```
>>> # 10/02/2023
>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}
>>> # states is a nested dictionary

>>> dir(states)  # Available methods
['__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__',
'__init__', '__init_subclass__', '__ior__', '__iter__', '__le__',
'__len__', '__lt__', '__ne__', '__new__', '__or__', '__reduce__',
'__reduce_ex__', '__repr__', '__reversed__', '__ror__',
'__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items',
'keys', 'pop', 'popitem', 'setdefault', 'update', 'values']

>>> states.pop('Telanagana')  # Return the value of the specified key
## and remove the key from dictionary
{'capital': 'Hyderabad', 'sweet': 'kurbanika meta'}

>>> states  # Check what has changed in the states dictionary
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}}
>>> states.keys()  # Check whether 'Telanagana' is still a key?
dict_keys(['Karnataka', 'Maharashtra'])

>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}

>>> states.popitem()  # removes the last key-value pair added and
                     # returns it as a tuple.
('Telanagana', {'capital': 'Hyderabad', 'sweet': 'kurbanika meta'})
>>> states["USA"] = "California"

>>> states["Japan"] = "Tokyo"
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'USA': 'California', 'Japan': 'Tokyo'}
>>> states.popitem()
```

```
('Japan', 'Tokyo')
>>> states.keys()
dict_keys(['Karnataka', 'Maharashtra', 'USA'])

>>> states.popitem()
('USA', 'California')
>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}

>>> #  Change key to correct spelling => Replace ''Telanagana'' with
'Telangana'
>>> # .pop(<key name>) => Returns the value & Removes the key from
the dict
>>> # dict_name[<key_name>] = <value_name>
>>> # case1 : <key_name> is present => value is updated
>>> # case2 : <key_name> is not present => a NEW key-value pair is
added to the dictionary

>>> states.keys()
dict_keys(['Karnataka', 'Maharashtra', 'Telanagana'])
>>> 'Telangana' in states.keys()
False

>>> 'Telanagana' in states.keys()
True
>>> # Case 2

>>> # Remove 'Telanagana' and then add 'Telangana' then assign the
    #previous value
>>> states['Telangana'] = states.pop('Telanagana')

>>> 'Telangana' in states.keys()
True
>>> 'Telanagana' in states.keys()
False
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telangana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> states.pop()
Traceback (most recent call last):
  File "<pyshell#30>", line 1, in <module>
```

```
    states.pop()
TypeError: pop expected at least 1 argument, got 0
>>> # Please Remember .pop method is available to both list and
    #dictionary but the way it works is very different
>>> [1,2,3].pop()
3
>>> s = [1,2,3]
>>> s.pop()  # Returns and removes the value at last index
3
>>> s
[1, 2]
>>> s.remove(2)  # List
>>> s
[1]

>>> states['Punjab']  # You can’t access a key that is not already
    # present
Traceback (most recent call last):
  File "<pyshell#37>", line 1, in <module>
    states['Punjab']
KeyError: 'Punjab'
>>> states[['Karnataka']]  # unhashable please research
Traceback (most recent call last):
  File "<pyshell#38>", line 1, in <module>
    states[['Karnataka']]
TypeError: unhashable type: 'list'
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telangana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> states.clear()  # clear all the items of dictionary and make it
an
    # empty dictionary
>>> states
{}
>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
```

```
>>> states['Telanagana']
{'capital': 'Hyderabad', 'sweet': 'kurbanika meta'}
>>> states['Telanagana'].keys()
dict_keys(['capital', 'sweet'])
>>> type(states['Telanagana'])
<class 'dict'>
>>> states['Telanagana'].pop('capital')
'Hyderabad'
>>> {'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> states['Telanagana']
{'sweet': 'kurbanika meta'}
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'sweet': 'kurbanika meta'}}
>>> states = {"Karnataka": {'capital': 'Bengaluru', 'sweet':'Mysore
Pak'}, "Maharashtra": {"capital": "Mumbai", "sweet": ("Jalebi",
"Gulab Jamun")}, 'Telanagana': {'capital': "Hyderabad",
'sweet':'kurbanika meta'}}

>>> states.get('Maharashtra')
{'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab Jamun')}
>>> states.get('Bihar')
>>> states['Bihar']
Traceback (most recent call last):
  File "<pyshell#54>", line 1, in <module>
    states['Bihar']
KeyError: 'Bihar'
>>> states.get('Bihar')
>>> a = states.get('Bihar')
>>> a
>>> print(a)
None
>>> a = states.get('Bihar', -1)  # returns the value specified as 2nd
      #argument
>>> a
-1
>>> a = states.get('Bihar', "State to be added ")
>>> a
'State to be added '

>>> d1 = {'a': 100, 'b': 200, 'c' : 300}
>>> d2 = {'b': "ABCD", 'd': "WXYZ"}
>>> d1
{'a': 100, 'b': 200, 'c': 300}
>>> d2
{'b': 'ABCD', 'd': 'WXYZ'}

# .update() Merges a dictionary with another dictionary or with an
# iterable of key-value pairs.
>>> d1.update(d2)
>>> d1  # The value corresponding to ‘b’ changed from  200 -> ‘ABCD’,
       # a new key ‘d’ is added to the dictionary
{'a': 100, 'b': 'ABCD', 'c': 300, 'd': 'WXYZ'}
>>> d2 = {'b': "ABCD", 'd': "WXYZ"}
>>> d1 = {'a': 100, 'b': 200, 'c' : 300}
>>> d2.update(d1)
>>> d2
{'b': 200, 'd': 'WXYZ', 'a': 100, 'c': 300}
>>> d1 = {'a': 100, 'b': 200, 'c' : 300}
>>> d1
{'a': 100, 'b': 200, 'c': 300}
>>> d1.update(b = 200, p = 1000)
>>> d1
{'a': 100, 'b': 200, 'c': 300, 'p': 1000}
>>> d1
{'a': 100, 'b': 200, 'c': 300, 'p': 1000}
>>> d1 = {'a': 100, 'b': 200, 'c' : 300}
>>> d1.update(b = 2000, p = 1000)
>>> d1
{'a': 100, 'b': 2000, 'c': 300, 'p': 1000}
>>> d2
{'b': 200, 'd': 'WXYZ', 'a': 100, 'c': 300}
>>> d2.update([('d', "Update"),('12', "Twelve")])
>>> d2
{'b': 200, 'd': 'Update', 'a': 100, 'c': 300, '12': 'Twelve'}
>>> states
{'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> d2.update(states)
>>> d2
```

```
{'b': 200, 'd': 'Update', 'a': 100, 'c': 300, '12': 'Twelve',
'Karnataka': {'capital': 'Bengaluru', 'sweet': 'Mysore Pak'},
'Maharashtra': {'capital': 'Mumbai', 'sweet': ('Jalebi', 'Gulab
Jamun')}, 'Telanagana': {'capital': 'Hyderabad', 'sweet': 'kurbanika
meta'}}
>>> # states['Telangana'] = states.pop('Telanagana')

>>> # dict_name[<new_key>] = dict_name.pop(<old_key>) Replace old key
>>> # with new key and preserve the value
>>> d = dict(a=1, b =2)  # one more way to create dict
>>> d
{'a': 1, 'b': 2}
>>>
```