“If you don’t like your destiny, don’t accept it.

Instead have the courage to change it the way you want it to be.” – Naruto

Python's collections module:

1. Package: a directory which has collections of modules. it also has __init__.py file. with this extension you can easily find out that it's a package.

2. Module: a python script, it has .py file extension. it has number of functions, classes, and variables.

3. script: a module can be called into a python script, which inturn uses module's functions and variables in it's code. we can also import

this script with the help of import statement.

4. collections module: it has different types of containers, containers are used to store different objects and we can also access them.

4. different data types in the collections module and their operations

--> Named tuples: it's a built in tuple data type and similar to the tuple data type, it has the extension .namedtuple.

1. these named tuples are immutable. (i.e., we can't add or change elements once they are assigned.)

2. it has keys which have their corresponding values.

3. we can access named tuples either by index or keys

4. it's used when we want to use several tuples in a application

--> deque: a double ended queue (deque), it supports append, pop elements from both sides of list

1. these are used as double-linked lists which are very useful for inserting,

deleting elements in O(1) time complexity.

--> Ordered dictionaries: a dictionary that preserves the order of the keys that are inserted.

1. if any application needs the key order, then we use OrderDict

--> defaultdict : it's a subclass of the built in dictionary class.

1. it's almost similar as dictionary class but only

difference is this never raises a key error as normal dictionary.

--> ChainMap: it's used to create a list of dictionaries. i.e., it combines multiple dictionaries into a single dictionary

1. when we search for key in the chainmap, it checks all the dictionaries one by one untill it finds the key

--> Counter: it's a hashable object, (i.e., it's value wont be changed throughout it's lifetime in the code once it's assigned.

counter is used to count the number of such hashable objects.

1. in counter, the dictionary key is hashable object but value is it's count.

2. when we want to see how many times each unique word is occuring in a string, then we use counter object.

--> UserDict : it wraps the dictionary objects.

1. we can add customized functions to dictionary.

2. when we want to add, update or modify the functionalities of dictionary, we use UserDict

--> UserList : it wraps the list objects.

1. we can add customized functions to lists.

2. when we want to add, update or modify the functionalities of lists, we use UserList.

--> UserString: it wraps the string objects.

1. we can add customized functions to strings.

2. when we want to add, update or modify the functionalities of strings, we use UserString