1) Atomic 2) Container

bool int float list

L = [10, 20, 30, 40]

index

L[0] L[1] L[2] L[3]

L_0 L_1 L_2 L_3

Name of container object

+

Subscript operator

+

Index

} = Individual Access to member

Container classes which provide index based individual access to their members are called as Sequential Containers.

---

① string ② tuple ③ list } → Sequential containers.

④ dict → Associative containers.

⑤ set

---

list :

0 1 2 3 index

L = [100, 200, 300, 400]

L[?] 100

L[?] 200

L[?] 300

L[?] 400

↓ indices

which are internally allocated by list

D = {'a' : 100, 'b' : 200, 'c' : 300, 'd' : 400}

D[?] 100

D[?] 200

D[?] 300

D[?] 400

D['a'] 100

D['b'] 200

D['c'] 300

D['d'] 400

D = {1.1 : 100, 'xyz' : 200, False : 300, 'HAHAHA' : 400}

Key

D[1.1] 100 $D_{1.1}$

D['xyz'] 200 $D_{'xyz'}$

D[False] 300 D_{False}

D['HAHAHA'] 400 $D_{'HAHAHA'}$

```
>>> D = {1.1 : 100, 'xyz' : 200, False : 300, 'HAHAHA' : 400}
>>> D[1.1]
```

```
>>> D['xyz']
100
>>> D[False]
200
>>> D['ΗΑΗΑΗΑ']
300
>>> D[1.1]
400
```

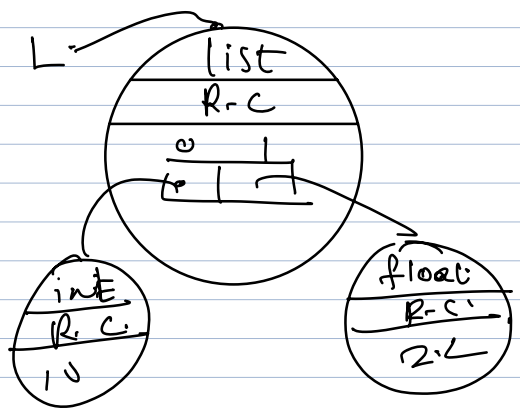
1.1 : 100

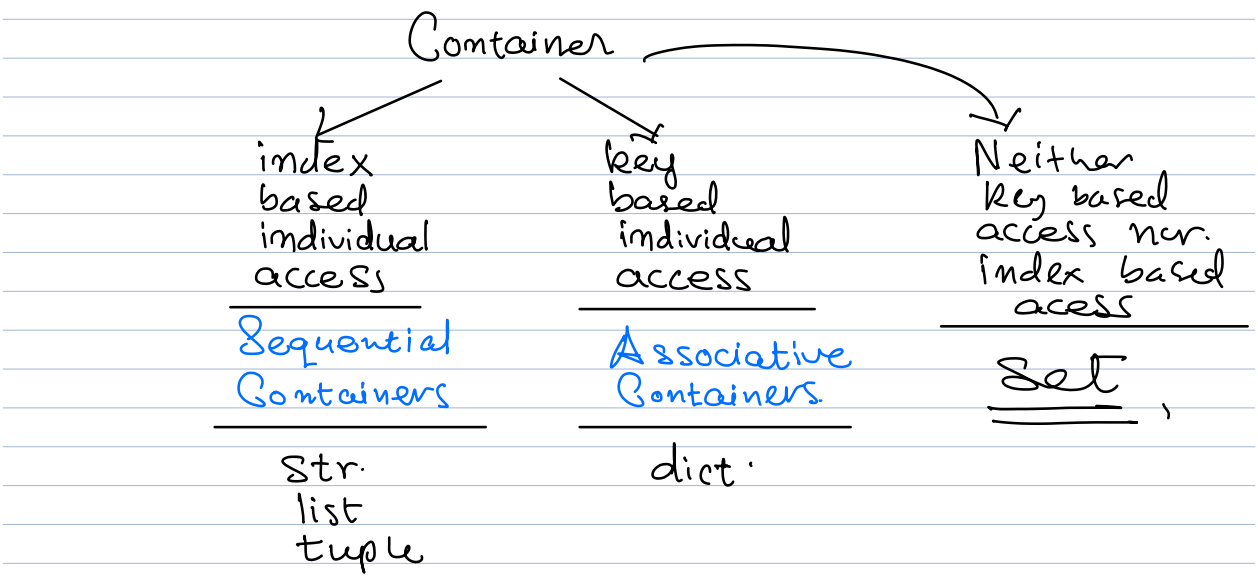
key : value

item

[Dictionary is a collection of items, every item is a key : value pair where key can be an object of any IMMUTABLE type & value can be an object of ANY object.]

L = [10, 2.2]






```
for i in range(len(L)):
    L[i]
```

```
for x in L:
    x
```

L[i]
O(1)

D[key]
O(1).

touch faster than dict.

```
for x in L:
    print(x)
```

```
for key in D:
    print(D[key])
```

=

Iterator level.

Gym

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

6 to 7.

Key Key-1