Topics

- Dictionaries

They are another type of Data Structures. It is also called as *map*.

Lists are represented by square brackets and
Tuples are represented by circular brackets and
Dictionaries are represented by curly brackets.

List	**[ ]**	**list ()**
Tuple	**( )**	**tuple ()**
Dictionary	**{ }**	**dict ()**

Dictionary has a property. It has entries in forms of keys & value.

In dictionary, data is represented by

{

key1: value1,
key2: value2,
key3: value3,

}

The first entry is called key and the next would be the corresponding value to the key. A key can only be a string or integer or float and values can be of any data type, it can be int, str, list, tuple or even a dictionary. Dictionaries are mutable.

>>>state_capital_dict = {

… “Delhi”: “Delhi”,

… “Haryana”: “Chandigarh”

… “MP”: “Bhopal”

… }

>>> state_capital_dict

>>> {“Delhi”: “Delhi”, “Haryana”: “Chandigarh”, “MP”: “Bhopal”}

Properties:

Property #1. The keys in a dictionary are always unique.

Property #2. To access a particular key value in a dictionary, we will use array notation.

INPUT:	**OUTPUT**:
`print(state_capital_dict["Haryana"])`	Chandigarh

How to put a new key or update a dictionary?

In list, we used the command, A = [0,1,2,3,4,5]; A [3] = 8

In dictionary,

>>> state_capital_dict = {

... "Delhi": “Delhi",

... "Haryana": “Chandigarh",

... "MP": “Bhopal",

... "Punjab": “Chandigarh",

... "Cities": [1,2,3,4,]

... }

>>> state_capital_dict [“*ABC*”] = “red”

We can add the key *ABC* and its value red to the dictionary if the key is not present early. But if the dictionary has the key *ABC* already in it, it will update its value to red.

Property #3. You can delete a key in a dictionary by using dict.pop(‘key_name’)

```
>>> state_capital_dict.pop('Delhi')
'Delhi'
>>> state_capital_dict
{'Haryana': 'Chandigarh', 'MP': 'Bhopal', 'Punjab': 'Chandigarh', 'Cities': [1, 2, 3,
4]}
```

Property #4. To get all the keys of the dictionary, you use, "**.**keys ()". To get all the values of a dictionary you use, "**.**value ()"

INPUT:

```
state_capital_dict = {
"Delhi": "Delhi",
"Haryana": "Chandigarh",
"MP": "Bhopal",
"Punjab": "Chandigarh",
"Cities": [1,2,3,4,]
}
print(state_capital_dict.keys())
```

OUTPUT:

dict_keys (['Delhi', 'Haryana', 'MP', 'Punjab', 'Cities'])

INPUT:

```
state_capital_dict = {
"Delhi": "Delhi",
"Haryana": "Chandigarh",
"MP": "Bhopal",
"Punjab": "Chandigarh",
"Cities": [1,2,3,4,]
}
print(state_capital_dict.values())
```

OUTPUT:

dict_values (['Delhi', 'Chandigarh', 'Bhopal', 'Chandigarh', [1, 2, 3, 4]])

Property #5: Len for a dictionary will give you the total number of keys in the dictionary.

INPUT:

```
state_capital_dict = {
"Delhi”: “Delhi",
"Haryana”: “Chandigarh",
"MP”: “Bhopal",
"Punjab”: “Chandigarh",
"Cities": [1,2,3,4,]
}
print(len(state_capital_dict))
```

OUTPUT:

5

Property #6: Looping

INPUT

```
state_capital_dict = {
"Delhi”: “Delhi",
"Haryana”: “Chandigarh",
"MP”: “Bhopal",
"Punjab”: “Chandigarh",
"Cities": [1,2,3,4,]
}
for key in state_capital_dict.keys():
    print ("for key", key, "value is", state_capital_dict[key])
```

OUTPUT

for key Delhi value is Delhi

for key Haryana value is Chandigarh

for key MP value is Bhopal

for key Punjab value is Chandigarh

for key Cities value is [1, 2, 3, 4]

Ex: timetable = {

“Monday”: [“English”, “Math”]

“Tuesday”: [“ABC”]

INPUT:

```
state_capital_dict = {
"Delhi”: “Delhi",
"Haryana”: “Chandigarh",
"MP”: “Bhopal",
"Punjab”: “Chandigarh",
"Cities": [1,2,3,4,]
}

for key, value in state_capital_dict.items():
    print (key, value)
```

OUTPUT:

Delhi Delhi

Haryana Chandigarh

MP Bhopal

Punjab Chandigarh

Cities [1, 2, 3, 4]

Keys are unique. You cannot change the name of key; you can delete a key and add another in its place.

Dictionary is always use whenever we want to do some mapping.

01. Given a list with values, [1,2,1,1,1,2,3,4,5,5]

What is the frequency of each value?

1 → 4, 2 → 2, 3 → 1, 4 → 1, 5 → 2

Keyword – 'in'

[1,2,3,4]. If you want to search for a particular element in this array.

3 in [1,2,3,4] → True

33 in [1,2,3,4] → False

It will check if a particular key is present in a dictionary or not. You can only check for keys and not for values. It is case sensitive.

Let's make an empty dictionary and key will be number & value will be frequency. We have to iterate the list first. So,

```
For x in A:
    if x not in dict:
        dict [x] = 1
    else:
        dict [x] += 1
```

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

Explanation

Step	x = **(value)**	Description
1	1	The program will check if the x =1 is in the dict. Since this is the first time, it will add the x=1 as key and its value as 1
2	2	The program will now check for x = 2. Since this is the first time, it will add

		x = 2 as key and its value as 1
3	1	The program will see that x=1 is already present in the dict, so it will go the else statement and increment the value of key 1 as 2.
4	1	The program will see that x=1 is already present in the dict, so it will go the else statement and increment the value of key 1 as 3.
5	1	The program will see that x=1 is already present in the dict, so it will go the else statement and increment the value of key 1 as 4.

02. Given an array, find the maximum value in the array.

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

- I'll take the first value as the maximum value.
- I'll run a loop from 1 to size of this array. I'll check the second value and see if it is great than the first value.
- If it is then I'll keep my first ***val*** as maximum value and will compare the next index in the list.

INPUT

```
for i in range (1, len(a)):
    if a[i] > max_val:
        max_val = a[i]


print(max_val)
```

["Congress", "BJP", "BJP", "BJP", "AAP", "AAP"]

{

"Congress": 1
"BJP": 3
"AAP": 2

}

```
party_dict = dict ()
for party in parties:
    if party in party_dict:
        party_dict[party] += 1
    else:
        party_dict[party] = 1

winning_part = “”
max_vote = 0

for key in party_dict.keys():
    if max_votes < party_dict[key]:
        max_votes = party_dict[key]
        winning_party = key
```