

Dictionary

python Dictionary is an unordered collection of items. While other compound data types have only values as an element, a dictionary has a key: value pair.its similiar as Hash table or hash data structure.

Dic creation

```
my_dic = {} # empty
print(my_dic)
print(type(my_dic))

my_dic = {1: "abc", 2: 'xyz'} #interger keys
print(my_dic)

my_dic = {'name': 'Harsh', 1: ['abe', 'xyz']} # dic with mixed keys
print(my_dic)

my_dic = dict() # create empty dic using dict().

my_dic = dict([(1, 'abc'), (2, 'xyz')]) # element as a list tuple
print(my_dic)

{}
<class 'dict'>
{1: 'abc', 2: 'xyz'}
{'name': 'Harsh', 1: ['abe', 'xyz']}
{1: 'abc', 2: 'xyz'}
```

Dic Access

```
my_dic = {1: 'harsh', 2: 'raj', 3: 'singh'}
print(my_dic[2])
```

raj

```
#if key is not present it gives keyerror
print(my_dic[4])
```

```
-----
-----
KeyError                                Traceback (most recent call
last)
<ipython-input-4-e57cb0c056cd> in <module>
      1 #if key is not present it gives keyerror
----> 2 print(my_dic[4])
```

KeyError: 4

another way of accessing key

```
print(my_dic.get(2))
```

raj

Dic Add and modify elements

```
my_dic = {1: 'harsh', 2: 'raj', 3: 'singh'}
```

```
my_dic[1] = 'shivam'
```

```
print(my_dic)
```

```
my_dic['degree'] = 'PhD'
```

```
print(my_dic)
```

```
{1: 'shivam', 2: 'raj', 3: 'singh'}
```

```
{1: 'shivam', 2: 'raj', 3: 'singh', 'degree': 'PhD'}
```

Dic Delete and Remove Element

```
dic = {1: 'Harsh', 2: 'raj', 3: 'singh', 'degree': 'PhD'}
```

```
print(dic.pop('degree'))
```

```
print(dic)
```

PhD

```
{1: 'Harsh', 2: 'raj', 3: 'singh'}
```

```
dic = {1: 'Harsh', 2: 'raj', 3: 'singh', 'degree': 'PhD'}
```

```
dic.popitem() # popitem() remove an arbitrary key
```

```
print(dic)
```

```
dic.popitem()
```

```
print(dic)
```

```
{1: 'Harsh', 2: 'raj', 3: 'singh'}
```

```
{1: 'Harsh', 2: 'raj'}
```

```
squares = { 2:4, 3:9, 4:16, 5:25, 6:36 }
```

```
del squares[5]
```

```
print(squares)
```

```
{2: 4, 3: 9, 4: 16, 6: 36}
```

```
squares.clear()
```

```
print(squares)
```

```
{}
```

```
squares = { 2:4, 3:9, 4:16, 5:25, 6:36 }
a = squares
del squares
print(a)
print(squares) # name error becасue dict is deleted

{2: 4, 3: 9, 4: 16, 5: 25, 6: 36}
```

```
-----
-----
NameError                                Traceback (most recent call
last)
<ipython-input-13-6c03d66c7227> in <module>
      3 del squares
      4 print(a)
----> 5 print(squares) # name error becасue dict is deleted

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

Dictionary Methods

```
squares = { 2:4, 3:9, 4:16, 5:25, 6:36 }
dic = squares.copy()
print(dic)
```

```
{2: 4, 3: 9, 4: 16, 5: 25, 6: 36}
```

```
# fromkeys[seq[,v]] -> return a new dictionary with key seq and values
subjects = {}.fromkeys(['maths', 'English', 'SST'],5)
print(subjects)
print(subjects['English'])
```

```
{'maths': 5, 'English': 5, 'SST': 5}
5
```

```
subjects = { 2:4, 3:9, 4:16, 5:25, 6:36 }
print(subjects.items())
```

```
dict_items([(2, 4), (3, 9), (4, 16), (5, 25), (6, 36)])
```

```
print(subjects.keys())
```

```
dict_keys([2, 3, 4, 5, 6])
```

```
# get list of all variable methods and attributes of dictionary
```

```
d = {}
print(dir(d))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__',
'__doc__', '__eq__', '__format__', '__ge__', '__getattr__',
'__getitem__', '__gt__', '__hash__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__',
```

```
'__reduce__', '__reduce_ex__', '__repr__', '__setattr__',  
'__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'clear',  
'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem',  
'setdefault', 'update', 'values']
```

Dic Comprehension

```
d = { 2:4, 3:9, 4:16, 5:25, 6:36 }
```

```
for pair in d.items():  
    print(pair)
```

```
(2, 4)  
(3, 9)  
(4, 16)  
(5, 25)  
(6, 36)
```

```
d = { 2:4, 3:9, 4:16, 5:25, 6:36 }
```

```
new_d = { k: v for k, v in d.items() if v > 9}  
print(new_d)
```

```
{4: 16, 5: 25, 6: 36}
```

```
d = { 2:4, 3:9, 4:16, 5:25, 6:36 }
```

```
new_dic = { k+2:v*2 for k, v in d.items() if v > 4}  
print(new_dic)
```

```
{5: 18, 6: 32, 7: 50, 8: 72}
```