

Dictionary:

1. It is ordered collection
2. It stores key:value pair
3. In dictionary keys should be unique
4. It is mutable
5. It is represented with { }

d.get(k,[default value])	It will check whether key exists, if found the return the value of the key If not found and default value is not given then it returns None, otherwise it returns given default value
d.setdefault(k,default value)	It will check whether key exists, if found the return the value of the key If not found, then it will add a new key: value pair and returns the default value
d.pop(key,[default value])	It will delete the given key:value pair if key found If key not found and if default value not given then it will throw exception If key not found and if default value is given then it will return default value
d.popitem()	It will delete the last key:value pair
d.items()	It will return a inerrable object which contains (key,value) as tuple
d.keys()	It will return dict_keys object, which contains list of all keys
d.values()	It will return dict_values object, which contains list of all values
dict.fromkeys(iterable, [value=None])	Return a new dictionary with all the values in the iterable as keys, and if value is not given the all keys will be assigned to None Otherwise all keys will be assigned to value
d1.update(d2)	It will add all keys of d2 in dictionary d1, if the keys are same then the d1 value will be overwritten by d2 value, otherwise new key:value will be added
d.clear()	It will delete all key:value pair, but keeps empty dictionary
d.copy()	It will create shallow copy of the dictionary

del(d["DAC"])-> to delete the key: value pair if key found, otherwise it will throw an exception.