Super keyword is used to call the parent class in method overriding especially usually called in child class bcoz of the function name of parent class is override by child class. The super() function is used to give access to methods and properties of a parent or sibling class.

In [3]:

```
class sample:
    def show(self):
        print("Parent Class")
class simple(sample):
    def show(self):
        super().show()
        print("Child Class")
s = simple()
s.show()

```

```
Parent Class
Child Class
```

In [6]:

```
class JustCounter:

    __secretCount=0

    def count(self):

        self.__secretCount =self.__secretCount+1

        print(self.__secretCount)

c = JustCounter()
c.count()
c.count()
c.count()
print("outside of the class")
print(c.__secretCount)
```

```
1
2
3
outside of the class

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [6], in <cell line: 16>()
     14 c.count()
     15 print("outside of the class")
---> 16 print(c.__secretCount)

AttributeError: 'JustCounter' object has no attribute '__secretCount'
```

Module 5 - Regular Expression

Regular expression is a powerful tool for various kinds of string manipulation these are basically a special text that is used for describing a search pattern to extract the information from text such as paragraphs,file,spreadsheet,logs

Regular Expression:Module match() search() sub() findAll()

In [7]:

```
import re
```

In [9]:

```
txt = "Welcome to SRET"
ptrn = "Welcome"
if re.match(ptrn,txt):
    print("Match found")
else:
    print("Match not found")
```

Match function will match only the first word in the given string

In [10]:

```
txt = "Welcome to SRET"
ptrn = "SRET"
if re.match(ptrn,txt):
    print("Match found")
else:
    print("Match not found")
```

In [14]:

```
txt = "Welcome to SRET"
ptrn = "SRET"
if re.search(ptrn,txt):
    print("Match found")
else:
    print("Match not found")
```

In [18]:

```
msg = " Hi Hi,Welcome to SRET"
ptrn = "Hello"
rpl = "Hi"
newMsg = re.sub(rpl,ptrn,msg,2)#2 is for replace the no.of times
print(newMsg)
```

Format for sub() sub("which word u want to replace","With which word u want to replace","In which msg","How many times") re.sub(rpl,ptrn,msg,2)

In [19]:

```
msg = input("Enter the Message:")
rpl = input("Which word you want to replace:")
p = input("Give the new word for changing:")
newMsg = re.sub(rpl,p,msg,1)
print(newMsg)
```

...

In [20]:

```
msg = input("Enter the Message:")
rpl = input("Which word you want to replace:")
p = input("Give the new word for changing:")
newMsg = re.sub(rpl,p,msg,1)
print(newMsg)
```

...

In [28]:

```
msg = " This is our 17 class"
ptrn = "Seventeen"
rpl = "17"
newMsg = re.sub(rpl,ptrn,msg,1)
print(newMsg)
```

...

FindAll() Msg beginning with one or more char followed by space followed by one or more digits

In [39]:

```
p = r'[a-zA-Z]+ \d+'
```

In [40]:

```
msg = "LXI 2015 VXI 2017 maruti cars in india"
re.findall(p,msg)
```

Out[40]:

```
['LXI 2015', 'VXI 2017']
```

Metacharcter Description \d+ Matches the digits \d{n} Matches exactly n digits ^ Matches at the beginning of the line $ Matches at the end of the line [] Matches any character inside the brackets [^] Matches any single character not inside the bracket \w Matches word character \W Matches non-word characters . Matches any single character except the new line character

program to extract an email adress from a text

In [50]:

```
p = r'[\w.-]+@[\w.-]+'
msg = "Hello my personal mail id is klmarjun@gmail.com and my official maid id is e0222
mail = re.findall(p,msg)
```

In [51]:

```
print("Mail IDs: ",mail)
```

```
Mail IDs:  ['klmarjun@gmail.com', 'e0222054@sret.edu.in', 'india@info-oxfor
d.com']
```

In [ ]:

```

```