Desconocido

### Inicializar un objeto

A veces, al crear un objeto, queremos establecer algunos valores (también llamados *propiedades* ) para su uso posterior. Cuando *inicializamos* un objeto, lo estamos preparando para ser utilizado.

Por ejemplo, supongamos que queremos establecer el número de manchas de nuestros objetos jirafa cuando se crean (o inicializan). Para ello, creamos una función \_\_init\_\_ (fíjate en los dos caracteres de subrayado a cada lado, para un total de cuatro). La función init establece las propiedades de un objeto cuando éste se crea por primera vez. Python llamará automáticamente a esta función cuando creemos un nuevo objeto. A continuación te explicamos cómo utilizarla.

>>> class Giraffe(Mammal):  
 def \_\_init\_\_(self, spots):  
 self.giraffe\_spots = spots

En primer lugar, definimos la función \_\_init\_\_ con los parámetros self y spots . Al igual que las otras funciones que hemos definido en la clase, la función \_\_init\_\_ también necesita tener self como primer parámetro. A continuación, establecemos el parámetro spots en una variable de objeto (su propiedad) llamada giraffe\_spots utilizando el parámetro self . Puedes pensar que esta línea de código dice: "Toma el valor del parámetro spots y guárdalo para más tarde (utilizando la variable de objeto giraffe\_spots )". Del mismo modo que una función de una clase puede llamar a otra función utilizando el parámetro self , también se accede a las variables de la clase utilizando self .

A continuación, si creamos un par de nuevos objetos jirafa (llamados ozwald y gertrude ) y mostramos su número de manchas, podrás ver la función de inicialización en acción:

>>> ozwald = Giraffe(100)  
>>> gertrude = Giraffe(150)  
>>> print(ozwald.giraffe\_spots)  
100  
>>> print(gertrude.giraffe\_spots)  
150

En primer lugar, creamos una instancia de la clase Giraffe utilizando el valor 100 como parámetro. Esto tiene el efecto de llamar a la función \_\_init\_\_ y utilizar 100 para el valor del parámetro spots . A continuación, creamos otra instancia de la clase Giraffe con un valor de 150. Por último, imprimimos la variable de objeto giraffe\_spots para cada uno de nuestros objetos jirafa, y vemos que los resultados son 100 y 150. ¡Ha funcionado!

Recuerda que cuando creamos un objeto de una clase, como ozwald en este ejemplo, podemos referirnos a sus variables o funciones utilizando el operador punto y el nombre de la variable o función que queremos utilizar (por ejemplo, ozwald.giraffe\_spots ). Pero cuando creamos funciones dentro de una clase, nos referimos a esas mismas variables (y a otras funciones) utilizando el parámetro self ( self.giraffe\_spots ).

[anterior](ch08_11.html)[Subtema 12 de 14: (Ver todo)](ch08.html)[siguiente](ch08_13.html)