

The image features a stack of books on the left side, with their spines and pages visible. The books have a light brown or tan color. The background is a solid, light blue color. The word 'Set' is written in a large, dark blue font in the center-right area of the image.

# Set

A set contains an unordered collection of unique and immutable objects. The set data type is, as the name implies, a Python implementation of the sets as they are known from mathematics.



# Operations and Manipulation

```
x= { 14,21,True,21,'abc'}
print ( x)
# Indexing not possible
#print ( x[1])
# Membership
if 21 in x:
    print ( True)
# Add a value
x.add ( 15)
print ( x)
# Remove any value
x.pop ()
print ( x)
# Remove specific value
x.remove ( 21)
print ( x)
# Unique function# Union combine two sets with |
print ( { 12,14,15,21,54} | { 21,15,23} )
# Intersection with &
print ( { 12,14,15,21,54} & { 21,15,23} )
# Defferentiation with - Return unique value of left set
print ( { 12,14,15,21,54} - { 21,15,23} )
# Symetry differentiation with ^ return none commn value from both
print ( { 12,14,15,21,54} ^ { 21,15,23} )
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017,
Type "copyright", "credits" or "license () " for i
>>>
===== RESTART: D:/Docu
{ True, 'abc', 21, 14}
True
{ True, 'abc', 14, 15, 21}
{'abc', 14, 15, 21}
{'abc', 14, 15}
{ 21, 54, 23, 12, 14, 15}
{ 21, 15}
{ 54, 12, 14}
{ 12, 14, 54, 23}
>>> |
```



Program to test whether every element in **S** is in **T** and every element in **T** is in **S**.

Set\_find\_subset\_superset.py - D:\Documents\Python\Set\_find\_subset\_superset.py (3.6.4)

File Edit Format Run Options Window Help

```
# Issubset & issuperset
# Program to test whether every element in S is in T and every element in T is in S.
setx = set(["apple", "mango"])
# we can define it also setx= {"apple", "mango"}
sety = set(["mango", "orange"])
setz = set(["mango"])
issubset = setx <= sety
print (issubset)
issuperset = setx >= sety
print (issuperset)
issubset = setz <= sety
print (issubset)
issuperset = sety >= setz
print (issuperset)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 0
Type "copyright", "credits" or "license () " for m
>>>
===== RESTART: D:\Documents\Python\
False
False
True
True
>>> |
```



## Program create a Shallow copy of a set

<p>Set_shallow_copy.py - D:/Documents/Python/Set_shallow_copy.py (3.6.4)</p> <p>File Edit Format Run Options Window Help</p> <pre># Shallow copy setx= { "Rajat",12,True,5.21, (1,"Cetpa") } scopy=setx.copy () print (scopy)</pre>	<p>Python 3.6.4 Shell</p> <p>File Edit Shell Debug Options Window Help</p> <p>Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) Type "copyright", "credits" or "license () " for more information</p> <pre>&gt;&gt;&gt; ===== RESTART: D:/Documents/Python/Set { True, 5.21, (1, 'Cetpa') , 'Rajat', 12}</pre>
---	---



## Program to create a frozenset

<pre>Set_frozen.py - D:/Documents/Python/Set_frozen.py (3.6.4) File Edit Format Run Options Window Help  # Program to use of frozensets. # Frozensets are like sets except that they cannot be changed, i.e. they are immutable: x = frozenset ([1, 2, 3, 4, 5]) y = frozenset ([3, 4, 5, 6, 7]) #use isdisjoint () . Return True if the set has no elements in common with other. print (x.isdisjoint (y) ) #use difference () . Return a new set with elements in the set that are not in the others. print (x.difference (y) ) #new set with elements from both x and y print (x   y)</pre>	<pre>Python 3.6.4 Shell File Edit Shell Debug Options Window Help  Python 3.6.4 (v3.6.4:d48eceb, Dec Type "copyright", "credits" or "license &gt;&gt;&gt; ===== RESTART: D:/I False frozenset ( { 1, 2} ) frozenset ( { 1, 2, 3, 4, 5, 6, 7} ) &gt;&gt;&gt;  </pre>
--	---



## Program to find maximum and minimum value of set

<pre>Set_max_min.py - D:/Documents/Python/Set_max_min.py (3.6.4) File Edit Format Run Options Window Help # Program to find maximum and the minimum value in a set. #Create a set seta = { 5, 10, 3, 15, 2, 20} #Find maximum value print ( max ( seta) ) #Find minimum value print ( min ( seta) )</pre>	<pre>Python 3.6.4 Shell File Edit Shell Debug Options Window Help Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04: Type "copyright", "credits" or "license () " for more in &gt;&gt;&gt; ===== RESTART: D:/Documents/Pyth 20 2 &gt;&gt;&gt;  </pre>
---	--