Sets in Python

```
Days={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}
```

```
print(Days) #Sets are unordered
```

```
{'Wednesday', 'Sunday', 'Thursday', 'Tuesday', 'Saturday', 'Friday', 'Monday'}
```

```
Days=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]  #as list
```

```
print(Days)              #Lists are ordered
```

```
['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
```

```
Days={"Monday",2,1.5,"Wednesday","Thursday","Friday","Saturday","Sunday"}
```

```
Days
```

```
{1.5, 2, 'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Wednesday'}
```

```
type(Days)
```

```
set
```

```
Days[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-aa23c6284ba9> in <module>()
----> 1 Days[0]

TypeError: 'set' object is not subscriptable
```

SEARCH STACK OVERFLOW

```
for i in Days:
  print(i)
```

```
Wednesday
1.5
2
Sunday
Friday
Saturday
Thursday
Monday
```

```
for i in range(len(Days)):
  print(Days[i])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-11-1cc3c12c9799> in <module>()
      1 for i in range(len(Days)):
----> 2   print(Days[i])

TypeError: 'set' object is not subscriptable
```

SEARCH STACK OVERFLOW

```
Days={1,2,"Hello","DAIICT",2.3,(1,2,3)}
```

```
Days
```

```
{(1, 2, 3), 1, 2, 2.3, 'DAIICT', 'Hello'}
```

```
Days={1,2,"Hello","DAIICT",2.3,(1,2,3),1,2,3}
```

```
Days      #sets have unique elements
```

```
{(1, 2, 3), 1, 2, 2.3, 3, 'DAIICT', 'Hello'}
```

```
#using set() method
Days=set([1,2,"Hello","DAIICT",2.3,(1,2,3),1,2,3])
```

```
Days
```

```
{(1, 2, 3), 1, 2, 2.3, 3, 'DAIICT', 'Hello'}
```

```
A=[1,2,"Hello","DAIICT",2.3,(1,2,3),1,2,3]
```

```
set(A)
```

```
{(1, 2, 3), 1, 2, 2.3, 3, 'DAIICT', 'Hello'}
```

```
A=[1,2,"Hello","DAIICT",(1,2),1,2,2.5,[1,2,3],1,2]
```

```
set(A)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-22-18cd15bec24f> in <module>()
----> 1 set(A)

TypeError: unhashable type: 'list'
```

SEARCH STACK OVERFLOW

```
A=[1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5]
print(set(A))
```

```
{1, 2, 3, 4, 5}
```

```
set1=set()
print(set1)
print(type(set1))
```

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

```
Days
```

```
{(1, 2, 3), 1, 2, 2.3, 3, 'DAIICT', 'Hello'}
```

```
Days[1]=2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-27-d0b30e8f5a4e> in <module>()
----> 1 Days[1]=2

TypeError: 'set' object does not support item assignment
```

SEARCH STACK OVERFLOW

```
#Adding the elements in set
Days=set(["Sunday","Monday","Tuesday","Wednesday","Thursday"])
print("Original Declared Set")
print(Days)
print("Adding the elements in the set Days")
Days.add("Friday")
Days.add("Saturday")
print("Modified Set")
print(Days)
Days.add("Hello","DAIICT")
print(Days)
```

```
Original Declared Set
{'Wednesday', 'Sunday', 'Tuesday', 'Thursday', 'Monday'}
Adding the elements in the set Days
Modified Set
{'Wednesday', 'Sunday', 'Tuesday', 'Friday', 'Saturday', 'Thursday', 'Monday'}
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-30-64bae78f6ba6> in <module>()
      8 print("Modified Set")
      9 print(Days)
---> 10 Days.add("Hello","DAIICT")
     11 print(Days)

TypeError: add() takes exactly one argument (2 given)
```

SEARCH STACK OVERFLOW

```
Days.add(set(["Hello","DAIICT"]))
print(Days)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-32-c1a86d42685a> in <module>()
----> 1 Days.add(set(["Hello","DAIICT"]))
      2 print(Days)

TypeError: unhashable type: 'set'
```

```
Days.add((1,2,3))
```

```
Days
```

```
{(1, 2, 3),
 'Friday',
 'Monday',
 'Saturday',
 'Sunday',
 'Thursday',
 'Tuesday',
 'Wednesday'}
```

```
#update() to add multiple elements
Days=set(["Monday","Tuesday","Wednesday","Thursday"])
print("original Declared Set: ", Days)
print("Adding Multiple Elements")
Days.update(["Friday","Saturday","Sunday"])
print("Modified set is: ",Days)
```

```
original Declared Set:  {'Wednesday', 'Tuesday', 'Monday', 'Thursday'}
Adding Multiple Elements
Modified set is:  {'Wednesday', 'Sunday', 'Thursday', 'Tuesday', 'Saturday', 'Friday', 'Monday
```

```
Days
```

```
{'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
```

```
Days.update("DAIICT")
```

```
Days
```

```
{'A',
 'C',
 'D',
 'Friday',
 'I',
 'Monday',
 'Saturday',
 'Sunday',
 'T',
 'Thursday',
 'Tuesday',
 'Wednesday'}
```

```
Days.update("1")
```

```
Days
```

```
{'1',
 'A',
 'C',
 'D',
 'Friday',
 'I',
 'Monday',
 'Saturday',
 'Sunday',
 'T',
 'Thursday',
 'Tuesday',
 'Wednesday'}
```

```
Numbers=set([1,2,3,4,5,6])
Numbers.update([7,8,9])
print(Numbers)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
Numbers.add(10)
```

```
Numbers
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
Numbers.update(["Hello","DAIICT",(1,2,3),4.5])
```

```
Numbers
```

```
{(1, 2, 3), 1, 10, 2, 3, 4, 4.5, 5, 6, 7, 8, 9, 'DAIICT', 'Hello'}
```

```
Numbers.update([11])
```

```
Numbers
```

```
{(1, 2, 3), 1, 10, 11, 2, 3, 4, 4.5, 5, 6, 7, 8, 9, 'DAIICT', 'Hello'}
```

```
Numbers.update('3456')
```

```
Numbers
```

```
{(1, 2, 3),
 1,
 10,
 11,
 2,
 3,
 '3',
 4,
 '4',
 4.5,
 5,
```

```
 '5',
 6,
 '6',
 7,
 8,
 9,
 'DAIICT',
 'Hello'}
```

```
Numbers.update(['3456','123'])
#'3456' --> '3','4','5','6'
#'3456','123' --> '3','4','5','6','1','2','3'
```

```
Numbers
```

```
{(1, 2, 3),
 1,
 '1',
 10,
 11,
 '123',
 2,
 '2',
 3,
 '3',
 '3456',
 4,
 '4',
 4.5,
 5,
 '5',
 6,
 '6',
 7,
 8,
 9,
 'DAIICT',
 'Hello'}
```

```
#Removing the values from the set--> discard()
Days={'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
print("Original Set: ",Days)
print("Removing some day")
Days.discard("Friday")
print("Modified set: ",Days)
```

```
Original Set:  {'Wednesday', 'Sunday', 'Tuesday', 'Friday', 'Saturday', 'Thursday', 'Monday'}
Removing some day
Modified set:  {'Wednesday', 'Sunday', 'Tuesday', 'Saturday', 'Thursday', 'Monday'}
```

```
Days.discard("Hello")
```

```
Days
```

```
{'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
```

```
#remove() to remove the element from the set
```

```
Days={'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
print("Original Set: ",Days)
print("Removing some day")
Days.remove('Monday')
Days.remove('Friday')
print("Modified Set: ",Days)
print("Removing the element not present in the set")
Days.remove("hello")
```

```
Original Set:  {'Wednesday', 'Sunday', 'Tuesday', 'Friday', 'Saturday', 'Thursday', 'Monday'}
Removing some day
Modified Set:  {'Wednesday', 'Sunday', 'Tuesday', 'Saturday', 'Thursday'}
Removing the element not present in the set
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-61-75ed65a27dfe> in <module>()
      7 print("Modified Set: ",Days)
      8 print("Removing the element not present in the set")
----> 9 Days.remove("hello")

KeyError: 'hello'
```

SEARCH STACK OVERFLOW

```
#pop() to remove the element --> we can't determine which element will be removed
Days={'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
print("Original Set: ",Days)
print("Removing some day")
Days.pop()
print("Modified Set: ",Days)
```

```
Original Set:  {'Wednesday', 'Sunday', 'Tuesday', 'Friday', 'Saturday', 'Thursday', 'Monday'}
Removing some day
Modified Set:  {'Sunday', 'Tuesday', 'Friday', 'Saturday', 'Thursday', 'Monday'}
```

```
Days
```

```
{'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday'}
```

```
#Removing all the values
Days.clear()
```

```
Days
```

```
set()
```

Mathematical Operations in Set

```
Days1={"Monday","Wednesday","Friday","Sunday"}
Days2={"Sunday","Tuesday","Thursday","Saturday"}
```

```
#union operation
Days1.union(Days2)
```

```
{'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
```

```
Days1|Days2
```

```
{'Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday', 'Wednesday'}
```

```
#intersection operation
Days1.intersection(Days2)
```

```
{'Sunday'}
```

```
Days1&Days2
```

```
{'Sunday'}
```

```
Days1={"Monday","Wednesday","Friday","Sunday"}
Days2={"Sunday","Tuesday","Thursday","Saturday"}
Days3={"Tuesday","Wedesday","Sunday"}
Days1.intersection(Days2,Days3)
```

```
{'Sunday'}
```

```
#symmetric difference operation
Days1-Days2
```

```
{'Friday', 'Monday', 'Wednesday'}
```

```
Days2
```

```
{'Saturday', 'Sunday', 'Thursday', 'Tuesday'}
```

```
Days1.symmetric_difference(Days2)
```

```
{'Friday', 'Monday', 'Saturday', 'Thursday', 'Tuesday', 'Wednesday'}
```

```
#Comparison between the sets
Days1={'A','B','C','D'}
Days2={'C','D','E','F'}
Days3={'A','G','H'}
Days4={'C','D'}
```

```
Days1>Days2
```

```
False
```

```
Days1>Days4  #Days1 is a superset of Days4
```

```
True
```

```
Days1.issuperset(Days4)
```

```
True
```

```
Days4<Days2  #Days4 is a subset of Days2
```

```
     True
```

```
Days4.issubset(Days2)
```

```
     True
```

```
Days1==Days2
```

```
     False
```

```
Days1.isdisjoint(Days2)
```

```
     False
```

```
Days3.isdisjoint(Days4)
```

```
     True
```

```
Days3
```

```
     {'A', 'G', 'H'}
```

```
'A' in Days3
```

```
     True
```

```
'a' in Days3
```

```
     False
```

```
#find the number of unplaced students
students=['Ram','Shyam','Mohan','Kiran','Riya']
placed_Students=['Ram','Riya','Shyam']
not_placed_students=set(students)-set(placed_Students)
print(not_placed_students)
print("number of students not placed: ",len(not_placed_students))
```

```
     {'Kiran', 'Mohan'}
     number of students not placed:  2
```

Application of NLP

```
#NLP=Natural Language Processing
#Fields: Data Science, Data Analytics, Power Electronics
ds_text='Data science is an interdisciplinary field that uses scientific methods processes algorithm
da_text="Data analysis is a process of inspecting cleansing transforming and modelling data with the
power_text='Power electronics is the application of solid state electronics to the control and conve
```

```
ds_set=set(ds_text.split(' '))
```

```
da_set=set(da_text.split(' '))
power_set=set(power_text.split(' '))
```

```
print("data science vs data analytics")
print(ds_set.intersection(da_set))
print(len(ds_set.intersection(da_set)))
```

```
data science vs data analytics
{'is', 'of', 'and', 'science', 'data', 'scientific', 'a', 'Data', 'domains'}
9
```

```
print("data science vs power electronics")
print(ds_set.intersection(power_set))
print(len(ds_set.intersection(power_set)))
```

```
data science vs power electronics
{'systems', 'is', 'of', 'and', 'application', 'data', 'to'}
7
```

```
print("data analytics vs power electronics")
print(da_set.intersection(power_set))
print(len(da_set.intersection(power_set)))
```

```
data analytics vs power electronics
{'with', 'is', 'of', 'In', 'and', 'the', 'data', 'in'}
8
```

Dictionary in Python

```
Dict={"Name":"Ram","Age": 25,"Institute":"DAIICT"}
```

```
Dict
```

```
{'Age': 25, 'Institute': 'DAIICT', 'Name': 'Ram'}
```

```
Dict['Name']
```

```
'Ram'
```

```
Dict['Age']
```

```
25
```

```
Dict['Age']=28
```

```
Dict
```

```
{'Age': 28, 'Institute': 'DAIICT', 'Name': 'Ram'}
```

```
#(key,value) format for dictionary
Dict=dict([(1,'One'),(2,'Two'),(3,'Three')])
```

```
# format for dictionary Dict[key]=value
```

```
Dict
```

```
{1: 'One', 2: 'Two', 3: 'Three'}
```

```
Dict[3]='Four'
```

```
Dict
```

```
{1: 'One', 2: 'Two', 3: 'Four'}
```

```
Dict[5]='five'
```

```
Dict
```

```
{1: 'One', 2: 'Two', 3: 'Four', 5: 'five'}
```

```
Dict[6]='six','SIX','Six'
```

```
Dict
```

```
{1: 'One', 2: 'Two', 3: 'Four', 5: 'five', 6: ('six', 'SIX', 'Six')}
```

```
Dict[7]=['six','SIX','Six']
```

```
Dict
```

```
{1: 'One',
 2: 'Two',
 3: 'Four',
 5: 'five',
 6: ('six', 'SIX', 'Six'),
 7: ['six', 'SIX', 'Six']}
```

```
Dict[7]
```

```
['six', 'SIX', 'Six']
```

```
Dict[7].append('Seven')
```

```
Dict
```

```
{1: 'One',
 2: 'Two',
 3: 'Four',
 5: 'five',
 6: ('six', 'SIX', 'Six'),
 7: ['six', 'SIX', 'Six', 'Seven']}
```

```
Dict={}
```

```
#Dictionary of student database
Dict['student_name']='Ram'
Dict['Age']=27
Dict['Affiliation']='DAIICT'
Dict['Residence']='Ahmedabad'
```

```
Dict
```

```
{'Affiliation': 'DAIICT',
 'Age': 27,
 'Residence': 'Ahmedabad',
 'student_name': 'Ram'}
```

```
#Dictionary of student database
Dict['student_name']=['Ram']
Dict['Age']=[27]
Dict['Affiliation']=['DAIICT']
Dict['Residence']=['Ahmedabad']
```

```
Dict
```

```
{'Affiliation': ['DAIICT'],
 'Age': [27],
 'Residence': ['Ahmedabad'],
 'student_name': ['Ram']}
```

```
#Dictionary of student database
Dict['student_name'].append('Shyam')
Dict['Age'].append(29)
Dict['Affiliation'].append('IIT Gandhinagar')
Dict['Residence'].append('Gandhinagar')
```

```
Dict
```

```
{'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [27, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
print(type(Dict['Age'][0]))
print(Dict['Age'][0])
```

```
<class 'int'>
27
```

```
Dict['Age'][0]=28
```

```
Dict
```

```
{'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [28, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
#delete the key from dictionary
del Dict['Age']
```

```
Dict
```

```
{'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
del Dict['Affiliation'][0]
```

```
Dict
```

```
{'Affiliation': ['IIT Gandhinagar'],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
#pop() deletes the element corresponding to a particular key value
Dict.pop('Affiliation')
```

```
['IIT Gandhinagar']
```

```
Dict
```

```
{'Residence': ['Ahmedabad', 'Gandhinagar'], 'student_name': ['Ram', 'Shyam']}
```

```
#Dictionary of student database
Dict['student_name']=['Ram','Shyam']
Dict['Age']=[27,29]
Dict['Affiliation']=['DAIICT','IIT Gandhinagar']
Dict['Residence']=['Ahmedabad','Gandhinagar']
```

```
Dict
```

```
{'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [27, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
#print keys in dict
for i in Dict:
  print(i)
```

```
student_name
Residence
Age
Affiliation
```

```
#print values in dict
for i in Dict:
  print(Dict[i])
```

```
['Ram', 'Shyam']
```

```
['Ahmedabad', 'Gandhinagar']
[27, 29]
['DAIICT', 'IIT Gandhinagar']
```

```
#to print the items of the dictionary
for i in Dict.items():
  print(i)
  print(type(i))
```

```
('student_name', ['Ram', 'Shyam'])
<class 'tuple'>
('Residence', ['Ahmedabad', 'Gandhinagar'])
<class 'tuple'>
('Age', [27, 29])
<class 'tuple'>
('Affiliation', ['DAIICT', 'IIT Gandhinagar'])
<class 'tuple'>
```

```
#cannot declare list as a key
Dict1={1:'One',2:['TWO','two'],[1,2,3]:'three'}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-149-86002888e49c> in <module>()
      1 #cannot declare list as a key
----> 2 Dict1={1:'One',2:['TWO','two'],[1,2,3]:'three'}

TypeError: unhashable type: 'list'
```

SEARCH STACK OVERFLOW

```
#cannot declare list as a key
Dict1={1:'One',2:['TWO','two'],(1,2,3):'three'}
```

```
Dict1
```

```
{(1, 2, 3): 'three', 1: 'One', 2: ['TWO', 'two']}
```

```
#to find the length of the dictionary
len(Dict1)
```

```
3
```

```
Dict2=Dict1.copy()
```

```
Dict2
```

```
{(1, 2, 3): 'three', 1: 'One', 2: ['TWO', 'two']}
```

```
Dict
```

```
{'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [27, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
Dict['Age']
```

```
[27, 29]
```

```
Dict['age']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-157-aa8447edc0a6> in <module>()
----> 1 Dict['age']

KeyError: 'age'
```

SEARCH STACK OVERFLOW

```
#get() to get the values from the key in the dictionary
Dict.get('Age')
```

```
[27, 29]
```

```
Dict.get('age')
```

```
Dict.get('Age','not found the key')
```

```
[27, 29]
```

```
Dict.get('age','not found the key')
```

```
'not found the key'
```

```
#to get the keys
Dict.keys()
```

```
dict_keys(['student_name', 'Residence', 'Age', 'Affiliation'])
```

```
#to get the values
Dict.values()
```

```
dict_values([['Ram', 'Shyam'], ['Ahmedabad', 'Gandhinagar'], [27, 29], ['DAIICT', 'IIT
Gandhinagar']])
```

```
Dict1
```

```
{(1, 2, 3): 'three', 1: 'One', 2: ['TWO', 'two']}
```

```
Dict.update(Dict1)
```

```
Dict
```

```
{(1, 2, 3): 'three',
 1: 'One',
 2: ['TWO', 'two'],
```

```
 'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [27, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
Dict.update(Dict1)
```

```
Dict[1]='DAIICT'
```

```
Dict
```

```
{(1, 2, 3): 'three',
 1: 'DAIICT',
 2: ['TWO', 'two'],
 'Affiliation': ['DAIICT', 'IIT Gandhinagar'],
 'Age': [27, 29],
 'Residence': ['Ahmedabad', 'Gandhinagar'],
 'student_name': ['Ram', 'Shyam']}
```

```
inventory={'shirts':25,'pents':25,'tshirts':50,'shoes':100}
print("inventory items: ",inventory)
```

```
inventory items:  {'shirts': 25, 'pents': 25, 'tshirts': 50, 'shoes': 100}
```

```
#popitem()
inventory.popitem()
```

```
('shoes', 100)
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'tshirts': 50}
```

```
inventory['shocks']=50
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'shocks': 50, 'tshirts': 50}
```

```
inventory['shocks']=inventory['shocks']-2
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'shocks': 48, 'tshirts': 50}
```

```
inventory['shocks']=inventory['shocks']-48
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'shocks': 0, 'tshirts': 50}
```

```
inventory.pop('shocks')
```

```
0
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'tshirts': 50}
```

```
inventory['total items']=len(inventory)
```

```
inventory
```

```
{'pents': 25, 'shirts': 25, 'total items': 3, 'tshirts': 50}
```

```

```