

dia3

May 29, 2024

1 LAMBDA , MAP , REDUCE , FILTER

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

```
[1]: (30, 100, 280)
```

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

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

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

```
[3]: 126
```

```
[4]: 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
```

```
[4]: [1, 3, 5, 7, 9]
```

2 CLASES Y OBJETOS

```
[5]: class mi_clase:
      var_1 = 100
      obj1 = mi_clase()
      print(obj1.var_1)
```

100

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

115

3 herencia

```
[7]: # multinivel , solo , herencia jerárquica
class persona:
    # Clase Padre
    def __init__(self, nombre , edad , genero):
        self.nombre = nombre
        self.edad = edad
        self.genero = genero
    def PersonaInfo(self):
        print('Nombre :- {}'.format(self.nombre))
        print('Edad :- {}'.format(self.edad))
        print('Genero :- {}'.format(self.genero))

class empleado(persona): # Clase hijo
    def __init__(self,nombre,edad,genero,empid,salario):
        persona.__init__(self,nombre,edad,genero)
        self.empid = empid
        self.salario = salario
    def EmpleadoInfo(self):
        print('ID Empleado :- {}'.format(self.empid))
        print('Salario :- {}'.format(self.salario))

class tiempoCompleto(empleado): # Clase gran hijo
    def __init__(self,nombre,edad,genero,empid,salario,ExperienciaLaboral):
        empleado.__init__(self,nombre,edad,genero,empid,salario)
        self.ExperienciaLaboral = ExperienciaLaboral
    def TiempoCompletoInfo(self):
        print('Experiencia Laboral :- {}'.format(self.ExperienciaLaboral))
```

```

class contractual(Empleado): # Clase gran hijo
    def __init__(self,nombre,edad,genero,empid,salario,ExpiracionContrato):
        Empleado.__init__(self,nombre,edad,genero,empid,salario)
        self.ExpiracionContrato = ExpiracionContrato
    def ContratoInfo(self):
        print('Expiracion de Contrato :- {}'.format(self.ExpiracionContrato))
        print('Detalle contractual de Empleado')
        print('*****')

contrato1 = contractual('Basit' , 36 , 'Masculino' , 456 , 80000,'21-12-2021')
contrato1.PersonaInfo()
contrato1.EmpleadoInfo()
contrato1.ContratoInfo()
print('\n \n')

```

```

Nombre :- Basit
Edad :- 36
Genero :- Masculino
ID Empleado :- 456
Salario :- 80000
Expiracion de Contrato :- 21-12-2021
Detalle contractual de Empleado
*****

```

```

[8]: # Super Clase Padre
class Father:
    def __init__(self):
        self.fathername = str()
    # Super Clase Madre
class Mother:
    def __init__(self):
        self.mothername = str()
    # Sub Clase Hijo
class Son(Father, Mother):
    name = str()
    def show(self):
        print('Mi Nombre :- ',self.name)
        print("Padre :", self.fathername)
        print("Madre :", self.mothername)
s1 = Son()
s1.name = 'Bill'
s1.fathername = "John"
s1.mothername = "Kristen"
s1.show()

```

Mi Nombre :- Bill
Padre : John
Madre : Kristen

```
[10]: class persona:      # Clase Padre
    def __init__(self, name , age , gender):
        self.name = name
        self.age = age
        self.gender = gender
    def PersonaInfo(self):
        print('Name :- {}'.format(self.name))
        print('Age :- {}'.format(self.age))
        print('Gender :- {}'.format(self.gender))

class estudiante(persona): # Clase Hijo
    def __init__(self,name,age,gender,studentid,fees):
        super().__init__(name,age,gender)
        self.studentid = studentid
        self.fees = fees
    def EstudianteInfo(self):
        super().PersonaInfo()
        print('Student ID :- {}'.format(self.studentid))
        print('Fees :- {}'.format(self.fees))
stud = estudiante('Asif' , 24 , 'Male' , 123 , 1200)
print('Detalle de Estudiante')
print('-----')
stud.EstudianteInfo()
```

Detalle de Estudiante

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

4 Iteradores

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

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

1
5
8

5 Decoradores

```
[13]: def install_decorator(func):  
    def wrapper():  
        print("Leer terminos y condiciones")  
        return func()  
    return wrapper()  
  
@install_decorator  
def A_user():  
    print("login")  
  
@install_decorator  
def B_user():  
    print("login_page")
```

Leer terminos y condiciones
login
Leer terminos y condiciones
login_page