### El método \_\_init\_\_()

Una función que forma parte de una clase es un *method*. Todo lo que aprendiste sobre las funciones se aplica también a los métodos; la única diferencia práctica por ahora es la forma en que llamaremos a los métodos. El método \_\_init\_\_() ❷ es un método especial que Python ejecuta automáticamente cada vez que creamos una nueva instancia basada en la clase Dog. Este método tiene dos guiones bajos iniciales y dos guiones bajos finales, una convención que ayuda a evitar que los nombres de métodos por defecto de Python entren en conflicto con tus nombres de métodos. Asegúrate de utilizar dos guiones bajos a cada lado de \_\_init\_\_(). Si utilizas sólo uno a cada lado, el método no será llamado automáticamente cuando utilices tu clase, lo que puede dar lugar a errores difíciles de identificar.

Definimos el método \_\_init\_\_() para que tenga tres parámetros: self, name, y age. El parámetro self es obligatorio en la definición del método, y debe ir en primer lugar, antes que los demás parámetros. Debe incluirse en la definición porque cuando Python llame a este método más adelante (para crear una instancia de Dog), la llamada al método pasará automáticamente el argumento self. Cada llamada a un método asociado a una instancia pasa automáticamente self, que es una referencia a la propia instancia; da acceso a la instancia individual a los atributos y métodos de la clase. Cuando hagamos una instancia de Dog, Python llamará al método \_\_init\_\_() de la clase Dog. Pasaremos a Dog() un nombre y una edad como argumentos; self se pasa automáticamente, por lo que no necesitamos pasarlo. Siempre que queramos crear una instancia a partir de la clase Dog, proporcionaremos valores sólo para los dos últimos parámetros, name y age.

Las dos variables definidas en el cuerpo del método \_\_init\_\_() tienen cada una el prefijo self ❸. Cualquier variable prefijada con self está disponible para todos los métodos de la clase, y también podremos acceder a estas variables a través de cualquier instancia creada a partir de la clase. La línea self.name = name toma el valor asociado al parámetro name y lo asigna a la variable name, que se adjunta a la instancia que se está creando. El mismo proceso ocurre con self.age = age. Las variables accesibles a través de instancias como ésta se denominan *attributes*.

La clase Dog tiene otros dos métodos definidos: sit() y roll\_over() ❹. Como estos métodos no necesitan información adicional para ejecutarse, sólo los definimos para que tengan un parámetro, self. Las instancias que creemos más adelante tendrán acceso a estos métodos. En otras palabras, podrán sentarse y darse la vuelta. Por ahora, sit() y roll\_over() no hacen gran cosa. Simplemente imprimen un mensaje diciendo que el perro se sienta o se da la vuelta. Pero el concepto puede ampliarse a situaciones realistas: si esta clase formara parte de un juego de ordenador, estos métodos contendrían código para hacer que un perro animado se sentara y se diera la vuelta. Si esta clase se escribiera para controlar un robot, estos métodos dirigirían los movimientos que hacen que un perro robótico se siente y se dé la vuelta.

[anterior](c09_3.html)[Subtema 4 de 34: (Ver todo)](c09.html)[siguiente](c09_5.html)