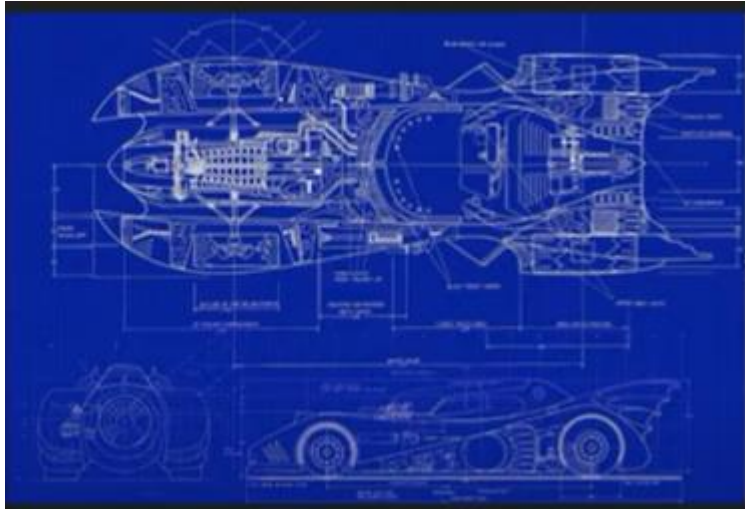


Variables de Clase

Clase



- Alto
- Ancho
- Velocidad maxima

Instancias



- Color
- Marca
- Modelo
- llave

Variables de Clase

Hemos visto como definimos atributos en una clase anteponiendo la palabra clave self:

```
class Persona:  
  
    def __init__(self,nombre):  
        self.nombre=nombre
```

Variables de Clase

Los atributos son independientes por cada objeto o instancia de la clase, es decir si definimos tres objetos de la clase persona, todas las personas tienen un atributo nombre pero cada uno tiene un valor independiente.

```
class Persona:

    def __init__(self,nombre):
        self.nombre=nombre

# bloque principal

persona1=Persona("Juan")
persona2=Persona("Ana")
persona3=Persona("Luis")

print(persona1.nombre) # Juan
print(persona2.nombre) # Ana
print(persona3.nombre) # Luis
```

Variables de Clase

En algunas situaciones necesitamos almacenar datos que sean compartidos por todos los objetos de dicha clase, en esas situaciones debemos emplear variables de clase.

Para definir una variable de clase lo hacemos dentro de la clase pero fuera de sus métodos

```
class Persona:

    variable=20

    def __init__(self,nombre):
        self.nombre=nombre

# bloque principal

persona1=Persona("Juan")
persona2=Persona("Ana")
persona3=Persona("Luis")

print(persona1.nombre) # Juan
print(persona2.nombre) # Ana
print(persona3.nombre) # Luis

print(persona1.variable) # 20
Persona.variable=5
print(persona2.variable) # 5
```

Independientemente que se definan muchos objetos de la clase Persona. La variable “variable” es compartida por todos los objetos persona1, persona2 y persona3.

Para modificar la variable de clase, hacemos referencia al nombre de la clase y seguidamente el nombre de la variable:

Persona.variable=5

Variables de Clase

Definir una clase cliente que almacene un código de cliente y un nombre.

En la clase cliente definir una variable de clase de tipo lista que almacene todos los clientes que tienen suspendidas sus cuentas corrientes.

Imprimir por pantalla todos los datos de clientes y el estado que se encuentra su cuenta corriente.

Solución :

```
class Cliente:
    suspendidos=[]
    def __init__(self, codigo, nombre):
        self.codigo=codigo
        self.nombre=nombre
```

Variables de Clase

```
def imprimir(self):  
    print("Codigo:", self.codigo)  
    print("nombre:", self.nombre)  
    self.esta_suspendido()  
  
def esta_suspendido(self):  
    if self.codigo in Cliente.suspendidos:  
        print("Esta suspendido")  
    else:  
        print("No esta suspendido")  
    print("_____")  
  
def suspender(self)  
    cliente.suspendidos.append(self.código)
```

Variables de Clase

Bloque Principal

```
cliente1=Cliente(1, “Juan”)
cliente2=Cliente(2, “Ana”)
cliente3=Cliente(3, “Diego”)
cliente4=Cliente(4, “Pedro”)
```

```
cliente3.suspender()
cliente4.suspender()
```

```
cliente1.imprimir()
cliente2.imprimir()
cliente3.imprimir()
cliente4.imprimir()
print(cliente.suspendidos)
```

Contador de Objetos Creados

Crea una clase llamada Vehículo que tenga una variable de clase llamada contador_vehiculos. Cada vez que se cree una instancia de la clase, esta variable deberá incrementarse en uno. Al final, se debe mostrar cuántos vehículos se han creado.

```
class Vehiculo:
    contador_vehiculos = 0

    def __init__(self, marca):
        self.marca = marca
        Vehiculo.contador_vehiculos += 1

# Crear varios objetos
v1 = Vehiculo("Toyota")
v2 = Vehiculo("Ford")
v3 = Vehiculo("Honda")

# Mostrar cuántos vehículos se crearon
print(f"Vehículos creados: {Vehiculo.contador_vehiculos}")
```


Variables de Clase

Problema Propuesto:

Plantear una clase llamada jugador.

Definir en la clase jugador los atributos nombre y puntaje y los métodos `__init__`, `imprimir` y `pasar_tiempo` (que debe reducir en uno la variable de clase).

Declarar dentro de la clase jugador una variable de clase que indique cuantos minutos falta para el fin de juego (iniciarla con el valor de 30).

Definir en el bloque principal dos objetos de la clase jugador.

Reducir dicha variable hasta llegar a cero.