Lists	Tuples	Sets	Dictionaries
Non-homogeneous data structure that stores data in columns of single or multiple rows	Non-homogeneous data structure that stores data in columns of single or multiple rows	Non-homogeneous data structure that stores data in a single row	Non-homogeneous data structure that stores key-value pairs
Represented by [ ] and elements seperated by comma	Represented by ( ) and elements seperated by comma	Represented by { } or set() and elements seperated by comma	Represented by { } and elements seperated by comma
It is mutable i.e. make changes	It is immutalbe i.e. can not make changes	It is mutable	It is mutable
It allows duplicate elements	It allows duplicate elements	It does not allow duplicate elements	It allows duplicate values but no duplicate keys
It is ordered	It is ordered	It is unordered	It is ordered (Python 3.7 and above)
It uses indexing that starts from 0	It also uses indexing that starts from 0	It does not use indexing	It uses key to access values and vice versa
sort() can be done if the data type is the same	Immutable so sort() can not be used	It is unordered	The keys can be sorted
It allows appending and insertion	It does not allow change	Allows adding	It allows adding with update()
reverse() can be done	Can not be changed	Unordered	Does not have an in built reverse function
count() return the number of occurances of the given value	count() return the number of occurances of the given value	count() is not defined	count() is not defined