

Dictionary

Python dictionary is an unordered collection of items. While other compound data types have only value as an element, a dictionary has a key: value pair.

Dictionary Creation

```
In [1]: #empty dictionary
my_dict = {}

#dictionary with integer keys
my_dict_intergers = {1: 'abc', 2: 'xyz'}
print(my_dict_intergers)

#dictionary with mixed keys
my_dict_diff_keys = {'name': 'satish', 1: ['abc', 'xyz']}
print(my_dict_diff_keys)

#create empty dictionary using dict()
my_dict = dict()

my_dict_tuples = dict([(1, 'abc'), (2, 'xyz')])    #create a dict with list of tuples
print(my_dict_tuples)

{1: 'abc', 2: 'xyz'}
{'name': 'satish', 1: ['abc', 'xyz']}
{1: 'abc', 2: 'xyz'}
```

Dict Access

```
In [4]: my_dict = {'name': 'satish', 'age': 27, 'address': 'guntur'}

#get name
print(my_dict['name'])

#get() also returns a value for the given key
print(my_dict.get('age'))

satish
27
```

Dict Add or Modify Elements

```
In [5]: my_dict = {'name': 'satish', 'age': 27, 'address': 'guntur'}

#update name
my_dict['name'] = 'raju'

print(my_dict)

{'name': 'raju', 'age': 27, 'address': 'guntur'}
```

```
In [6]: #add new key
my_dict['degree'] = 'M.Tech'

print(my_dict)

{'name': 'raju', 'age': 27, 'address': 'guntur', 'degree': 'M.Tech'}
```

Dict Delete or Remove Element

```
In [7]: #create a dictionary
my_dict = {'name': 'satish', 'age': 27, 'address': 'guntur'}

#remove a particular item
print(my_dict.pop('age'))

print(my_dict)

27
{'name': 'satish', 'address': 'guntur'}
```

```
In [8]: squares = {2: 4, 3: 9, 4: 16, 5: 25}

#delete particular key
del squares[2]

print(squares)

{3: 9, 4: 16, 5: 25}
```

```
In [9]: #remove all items
squares.clear()

print(squares)

{}
```

```
In [10]: squares = {2: 4, 3: 9, 4: 16, 5: 25}

#delete dictionary itself
del squares

print(squares) #NameError because dict is deleted

-----
NameError                                Traceback (most recent call last)
C:\Users\SHIVAR~1\AppData\Local\Temp\ipykernel_18520\2970631294.py in <module>
      4 del squares
      5
----> 6 print(squares) #NameError because dict is deleted

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

Dictionary Methods

```
In [11]: squares = {2: 4, 3: 9, 4: 16, 5: 25}

my_dict = squares.copy()
print(my_dict)

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

```
In [12]: #fromkeys[seq[, v]] -> Return a new dictionary with keys from seq and value equal to v (defaults to None).
subjects = {}.fromkeys(['Math', 'English', 'Hindi'], 0)
print(subjects)

{'Math': 0, 'English': 0, 'Hindi': 0}
```

```
In [13]: subjects = {2:4, 3:9, 4:16, 5:25}
print(subjects.items()) #return a new view of the dictionary items (key, value)

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

```
In [14]: subjects = {2:4, 3:9, 4:16, 5:25}
print(subjects.keys()) #return a new view of the dictionary keys

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

```
In [15]: subjects = {2:4, 3:9, 4:16, 5:25}
print(subjects.values()) #return a new view of the dictionary values

dict_values([4, 9, 16, 25])
```

Dict Comrehension

```
In [16]: #Dict comprehensions are just like list comprehensions but for dictionaries

d = {'a': 1, 'b': 2, 'c': 3}
for pair in d.items():
    print(pair)

('a', 1)
('b', 2)
('c', 3)
```

```
In [17]: #Creating a new dictionary with only pairs where the value is larger than 2
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
new_dict = {k:v for k, v in d.items() if v > 2}
print(new_dict)

{'c': 3, 'd': 4}
```