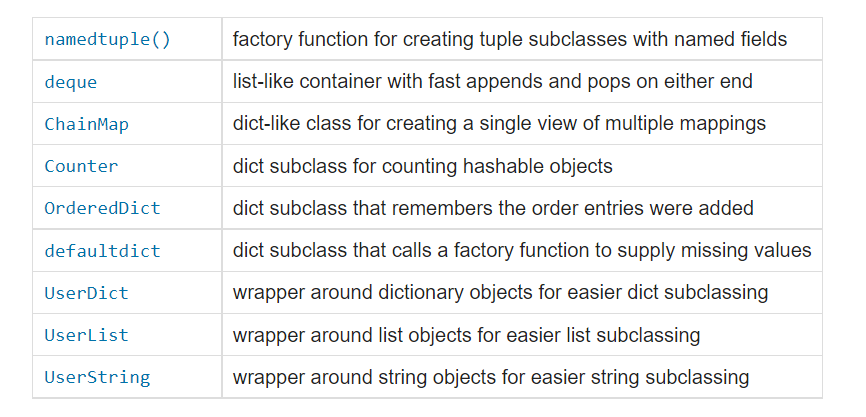
**Python Collections Module:**

* Collections module provides different type of container.
* A Container is an object that is used to store different objects and provide a way to access the contained objects and iterate over them.

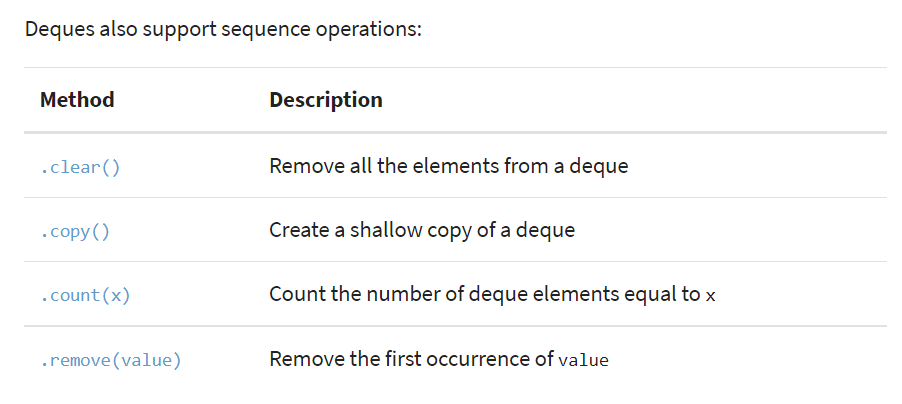


**Named Tuple:**

* A NamedTuple returns a tuple object with names for each position which the ordinary tuples lack.
* These fields give you direct access to the values in a given named tuple using the dot notation, like in obj.attr.
* **\_make()** : This function is used to return a namedtuple() from the iterable passed as argument.
* **\_asdict()** : This function returns the OrdereDict() as constructed from the mapped values of namedtuple().

**Deque:**

* In Python, append and pop operations on the beginning or left side of list objects are inefficient, with O(n) time complexity. In Python, append and pop operations on the beginning or left side of list objects are inefficient, with O(n) time complexity.
* Deque (Doubly Ended Queue) is the optimized list for quicker append and pop operations from both sides of the container.
* It provides O(1) time complexity for append and pop operations as compared to list with O(n) time complexity.
* The deque initializer takes two optional arguments:
  + **iterable** holds an iterable that serves as an initializer.
  + **maxlen** holds an integer number that specifies the maximum length of the deque.
* If you don’t provide an iterable, then you get an empty deque. If you supply a value to maxlen, then your deque will only store up to maxlen items.



**UserDict:**

* UserDict is a dictionary-like container that acts as a wrapper around the dictionary objects. This container is used when someone wants to create their own dictionary with some modified or new functionality.
* Python’s collections provides three convenient wrapper classes that mimic the behavior of the built-in data types.
  + UserString
  + UserList
  + UserDict

**UserList:**

* [UserList](https://www.geeksforgeeks.org/collections-userlist-in-python/)is a list like container that acts as a wrapper around the list objects. This is useful when someone wants to create their own list with some modified or additional functionality.

**UserString:**

* UserString is a string like container and just like UserDict and UserList it acts as a wrapper around string objects. It is used when someone wants to create their own strings with some modified or additional functionality.

**ChainMap:**

* A ChainMap encapsulates many dictionaries into a single unit and returns a list of dictionaries.

**Counter:**

* A counter is a sub-class of the dictionary.
* It is used to keep the **count of the elements in an iterable** in the form of an unordered dictionary where the key represents the element in the iterable and value represents the count of that element in the iterable.

**OrderedDict:**

* An **OrderedDict** is also a sub-class of dictionary but unlike dictionary, it remembers the order in which the keys were inserted.

**Defaultdict:**

* A **DefaultDict** is a sub-class to dictionary. It is used to provide some default values for the key that does not exist and never raises a KeyError.