

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  #cuando empleo funciones recuerdo 4 cosas
6  #1 Defino la funcion
7  #2 Defino si tiene entradas (ARGUMENTOS)
8  #3 Defino si tiene salidas (RETORNO)
9  #4 Invoco o uso la funcion (usarla)
10
11 #def nombre(argumentos):
12 #     proceso
13 #     return
```

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
```

In [1]:

```

1  # ejemplo de clases
2  # Profe Jose
3  # Gp4
4  # SIC2022
5  # 6/09/22
6
7  ##### CLASES #####
8
9  # _____CLASE PRINCIPAL_____
10
11 class seres_vivos: #Clase ppal(main) , clase abuelo
12
13
14
15     # SIEMPRE PRIMERO DEFINO UN CONSTRUCTOR
16     # EN EL CONSTRUCTOR ASIGNO LOS ATRIBUTOS DEL OBJETO QUE PERTENECE A ESA
17
18     def __init__(self,tipo,alimento,habitat,desplazamiento,color): #CONSTRUC
19         self.tipo=tipo
20         self.alimento=alimento
21         self.habitat=habitat
22         self.desplazamiento=desplazamiento
23         self.color=color
24
25     #-----GETTERS----- PARA OBTENER INFO
26
27     def get_tipo(self):
28         return self.tipo
29     def get_alimento(self):
30         return self.alimento
31     def get_habitat(self):
32         return self.habitat
33     def get_desplazamiento(self):
34         return self.desplazamiento
35     def get_color(self):
36         return self.color
37
38
39
40     #-----SETTERS----- PARA MODIFICAR INFO
41
42     def set_tipo(self,tipo):
43         self.tipo=tipo
44     def set_alimento(self,alimento):
45         self.alimento=alimento
46     def set_habitat(self,habitat):
47         self.habitat=habitat
48     def set_desplazamiento(self,desplazamiento):
49         self.desplazamiento=desplazamiento
50     def set_color(self,color):
51         self.color=color
52
53
54 # _____SUBCLASES_____
55
56 class humanos(seres_vivos): #subclase de la clase seres_vivos

```

```

57
58     #ejemplo 1 -> en este heredo el constructor de la clase principal (y los
59
60     #def __init__(self, tipo, alimento, habitat, desplazamiento, color):
61     #     seres_vivos.__init__(self, tipo, alimento, habitat, desplazamiento, colo
62
63     #ejemplo 2 -> en este creo nuevos atributos de mi subclase, y
64     #"por ahora no me interesan Los atributos que se heredaron"
65
66     def __init__(self, nombre, edad, estatura, peso, codigo):
67         self.nombre=nombre
68         self.edad=edad
69         self.estatura=estatura
70         self.peso=peso
71         self.codigo=codigo
72
73
74     #ejemplo 3 -> constructor completo en el que agrego nuevos atributos ade
75
76     #def __init__(self, nombre, edad, estatura, peso, codigo, tipo, alimento, habita
77     #     self.nombre=nombre
78     #     self.edad=edad
79     #     self.estatura=estatura
80     #     self.peso=peso
81     #     self.codigo=codigo
82     #     seres_vivos.__init__(self, tipo, alimento, habitat, desplazamiento, colo
83
84
85     #-----GETTERS  SUBCLASE ----- PARA OBTENER
86
87     def get_nombre(self):
88         return self.nombre
89     def get_edad(self):
90         return self.edad
91     def get_estatura(self):
92         return self.estatura
93     def get_peso(self):
94         return self.peso
95     def get_codigo(self):
96         return self.codigo
97
98     #-----SETTERS  SUBCLASE----- PARA MODIFI
99
100    def set_nombre(self, nombre):
101        self.nombre=nombre
102    def set_edad(self, edad):
103        self.edad=edad
104    def set_estatura(self, estatura):
105        self.estatura=estatura
106    def set_peso(self, peso):
107        self.peso=peso
108    def set_codigo(self, codigo):
109        self.codigo=codigo
110
111
112
113    ##### VARIABLES GLOBALES, FUNCIONES , ETC #####

```

```

114
115
116 ##### MAIN #####
117
118 # _____ ejemplo de como definir objetos para la clase padre _____
119
120
121 # crear los objetos de la clase seres_vivos
122 ser_vivo1=seres_vivos("humanos","variado","panama","bipedo","amarillo")
123 ser_vivo2=seres_vivos("animales","variado","panama","depende","variado")
124 ser_vivo3=seres_vivos("plantas","luz","panama","estatico","verde")
125
126 #accedemos a los atributos mediante los metodos set y get
127 print(ser_vivo1.get_habitat())
128 ser_vivo1.set_habitat("colombia")
129 print(ser_vivo1.get_habitat())
130 print("_____")
131
132 # _____ ejemplo de como definir objetos para la clase hijo usando
133
134 #ejemplo 1 -> solo heredo los atributos de la clase padre y sus metodos
135 #          tipo ,alimento ,habitat,desplazamiento,color
136 #persona1=humanos("humano","variado","panama","bipedo","amarillo")
137 #print(persona1.get_tipo())
138
139
140 #ejemplo 2 -> creo mis nuevos atributos sin usar los del padre, PERO!!!! SI
141
142 #          nombre,edad,estatura,peso,codigo
143 persona1=humanos("jose","28","170","63","1234")
144 print(persona1.nombre)
145 persona1.set_habitat("colombia") # Aunque en este ejemplo no habia atributo
146 print("el habitat es : ", persona1.get_habitat())
147
148 #ejemplo 3 -> creo lo nuevo + lo que herede
149 #          nombre,edad,estatura,peso,codigo,tipo,alimento ,habitat,de
150 #persona1=humanos("jose","28","170","63","1234","humano","variado","colombia

```

panama
colombia

jose
el habitat es : colombia