Python Data Structures

List:

1. List Creation:

i. emptyList = [ ]
ii. ListName = [1,2,3,4,5]
iii. ListName = [[1,2],[3,4]] -> list of lists

2. Adding Elements:

i. append( ) -> listname.append('element')
ii. insert( ) -> listname.insert(index,'element')
iii. extend( ) -> listname1.extend(listname2)

3. Removing elements:

i. remove( ) -> lst.remove('element')
ii. del -> del listname[index]
iii. pop( ) -> listname.pop(index)

4. Altering elements:

i. reverse( ) -> listname.reverse( )
ii. sorted( ) -> sorted(listname), sorted(listname,reverse=True) -> to sort in descending order
iii. sort() -> listname.sort( ) -> to sort the list in itself

5. Accessing elements:

i. listname[index]
ii. List slicing:
    a. listname[:] -> print whole list
    b. listname[start_index:end_index:stepsize]

6. Other operations:

i. listname1 = listname2 + listname3 -> list concatination
ii. membership: in, not in
iii. count( )
iv. len( )

Tuple:

1. Tuple Creation:

i. tuple_name = (1,2,3,4,5)
ii. tuple_name = (1, (2,3,4), [1,'raju',28,'abc']) -> nested tuple, contains tuple and list within a tuple
iii. tuple_name = ('String',) -> comma is needed otherwise it will take as string datatype

2. Operations like Adding, Removing, Altering method does not work as tuples are immutable.

3. Tuple deletion works but it deletes the whole tuple: del tuple_name

4. However we can modify tuple by using nested tuple, i.e. list inside the tuple and modify the list.

tuple_name = (1, (2,3,4), [1,'raju',28,'abc'])
tuple_name[2][1] = 'pallab'

5. Accessing elements:

i. tuple_name[index]
ii. Tuple slicing:
    a. tuple_name[:] -> print whole list
    b. tuple_name[start_index:end_index:stepsize]

6. Other operations:

i. count( ) -> counts number of occurence of a particular element
ii. index(element) -> returns index of the first occurence of the element mentioned
iii. membership -> in, not in
iv. len( )
v. sorted( ) -> does not sort the tuple itself. It stores in a different variable
vi. max( )
vii. min( )
viii. sum( )

Set:

1. Set creation:

i. set_name = {1,2,3,4}
ii. set_name = ([1,2,3,4]) -> set from a list
iii. set_name = set( ) -> initializing a set

2. Adding Elements:

i. add( ) -> set_name.add(element) -> adding single element
ii. update( ) -> set_name.update([5,6,7]) -> adding multiple elements

3. Removing elements:

i. discard( ) -> set_name.discard(element)
ii. remove( ) -> set_name.remove(element)
iii. pop( ) -> set_name.pop( ) -> removes random elements
iv. clear( ) -> set_name.clear( ) -> removes all elements in the set

4. As indexing does not work in set so altering and accessing any particular element is not possible

5. Other operations:

i. set_name1 | set_name2 -> union
    Alt. set_name1.union(set_name2)
ii. set_name1 & set_name2 -> intersection
    Alt. set_name1.intersection(set_name2)
iii. set_name1 - set_name2 -> set difference
    Alt. set_name1.difference(set_name2)
iv. set_name1 ^ set_name2 -> symmetric difference
    Alt. set_name1.symmetric_difference(set_name2)
v. issubset( )
vi. issuper( )
vii. forzenset( ) -> to creates immutable sets

Dictionary:

1. Dictionary creation:

i. dictionary_name = {key1:value1, key2:value2}
ii. dictionary_name = {'name':'xyz', 1:['abc','xyz']} -> dictionary with mixed keys
iii. dictionary_name = dict([(1,'abc'), (2,'xyz')]) -> creating a dictionary with list of tuple

2. Adding/Altering Elements:

i. dictionary_name[key] = value

3. Removing Elements:

i. pop(key) -> dictionary_name.pop(key)
ii. popitem( ) -> dictionary_name.popitem( ) -> removes an arbitary key
iii. del -> del dictionary_name[key], del dictionary_name
iv. clear( )

4. Accessing Elements:

i. dictionary_name[key]
ii. get( ) -> dictionary_name.get(key)

5. Other operations:

i. copy( )
ii. fromkeys([sequence of keys], value)
iii. item( ) -> converts to a new view -> (key,value)
iv. keys( ) -> prints only the keys
v. values( ) -> prints only the values
vi. dir(dictionary_name) ->get list of all available methods and attributes of dictionary

String:

1. String creation:

i. string_name = 'Hello'
ii. string_name = "Hello"
iii. string_name = '"Hello"'

2. Operations like Adding, Removing, Altering method does not work as strings are immutable

3. We cannot delete a particular element(word/alphabet), however we can delete the entrie string
del string_name

4. Other operations:

i. string_name1 + string_name2 -> concatination
ii. membership -> in, not in
iii. lower( )
iv. upper( )
v. split( ) -> very useful
vi. find( )
vii. replace( ) -> it replaces and put it in different variable
viii. reversed( )