SET

DATA STRUCTURE

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

- If we want to represent a group of unique values as a single entity then we should go for set.
- Duplicates are not allowed.
- Insertion order is not preserved.But we can sort the elements.
- Indexing and slicing not allowed for the set.
- Heterogeneous elements are allowed.
- Set objects are mutable i.e once we creates set object we can perform any changes in that object based on our requirement.
- We can represent set elements within curly braces and with comma seperation
- We can apply mathematical operations like union, intersection, difference etc on set objects.

Creation of Set Objects:

```
1) s={10,20,30,40}
2) print(s)
3) print(type(s))
```

Output

```
{40, 10, 20, 30}
<class 'set'>
```

We can create set objects by using set() Function `s = set(any sequence)`

Eg 1:

```
1) l = [10,20,30,40,10,20,10]
2) s=set(l)
3) print(s) # {40, 10, 20, 30}
```

Eg 2:

```
1) s=set(range(5))
2) print(s) #{0, 1, 2, 3, 4}
```

Note:

- While creating empty set we have to take special care.
- Compulsory we should use set() function.
- s = {} → It is treated as dictionary but not empty set.

```
1) s={}
2) print(s)
3) print(type(s))
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Output

```
{}
<class 'dict'>
```

Eg:

```
1) s=set()
2) print(s)
3) print(type(s))
```

Output

```
set()
<class 'set'>
```

Important Functions of Set:

1) add(x):

Adds item x to the set.

```
1) s={10,20,30}
2) s.add(40);
3) print(s)  #{40, 10, 20, 30}
```

2) update(x,y,z):

- To add multiple items to the set.
- Arguments are not individual elements and these are Iterable objects like List, Range etc.
- All elements present in the given Iterable objects will be added to the set.

```
1) s={10,20,30}
2) l=[40,50,60,10]
3) s.update(l,range(5))
4) print(s)
```

Output: {0, 1, 2, 3, 4, 40, 10, 50, 20, 60, 30}

Q) What is the difference between add() and update() Functions in Set?

- We can use add() to add individual item to the Set,where as we can use update() function to add multiple items to Set.
- add() function can take only one argument where as update() function can take any number of arguments but all arguments should be iterable objects.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Q) Which of the following are valid for set s?

1) s.add(10)
2) s.add(10,20,30) → TypeError: add() takes exactly one argument (3 given)
3) s.update(10) → TypeError: 'int' object is not iterable
4) s.update(range(1,10,2),range(0,10,2))

3) copy():

- Returns copy of the set.
- It is cloned object.

```
1) s = {10,20,30}
2) s1 = s.copy()
3) print(s1)
```

4) pop():

It removes and returns some random element from the set.

```
1) s={40,10,30,20}
2) print(s)
3) print(s.pop())
4) print(s)
```

Output

```
{40, 10, 20, 30}
40
{10, 20, 30}
```

5) remove(x):

- It removes specified element from the set.
- If the specified element not present in the Set then we will get KeyError.

```
1) s = {40, 10, 30, 20}
2) s.remove(30)
3) print {◊(s) 40, 10, 20}
4) s.remove(50 KeyError: ◊) 50
```

6) discard(x):

1) It removes the specified element from the set.
2) If the specified element not present in the set then we won't get any error.

```
1) s = {10, 20, 30}
2) s.discard(10)
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

```
3) print {◊(s)  20, 30}
4) s.discard(50)
5) print {◊(s) 20, 30}
```

Q) What is the difference between remove() and discard() functions in Set?
Q) Explain differences between pop(),remove() and discard() functionsin Set?

7) clear():

To remove all elements from the Set.

```
1) s={10,20,30}
2) print(s)
3) s.clear()
4) print(s)
```

Output
{10, 20, 30}
set()

Mathematical Operations on the Set:

1) union():

- **x.union(y) → We can use this function to return all elements present in both sets**
- **x.union(y) OR x|y.**

```
1) x = {10, 20, 30, 40}
2) y = {30, 40, 50, 60}
3) print (x.union(y))  → {10, 20, 30, 40, 50, 60}
4) print (x|y) → {10, 20, 30, 40, 50, 60}
```

2) intersection():

- **x.intersection(y) OR x&y.**
- **Returns common elements present in both x and y.**

```
1) x = {10, 20, 30, 40}
2) y = {30, 40, 50, 60}
3) print (x.intersection(y)) → {40, 30}
4) print(x&y) → {40, 30}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

3) difference():

- x.difference(y) OR x-y.
- Returns the elements present in x but not in y.

```
1) x = {10, 20, 30, 40}
2) y = {30, 40, 50, 60}
3) print (x.difference(y)) → 10, 20
4) print (x-y) → {10, 20}
5) print (y-x) → {50, 60}
```

4) symmetric_difference():

- x.symmetric_difference(y) OR x^y.
- Returns elements present in either x OR y but not in both.

```
1) x = {10, 20, 30, 40}
2) y = {30, 40, 50, 60}
3) print (x.symmetric_difference(y)) → {10, 50, 20, 60}
4) print(x^y) → {10, 50, 20, 60}
```

Membership Operators: (in, not in)

```
1) s=set("durga")
2) print(s)
3) print('d' in s)
4) print('z' in s)
```

Output

```
{'u', 'g', 'r', 'd', 'a'}
True
False
```

Set Comprehension:

Set comprehension is possible.

```
1) s = {x*x  for x in range(5)}
2) print (s)  → {0, 1, 4, 9, 16}
3)
4) s = {2**x  for x in range(2,10,2)}
5) print (s) → {16, 256, 64, 4}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Set Objects won't support indexing and slicing:

```
1) s = {10,20,30,40}
2) print(s[0]) → TypeError: 'set' object does not support indexing
3) print(s[1:3]) → TypeError: 'set' object is not subscriptable
```

Q) Write a Program to eliminate Duplicates Present in the List?

Approach - 1

```
1) l=eval(input("Enter List of values: "))
2) s=set(l)
3) print(s)
```

```
D:\Python_classes>py test.py
Enter List of values: [10,20,30,10,20,40]
{40, 10, 20, 30}
```

Approach - 2

```
1) l=eval(input("Enter List of values: "))
2) l1=[]
3) for x in l:
4)     if x not in l1:
5)         l1.append(x)
6) print(l1)
```

```
D:\Python_classes>py test.py
Enter List of values: [10,20,30,10,20,40]
[10, 20, 30, 40]
```

Q) Write a Program to Print different Vowels Present in the given Word?

```
1) w=input("Enter word to search for vowels: ")
2) s=set(w)
3) v={'a','e','i','o','u'}
4) d=s.intersection(v)
5) print("The different vowel present in",w,"are",d)
```

```
D:\Python_classes>py test.py
Enter word to search for vowels: durga
The different vowel present in durga are {'u', 'a'}
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

DICTIONARY
DATA STRUCTURE

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

- We can use List, Tuple and Set to represent a group of individual objects as a single entity.
- If we want to represent a group of objects as key-value pairs then we should go for Dictionary.

Eg:

- rollno ---- name
- phone number -- address
- ipaddress --- domain name

- Duplicate keys are not allowed but values can be duplicated.
- Hetrogeneous objects are allowed for both key and values.
- Insertion order is not preserved
- Dictionaries are mutable
- Dictionaries are dynamic
- indexing and slicing concepts are not applicable

Note: In C++ and Java Dictionaries are known as "Map" where as in Perl and Ruby it is known as "Hash"

How to Create Dictionary?

- d = {} OR d = dict()
- We are creating empty dictionary. We can add entries as follows

```
1) d[100]="durga"
2) d[200]="ravi"
3) d[300]="shiva"
4) print(d) → {100: 'durga', 200: 'ravi', 300: 'shiva'}
```

- If we know data in advance then we can create dictionary as follows

- d = {100:'durga' ,200:'ravi', 300:'shiva'}
- d = {key:value, key:value}

How to Access Data from the Dictionary?

We can access data by using keys.

```
1) d = {100:'durga',200:'ravi', 300:'shiva'}
2) print(d[100])   #durga
3) print(d[300])   #shiva
```

If the specified key is not available then we will get KeyError

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

print(d[400]) → KeyError: 400

We can prevent this by checking whether key is already available or not by using has_key() function or by using in operator.

d.has_key(400) → Returns 1 if key is available otherwise returns 0

But has_key() function is available only in Python 2 but not in Python 3. Hence compulsory we have to use in operator.

```
if 400 in d:
        print(d[400])
```

Q) Write a Program to Enter Name and Percentage Marks in a Dictionary and Display Information on the Screen

```
1)  rec={}
2)  n=int(input("Enter number of students: "))
3)  i=1
4)  while i <=n:
5)      name=input("Enter Student Name: ")
6)      marks=input("Enter % of Marks of Student: ")
7)      rec[name]=marks
8)      i=i+1
9)  print("Name of Student","\t","% of marks")
10) for x in rec:
11)     print("\t",x,"\t\t",rec[x])
```

```
D:\Python_classes>py test.py
Enter number of students: 3
Enter Student Name: durga
Enter % of Marks of Student: 60%
Enter Student Name: ravi
Enter % of Marks of Student: 70%
Enter Student Name: shiva
Enter % of Marks of Student: 80%

Name of Student          % of marks
---------------         ------------
        durga            60%
        ravi             70 %
        shiva            80%
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

How to Update Dictionaries?

- d[key] = value
- If the key is not available then a new entry will be added to the dictionary with the specified key-value pair
- If the key is already available then old value will be replaced with new value.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d)
3) d[400]="pavan"
4) print(d)
5) d[100]="sunny"
6) print(d)
```

Output

```
{100: 'durga', 200: 'ravi', 300: 'shiva'}
{100: 'durga', 200: 'ravi', 300: 'shiva', 400: 'pavan'}
{100: 'sunny', 200: 'ravi', 300: 'shiva', 400: 'pavan'}
```

How to Delete Elements from Dictionary?

1) del d[key]

- It deletes entry associated with the specified key.
- If the key is not available then we will get KeyError.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d)
3) del d[100]
4) print(d)
5) del d[400]
```

Output

```
{100: 'durga', 200: 'ravi', 300: 'shiva'}
{200: 'ravi', 300: 'shiva'}
KeyError: 400
```

2) d.clear()

To remove all entries from the dictionary.

```
1)  d={100:"durga",200:"ravi",300:"shiva"}
2) print(d)
3) d.clear()
4) print(d)
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Output

```
{100: 'durga', 200: 'ravi', 300: 'shiva'}
{}
```

3) del d

To delete total dictionary.Now we cannot access d.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d)
3) del d
4) print(d)
```

Output

```
{100: 'durga', 200: 'ravi', 300: 'shiva'}
NameError: name 'd' is not defined
```

Important Functions of Dictionary:

1) dict():

To create a dictionary

- d = dict() → It creates empty dictionary
- d = dict({100:"durga",200:"ravi"}) → It creates dictionary with specified elements
- d = dict([(100,"durga"),(200,"shiva"),(300,"ravi")])
  → It creates dictionary with the given list of tuple elements

2) len()

Returns the number of items in the dictionary.

3) clear():

To remove all elements from the dictionary.

4) get():

To get the value associated with the key

d.get(key)

If the key is available then returns the corresponding value otherwise returns None.It wont raise any error.

DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

d.get(key,defaultvalue)

If the key is available then returns the corresponding value otherwise returns default value.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d[100]) → durga
3) print(d[400]) → KeyError:400
4) print(d.get(100)) → durga
5) print(d.get(400)) → None
6) print(d.get(100,"Guest"))  → durga
7) print(d.get(400,"Guest"))  → Guest
```

5) pop():

d.pop(key)

- **It removes the entry associated with the specified key and returns the corresponding value.**
- **If the specified key is not available then we will get KeyError.**

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d.pop(100))
3) print(d)
4) print(d.pop(400))
```

Output

```
durga
{200: 'ravi', 300: 'shiva'}
KeyError: 400
```

6) popitem():

It removes an arbitrary item(key-value) from the dictionaty and returns it.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d)
3) print(d.popitem())
4) print(d)
```

Output

```
{100: 'durga', 200: 'ravi', 300: 'shiva'}
(300, 'shiva')
{100: 'durga', 200: 'ravi'}
If the dictionary is empty then we will get KeyError
d={}
print(d.popitem()) ==>KeyError: 'popitem(): dictionary is empty'
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

7) keys():

It returns all keys associated eith dictionary.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d.keys())
3) for k in d.keys():
4)     print(k)
```

Output

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

8) values():

It returns all values associated with the dictionary.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d.values())
3) for v in d.values():
4)     print(v)
```

Output

```
dict_values(['durga', 'ravi', 'shiva'])
durga
ravi
shiva
```

9) items():

It returns list of tuples representing key-value pairs.

[(k,v),(k,v),(k,v)]

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) for k,v in d.items():
3)     print(k,"--",v)
```

Output

```
100 -- durga
200 -- ravi
300 -- shiva
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

10) copy():

To create exactly duplicate dictionary (cloned copy)

```
d1 = d.copy();
```

11) setdefault():

```
d.setdefault(k,v)
```

- If the key is already available then this function returns the corresponding value.
- If the key is not available then the specified key-value will be added as new item to the dictionary.

```
1) d={100:"durga",200:"ravi",300:"shiva"}
2) print(d.setdefault(400,"pavan"))
3) print(d)
4) print(d.setdefault(100,"sachin"))
5) print(d)
```

Output

```
pavan
{100: 'durga', 200: 'ravi', 300: 'shiva', 400: 'pavan'}
durga
{100: 'durga', 200: 'ravi', 300: 'shiva', 400: 'pavan'}
```

12) update():

d.update(x)

All items present in the dictionary x will be added to dictionary d

Q) Write a Program to take Dictionary from the Keyboard and print the Sum of Values?

```
1) d=eval(input("Enter dictionary:"))
2) s=sum(d.values())
3) print("Sum= ",s)
```

Output

```
D:\Python_classes>py test.py
Enter dictionary:{'A':100,'B':200,'C':300}
Sum=  600
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Q) Write a Program to find Number of Occurrences of each Letter present in the given String?

```
1) word=input("Enter any word: ")
2) d={}
3) for x in word:
4)     d[x]=d.get(x,0)+1
5) for k,v in d.items():
6)     print(k,"occurred ",v," times")
```

Output

```
D:\Python_classes>py test.py
Enter any word: mississippi
m occurred  1  times
i occurred  4  times
s occurred  4  times
p occurred  2  times
```

Q) Write a Program to find Number of Occurrences of each Vowel present in the given String?

```
1) word=input("Enter any word: ")
2) vowels={'a','e','i','o','u'}
3) d={}
4) for x in word:
5)     if x in vowels:
6)         d[x]=d.get(x,0)+1
7) for k,v in sorted(d.items()):
8)     print(k,"occurred ",v," times")
```

Output

```
D:\Python_classes>py test.py
Enter any word: doganimaldoganimal
a occurred  4  times
i occurred  2  times
o occurred  2  times
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

Q) Write a Program to accept Student Name and Marks from the Keyboard and creates a Dictionary. Also display Student Marks by taking Student Name as Input?

```
1)  n=int(input("Enter the number of students: "))
2)  d={}
3)  for i in range(n):
4)      name=input("Enter Student Name: ")
5)      marks=input("Enter Student Marks: ")
6)      d[name]=marks
7)  while True:
8)      name=input("Enter Student Name to get Marks: ")
9)      marks=d.get(name,-1)
10)     if marks== -1:
11)         print("Student Not Found")
12)     else:
13)         print("The Marks of",name,"are",marks)
14)     option=input("Do you want to find another student marks[Yes|No]")
15)     if option=="No":
16)         break
17) print("Thanks for using our application")
```

Output

```
D:\Python_classes>py test.py
Enter the number of students: 5

Enter Student Name: sunny
Enter Student Marks: 90

Enter Student Name: banny
Enter Student Marks: 80

Enter Student Name: chinny
Enter Student Marks: 70

Enter Student Name: pinny
Enter Student Marks: 60

Enter Student Name: vinny
Enter Student Marks: 50

Enter Student Name to get Marks: sunny
The Marks of sunny are 90
```

Do you want to find another student marks[Yes|No]Yes

Enter Student Name to get Marks: durga
Student Not Found

Do you want to find another student marks[Yes|No]No
Thanks for using our application

Dictionary Comprehension:

Comprehension concept applicable for dictionaries also.

```
1) squares={x:x*x for x in range(1,6)}
2) print(squares)
3) doubles={x:2*x for x in range(1,6)}
4) print(doubles)
```

Output

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
{1: 2, 2: 4, 3: 6, 4: 8, 5: 10}