

In [ ]:

```
1  #como programador defino funciones (metodos) que se van a usar repetidamente
2  #por ejemplo tenemos la funcion print que se definio para usarla repetidamen
3  #a las funciones tambien se les conoce como METODOS()
4
5
6  #cuando empleo funciones recuerdo 4 cosas
7
8  #1 Defino la funcion
9  #2 Defino si tiene entradas (ARGUMENTOS)
10 #3 Defino si tiene salidas (RETORNO)
11 #4 Invoco o uso la funcion
```

In [ ]:

```
1  #Programacion orientada a objetos POO
2
3  # CLASES : es un algo que encierra un conjunto de objetos -sirve para catalo
4  # OBJETOS : un objeto es un elemento (UNA INSTANCIA) que pertenece a una cla
5  #           el elemento dentro del conjunto
6  # ATRIBUTOS : los elementos que pueden diferenciar un objeto de otro
7  # METODOS: funciones -> una cosa que hace algo que necesito para mi objeto
8
9  #diferencia entre objeto e instancia:
10 #son "lo mismo" un objeto es una instancia de una clase
11
12
13 #constructores: inicializa/crea/hace que exista el objeto de la clase y le
14 #               correspondientes a su clase
15 #SELF: autoparametros -> puntero -> el codigo entiende sobre quien estan ha
16 #               toma los valores del objeto especifico que estoy tr
17
18 #JERARQUIA O HERENCIA -> hay un orden, y hay cosas que se pueden heredar
19 #               -> hay abuelos, papas, hijos, etc
20
21
```

In [17]:

```

1 #####
2 #Primer ejemplo de Calses
3
4 ##### CLASE PADRE
5 ##### Defino mis Calses #####
6 class profesoresSIC:
7
8     def __init__(self,nombre,apellido,edad,codigo): #constructor
9         self.nombre = nombre
10        self.apellido = apellido
11        self.edad = edad
12        self.codigo = codigo
13
14
15 #####Setters y getters
16
17 ##### con el get, accedo a toda la informacion sobre los atributos del
18
19     def get_nombre(self):
20         return self.nombre
21     def get_apellido(self):
22         return self.apellido
23     def get_edad(self):
24         return self.edad
25     def get_codigo(self):
26         return self.codigo
27
28 ##### con el set, modifico toda la informacion sobre los atributos del
29
30     def set_nombre(self,nombre):
31         self.nombre = nombre
32     def set_apellido(self,apellido):
33         self.apellido = apellido
34     def set_edad(self,edad):
35         self.edad = edad
36     def set_codigo(self,codigo):
37         self.codigo = codigo
38
39
40 ##### clase hijo
41
42 class tutor(profesoresSIC):
43
44     def __init__(self,nombre,apellido,edad,codigo):
45         profesoresSIC.__init__(self,nombre,apellido,edad,codigo)
46
47
48
49
50
51
52 ##### MAIN ##### aqui hago el llamado mis funciones,cl
53
54
55 ##### cree los profesores principales de la clase profesor
56 profe1 = profesoresSIC("jose", "burgos", 28, 1234)

```

```
57 profe2 = profesoresSIC("rafa", "puche", 30, 5678)
58 profe3 = profesoresSIC("argenis", "bouzas", 32, 9012)
59 profe4 = profesoresSIC("cristiam", "loaiza", 33, 3456)
60
61 print(profe1.get_nombre())
62 profe1.set_nombre("erick")
63 print(profe1.get_nombre())
64
65 ##### voy a crear a los tutores
66
67 tutor1 = tutor("luis", "gutierrez", 23, 1234)
68 tutor2 = tutor("ana", "bermudez", 23, 1234)
69 tutor3 = tutor("fredy", "gonzales", 23, 1234)
70 tutor4 = tutor("arturo", "morales", 23, 1234)
71
72 print(tutor1.get_nombre())
73
74
75
```

jose  
erick  
luis