J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

# 9. LISTS, TUPLES, SETS, AND DICTIONARY

## Section – A

**Choose the best answer** **(1 Mark)**

1. Pick odd one in connection with collection data type

(a) List (b) Tuple (c) Dictionary **(d) Loop**

2. Let list1=[2,4,6,8,10], then print(List1[-2]) will result in

(a) 10 **(b) 8** (c) 4 (d) 6

3. Which of the following function is used to count the number of elements in a list?

(a) count() (b) find() **(c)len()** (d) index()

4. If List=[10,20,30,40,50] then List[2]=35 will result

(a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35] **(c) [10,20,35,40,50]** (d) [10,35,30,40,50]

5. If List=[17,23,41,10] then List.append(32) will result

(a) [32,17,23,41,10] **(b) [17,23,41,10,32]** (c) [10,17,23,32,41] (d) [41,32,23,17,10]

6. Which of the following Python function can be used to add more than one element within an Existing list?

(a) append() (b) append_more() **(c)extend()** (d) more()

7. What will be the result of the following Python code?

```
S=[x**2 for x in range(5)]
print(S)
```

(a) [0,1,2,4,5] **(b) [0,1,4,9,16]** (c) [0,1,4,9,16,25] (d) [1,4,9,16,25]

8. What is the use of type() function in python?

(a) To create a Tuple (b) To know the type of an element in tuple.

**(c) To know the data type of python object.** (d) To create a list.

9. Which of the following statement is not correct?

(a) A list is mutable

(b) A tuple is immutable.

(c) The append() function is used to add an element.

**(d) The extend() function is used in tuple to add elements in a list.**

10. Let setA={3,6,9}, setB={1,3,9}. What will be the result of the following snippet?

```
print(setA|setB)
```

(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} **(d) {1,3,6,9}**

11. Which of the following set operation includes all the elements that are in two sets but not the one that are common to two sets?

**(a) Symmetric difference** (b) Difference (c) Intersection (d) Union

12. The keys in Python, dictionary is specified by

(a) = (b) ; (c)+ **(d) :**

## Section-B

**Answer the following questions** **(2 Marks)**

**1. What is List in Python?**

- A list is an ordered collection of values enclosed within square brackets [ ] also known as a "sequence data type".
- Each value of a list is called as element.
- Elements can be a numbers, characters, strings and even the nested lists.
- **Syntax:** Variable = [element-1, element-2, element-3 …… element-n]

**2. How will you access the list elements in reverse order?**

- Python enables reverse or negative indexing for the list elements.
- A negative index can be used to access an element in reverse order.
- Thus, python lists index in opposite order.
- The python sets -1 as the index value for the last element in list and -2 for the preceding element and so on.
- This is called as **Reverse Indexing.**

**3. What will be the value of x in following python code?**

```
List1=[2,4,6,[1,3,5]]
x=len(List1)
print(x)
```

**OUTPUT:**

```
====== RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/LI.py ======
4
>>>
```

**4. Differentiate del with remove( ) function of List.**

| del | remove( ) |
|---|---|
| **del** statement is used to delete known elements | **remove( )** function is used to delete elements of a list if its index is unknown. |
| The del statement can also be used to delete entire list. | The remove is used to delete a particular element |

**5. Write the syntax of creating a Tuple with n number of elements.**

**Syntax:**

**Tuple_Name = (E1, E2, E2 ……. En)** # Tuple with n number elements
**Tuple_Name = E1, E2, E3 ….. En** # Elements of a tuple without parenthesis

**6. What is set in Python?**

- In python, a set is another type of collection data type.
- A Set is a mutable and an unordered collection of elements without duplicates or repeated element.
- This feature used to include membership testing and eliminating duplicate elements.

## Section-C

**Answer the following questions** **(3 Marks)**

**1. What are the advantages of Tuples over a list?**

- The elements of a list are changeable (mutable) whereas the elements of a tuple are unchangeable (immutable), this is the key difference between tuples and list.
- The elements of a list are enclosed within square brackets. But, the elements of a tuple are enclosed by paranthesis.
- Iterating tuples is faster than list.

**2. Write a short note about sort( ).**

**sort ( ):**

- It sorts the element in list.
- sort( ) will affect the original list.

**Syntax :** List.sort(reverse=True|False, key=myFunc)

**Description of the Syntax:**

Both arguments are optional ,

- If reverse is set as True, list sorting is in descending order.
- Ascending is default.
- Key=myFunc; “myFunc” - the name of the user defined function that specifies the sorting criteria.

**3. What will be the output of the following code?**

```
list = [2**x for x in range(5)]
print(list)
```

**OUTPUT:** [1, 2, 4, 8, 16]

**4. Explain the difference between del and clear( ) in dictionary with an example.**

| del | clear( ) |
|---|---|
| The **del** statement is used to delete known elements | The function **clear( )** is used to delete all the elements in list |
| The del statement can also be used to delete entire list. | It deletes only the elements and retains the list. |

**5. List out the set operations supported by python.**

**Set Operations:**

**(i) Union:** It includes all elements from two or more sets.
**(ii) Intersection:** It includes the common elements in two sets.
**(iii) Difference:** It includes all elements that are in first set (say set A) but not in the second set (say set B).
**iv) Symmetric difference:** It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.

**6. What are the difference between List and Dictionary?**

| List | Dictionary |
|---|---|
| • A list is an ordered collection of values or elements of any type . | • A dictionary is a mixed collection of elements and it stores a key along with its element. |
| • It is enclosed within square brackets [ ] | • The key value pairs are enclosed with curly braces { }. |
| • **Syntax:**<br>Variable = [element-1, element-2, element-3 …… element-n] | • **Syntax of defining a dictionary:**<br>Dictionary_Name = { Key_1: Value_1,<br>Key_2:Value_2,<br>……..<br>Key_n:Value_n<br>} |
| • The commas work as a separator for the elements. | • The keys in a Python dictionary is separated by a colon ( : ) while the commas work as a separator for the elements. |

## Section - D

**Answer the following questions:** **(5 Marks)**

**1. What the different ways to insert an element in a list. Explain with suitable example.**

**Inserting elements in a list using insert():**

- The **insert ( )** function helps you to include an element at your desired position.
- The **insert( )** function is used to insert an element at any position of a list.

**Syntax:**

List.insert (position index, element)

**Example:**

```
>>> MyList=[34,98,47,'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan' ]
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
```

**Output:** [34, 98, 47, 'Ramakrishnan', 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']

- In the above example, insert( ) function inserts a new element 'Ramakrishnan' at the index value 3, ie. at the $4^{th}$ position.
- While inserting a new element, the existing elements shifts one position to the right.

**Adding more elements in a list using append():**

- The **append( )** function is used to add a single element in a list.
- But, it includes elements at the end of a list.

**Syntax:**

List.append (element to be added)

**Example:**

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
```

**Output:** [34, 45, 48, 90]

**Adding more elements in a list using extend():**

- The **extend( )** function is used to add more than one element to an existing list.
- In extend( ) function, multiple elements should be specified within square bracket as arguments of the function.

**Syntax:**

List.extend ( [elements to be added])

**Example:**

```
>>> Mylist=[34, 45, 48]
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
```

**Output:** [34, 45, 48, 90, 71, 32, 29]

**2. What is the purpose of range( )? Explain with an example.**

**range():**

- The range( ) is a function used to generate a series of values in Python.
- Using range( ) function, you can create list with series of values.
- The range( ) function has three arguments.

**Syntax of range ( ) function:**

range (start value, end value, step value)

where,

- **start value** – beginning value of series. Zero is the default beginning value.
- **end value** – upper limit of series. Python takes the ending value as upper limit – 1.
- **step value** – It is an optional argument, which is used to generate different interval of values.

**Example : Generating whole numbers upto 10**

```
for x in range (1, 11):
print(x)
```

**Output:**

```
1
2
3
4
5
6
7
8
9
10
```

**Creating a list with series of values**

- Using the range( ) function, you can create a list with series of values.
- To convert the result of range( ) function into list, we need one more function called list( ).
- The list( ) function makes the result of range( ) as a list.

**Syntax:**

```
List_Varibale = list ( range ( ) )
```

**Example :**

```
>>> Even_List = list(range(2,11,2))
>>> print(Even_List)
```

**Output:** [2, 4, 6, 8, 10]

- In the above code, list( ) function takes the result of range( ) as Even_List elements.
- Thus, Even_List list has the elements of first five even numbers.

**3. What is nested tuple? Explain with an example.**

**Tuple:**

- Tuples consists of a number of values separated by comma and enclosed within parentheses.
- Tuple is similar to list, values in a list can be changed but not in a tuple.

**Nested Tuples:**

- In Python, a tuple can be defined inside another tuple; called Nested tuple.
- In a nested tuple, each tuple is considered as an element.
- The for loop will be useful to access all the elements in a nested tuple.

**Example:**

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5), ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
for i in Toppers:
    print(i)
```

**Output:**

```
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)
```

**4. Explain the different set operations supported by python with suitable example.**

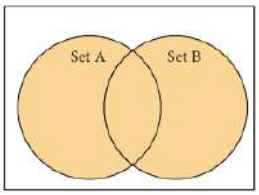
➢ A Set is a mutable and an unordered collection of elements without duplicates.

**Set Operations:**

➢ The set operations such as Union, Intersection, difference and Symmetric difference.

**(i) Union:**

- It includes all elements from two or more sets.
- The **operator** | is used to union of two sets.
- The function union( ) is also used to join two sets in python.




**Example:**

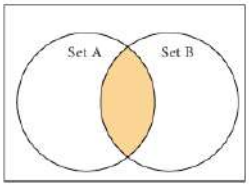
```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

**Output:**

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

**(ii) Intersection:**

- It includes the common elements in two sets.
- The **operator &** is used to intersect two sets in python.
- The function **intersection( )** is also used to intersect two sets in python.




**Example:**

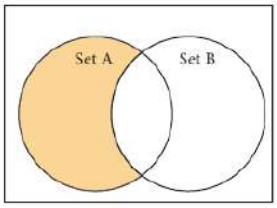
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

**Output:**

```
{'A', 'D'}
```

**(iii) Difference:**

- It includes all elements that are in first set (say set A) but not in the second set (say set B).
- The minus **(-) operator** is used to difference set operation in python.
- The function **difference( )** is also used to difference operation.




**Example:**

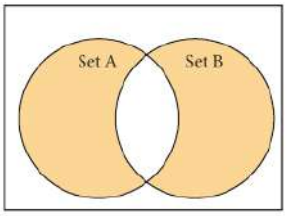
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

**Output:**

```
{2, 4}
```

**(iv) Symmetric difference**

- It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.
- The caret **(^) operator** is used to symmetric difference set operation in python.
- The function **symmetric_difference( )** is also used to do the same operation.




**Example:**

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```

**Output:**

```
{2, 4, 'B', 'C'}
```

**PREPARED BY**

**J. BASKARAN M.Sc., B.Ed. (C.S)**
**jbaskaran89@gmail.com**

**J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)**
**jilakkia@gmail.com**