### Importar un módulo dentro de otro módulo

A veces querrás distribuir tus clases en varios módulos para evitar que un archivo crezca demasiado y para evitar almacenar clases no relacionadas en el mismo módulo. Cuando almacenas tus clases en varios módulos, puedes encontrar que una clase de un módulo depende de una clase de otro módulo. Cuando esto ocurra, puedes importar la clase necesaria en el primer módulo.

Por ejemplo, almacenemos la clase Car en un módulo y las clases ElectricCar y Battery en otro módulo. Crearemos un nuevo módulo llamado *electric\_car.py*-reemplazando el archivo *electric\_car.py* que creamos antes- y copiaremos sólo las clases Battery y ElectricCar en este archivo:

**electric\_car.py**

"""A set of classes that can be used to represent electric cars."""  
  
from car import Car  
  
class Battery:  
 --snip--  
  
class ElectricCar(Car):  
 --snip--

La clase ElectricCar necesita acceder a su clase padre Car, así que importaremos Car directamente al módulo. Si olvidamos esta línea, Python emitirá un error cuando intentemos importar el módulo electric\_car. También tenemos que actualizar el módulo Car para que sólo contenga la clase Car:

**car.py**

"""A class that can be used to represent a car."""  
  
class Car:  
 --snip--

Ahora podemos importar de cada módulo por separado y crear la clase que necesitemos:

**my\_cars.py**

from car import Car  
from electric\_car import ElectricCar  
  
my\_mustang = Car('ford', 'mustang', 2024)  
print(my\_mustang.get\_descriptive\_name())  
  
my\_leaf = ElectricCar('nissan', 'leaf', 2024)  
print(my\_leaf.get\_descriptive\_name())

Importamos Car de su módulo, y ElectricCar de su módulo. A continuación, creamos un coche normal y un coche eléctrico. Ambos coches se crean correctamente:

2024 Ford Mustang  
2024 Nissan Leaf

[anterior](c09_25.html)[Subtema 26 de 34: (Ver todo)](c09.html)[siguiente](c09_27.html)