

Curso de Python

Decoradores `@property` , `@<nombre>.setter` , `@<nombre>.deleter`:

A cualquier conjunto de métodos que se inicien con '`@property`' se les conoce como Propiedades.

```
class MamiferoCuadrupedo:
    def __init__(self, nombre, edad, peso):
        self.__mi_nombre = nombre
        .....
```

```
@property
def mi_nombre(self):
    return self.__mi_nombre
```

```
@ mi_nombre.setter
def mi_nombre(self , valor):
    self.__mi_nombre = valor
```

```
@ mi_nombre.deleter
def mi_nombre(self ):
    del self.__mi_nombre
```




ProfesorenVideo.com

Curso de Python

@property

```
def mi_nombre(self):  
    return self.__mi_nombre
```

@ mi_nombre.setter

```
def mi_nombre(self , valor):  
    self.__mi_nombre = valor
```

@ mi_nombre.deleter

```
def mi_nombre(self , valor):  
    del self.__mi_nombre
```

@property:

- Convierte un método en una propiedad de solo lectura.
- Permite acceder a los datos de forma controlada.

@<nombre>.setter:

- Se usa junto con @property para definir un método que establece (cambia) el valor de la propiedad.
- Permite añadir lógica adicional durante la asignación de un valor.

@<nombre>.deleter:

- Se usa junto con @property para definir un método que elimina la propiedad.
- Permite añadir lógica adicional durante la eliminación de la propiedad.

Curso de Python

Decorador de propiedad @deleter :

Es usado para definir un método de instancia que **borra** algún atributo de instancia.

```
# Deleter para nombre
@mi_nombre.deleter
def mi_nombre(self):
    del self.__mi_nombre
```



Para llamar al decorador: @deleter
del objeto.mi_nombre

Curso de Python

Decorador de clase @classmethod :

Es usado para definir un método de clase que **acceda** a algún atributo de clase. Puede acceder para: **mostrar**, **cambiar**, **borrar** o **agregar**.

Decorador de clase

@classmethod

```
def borrar_extremidades(cls):  
    del cls.extremidades
```



Para llamar al decorador de **clase** :

objeto.borrar_extremidades()



ProfesorenVideo.com

Curso de Python

Basta de teoría, mejor programemos ...

... pero después volvemos