

Dictionary

1. It is based on the concept of hash table or it is the hash table i.e. a table which stores the data in key and value pairs.
2. It is unordered collection of data because there is no indexing.
3. Multiple data type keys are allowed.
4. Can be created by using LISTS and TUPLES.
5. Empty Dict is created by dict(). However, an empty set is only created as set().

```
In [1]: empty_dict = dict()
```

```
In [2]: type(empty_dict)
```

```
Out[2]: dict
```

```
In [3]: len(empty_dict)
```

```
Out[3]: 0
```

```
In [4]: empty_set = {}
```

```
In [6]: type(empty_set), len(empty_set)
```

```
Out[6]: (dict, 0)
```

```
In [1]: players_details = {'name': ['sachin', 'sehwag', 'dhoni', 'warner', 'stokes'],  
                           'country': ['india', 'india', 'india', 'aus', 'eng']}
```

```
In [87]: players_details
```

```
Out[87]: {'name': ['sachin', 'sehwag', 'dhoni', 'warner', 'stokes'],  
          'country': ['india', 'india', 'india', 'aus', 'eng']}
```

```
In [88]: players_details.keys()
```

```
Out[88]: dict_keys(['name', 'country'])
```

```
In [89]: players_details.values()
```

```
Out[89]: dict_values(['sachin', 'sehwag', 'dhoni', 'warner', 'stokes'], ['india', 'india',  
                                'india', 'aus', 'eng'])
```

```
In [90]: players_details['name']
```

```
Out[90]: ['sachin', 'sehwag', 'dhoni', 'warner', 'stokes']
```

Benefit of get method in dict

```
In [91]: players_details.get('name')
```

```
Out[91]: ['sachin', 'sehwag', 'dhoni', 'warner', 'stokes']
```

```
In [92]: players_details['kind']
```

```

KeyError                                Traceback (most recent call last)
<ipython-input-92-6df4817171ba> in <module>
----> 1 players_details['kind']

KeyError: 'kind'

```

```
In [93]: print(players_details.get('kind'))
```

```
None
```

```
In [94]: bowlers = {(1, 'bum bum'),(2, 'lee'),(3,'donald')}
```

```
In [95]: bowlers
```

```
Out[95]: {(1, 'bum bum'), (2, 'lee'), (3, 'donald')}
```

```
In [96]: type(bowlers)
```

```
Out[96]: set
```

As shown in the above scenario, if we don't use a key:value pair inside the flower braces then it will create a SET.

```
In [97]: new_bowlers = dict([(1, 'bum bum'),(2, 'lee'),(3,'donald')])
```

```
In [98]: new_bowlers
```

```
Out[98]: {1: 'bum bum', 2: 'lee', 3: 'donald'}
```

This is the way by which we can create a dictionary using TUPLES and enclosing them inside a LIST.

Add and modify elements in a DICT

```
In [99]: new_bowlers
```

```
Out[99]: {1: 'bum bum', 2: 'lee', 3: 'donald'}
```

```
In [100... new_bowlers[2] = 'bolt'
```

```
In [101... new_bowlers
```

```
Out[101... {1: 'bum bum', 2: 'bolt', 3: 'donald'}
```

```
In [102... new_bowlers[3] = ['donald','nel','ntini','pollock']
```

```
In [103... new_bowlers
```

```
Out[103... {1: 'bum bum', 2: 'bolt', 3: ['donald', 'nel', 'ntini', 'pollock']}
```

```
In [104... new_bowlers[4] = ('whokes','stokes','anderson','broad')
```

```
In [105... new_bowlers
```

```
Out[105... {1: 'bum bum',
 2: 'bolt',
 3: ['donald', 'nel', 'ntini', 'pollock'],
 4: ('whokes', 'stokes', 'anderson', 'broad')}
```

```
In [106... new_bowlers['aus'] = ['warne','lee','mcgrath']
```

```
In [107...] new_bowlers
```

```
Out[107...] {1: 'bum bum',  
            2: 'bolt',  
            3: ['donald', 'nel', 'ntini', 'pollock'],  
            4: ('whokes', 'stokes', 'anderson', 'broad'),  
            'aus': ['warne', 'lee', 'mcgrath']}
```

Delete elements in a DICT

```
In [108...] new_bowlers.pop(2)
```

```
Out[108...] 'bolt'
```

```
In [109...] new_bowlers
```

```
Out[109...] {1: 'bum bum',  
            3: ['donald', 'nel', 'ntini', 'pollock'],  
            4: ('whokes', 'stokes', 'anderson', 'broad'),  
            'aus': ['warne', 'lee', 'mcgrath']}
```

```
In [110...] new_bowlers.popitem()
```

```
Out[110...] ('aus', ['warne', 'lee', 'mcgrath'])
```

```
In [111...] new_bowlers
```

```
Out[111...] {1: 'bum bum',  
            3: ['donald', 'nel', 'ntini', 'pollock'],  
            4: ('whokes', 'stokes', 'anderson', 'broad')}
```

```
In [112...] new_bowlers.update({1: 'bumrah'})
```

```
In [113...] new_bowlers
```

```
Out[113...] {1: 'bumrah',  
            3: ['donald', 'nel', 'ntini', 'pollock'],  
            4: ('whokes', 'stokes', 'anderson', 'broad')}
```

```
In [114...] new_bowlers.update({1: ['bumrah', 'ishu', 'nehra', 'irfan']})
```

```
In [115...] new_bowlers
```

```
Out[115...] {1: ['bumrah', 'ishu', 'nehra', 'irfan'],  
            3: ['donald', 'nel', 'ntini', 'pollock'],  
            4: ('whokes', 'stokes', 'anderson', 'broad')}
```

```
In [116...] temp_dict = new_bowlers
```

```
In [117...] id(new_bowlers)
```

```
Out[117...] 2301623500944
```

```
In [118...] id(temp_dict)
```

```
Out[118...] 2301623500944
```

```
In [119...] dp_dict = new_bowlers.copy()
```

```
In [120...] id(dp_dict)
```

Out[120...] 2301623019776

In [121...] dp_dict

Out[121...] {1: ['bumrah', 'ishu', 'nehra', 'irfan'],
3: ['donald', 'nel', 'ntini', 'pollock'],
4: ('whokes', 'stokes', 'anderson', 'broad')}

In [122...] dp_dict.values() *# return all the values of the keys*

Out[122...] dict_values(['bumrah', 'ishu', 'nehra', 'irfan'], ['donald', 'nel', 'ntini', 'pollock'], ('whokes', 'stokes', 'anderson', 'broad'))

In [123...] dp_dict.items() *# return new view of the dictionary*

Out[123...] dict_items([(1, ['bumrah', 'ishu', 'nehra', 'irfan']), (3, ['donald', 'nel', 'ntini', 'pollock']), (4, ('whokes', 'stokes', 'anderson', 'broad'))])

In [124...] dp_dict.fromkeys([1,2,3],[7,8,9]) *# return new dictionary*

Out[124...] {1: [7, 8, 9], 2: [7, 8, 9], 3: [7, 8, 9]}

In [125...] dp_dict

Out[125...] {1: ['bumrah', 'ishu', 'nehra', 'irfan'],
3: ['donald', 'nel', 'ntini', 'pollock'],
4: ('whokes', 'stokes', 'anderson', 'broad')}

LOOP a dictionary

In [126...] **for** pair **in** dp_dict.items():
 print(pair)
 print('-----')

```
(1, ['bumrah', 'ishu', 'nehra', 'irfan'])
-----
(3, ['donald', 'nel', 'ntini', 'pollock'])
-----
(4, ('whokes', 'stokes', 'anderson', 'broad'))
-----
```

In [127...] **for** pair , value **in** new_bowlers.items():
 print(pair)
 print('-----')
 print(value)

```
1
-----
['bumrah', 'ishu', 'nehra', 'irfan']
3
-----
['donald', 'nel', 'ntini', 'pollock']
4
-----
('whokes', 'stokes', 'anderson', 'broad')
```

DICT Comprehension

In [128...] dict_comp_1 = {k+100:v **for** k , v **in** new_bowlers.items()}

In [129...] dict_comp_1

Out[129...] {101: ['bumrah', 'ishu', 'nehra', 'irfan'],
103: ['donald', 'nel', 'ntini', 'pollock'],
104: ('whokes', 'stokes', 'anderson', 'broad')}

```
In [130...] res = {k+200:v.append('rama') for k , v in new_bowlers.items() if k < 4}

In [131...] res

Out[131...] {201: None, 203: None}

In [132...] new_bowlers

Out[132...] {1: ['bumrah', 'ishu', 'nehra', 'irfan', 'rama'],
3: ['donald', 'nel', 'ntini', 'pollock', 'rama'],
4: ('whokes', 'stokes', 'anderson', 'broad')}

In [133...] del dict_comp_1[101]           # delete or remove the particular index data

In [134...] dict_comp_1

Out[134...] {103: ['donald', 'nel', 'ntini', 'pollock', 'rama'],
104: ('whokes', 'stokes', 'anderson', 'broad')}

In [135...] dict_comp_1.clear()          # clears the dictionary content

In [136...] dict_comp_1

Out[136...] {}

In [137...] new_bowlers

Out[137...] {1: ['bumrah', 'ishu', 'nehra', 'irfan', 'rama'],
3: ['donald', 'nel', 'ntini', 'pollock', 'rama'],
4: ('whokes', 'stokes', 'anderson', 'broad')}

In [138...] del new_bowlers             # Deletes the dictionary
```

STRINGS

1. These are also immutable just like the tuples. So, we cannot change the single or multiple characters of a string.

```
In [139...] name = 'My Name is Rajesh Sharma'

In [140...] name

Out[140...] 'My Name is Rajesh Sharma'

In [141...] print(name)

My Name is Rajesh Sharma

In [144...] name[2:14]

Out[144...] ' Name is Raj'

In [149...] ''.join(name)

Out[149...] 'My Name is Rajesh Sharma'

In [150...] ' '.join(name)

Out[150...] 'M y   N a m e   i s   R a j   e s h   S h a r m a'
```

```
In [154... '@ '.join(name)
```

```
Out[154... 'M@ y@ @ N@ a@ m@ e@ @ i@ s@ @ R@ a@ j@ e@ s@ h@ @ S@ h@ a@ r@ m@ a '
```

```
In [158... ' '.join(name.split(' '))
```

```
Out[158... 'My Name is Rajesh Sharma'
```

```
In [161... '$$'.join(name.split(' '))
```

```
Out[161... 'My$$Name$$is$$Rajesh$$Sharma'
```

```
In [162... ' $$ '.join(name.split(' '))
```

```
Out[162... 'My $$ Name $$ is $$ Rajesh $$ Sharma'
```

```
In [163... name
```

```
Out[163... 'My Name is Rajesh Sharma'
```

```
In [164... name.replace('Rajesh','Raja Babu') # This will create a new string not perform the c
```

```
Out[164... 'My Name is Raja Babu Sharma'
```

```
In [165... name
```

```
Out[165... 'My Name is Rajesh Sharma'
```

```
In [166... del name # Removes or deletes the dictionary
```

```
In [167... name
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-167-9bc0cb2ed6de> in <module>
----> 1 name

NameError: name 'name' is not defined
```