

DICTIONARY

If we want to represent a set/a group of entities as a key value pair, then we should go for a python dictionary.

Properties:

1. Heterogeneous objects are allowed for both keys and values
2. Dynamic and growable in nature
3. Insertion order is not preserved. Therefore indexing and slicing is not allowed.
4. Duplicate values are allowed.

- Creating a dictionary:

```
d={100:"Kabin",200:"Pragyan",300:"Nikita"}
print(type(d))
print(d)
```

Output:

```
<class 'dict'>
{100: 'Kabin', 200: 'Pragyan', 300: 'Nikita'}
```

Note:

In this data type, duplicate values are allowed but duplicate keys are not allowed.

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",100:"isha"}
print(type(d))
print(d)
```

Output:

```
<class 'dict'>
{100: 'isha', 200: 'Pragyan', 300: 'Nikita'}
```

- Accessing elements of a dictionary:

```
d={100:"kabin",200:"Pragyan",399:"Nikita",400:"Pragyan"}
print(d[400])
print(d[399])
```

Output:

```
Pragyan
Nikita
```

- Accessing list as a dictionary:

```
l=[(100,"kabin"),(200,"Isha"),(300,"aayush")]
d=dict(l)
print(d)
print(type(d))
```

Output:

```
{100: 'kabin', 200: 'Isha', 300: 'aayush'}
<class 'dict'>
```

Example:

```
rec={}
n=int(input("enter the number of students"))
i=1
while i<=n:
    name=input("enter the name of students")
    marks=input("enter the marks of the students")
    rec[name]=marks
    i=i+1
print("the records of the students are:")
print(rec)
```

Output:

```
enter the number of students 3
enter the name of students ram
enter the marks of the students 45
enter the name of students shyam
enter the marks of the students 87
enter the name of students hari
enter the marks of the students 27
the records of the students are:
{'ram': '45', 'shyam': '87', 'hari': '27'}
```

- Insertion and Deletion

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}
print(d)
d[500]="vaskar"
print(d)
del d[100]
print(d)
```

Output:

```
{100: 'Kabin', 200: 'Pragyan', 300: 'Nikita'}  
{100: 'Kabin', 200: 'Pragyan', 300: 'Nikita', 500: 'vaskar'}  
{200: 'Pragyan', 300: 'Nikita', 500: 'vaskar'}
```

- **Mathematical Operators:**

1. concatenation operator is not valid
2. repetition/multiplication operator is not valid
3. comparison/relational operator is not valid
4. equality operator is valid (For this to be true, the number of keys, keys and element; all should be equal)
5. membership operator is valid but only for keys

- **Functions in Dictionary:**

1. dict(): to create an empty dictionary.
2. len(): to find the length of the dictionary.
3. clear(): to empty the dictionary.

4. get():

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}  
print(d.get(100))  
print(d.get(600))
```

Output:

```
Kabin  
None
```

5. pop(): to pop out a specific element using a key.

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}  
print(d.pop(100))  
print(d)
```

Output:

```
Kabin  
{200: 'Pragyan', 300: 'Nikita'}
```

6. popitem(): to pop out a random item from the dictionary, no need to use key.
7. keys(): to get the list of keys in dictionary

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}  
print(d.keys())
```

Output:

```
dict_keys([100, 200, 300])
```

8. values(): to get the list of values in dictionary
9. items(): to get the list of both keys and their associated values

Example:

```
rec={}  
n=int(input("enter the number of students : "))  
i=1  
while i<=n:  
    name=input("enter the name of students : ")  
    marks=input("enter the marks of the students : ")  
    rec[name]=marks  
    i=i+1  
print("the records of the students are : ")  
for k,v in rec.items():  
    print(f"The marks for {k} is {v}")
```

Output:

```
enter the number of students : 3  
enter the name of students : ram  
enter the marks of the students : 56  
enter the name of students : shyam  
enter the marks of the students : 67  
enter the name of students : hari  
enter the marks of the students : 96  
the records of the students are :  
The marks for ram is 56  
The marks for shyam is 67  
The marks for hari is 96
```

- Aliasing:

- Copy function:

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}  
e=d.copy()  
d[500]="dipesh"
```

```
print(d)
print(e)
```

Output:

```
{100: 'Kabin', 200: 'Pragyan', 300: 'Nikita', 500: 'dipesh'}
{100: 'Kabin', 200: 'Pragyan', 300: 'Nikita'}
```

- Update function:

```
d={100:"Kabin",200:"Pragyan",300:"Nikita",}
e={100:"dipesh",500:"isha",600:"vaskar",}
d.update(e)
print(d)
```

Output:

```
{100: 'dipesh', 200: 'Pragyan', 300: 'Nikita', 500: 'isha', 600:
'vaskar'}
```

- Dictionary Comprehension:

The process of creating a new dictionary.

```
d= {x:x*x for x in range(10)}
print(d)
```

Output:

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```