

# Dict

August 16, 2024

## 1 Dictionary

1. Dictionary is a mutable data type in Python.
2. A python dictionary is a collection of key and value pairs separated by a colon (:) & enclosed in curly braces {}.
2. KEYS must be unique & VALUES can be duplicate .

### 1.1 Create Dictionary

```
[6]: mydict = dict()  #Empty dictionary
mydict
```

```
[6]: {}
```

```
[10]: mydict = {1:'one',2:'two',3:'three',4:'four'}  # dictionary with integer keys
mydict
```

```
[10]: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

#### 1.1.1 Create dictionary using dict()

```
[15]: newdict = dict({1:'one',2:'two',3:'three',4:'four'})
newdict
```

```
[15]: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
[20]: mydict = {'A':'one' , 'B':'two' , 'C':'three'}  # dictionary with character keys
mydict
```

```
[20]: {'A': 'one', 'B': 'two', 'C': 'three'}
```

```
[24]: mydict = {1:'one', 'A':'Apple',55.55:'float'}  # dictionary with mixed keys
mydict
```

```
[24]: {1: 'one', 'A': 'Apple', 55.55: 'float'}
```

Return Dictionary Keys using keys() method

```
[26]: mydict.keys()
```

```
[26]: dict_keys([1, 'A', 55.55])
```

### Return Dictionary Values using values() method

```
[30]: mydict.values()
```

```
[30]: dict_values(['one', 'Apple', 'float'])
```

### Access each key-value pair within a dictionary

```
[33]: mydict.items()
```

```
[33]: dict_items([(1, 'one'), ('A', 'Apple'), (55.55, 'float')])
```

```
[39]: mydict={1:'one',2:'two','A':['Tanishq','Man','Earth'],'B':['Age','Red']}  
mydict
```

```
[39]: {1: 'one', 2: 'two', 'A': ['Tanishq', 'Man', 'Earth'], 'B': ['Age', 'Red']}
```

### Creating dictionary from a sequence of keys

```
[3]: keys={'a','b','c','d','e'}  
mydict3 = dict.fromkeys(keys)  
mydict3
```

```
[3]: {'a': None, 'e': None, 'b': None, 'c': None, 'd': None}
```

### Creating dictionary from a sequence of keys and values

```
[12]: keys={'a','b','c','d','e'}  
value = 20  
mydict3 = dict.fromkeys(keys,value)  
mydict3
```

```
[12]: {'a': 20, 'e': 20, 'b': 20, 'c': 20, 'd': 20}
```

```
[20]: keys = {1,2,3,4,5}  
value = ['A','B','C','D']  
mydict3 = dict.fromkeys(keys,value)  
mydict3
```

```
[20]: {1: ['A', 'B', 'C', 'D'],  
      2: ['A', 'B', 'C', 'D'],  
      3: ['A', 'B', 'C', 'D'],  
      4: ['A', 'B', 'C', 'D'],  
      5: ['A', 'B', 'C', 'D']}
```

```
[24]: value.append('XYZ')  
mydict3
```

```
[24]: {1: ['A', 'B', 'C', 'D', 'XYZ'],
      2: ['A', 'B', 'C', 'D', 'XYZ'],
      3: ['A', 'B', 'C', 'D', 'XYZ'],
      4: ['A', 'B', 'C', 'D', 'XYZ'],
      5: ['A', 'B', 'C', 'D', 'XYZ']}
```

## 1.2 Accessing Items

```
[30]: mydict = {1:'one' , 2:'two' , 3:'three' , 4:'four'}
      mydict
```

```
[30]: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
[52]: mydict[4] # Access item using key
```

```
[52]: 'four'
```

```
[56]: mydict.get(4) # Access item using get() method
```

```
[56]: 'four'
```

.get( ) will not show error if the given key is not in dictionary

## 1.3 Add, Remove, & Change Items

```
[68]: mydict1 = {'Name':'Tanishq', 'ID':1903, 'Address':'USA'}
      mydict1
```

```
[68]: {'Name': 'Tanishq', 'ID': 1903, 'Address': 'USA'}
```

### Changing Dictionary Items

```
[72]: mydict1['Address'] = 'India'
      mydict1['ID'] = 1907
      mydict1
```

```
[72]: {'Name': 'Tanishq', 'ID': 1907, 'Address': 'India'}
```

```
[75]: dict1 = {'Name': 'TANISHQ'}
      mydict1.update(dict1)
      mydict1
```

```
[75]: {'Name': 'TANISHQ', 'ID': 1907, 'Address': 'India'}
```

```
[79]: mydict1['Job']='CEO' # Adding items in the dictionary
      mydict1
```

```
[79]: {'Name': 'TANISHQ', 'ID': 1907, 'Address': 'India', 'Job': 'CEO'}
```

dict.pop(key) -> removes element with given key from dict

dict.popitem() -> removes last element from dict

```
[86]: mydict4= {1:'one',2:'two',3:'three',4:'four',5:'five'}  
mydict4
```

```
[86]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

```
[90]: mydict4.popitem()
```

```
[90]: (5, 'five')
```

```
[94]: mydict4
```

```
[94]: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
[97]: mydict4.popitem()
```

```
[97]: (4, 'four')
```

```
[99]: mydict4
```

```
[99]: {1: 'one', 2: 'two', 3: 'three'}
```

```
[101]: mydict4.pop(1)
```

```
[101]: 'one'
```

```
[103]: mydict4
```

```
[103]: {2: 'two', 3: 'three'}
```

```
[129]: dict5 = {'Name':'Tanishq', 'ID':1903, 'Address':'USA'}  
dict5
```

```
[129]: {'Name': 'Tanishq', 'ID': 1903, 'Address': 'USA'}
```

```
[133]: del[dict5['ID']] # Removing item using del method  
dict5
```

```
[133]: {'Name': 'Tanishq', 'Address': 'USA'}
```

```
[137]: dict5.clear() # Delete all items of the dictionary using clear method  
dict5
```

```
[137]: {}
```

```
[149]: del dict5 # Delete the dictionary object  
dict5
```

```

-----
NameError                                Traceback (most recent call last)
Cell In[149], line 1
----> 1 del dict5          # Delete the dictionary object
      2 dict5

NameError: name 'dict5' is not defined

```

## 1.4 Copy Dictionary

```

[153]: fruits= {'Apple':'Red', 'Orange':'orange', 'Banana':'Yellow', 'Watermelon':
        ↪'Green'}
        fruits

[153]: {'Apple': 'Red', 'Orange': 'orange', 'Banana': 'Yellow', 'Watermelon': 'Green'}

[155]: ABC = fruits

[159]: id(ABC),id(fruits)

[159]: (2337251419008, 2337251419008)

[161]: # The address of both mydict & mydict1 will be the same because they are equal
        ↪to eachother

[163]: ABC = fruits.copy()

[165]: id(ABC),id(fruits)

[165]: (2337252327744, 2337251419008)

[167]: #Both addresses are different bcuz now both are individually different dict

[169]: fruits['Cherry']='Pink'
        fruits

[169]: {'Apple': 'Red',
        'Orange': 'orange',
        'Banana': 'Yellow',
        'Watermelon': 'Green',
        'Cherry': 'Pink'}

[173]: ABC # Copy of list won't be impacted due to the changes made in the original

[173]: {'Apple': 'Red', 'Orange': 'orange', 'Banana': 'Yellow', 'Watermelon': 'Green'}

```

## 1.5 Loop through a Dictionary

```
[177]: fruits
```

```
[177]: {'Apple': 'Red',  
       'Orange': 'orange',  
       'Banana': 'Yellow',  
       'Watermelon': 'Green',  
       'Cherry': 'Pink'}
```

```
[179]: for i in fruits:  
       print(i, 'colour is', fruits[i])
```

```
Apple colour is Red  
Orange colour is orange  
Banana colour is Yellow  
Watermelon colour is Green  
Cherry colour is Pink
```

here, i-> key & fruits[i] -> values

```
[183]: for i in fruits:  
       print(fruits[i])
```

```
Red  
orange  
Yellow  
Green  
Pink
```

## 1.6 Dictionary Membership

Test if a key is in a dictionary or not.

Membership test can be only done for keys.

```
[188]: fruits
```

```
[188]: {'Apple': 'Red',  
       'Orange': 'orange',  
       'Banana': 'Yellow',  
       'Watermelon': 'Green',  
       'Cherry': 'Pink'}
```

```
[194]: 'Apple' in fruits
```

```
[194]: True
```

```
[192]: 'Kiwi' in fruits
```

```
[192]: False
```

## 1.7 All/Any

1. All: Similar to AND Gate \*

True: If all keys are true

False: If any key is false

2. Any: Similar to OR Gate +

True: If any key is true

False: All keys are false

```
[203]: fruits
```

```
[203]: {'Apple': 'Red',  
       'Orange': 'orange',  
       'Banana': 'Yellow',  
       'Watermelon': 'Green',  
       'Cherry': 'Pink'}
```

```
[205]: all(fruits)
```

```
[205]: True
```

```
[211]: att = {'A':0 , 'B':22, 'C':100, 0:'the'}  
       all(att)
```

```
[211]: False
```

```
[217]: any(fruits)
```

```
[217]: True
```

```
[219]: any(att)
```

```
[219]: True
```

```
[221]: abc = {False:'20', 0:'Zero'}  
       any(abc)
```

```
[221]: False
```