



New  
syllabus  
2020-21



Chapter 14  
Dictionary

Computer Science  
Class XI ( As per CBSE Board)

Visit : [python.mykvs.in](http://python.mykvs.in) for regular updates

# Dictionary



It is an unordered collection of items where each item consist of a key and a value.

It is mutable (can modify its contents ) but Key must be unique and immutable.



```
{'key': 'value'}
```

## Creating A Dictionary

It is enclosed in curly braces {} and each item is separated from other item by a comma(,). Within each item, key and value are separated by a colon (:). Passing value in dictionary at declaration is dictionary initialization.get() method is used to access value of a key

e.g.

```
dict = {'Subject': 'Informatic Practices', 'Class': '11'}
```

## Accessing List Item

```
dict = {'Subject': 'Informatics Practices', 'Class': 11}
```

```
print(dict)
```

```
print ("Subject : ", dict['Subject'])
```

```
print ("Class : ", dict.get('Class'))
```

OUTPUT

```
{'Class': '11', 'Subject': 'Informatics Practices'}
```

```
('Subject : ', 'Informatics Practices')
```

```
('Class : ', 11)
```

# Dictionary

## Iterating / Traversing through A Dictionary

Following example will show how dictionary items can be accessed through loop.

e.g.

```
dict = {'Subject': 'Informatics Practices', 'Class': 11}
for i in dict:
    print(dict[i])
```

OUTPUT

11

Informatics Practices

## Updating/Manipulating Dictionary Elements

We can change the individual element of dictionary.

e.g.

```
dict = {'Subject': 'Informatics Practices', 'Class': 11}
dict['Subject']='computer science'
print(dict)
```

OUTPUT

```
{'Class': 11, 'Subject': 'computer science'}
```



## Deleting Dictionary Elements

**del, pop() and clear()** statement are used to remove elements from the dictionary.

del e.g.

```
dict = {'Subject': 'Informatics Practices', 'Class': 11}
print('before del', dict)
del dict['Class'] # delete single element
print('after item delete', dict)
del dict #delete whole dictionary
print('after dictionary delete', dict)
```

### Output

```
('before del', {'Class': 11, 'Subject': 'Informatics Practices'})
('after item delete', {'Subject': 'Informatics Practices'})
('after dictionary delete', <type 'dict'>)
```

# Dictionary



**pop()** method is used to remove a particular item in a dictionary. **clear()** method is used to remove all elements from the dictionary.

e.g.

```
dict = {'Subject': 'Informatics Practices', 'Class': 11}
print('before del', dict)
dict.pop('Class')
print('after item delete', dict)
dict.clear()
print('after clear', dict)
```

Output

```
('before del', {'Class': 11, 'Subject': 'Informatics Practices'})
('after item delete', {'Subject': 'Informatics Practices'})
('after clear', {})
```



## Built-in Dictionary Functions

S.No.	Function & Description
1	<p><a href="#"><u>len(dict)</u></a> Gives the total length of the dictionary. It is equal to the number of items in the dictionary.</p> <pre><b>dict = {'Name': 'Aman', 'Age': 37}; print ("Length : %d" % len (dict))</b></pre> <p><b>OUTPUT -&gt;2</b></p>
2	<p><a href="#"><u>str(dict)</u></a> Return a printable string representation of a dictionary</p>
3	<p><a href="#"><u>type(variable)</u></a> If variable is dictionary, then it would return a dictionary type.</p>



## Built-in Dictionary Methods

S.No.	Method & Description
1	<b>dict()</b> - creates dictionary <code>x = dict(name = "Aman", age = 37, country = "India")</code> Here x is created as dictionary
2	<b>keys()</b> - returns all the available keys <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>print(x.keys())</code> OUTPUT-> <code>dict_keys(['country', 'age', 'name'])</code>
3	<b>values()</b> - returns all the available values <code>x = dict(name = "Aman", age = 37, country = "India")</code> <code>print(x.values())</code> OUTPUT-> <code>dict_values(['India', 37, 'Aman'])</code>





## Built-in Dictionary Methods

S.No.	Method & Description
4	<p><b>items()</b> - return the list with all dictionary keys with values.</p> <pre>x = dict(name = "Aman", age = 37, country = "India") print(x.items())</pre> <p>OUTPUT-&gt;dict_items([('country', 'India'), ('age', 37), ('name', 'Aman')])</p>
5	<p><b>update()</b>-used to change the values of a key and add new keys</p> <pre>x = dict(name = "Aman", age = 37, country = "India") d1 = dict(age= 39) x.update(d1,state="Rajasthan") print(x)</pre> <p>OUTPUT-{'country': 'India', 'age': 39,'name': 'Aman', 'state': 'Rajasthan'}</p>



## Built-in Dictionary Methods

S.No.	Method & Description
6	<p><b>del</b> -used to remove key</p> <pre>x = dict(name = "Aman", age = 37, country = "India") del x['age'] print(x)</pre> <p>OUTPUT-&gt;{'country': 'India', 'name': 'Aman'}</p> <p>del x -&gt; will remove complete dictionary</p>
7	<p><b>fromkeys()</b> – is used to create dictionary from keys</p> <pre>keys = {'a', 'e', 'i', 'o', 'u' } value = "Vowel" vowels = dict.fromkeys(keys, value) print(vowels)</pre> <p>OUTPUT-&gt; {'i': 'Vowel', 'u': 'Vowel', 'e': 'Vowel', 'a': 'Vowel', 'o': 'Vowel'}</p>

## Built-in Dictionary Methods

S.No.	Method & Description
8	<p><code>copy()</code> - returns a shallow copy of the dictionary.</p> <pre>x = dict(name = "Aman", age = 37, country = "India") y=x.copy() print(y) print(id(x)) print(id(y))</pre> <p>OUTPUT - &gt;{'country': 'India', 'age': 37, 'name': 'Aman'}</p> <p>33047872</p> <p>33047440</p>
9	<p><code>popitem()</code> – removes last item from dictionary</p> <pre>x = dict(name = "Aman", age = 37, country = "India") x.popitem() print(x)</pre> <p>OUTPUT-&gt; {'age': 37, 'name': 'Aman'}</p>

## Built-in Dictionary Methods

S.No.	Method & Description
10	<p><code>setdefault()</code> method returns the value of the item with the specified key. If the key does not exist, insert the key, with the specified value.</p> <pre>x = dict(name = "Aman", country = "India") y=x.setdefault('age',39) print(y) OUTPUT-&gt; 39</pre>
11	<p><code>max()</code> – returns key having maximum value</p> <pre>Tv = {'a':100, 'b':1292, 'c' : 88} Keymax = max(Tv, key=Tv.get) print(Keymax) OUTPUT-&gt; b</pre>
12	<p><code>min()</code>- returns key having minimum value</p>



## Built-in Dictionary Methods

S.No.	Method & Description
13	<pre>sorted- sort by key or value dict1 = {'b':100, 'a':12, 'c' : 88} y = sorted(dict1.items(),key=lambda x: x[1],reverse=True) print(y) OUTPUT-&gt; [('b', 100), ('c', 88), ('a', 12)]</pre>

# Dictionary



**count the number of times a character appears in a given string using a dictionary**

```
input_string = "python.mykvs.in"
```

```
frequencies = {}
```

```
for char in input_string:
```

```
    if char in frequencies:
```

```
        frequencies[char] += 1
```

```
    else:
```

```
        frequencies[char] = 1
```

```
print ("Per char frequency in '{}' is :\n {}".format(input_string,  
str(frequencies)))
```

OUTPUT

Per char frequency in 'python.mykvs.in' is :

```
{'i': 1, 'h': 1, 'k': 1, 'm': 1, 'o': 1, 'n': 2, 'p': 1, 's': 1, 't': 1, 'v': 1, 'y': 2, '!': 2}
```

# Dictionary



**create a dictionary with names of employees, their salary and access them**

```
employees =
```

```
{'Aman':{'salary':'10000'},'Mayur':{'salary':'51000'}}
```

```
employee1 = employees['Aman']
```

```
print(employee1)
```

**OUTPUT**

```
{'salary': '10000'}
```

# Dictionary



## Questions.

1. Create dictionary to store 4 student details with rollno,name,age field.Search student in list.
2. Create dictionary for month and noofdays for a year. User is asked to enter month name and system will show no of days of that month.