

Dictionary:-

It is an unordered collection of key and value.

```
dict1 = {'name': 'piyush', 'id': 12, 'subject': 'computer'}
```

```
print(dict1)
```

```
print(dict1['name']) #get value by key
```

```
dict1['age'] = 25 # update value by key
```

```
print(dict1)
```

```
del dict1['name'] # delete value by key
```

```
print(dict1)
```

```
dict1.clear() # remove all entries in dict
```

```
print(dict1)
```

```
del dict1 # delete entire dictionary
```

```
print(dict1)
```

```
dict1 = {'name': 'piyush', 'Age': 43, 'Name': 'Majid'}
```

```
print (dict1['name']) #case-sensitive
```

#you can use strings, numbers or tuples as dictionary keys

#but list and sets can't

```
dict1 = {(1,2): 'piyush', 'Age': 43, 'Name': 'Majid'}
```

```
print(dict1)
```

Dictionary functions:-

fromkeys():-

```
x = ('key1', 'key2', 'key3')
```

```
y = 0
```

```
thisdict = dict.fromkeys(x, y)
```

```
print(thisdict)
```

get():-

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

```
x = car.get("model")
```

```
print(x)
```

items():-

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

keys():-

```
x = car.items()
```

```
print(x)
```

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

```
x = car.keys()
```

```
print(x)
```

values():-

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

```
x = car.values()
```

```
print(x)
```

update():-

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

```
car.update({"color": "White"})
```

```
#car1= {"color": "White"}
```

```
#car=car | car1
```

```
print(car)
```

pop():-

```
car = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

```
x = car.pop("model")
```

```
print(x)
```

```
print(car)
```

setdefault():-

```
car = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
x = car.setdefault("model", "Bronco")
print(x)

print(car)

car = {
    "brand": "Ford",
    "year": 1964
}
x = car.setdefault("model", "Bronco")
print(x)

print(car)
```

sets:-

Python also includes a data type for *sets*. A set is an unordered collection with no duplicate elements. Basic uses include membership testing and eliminating duplicate entries. Set objects also support mathematical operations like union, intersection, difference, and symmetric difference

```
>>> basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
```

```
>>> print(basket) # show that duplicates have been removed
```

```
{'orange', 'banana', 'pear', 'apple'}
```

```
>>> 'orange' in basket # fast membership testing
```

```
True
```

```
>>> 'crabgrass' in basket
```

```
False
```

```
>>> fib={1,1,2,3,5,8,13}
```

```
>>> prime={2,3,5,7,11,13}
```

```
>>> fib
{1, 2, 3, 5, 8, 13}

>>> prime
{2, 3, 5, 7, 11, 13}

>>> fib | prime #Union
{1, 2, 3, 5, 7, 8, 11, 13}

>>> fib & prime #intersection
{2, 3, 5, 13}

>>> fib - prime #difference
{8, 1}

>>> fib ^ prime #symmetric difference
{8,1,11,7}
```

Type Casting:-

```
>>> list1=[2,4,6]

>>> t1=tuple(list1)

>>> t1
(2, 4, 6)

>>> s1=set(t1)

>>> s1
{2, 4, 6}

>>> list1=list(s1)

>>> list1
[2, 4, 6]

>>> type(list1)
```

