

# 7 - Clases

Horas de Libre Configuración

Curso 2023 - 2024



# Ejemplo de clase



```
class Punto():  
    def __init__(self, x=0, y=0):  
        self.x = x  
        self.y = y  
    def mostrar(self):  
        return self.x, self.y
```

# Uso de una clase

```
p1 = Punto()          # Crea (0,0)
p2 = Punto(1,2)        # Crea (1,2)

print(p1.mostrar())    # -> (0,0)
print(p2.x)            # -> 1
```

# Constructor



- ▶ Se identifica con el nombre `__init__`.
- ▶ Un solo constructor por clase.
- ▶ El primer parámetro es `self` (referencia al objeto que se crea).
- ▶ Opcionalmente puede añadir más parámetros de inicialización.

# Atributos



- ▶ Se definen e inicializan los atributos de la clase.
- ▶ Los atributos son por defecto públicos, pero se pueden hacer privados poniendo en su nombre el prefijo \_\_ (doble subrayado).
- ▶ Se pueden establecer propiedades para acceder a atributos privados.

# Atributos



```
class Punto():  
    def __init__(self, x=0, y=0):  
        self.__x = x  
        self.__y = y  
    @property      # Getter (lectura)  
    def x(self):  
        return self.__x
```

# Atributos



```
@x.setter    # Setter (escritura)
def x(self, pos_x):
    self.__x = pos_x
```

- Se pueden establecer propiedades de sólo lectura (sin setter) o sólo escritura (sin getter).

# Métodos



- ▶ Es cualquier función anidada en la definición de la clase.
- ▶ Al igual que para los atributos, por defecto son públicos y se pueden hacer privados poniendo en su nombre el prefijo \_\_ (doble subrayado).



# Métodos



```
def sumar(self, n):  
    self.__x = self.__x + n  
    self.__y = self.__y + n
```

```
def __reset(self):  
    self.__x = 0  
    self.__y = 0
```