating the following code. You need to correct the code? Select two

```
01. numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
02. index = 0
03. while [index] < 10
04.         print(numbers[index])
05.   if numbers(index) = 6
06.         break
07.   else:
08.         index += 1
```

Options:

A. line 03 should be while(index < 10):
B. line 03 should be while(index <=10):
C. line 05 should be if numbers[index] == 6:
D. line 05 should be if numbers(index) = 6

Solution:

Q13) What is the output of the following?

```
num = (10, 50, 20)
num[1] = 40
print(num)
```

Options:

a) (10, 50, 20)
b) Error
c) (10, 40, 20)
d) (10, 40, 50, 20)

Solution:

Q14) What is the output of the following?

```
set = {'amit', 'sumit', 'anil',}
set.add('amit')
set.discard('amit')
set.update(['amit', 'rahul'])
set.discard('rahul')
print(set)
```

Options:

a) {'amit', 'sumit', 'anil'}
b) {'sumit', 'anil'}
c) {'sumit', 'amit', 'amit'}
d) {'sumit', 'amit'}

Solution:

Q15) What is the output of the following?

```
s1 = {40, 20, 60}
s2 = {50, 20}
r1 = s1 | s2
r2 = s1 & s2
r3 = s1 - s2
print(r1, r2, r3)
```

Options:

a) {40, 50, 20, 60} {50} {40, 60}
b) {40, 50, 20, 60} {20} {40}
c) {40, 50, 60} {20} {40, 60}
d) {40, 50, 20, 60} {20} {40, 60}

Solution:

Q16) What is the output of the following?

```
d = {10: 'a', 20: 'b', 40: 'c'}
d[50] =  'e'
d.pop(20)
d.update({20:'e'})
for i in d:
    print(i, end = ' ')
```

Options:

a) 40 10
b) 10 40 50 20
c) 10 20
d) 10 40 50

Solution:

Q17) You create the following program to locate a conference room and display the room name.

```
1. rooms = {1:'Foyer', 2:'Conference Room'}
2. room = input("enter the room number ")
3. if not room in rooms:
4.     print("yes")
5. else:
6.     print("no")
```

Which two data types are stored in the rooms list at line 1?

1) bool and string
2) float and bool
3) int and string
4) float and int

What is the data type of room at line 2?

1) bool
2) float
2) int
4) string

Solution: