### Valores por defecto

Al escribir una función, puedes definir un *default value* para cada parámetro. Si se proporciona un argumento para un parámetro en la llamada a la función, Python utiliza el valor del argumento. Si no, utiliza el valor por defecto del parámetro. Así que cuando defines un valor por defecto para un parámetro, puedes excluir el argumento correspondiente que normalmente escribirías en la llamada a la función. Utilizar valores por defecto puede simplificar tus llamadas a funciones y aclarar la forma en que se utilizan normalmente tus funciones.

Por ejemplo, si observas que la mayoría de las llamadas a describe\_pet() se utilizan para describir perros, puedes establecer el valor por defecto de animal\_type en 'dog'. Ahora cualquiera que llame a describe\_pet() para describir un perro puede omitir esa información:

def describe\_pet(pet\_name, animal\_type='dog'):  
 """Display information about a pet."""  
 print(f"\nI have a {animal\_type}.")  
 print(f"My {animal\_type}'s name is {pet\_name.title()}.")  
  
describe\_pet(pet\_name='willie')

Cambiamos la definición de describe\_pet() para incluir un valor por defecto, 'dog', para animal\_type. Ahora, cuando se llama a la función sin especificar animal\_type, Python sabe que debe utilizar el valor 'dog' para este parámetro:

I have a dog.  
My dog's name is Willie.

Ten en cuenta que ha habido que cambiar el orden de los parámetros en la definición de la función. Como el valor por defecto hace innecesario especificar un tipo de animal como argumento, el único argumento que queda en la llamada a la función es el nombre de la mascota. Python sigue interpretándolo como un argumento posicional, por lo que si se llama a la función sólo con el nombre de la mascota, ese argumento coincidirá con el primer parámetro de la definición de la función. Esta es la razón por la que el primer parámetro debe ser pet\_name.

La forma más sencilla de utilizar esta función ahora es proporcionar sólo el nombre de un perro en la llamada a la función:

describe\_pet('willie')

Esta llamada a la función tendría el mismo resultado que el ejemplo anterior. El único argumento proporcionado es 'willie', por lo que se empareja con el primer parámetro de la definición, pet\_name. Como no se proporciona ningún argumento para animal\_type, Python utiliza el valor por defecto 'dog'.

Para describir un animal que no sea un perro, podrías utilizar una llamada a una función como ésta:

describe\_pet(pet\_name='harry', animal\_type='hamster')

Como se proporciona un argumento explícito para animal\_type, Python ignorará el valor por defecto del parámetro.

[anterior](c08_10.html)[Subtema 11 de 40: (Ver todo)](c08.html)[siguiente](c08_12.html)