### Utilizar get() para acceder a los valores

Utilizar claves entre corchetes para recuperar el valor que te interesa de un diccionario puede causar un problema potencial: si la clave que pides no existe, obtendrás un error.

Veamos qué ocurre cuando pides el valor en puntos de un alienígena que no tiene un valor en puntos establecido:

**alien\_no\_points.py**

alien\_0 = {'color': 'green', 'speed': 'slow'}  
print(alien\_0['points'])

Esto da lugar a un rastreo, que muestra un KeyError:

Traceback (most recent call last):  
 File "alien\_no\_points.py", line 2, in <module>  
 print(alien\_0['points'])  
 ~~~~~~~^^^^^^^^^^  
KeyError: 'points'

En el Capítulo 10 aprenderás más sobre cómo tratar errores como éste en general. En el caso concreto de los diccionarios, puedes utilizar el método get() para establecer un valor por defecto que se devolverá si la clave solicitada no existe.

El método get() requiere una clave como primer argumento. Como segundo argumento opcional, puedes pasar el valor que se devolverá si la clave no existe:

alien\_0 = {'color': 'green', 'speed': 'slow'}  
  
point\_value = alien\_0.get('points', 'No point value assigned.')  
print(point\_value)

Si la clave 'points' existe en el diccionario, obtendrás el valor correspondiente. Si no existe, obtendrás el valor por defecto. En este caso, points no existe, y obtenemos un mensaje limpio en lugar de un error:

No point value assigned.

Si existe la posibilidad de que la clave que pides no exista, considera la posibilidad de utilizar el método get() en lugar de la notación de corchetes.

[anterior](c06_11.html)[Subtema 12 de 28: (Ver todo)](c06.html)[siguiente](c06_13.html)