LAMBDA , MAP , REDUCE , FILTER

In [2]:
```
res = (lambda *args: sum(args))
res(10,20) , res(10,20,30,40) ,  res(10,20,30,40,50,60,70)
```

Out[2]:
```
(30, 100, 280)
```

In [4]:
```
odd_num=[1,8,7,5,9,33]
def twice(n):
    return n*2
doubles = list(map(twice,odd_num))
doubles
```

Out[4]:
```
[2, 16, 14, 10, 18, 66]
```

In [5]:
```
from functools import reduce
def add(a,b):
    return a+b
sum_all = reduce(add,doubles)
sum_all
```

Out[5]:
```
126
```

In [6]:
```
list1 = [1,2,3,4,5,6,7,8,9]
def odd(n):
    if n%2 ==1: return True
    else: return False
odd_num = list(filter(odd,list1))
odd_num
```

Out[6]:
```
[1, 3, 5, 7, 9]
```

CLASS & OBJECT

In [7]:
```
class my_class:
    var_1 = 100
obj1 = my_class()
print(obj1.var_1)
```

```
100
```

In [10]:
```
class myclass:
    def __init__(self,a):
        self.var_1 = 100+a
obj1 = myclass(15)
print(obj1.var_1)
```

```
115
```

Inheritance

In [11]:
```python
# multi level , single , hierarchical inheritance
class person:
# Parent Class
    def __init__(self, name , age , gender):
        self.name = name
        self.age = age
        self.gender = gender
    def PersonInfo(self):
        print('Name :- {}'.format(self.name))
        print('Age :- {}'.format(self.age))
        print('Gender :- {}'.format(self.gender))

class employee(person): # Child Class
    def __init__(self,name,age,gender,empid,salary):
        person.__init__(self,name,age,gender)
        self.empid = empid
        self.salary = salary
    def employeeInfo(self):
        print('Employee ID :- {}'.format(self.empid))
        print('Salary :- {}'.format(self.salary))

class fulltime(employee): # Grand Child Class
    def __init__(self,name,age,gender,empid,salary,WorkExperience):
        employee.__init__(self,name,age,gender,empid,salary)
        self.WorkExperience = WorkExperience
    def FulltimeInfo(self):
        print('Work Experience :- {}'.format(self.WorkExperience))

class contractual(employee): # Grand Child Class
    def __init__(self,name,age,gender,empid,salary,ContractExpiry):
        employee.__init__(self,name,age,gender,empid,salary)
        self.ContractExpiry = ContractExpiry
    def ContractInfo(self):
        print('Contract Expiry :- {}'.format(self.ContractExpiry))
        print('Contractual Employee Details')
        print('****************************')

contract1 = contractual('Basit' , 36 , 'Male' , 456 , 80000,'21-12-2021')
contract1.PersonInfo()
contract1.employeeInfo()
contract1.ContractInfo()
print('\n \n')
```

```
Name :- Basit
Age :- 36
Gender :- Male
Employee ID :- 456
Salary :- 80000
Contract Expiry :- 21-12-2021
Contractual Employee Details
****************************
```

In [13]:
```python
# Super Class
class Father:
    def __init__(self):
        self.fathername = str()
 # Super Class
class Mother:
    def __init__(self):
        self.mothername = str()
 # Sub Class
class Son(Father, Mother):
    name = str()
    def show(self):
        print('My Name :- ',self.name)
        print("Father :", self.fathername)
        print("Mother :", self.mothername)
s1 = Son()
s1.name = 'Bill'
s1.fathername = "John"
s1.mothername = "Kristen"
s1.show()
```

```
My Name :-  Bill
Father : John
Mother : Kristen
```

In [1]:
```python
class person:   # Parent Class
    def __init__(self, name , age , gender):
        self.name = name
        self.age = age
        self.gender = gender
    def PersonInfo(self):
        print('Name :- {}'.format(self.name))
        print('Age :- {}'.format(self.age))
        print('Gender :- {}'.format(self.gender))

class student(person): # Child Class
    def __init__(self,name,age,gender,studentid,fees):
        super().__init__(name,age,gender)
        self.studentid = studentid
        self.fees = fees
    def StudentInfo(self):
        super().PersonInfo()
        print('Student ID :- {}'.format(self.studentid))
        print('Fees :- {}'.format(self.fees))
stud = student('Asif' , 24 , 'Male' , 123 , 1200)
print('Student Details')
print('---------------')
stud.StudentInfo()
```

```
Student Details
---------------
Name :- Asif
Age :- 24
Gender :- Male
Student ID :- 123
Fees :- 1200
```

Iterator

In [ ]:
```
# iter()
# next() (StopIteration) -> .__next__()
```

In [14]:
```
m=[1,5,8,2,7,6]
x=iter(m)
print(x.__next__())
print(x.__next__())
print(x.__next__())
```

```
1
5
8
```

Decorator

In [15]:
```
def install_decorator(func):
    def wrapper():
        print("read terms and conditions")
        return func()
    return wrapper()

@install_decorator
def A_user():
    print("login")

@install_decorator
def B_user():
    print("login_page")
```

```
read terms and conditions
login
read terms and conditions
login_page
```