In [ ]:
```
#It describes the idea of wrapping data
 #and the methods that work on data within one unit.
 #This puts restrictions on accessing variables and methods directly
 #and can prevent the accidental modification of data
 #public attribute
class encap:
  __a=10
  def hello(self):
    print("hello")
obj=encap()
obj.hello()
print(obj.__a)
```

```
hello

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call l
ast)
<ipython-input-1-d8ac727724a9> in <module>()
     10 obj=encap()
     11 obj.hello()
---> 12 print(obj.__a)

AttributeError: 'encap' object has no attribute '__a'
```

In [ ]:
```
class encap:
  __a=10
  def hello(self):
    print("hello")
obj=encap()
obj.hello()
print (obj.__a)
```

```
hello

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call l
ast)
<ipython-input-3-c75b7f342134> in <module>()
      5 obj=encap()
      6 obj.hello()
----> 7 print (obj.__a)

AttributeError: 'encap' object has no attribute '__a'
```

In [ ]:
```
#private attribute inside the class
class encap:
  __a=10
  def hello(self):
    print("hello")
    print (self.__a)
obj=encap()
obj.hello()
```

```
hello
10
```

In [ ]:
```
#by default all methods are public
class disp:
  def disp1(self):
    print("u r in disp1")
  def disp2(self):
    print("u r in disp2")
obj1=disp()
obj1.disp1()
obj1.disp2()
```

```
u r in disp1
u r in disp2
```

In [ ]:
```
#encapsulating methods
class disp:
  def __disp1(self):
    print("u r in disp1")
  def disp2(self):
    print("u r in disp2")

obj1=disp()
obj1.disp1()
obj1.disp2()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call l
ast)
<ipython-input-14-40a52dcef90f> in <module>()
      5     print("u r in disp2")
      6 obj1=disp()
----> 7 obj1.disp1()
      8 obj1.disp2()

AttributeError: 'disp' object has no attribute 'disp1'
```

In [ ]:
```python
#CALLING A PRIVATE METHOD IN A PUBLIC CLASS
class disp:
  def __disp1(self):
    print("u r in disp1")
  def disp2(self):
    print("u r in disp2")
    self.__disp1()
obj1=disp()
obj1.disp2()
```

```
u r in disp2
u r in disp1
```