

Collection Data types (OR) Data Structures

Python data types are classified into 2 categories

1. Scalar Data types (5)
 - a. Int
 - b. Float
 - c. Complex
 - d. Boolean
 - e. NoneType
2. Collection Data types (9)
 - a. Sequences
 - i. List
 - ii. Tuple
 - iii. Range
 - iv. String
 - v. Bytes
 - vi. bytearray
 - b. Sets
 - i. Set
 - ii. frozenset
 - c. Mapping
 - i. Dictionary

Scalar data types are used to represent single value or one value. In order to represent group of values or collection of values we use collection data types.

Why Collection data types?

1. To represent more than one value or object
2. Grouping individual objects/values into single object and referring with one name, this avoid creating multiple variables
3. By grouping all individual objects/values, we can perform aggregate operations (maximum, minimum, avg, sum, median, std,...)
4. Collections are dynamic in size
5. Collections can be homogeneous (similar type) or heterogeneous (different type)
6. Collections uses data structure for organizing data. Data structures define set of rules and regulations for representing data in memory.

Based on the organization of data, collections are classified into 3 types.

1. Sequences
 - a. List
 - b. Tuple
 - c. Range
 - d. String
 - e. Bytes
 - f. Bytearray
2. Sets
 - a. Set
 - b. Frozenset
3. Mapping
 - a. Dictionary

Sequences

Sequences are ordered collection, where data is organized in memory in sequential order. In ordered collections insertion is preserved, the order which in which data inserted, in same order it stores in memory.

Sequences are index based collections, where reading and writing can be done using index (position).

Sequences allow duplicate values or duplicate items/elements.

Python support 6 sequence data types, in which 2 mutable sequences and 4 are immutable sequences

Mutable sequences

1. List
2. Bytearray

Immutable sequences

1. Tuple
2. String
3. Range
4. Bytes

List data type

List is mutable sequence data type or data structure.

Mutable means after creating list changes can be done.

List is can be homogeneous or heterogeneous
List allows duplicate values or items/element

In application development list is used to group individual objects where duplicates are allowed and reading, writing is done sequentially and randomly.

How to create list?

List can be created in several ways

1. Empty list is created using empty square brackets []

```
>>> A=[]
>>> print(A)
[]
>>> print(type(A))
<class 'list'>
```

2. List created with items/elements/values separated with comma , with in square brackets

Syntax: list_name=[ele,ele,ele,ele,..]

```
>>> B=[10]
>>> print(B,type(B))
[10] <class 'list'>
>>> C=[10,20,30,40,50]
>>> print(C,type(C))
[10, 20, 30, 40, 50] <class 'list'>
>>> D=[10,1.5,1+2j,True,"Python"]
>>> print(D,type(D))
[10, 1.5, (1+2j), True, 'Python'] <class 'list'>
```

3. Using list() function (Type conversion function)

This function is used to create list by converting other collections into list type.

Note: every collection data type is called iterable (which allow to iterate or read individual values)

Syntax1: list() → This create empty list

Syntax2: list(iterable) → This create list by reading values from existing iterable or collection

```
>>> E=list()
>>> print(E)
[]
>>> F=list(range(1,6))
>>> print(F)
[1, 2, 3, 4, 5]
>>> G=list(range(10,60,10))
>>> print(G)
[10, 20, 30, 40, 50]
>>> I=list("ABCD")
>>> print(I)
['A', 'B', 'C', 'D']
>>> J=list("PYTHON")
>>> print(J)
['P', 'Y', 'T', 'H', 'O', 'N']
>>> K=list([10,20,30])
>>> print(K)
[10, 20, 30]
```

```
>>> L=list(10)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    L=list(10)
TypeError: 'int' object is not iterable
```

How to read content of list or sequence?

The content of list or sequence is read in different ways

1. Index
2. Slicing
3. for loop
4. iterator
5. enumerate