Object-oriented Programming (Optional)

What is Object-oriented programming? (Optional)

Las clases representan y definen conceptos, mientras que los objetos son instancias de clases.

concepto de programación orientada a objetos se reduce a atributos y métodos asociados con un tipo. Los atributos son las características asociadas a un tipo, y los métodos son las funciones asociadas a un tipo

Object-Oriented Programming Defined

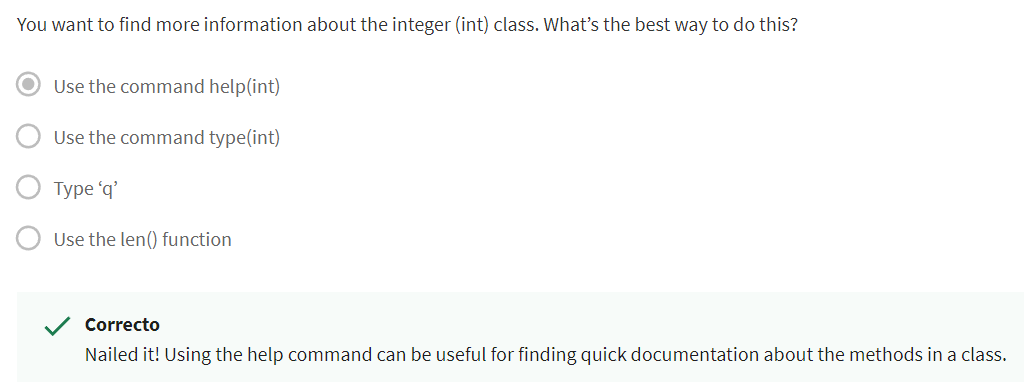
En la programación orientada a objetos, los conceptos se modelan como clases y objetos. Una idea se define usando una clase, y una instancia de esta clase se llama objeto. Casi todo en Python es un objeto, incluidas cadenas, listas, diccionarios y números. Cuando creamos una lista en Python, estamos creando un objeto que es una instancia de la clase de lista, que representa el concepto de una lista. Las clases también tienen atributos y métodos asociados con ellas. Los atributos son las características de la clase, mientras que los métodos son funciones que forman parte de la clase.

Classes and Objects in Python (Optional)

type(“”) saber el tipo

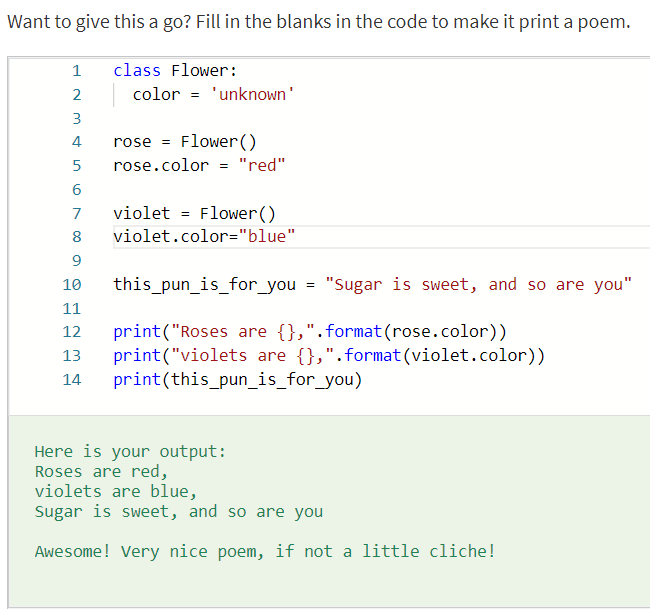
dir(“”) saldran todos los métodos del objeto String

help(“”) saldran de forma mas detallada todos los metodos



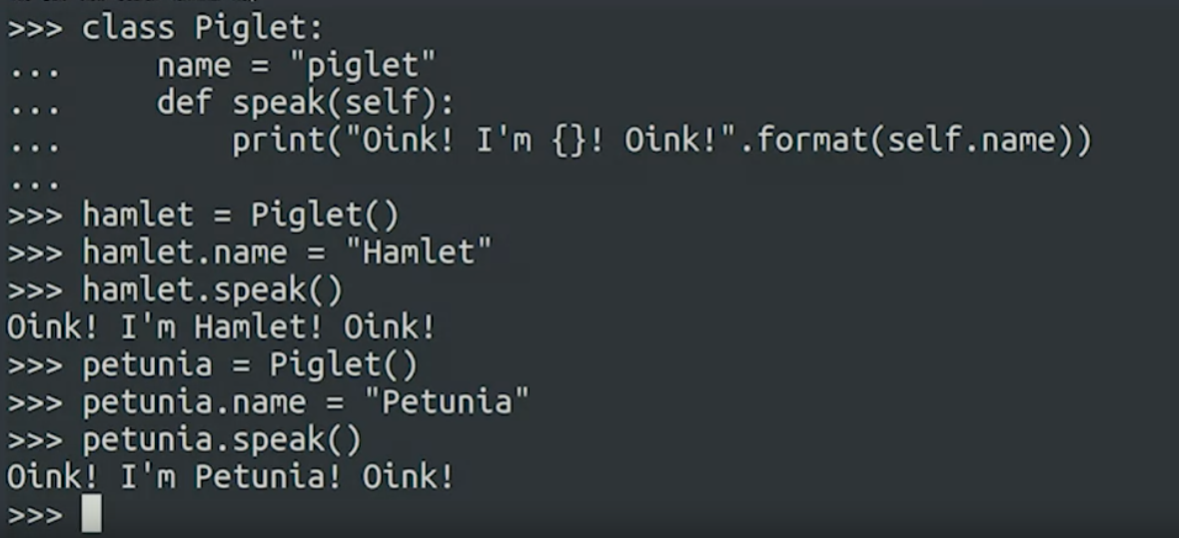
Defining New Classes (Optional)



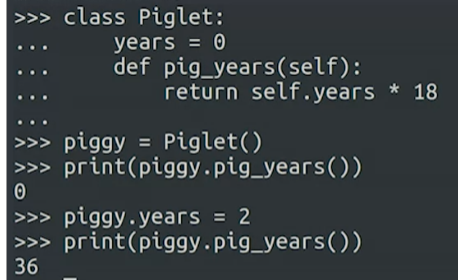


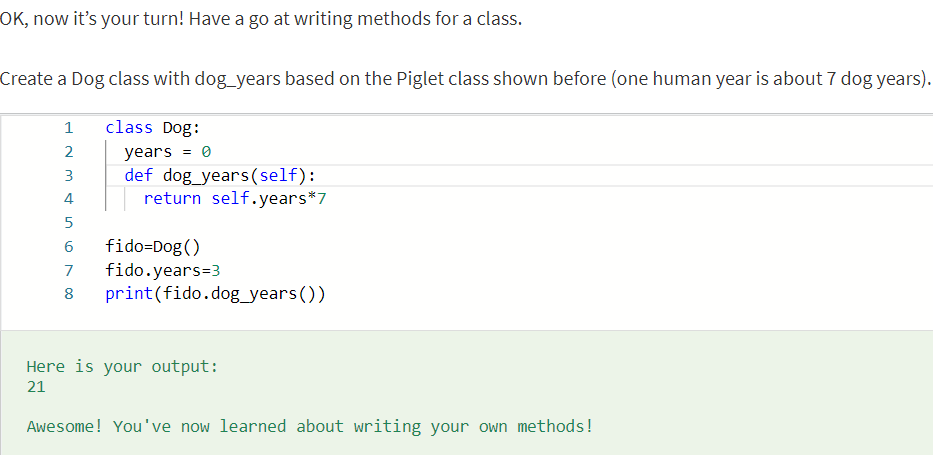
Classes and Methods (Optional)

Instance Methods (Optional)

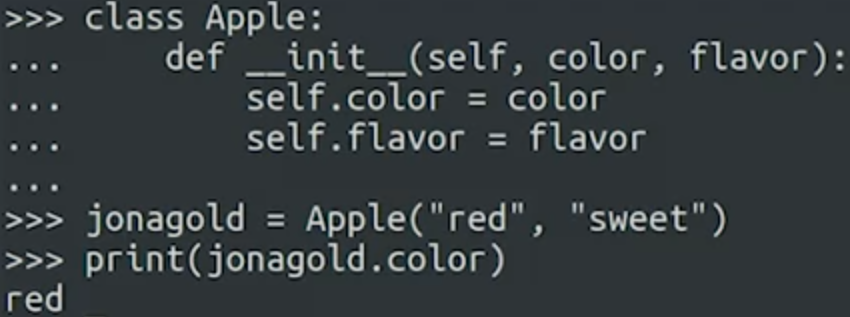


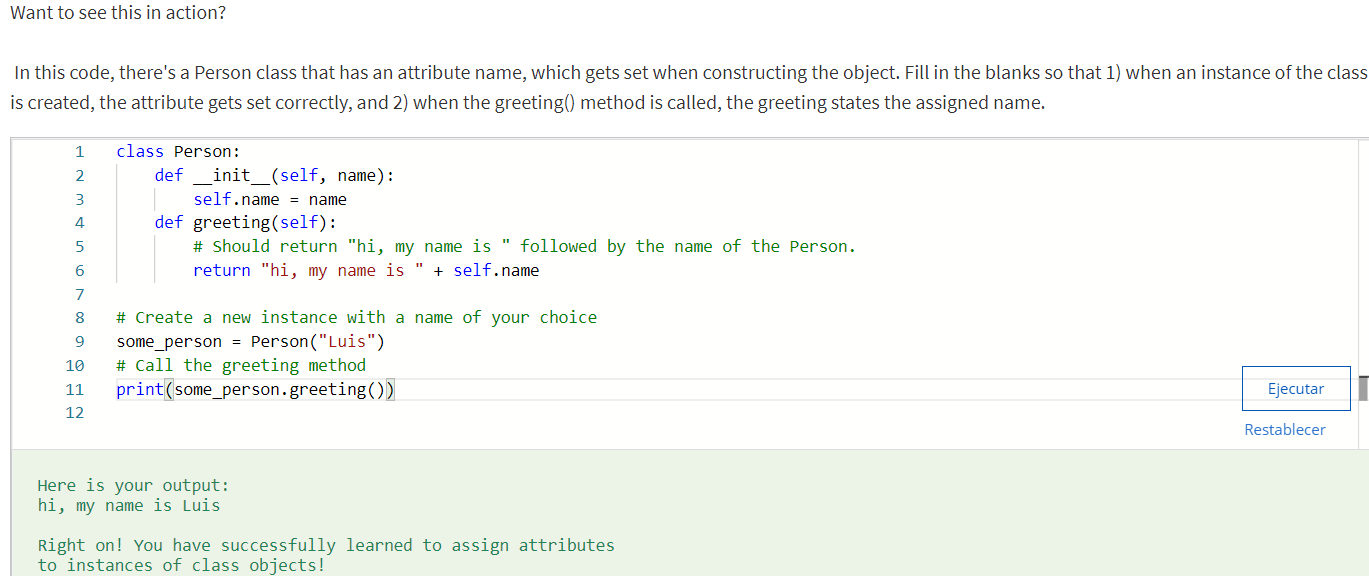
Las variables que tienen valores diferentes para diferentes instancias de la misma clase se denominan variables de instancia

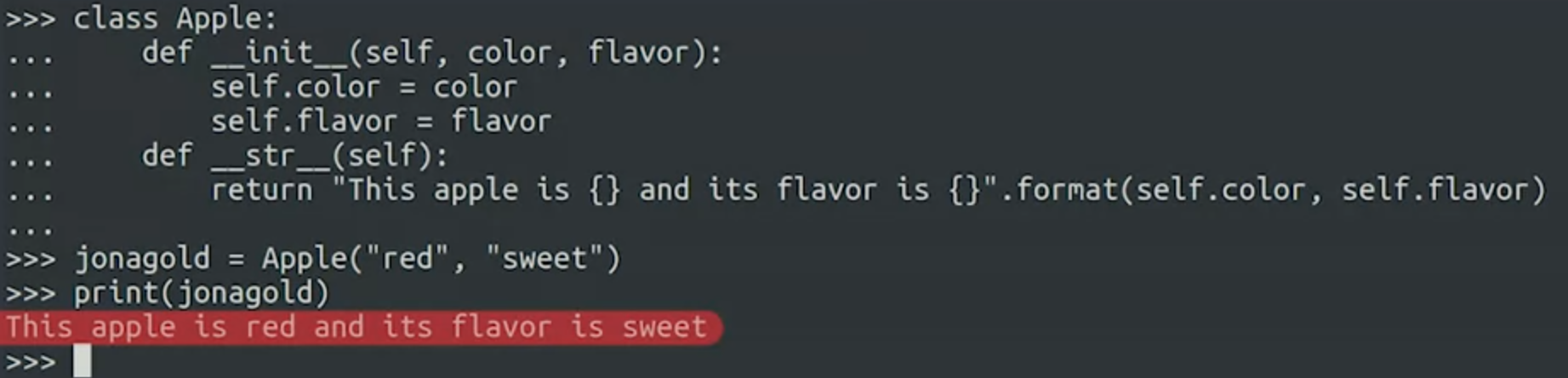




Constructors and Other Special Methods (Optional)

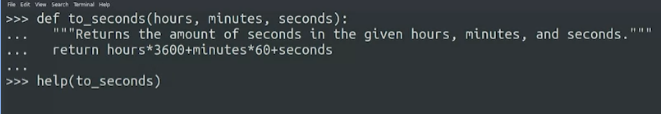


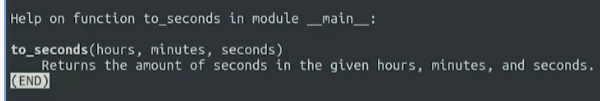




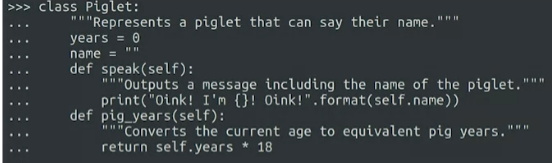
Documenting Functions, Classes, and Methods (Optional)

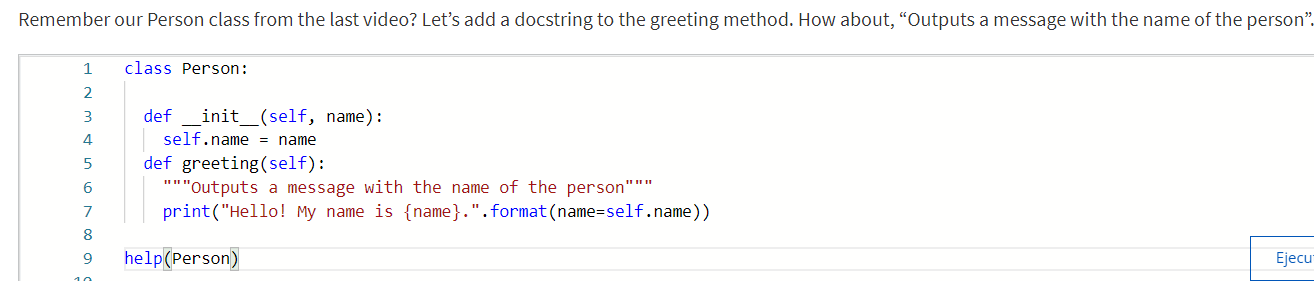
Queremos que nuestros métodos, clases y funciones nos den más información cuando nosotros o alguien más usamos la función de ayuda. Podemos hacer eso agregando una docstring. Una docstring es un texto breve que explica lo que hace algo

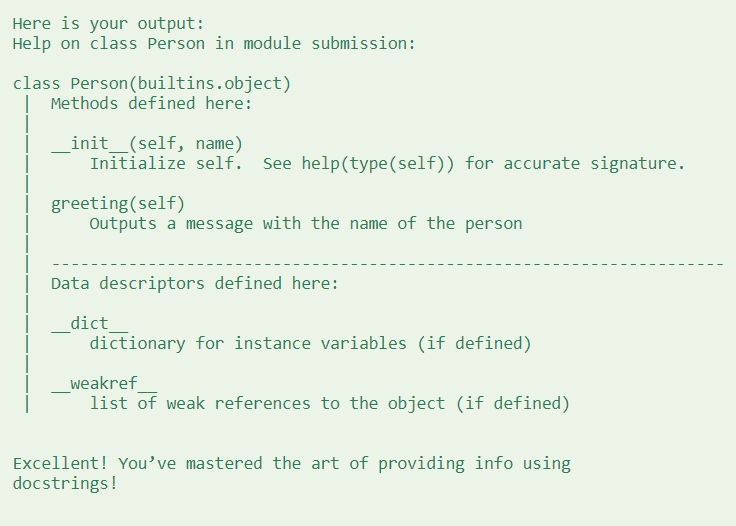




Cómo llamamos anteriormente, también podemos agregar docstrings a clases y métodos.





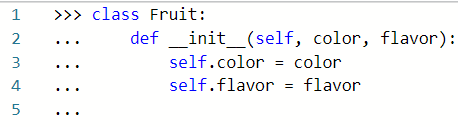


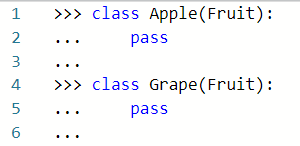
shift + enter

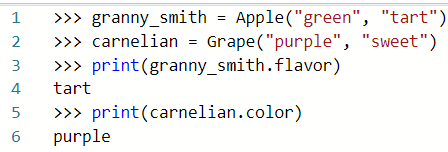
Code Reuse

Inheritance (Optional)

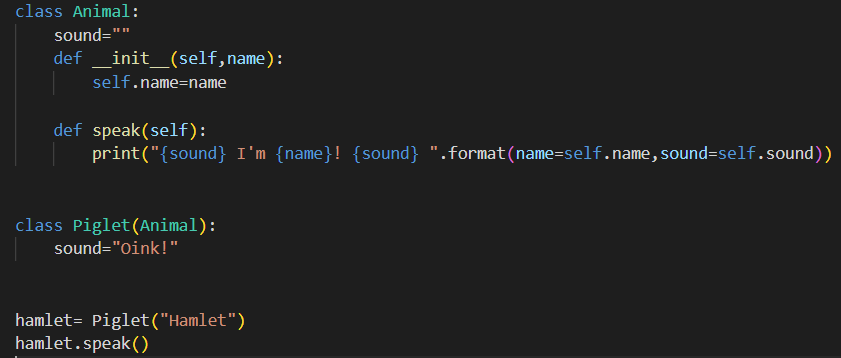
Relación de herencia se usa paréntesis



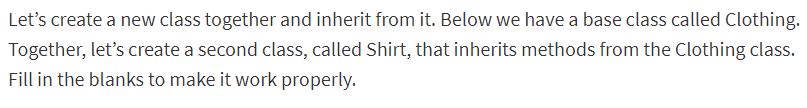


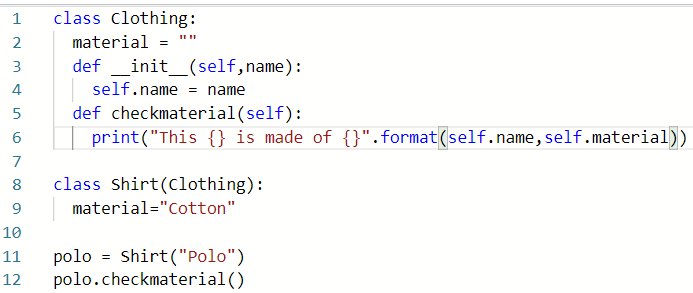


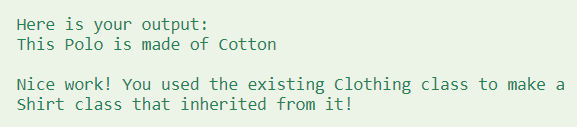
Se usa “()” cuando es herencia de una clase







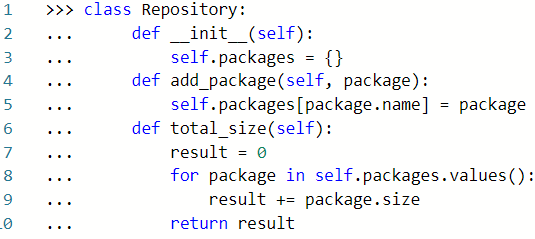


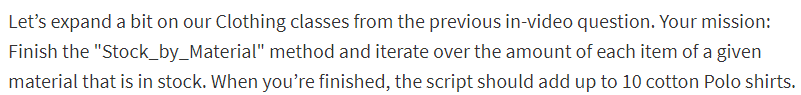


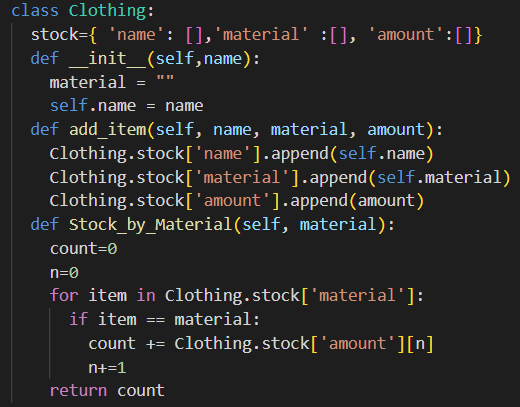
Object Composition

Cuando dos clases están relacionadas pero no hay herencia se conoce como **composición**, donde una clase hace uso del código contenido en otra clase.

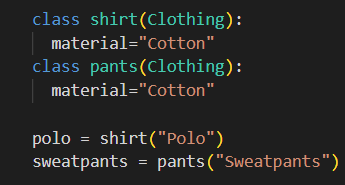
En este ejemplo supongamos que Repository y Packages son clases distintas, pero Repository contiene a Package



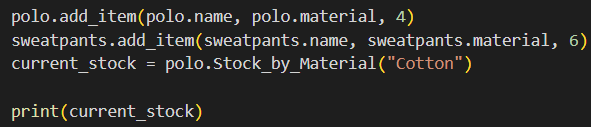
En este ejemplo supongamos la clase Clothing,,

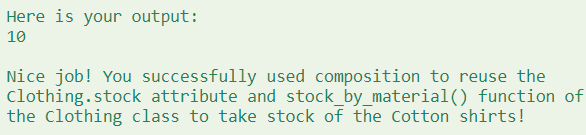


shirt y pants son clases que heredan de “Clothing”, se crean dos objetos polo y sweatpants



ambos objetos están relacionados, de tal forma que cuando se agrega un elemento en un objeto “polo” también se actualiza en el otro objeto sweatpants. En conclusion polo y sweatpants estan relacionados mediante Clothing, no existe herencia la una con la otra y ambas contienen el uno con el otro.





Python Modules (Optional)

Son librerías dentro de python, por ello para hacer uso de ellas es necesario antes importarlas

