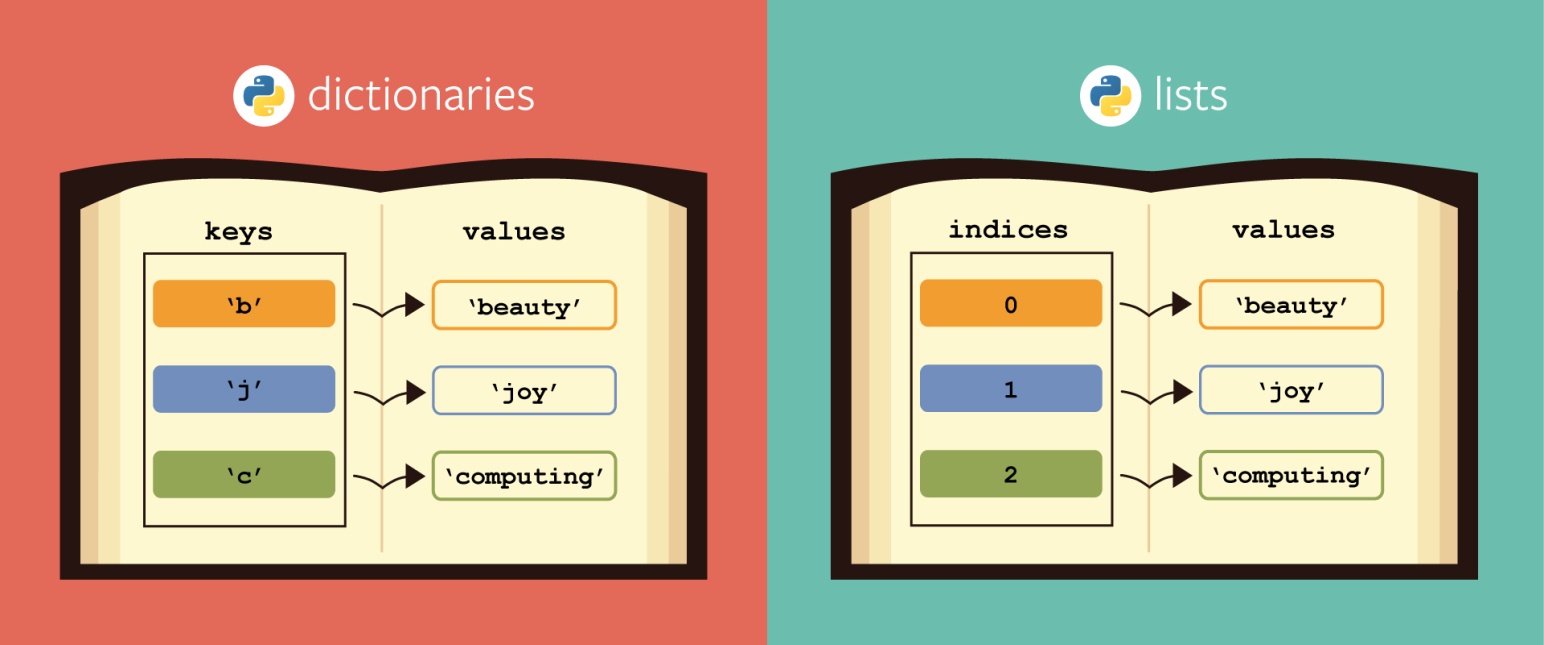
**Working with Lists & Dictionaries in Python**

[Beginner](https://www.analyticsvidhya.com/blog/category/beginner/) Programming [Python](https://www.analyticsvidhya.com/blog/category/python-2/)

This article was published as a part of the Data Science Blogathon

Introduction

Python is a high-level general-purpose programming language. Python is one of the most popular and widely used programming languages in the world Python is used for data analytics, machine learning, and even design. Python is used in a wide range of domains owing to its simple yet powerful nature. In this article, we will be discussing lists & dictionaries datatypes.



*Source: Google Images*

Lists

Lists are mutable data types in Python. Lists is a 0 based index datatype meaning the index of the first element starts at 0. Lists are used to store multiple items in a single variable.

Append the lists

Append is used when you want to add one item to the list.

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

city\_list.append('Bangalore')

print(city\_list)

Output

['Agra', 'Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore', 'Bangalore']

Extend the list

Extend is used when you want to add more than one item to the list

num\_list=[10,20,30,40,50,60,70,80,90]

num\_list.extend([100,110,120,130,140,150])

print(num\_list)

Output

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]

Change the lists

Change is used when you want to update a specific item in a list.

num\_list[7]= 200

print(num\_list)

Output

[10, 20, 30, 40, 50, 60, 70, 200, 90, 100, 110, 120, 130, 140, 150]

Insert in the lists

It is used to insert values at the provided index

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

city\_list.insert(4,'Vadodara')

print(city\_list)

Output

['Agra', 'Chennai', 'Ahmedabad', 'Surat', 'Vadodara', 'Lucknow', 'Coimbatore']

Deleting items in a list

You can remove an item in a list by providing the item index or range that needs to be removed

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

del city\_list[4]

print(city\_list)

Output

['Agra', 'Chennai', 'Ahmedabad', 'Surat', 'Coimbatore']

del city\_list[4:6]

print(city\_list)

Output

['Agra', 'Chennai', 'Ahmedabad', 'Surat']

You can also delete the complete list using the below command

del city\_list

Emptying a list

This is very easy to do as shown below

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

city\_list=[]

print(city\_list)

Output

[]

Methods present in a list

Length

Lenght is used to find how many elements are present inside the list

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

print(len(city\_list))

Output

6

Clear

Clear is used to delete elements in a list

num\_list=[1,2,2,2,3,4,5,6,6,7,6,7]

print(num\_list.clear())

Output

None

Reverse

The reverse function is used to reverse the elements of a list as shown below

city\_list=['Agra','Chennai', 'Ahmedabad', 'Surat', 'Lucknow', 'Coimbatore']

city\_list.reverse()

print(city\_list)

Output

['Coimbatore', 'Lucknow', 'Surat', 'Ahmedabad', 'Chennai', 'Agra']

Sort

Sort is used to sort the elements in a list. By default, it sorts in ascending order, to sort in descending order one parameter needs to be provided in the sort function as shown below

num\_list=[1,2,2,2,3,4,5,6,6,7,6,7]

num\_list.sort(reverse=True)

print(num\_list)

Output

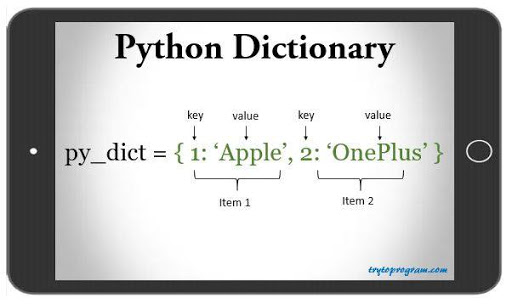
[7, 7, 6, 6, 6, 5, 4, 3, 2, 2, 2, 1]

Dictionary

Dictionaries are mutable data types in nature meaning they can be updated after they are created. Syntactically they are written in a key, value pair format inside curly braces.

{Key1:Value1,Key2:Value2,Key3:Value3,Key4:Value4}

Keys are always unique and there cannot be any duplicates. There is no index in the dictionary meaning they are not ordered. Key is the default iterator and is used to retrieve the value.



How a Dictionary is created?

There are four ways in which a dictionary can be created

1) Empty Dictionary

dict\_emp={}

print(dict\_emp)

Output:

{}

2) When keys are integer values

dict\_city={1: 'Ahmedabad', 2: 'Chennai', 3: 'Coimbatore',4:'Surat',5:'Agra'}

print(dict\_city)

Output

{1: 'Ahmedabad', 2: 'Chennai', 3: 'Coimbatore', 4: 'Surat', 5: 'Agra'}

3) When keys are string values

dict\_travel={'Country1':'USA', 'Country2': 'India', 'Country3':'Japan', 'Country4':'UK', 'Country5': 'Australia'}

print(dict\_travel)

Output

{'Country1': 'USA', 'Country2': 'India', 'Country3': 'Japan', 'Country4': 'UK', 'Country5': 'Australia'}

4) When mixed keys are present

mixed\_dict= {'Country':'India', 1456:'Free','list':['city', 'road',12]}

print(mixed\_dict)

Output

{'Country': 'India', 1456: 'Free', 'list': ['city', 'road', 12]}

Different ways to access list elements

Let’s create two dictionary

dict\_salesid= {'SID1': Fiat

'SID2': Mercedes

'SID3': Maruti

'SID4': Volkswagen

'SID5': Kia}

dict\_salesinfo= {'SID':Fiat,

'Sales': 20000

'LaunchDay':'Wed'

'Cost': 500000}

Length of the dictionary

print(len(dict\_salesid))

Output

5

Extracting dictionary element using its key

sales\_id='SID2'

if sales\_id in dict\_salesid:

name= dict\_salesid[sales\_id]

print('Sales ID is {}, Sales name is {}'. format(sales\_id,name))

else:

print('Sales ID {} not found'.format(sales\_id))

Output

Sales ID is SID2, Sales name is Mercedes

Setting dictionary element using its key

First we have set the dictionary element and then we have retrieved it using get function

dict\_salesinfo['LaunchDay']='Thurs'

dict\_salesinfo['Cost']=6000000

LaunchDay= dict\_salesinfo.get('LaunchDay')

Cost=dict\_salesinfo.get('Cost')

print('Launchday is {}, Cost is {}'. format(LaunchDay,Cost))

Output

Launchday is Thurs, Cost is 6000000

Delete

A dictionary object can be deleted using the del statement

del dict\_salesinfo

Output:

The dictionary object is deleted

Deleting a specific item

To delete a specific item pass on the dictionary key in the del statement as shown below

del dict\_salesinfo['SID']

print(dict\_salesinfo)

Output

{'Sales': 20000, 'LaunchDay': 'Wed', 'Cost': 500000}

Another way to delete a specific item is using the pop function

print(dict\_salesinfo.pop('SID'))

Output

Fiat

The pop function also returns the key value that is being deleted. In this case, it is Fiat.

The third method to delete a dictionary object is using the clear method

print(dict\_salesinfo.clear())

Output

None

Looping through a dictionary object

We will use the same dictionary object dict\_salesinfo. Using the keys function first save all the keys in a dict\_key variable

dict\_keys= dict\_salesinfo.keys()

print(dict\_keys)

print(type(dict\_keys))

Output

dict\_keys(['SID', 'Sales', 'LaunchDay', 'Cost'])

<class 'dict\_keys'>

for var in dict\_keys:

print(var + “:” + str(dict\_salesinfo[var]))

Output

SID:Fiat

Sales:20000

LaunchDay:Wed

Cost:500000

Printing dictionary object values in key, value pair

dict\_values= dict\_salesinfo.values()

print(dict\_values)

Output

dict\_values(['Fiat', 20000, 'Wed', 500000])

Using the items() function

The item function converts a dictionary item into a tuple

dict\_items= dict\_salesinfo.items()

print(dict\_items)

print(type(dict\_items))

Output

dict\_items([('SID', 'Fiat'), ('Sales', 20000), ('LaunchDay', 'Wed'), ('Cost', 500000)])

<class 'dict\_items'>

Looping through the items function

for key, value in dict\_salesinfo.items():

print(key +"-"+str(value))

Output

SID-Fiat

Sales-20000

LaunchDay-Wed

Cost-500000

Converting a list into a dictionary object

Dictionary object contains key-value pairs, and the list must adhere to this format or else it will throw an error.

sales\_infolist=[['SID','Fiat'],['Sales','20000'],['LaunchDay','Wed'],['Cost','500000']]

print(type(sales\_infolist))

sales\_infolist\_dict= dict(sales\_infolist)

print(type(sales\_infolist\_dict))

Output

<class 'list'>

<class 'dict'>

Copying a dictionary into a new dictionary

dict\_salesinfo\_new= dict\_salesinfo.copy()

print(dict\_salesinfo\_new)

Output

{'SID': 'Fiat', 'Sales': 20000, 'LaunchDay': 'Wed', 'Cost': 500000}

Updating the dictionary object

The update method is used to update the dictionary object with a

dict\_salesinfo= {'SID':'Fiat','Sales': 20000,'LaunchDay':'Wed','Cost': 500000}

dict\_salesinfo.update({'Profit':'50000'})

print(dict\_salesinfo)

Output

{'SID': 'Fiat', 'Sales': 20000, 'LaunchDay': 'Wed', 'Cost': 500000, 'Profit': '50000'}