

Python Dictionary

Dictionary in Python is a collection of keys values, used to store data values like a map, which, unlike other data types which hold only a single value as an element.

In some languages it is known as map or associative arrays.

```
dict = { 'name' : 'nitish' , 'age' : 33 , 'gender' : 'male' }
```

Characterstics:

- Mutable
- Indexing has no meaning
- keys can't be duplicated
- keys can't be mutable items

Creating dictionary

```
In [15]: # empty dictionary
d = {}
print(d)
# 1D dictionary
d1 = { 'name' : 'nitish' , 'gender' : 'male' }
print(d1)
# with mixed keys
d2 = {(1,2,3):1, 'hello':'world'}
print(d2)
# 2D dictionary -> JSON
s = {
    'name':'nitish',
    'college':'bit',
    'sem':4,
    'subjects':{
        'dsa':50,
```

```

        'maths':67,
        'english':34
    }
}
print(s)
# using sequence and dict function
d4 = dict([('name','nitish'),('age',32),(3,3)])
print(d4)
# duplicate keys
d5 = {'name':'nitish','name':'rahul'}
print(d5)
# mutable items as keys
d6 = {'name':'nitish',(1,2,3):2}
print(d6)

```

```

{}
{'name': 'nitish', 'gender': 'male'}
{(1, 2, 3): 1, 'hello': 'world'}
{'name': 'nitish', 'college': 'bit', 'sem': 4, 'subjects': {'dsa': 50, 'maths': 67, 'english': 34}}
{'name': 'nitish', 'age': 32, 3: 3}
{'name': 'rahul'}
{'name': 'nitish', (1, 2, 3): 2}

```

Accessing items

```

In [22]: my_dict = {'name': 'Jack', 'age': 26}
# [] - like an indexing
print(my_dict['age'])

# using get function
print(my_dict.get('age'))

print(s['subjects']['maths'])

```

```

26
26
67

```

Adding key-value pair

```
In [28]: d4['gender'] = 'male'
print(d4)
d4['weight'] = 72
print(d4)
s['subjects']['ds'] = 75
print(s)
```

```
{'name': 'nitish', 'age': 32, 3: 3, 'gender': 'male', 'weight': 72}
{'name': 'nitish', 'age': 32, 3: 3, 'gender': 'male', 'weight': 72}
{'name': 'nitish', 'college': 'bit', 'sem': 4, 'subjects': {'dsa': 50, 'maths': 67, 'english': 34, 'ds': 75}}
```

Removing key-value pair

```
In [62]: d = {'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72}
# pop - remove key based pair
d.pop(3)
print(d)

# popitem - remove last key value if dict exist else throw error
d.popitem()
d.popitem()
print(d)

# del - delete key-value or whole dictionary
del d['name']
print(d)
del d

s = {'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72, 'subjects': {'maths': 23, 'gk': 34}}
# clear
s.clear()
print(s)

s = {'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72, 'subjects': {'maths': 23, 'gk': 34}}

del s['subjects']['maths']
print(s)
```

```
{'name': 'nitish', 'age': 32, 'gender': 'male', 'weight': 72}  
{'name': 'nitish', 'age': 32}  
{'age': 32}  
{}  
{'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72, 'subjects': {'gk': 34}}
```

Editing key-value pair

```
In [66]: s = {'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72, 'subjects': {'maths': 23, 'gk': 34}}  
s['subjects']['dsa'] = 80  
s
```

```
Out[66]: {'name': 'nitish',  
          'age': 32,  
          3: 31,  
          'gender': 'male',  
          'weight': 72,  
          'subjects': {'maths': 23, 'gk': 34, 'dsa': 80}}
```

Dictionary operations

- Membership
- Iteration

```
In [75]: print(s)  
         'name' in s  
  
{'name': 'nitish', 'age': 32, 3: 31, 'gender': 'male', 'weight': 72, 'subjects': {'maths': 23, 'gk': 34, 'dsa': 80}}
```

```
Out[75]: True
```

```
In [77]: d = {'name': 'nitish', 'gender': 'male', 'age': 33}  
  
for i in d:  
    print(i, d[i])
```

```
name nitish  
gender male  
age 33
```

Dictionary functions

```
In [81]: # Len/sorted  
d = {'name': 'nitish', 'gender': 'male', 'age': 33}  
print(len(d), sorted(d, reverse=True), max(d))
```

```
3 ['name', 'gender', 'age'] name
```

```
In [83]: # items/keys/values  
print(d)  
print(d.items())  
print(d.keys())  
print(d.values())
```

```
{'name': 'nitish', 'gender': 'male', 'age': 33}  
dict_items([('name', 'nitish'), ('gender', 'male'), ('age', 33)])  
dict_keys(['name', 'gender', 'age'])  
dict_values(['nitish', 'male', 33])
```

```
In [85]: # update  
d = {'name': 'nitish', 'gender': 'male', 'age': 33}  
print(d)  
d1 = {'name': 'Abhishek', 3: 4, 4: 5, 'age': 26}  
  
d.update(d1)  
print(d)
```

```
{'name': 'nitish', 'gender': 'male', 'age': 33}  
{'name': 'Abhishek', 'gender': 'male', 'age': 26, 3: 4, 4: 5}
```

Dictionary comprehension

```
{ key: value for vars in iterable }
```

```
In [93]: # print 1st 10 numbers and their squares
        {i:i**2 for i in range(1,11)}
```

```
Out[93]: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

```
In [95]: distances = {'delhi':1000,'mumbai':2000,'bangalore':3000}
        print(distances.items())
```

```
dict_items([('delhi', 1000), ('mumbai', 2000), ('bangalore', 3000)])
```

```
In [97]: # using existing dict
        distances = {'delhi':1000,'mumbai':2000,'bangalore':3000}
        {key:value*0.62 for (key,value) in distances.items() }
```

```
Out[97]: {'delhi': 620.0, 'mumbai': 1240.0, 'bangalore': 1860.0}
```

```
In [100... # using zip
        days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]
        temp_C = [30.5,32.6,31.8,33.4,29.8,30.2,29.9]

        {i:j for (i,j) in zip(days,temp_C)}
```

```
Out[100... {'Sunday': 30.5,
            'Monday': 32.6,
            'Tuesday': 31.8,
            'Wednesday': 33.4,
            'Thursday': 29.8,
            'Friday': 30.2,
            'Saturday': 29.9}
```

```
In [104... # using if condition
        products = {'phone':10,'laptop':0,'charger':32,'tablet':0}
        {key:value for (key,value) in products.items() if value>0}
```

```
Out[104... {'phone': 10, 'charger': 32}
```

```
In [106... # Nested Comprehension
        # print tables of number from 2 to 4
        {i:{j:i*j for j in range(1,11)} for i in range(2,5)}
```

```
Out[106... {2: {1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18, 10: 20},  
            3: {1: 3, 2: 6, 3: 9, 4: 12, 5: 15, 6: 18, 7: 21, 8: 24, 9: 27, 10: 30},  
            4: {1: 4, 2: 8, 3: 12, 4: 16, 5: 20, 6: 24, 7: 28, 8: 32, 9: 36, 10: 40}}
```

END