Sets

- Set is a unordered collection of items.
- Every set element must be unique,set doesnot allows duplicates

In [2]:

```
# creating a set

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

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

In [3]:

```
# adding a single item to a set
s.add(50)
print(s)
```

```
{40, 10, 50, 20, 30}
```

In [5]:

```
# to add multiple elements to a set
s.update((90,80))
s
```

Out[5]:

```
{10, 20, 30, 40, 50, 80, 90}
```

In [6]:

```
# removing element from a set
s.remove(90)
s
```

Out[6]:

```
{10, 20, 30, 40, 50, 80}
```

In [7]:

```
print(dir(s),end=" ")
```

...

In [8]:

```
s1={1,2,3}
s2={3,4,5,6}
```

Out[8]:

```
{1, 2, 3, 4, 5, 6}
```

In [9]:

```
s1.intersection(s2)
```

Out[9]:

```
{3}
```

In [11]:

```
s1={1,2,3}
s2={3,4,5,6}
s1-s2
```

Out[11]:

```
{1, 2}
```

In [12]:

```
s2-s1
```

Out[12]:

```
{4, 5, 6}
```

In [13]:

```
s1={1,2,3}
s2={1,2,3,4,5,6}
s1.issubset(s2)
```

Out[13]:

```
True
```

In [14]:

```
s2.issuperset(s1)
```

Out[14]:

```
True
```

In [15]:

```
s1={1,2,3}
s2={1,2,3,4,5,6}
s1.symmetric_difference(s2)
```

...

Strings

- String is a group of characters
- enclosed with either single quotes or double quotes

In [16]:

```
s="hello"
print(s)
print(type(s))
```

...

In [18]:

```
# accessing strings
# forward index 0,1,2,..
# backward index -1,-2,-3.
print(s[0])
print(s[1])
print(s[-1])
print(s[4])
```

...

In [19]:

```
# Length of a string
print(len(s))
```

```
5
```

In [21]:

```
# slicing [start:stop:step]
# i/p: python -> o/p:yth
#      012345
s1="python"
print(s1[1:4:1])
```

```
yth
```

In [22]:

```
print(dir(s),end=" ")
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__siz
eof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'cou
nt', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'inde
x', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'i
slower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesu
ffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip',
'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate',
'upper', 'zfill']
```

In [23]:

```
s="HELLO WORLD"
s.lower()
#to convert string into lowercase
```

...

In [24]:

```
s1="java"
s1.upper() # to convert into uppercase
```

Out[24]: 'JAVA'

In [25]:

```
s="python programming"
s.count('p')
```

...

In [28]:

```
print(s.index('t'))
print(s.index('y'))
```

...

In [31]:

```
s="machine learning"
s.title()
```

...

In [32]:

```
s="machine learning"
s.capitalize()
```

...

In [33]:

```
s="coding"
s.replace('i','$')
```

...

In [34]:

```
s="home"
s.islower() # returns true if the
# given string is in lowercase,
# otherwise return false
```

Out[34]:

```
True
```

In [35]:

```
s="HOUSE"
s.isupper()
 # returns true if the
# given string is in uppercase,
# otherwise return false
```

Out[35]:

```
True
```

functions

- function is a block of code, which is used to perform a specific task

In [37]:

```
# addition of 2 numbers
def add():
    n1=int(input("enter a value::"))
    n2=int(input("enter b value:"))
    s=n1+n2
    print("addition of two numbers is: ",s)
add()
```

```
enter a value::34
enter b value:789
addition of two numbers is:  823
```

In [ ]:

```

```