### Recorrer todos los pares clave-valor

Antes de explorar los distintos enfoques del bucle, consideremos un nuevo diccionario diseñado para almacenar información sobre un usuario en un sitio web. El siguiente diccionario almacenaría el nombre de usuario, el nombre y los apellidos de una persona:

**user.py**

user\_0 = {  
 'username': 'efermi',  
 'first': 'enrico',  
 'last': 'fermi',  
 }

Puedes acceder a cualquier dato de user\_0 basándote en lo que ya has aprendido en este capítulo. Pero, ¿y si quisieras ver todo lo almacenado en el diccionario de este usuario? Para ello, podrías recorrer el diccionario utilizando un bucle for:

user\_0 = {  
 'username': 'efermi',  
 'first': 'enrico',  
 'last': 'fermi',  
 }  
  
for key, value in user\_0.items():  
 print(f"\nKey: {key}")  
 print(f"Value: {value}")

Para escribir un bucle for para un diccionario, crea nombres para las dos variables que contendrán la clave y el valor de cada par clave-valor. Puedes elegir los nombres que quieras para estas dos variables. Este código funcionaría igual de bien si hubieras utilizado abreviaturas para los nombres de las variables, como ésta:

for k, v in user\_0.items()

La segunda mitad de la sentencia for incluye el nombre del diccionario seguido del método items(), que devuelve una secuencia de pares clave-valor. A continuación, el bucle for asigna cada uno de estos pares a las dos variables proporcionadas. En el ejemplo anterior, utilizamos las variables para imprimir cada key, seguido del value asociado. El "\n" de la primera llamada a print() garantiza que se inserte una línea en blanco antes de cada par clave-valor en la salida:

Key: username  
Value: efermi  
  
Key: first  
Value: enrico  
  
Key: last  
Value: fermi

El bucle a través de todos los pares clave-valor funciona especialmente bien para diccionarios como el ejemplo *favorite\_languages.py* de la página 96, que almacena el mismo tipo de información para muchas claves diferentes. Si recorres el diccionario favorite\_languages, obtendrás el nombre de cada persona del diccionario y su lenguaje de programación favorito. Como las claves siempre se refieren al nombre de una persona y el valor siempre es un lenguaje, utilizaremos las variables name y language en el bucle en lugar de key y value. Así será más fácil seguir lo que ocurre dentro del bucle:

**favorite\_languages.py**

favorite\_languages = {  
 'jen': 'python',  
 'sarah': 'c',  
 'edward': 'rust',  
 'phil': 'python',  
 }  
  
for name, language in favorite\_languages.items():  
 print(f"{name.title()}'s favorite language is {language.title()}.")

Este código le dice a Python que recorra cada par clave-valor del diccionario. A medida que recorre cada par, la clave se asigna a la variable name, y el valor se asigna a la variable language. Estos nombres descriptivos hacen que sea mucho más fácil ver lo que hace la llamada a print().

Ahora, en unas pocas líneas de código, podemos mostrar toda la información del sondeo:

Jen's favorite language is Python.  
Sarah's favorite language is C.  
Edward's favorite language is Rust.  
Phil's favorite language is Python.

Este tipo de bucle funcionaría igual de bien si nuestro diccionario almacenara los resultados de encuestar a mil o incluso a un millón de personas.

[anterior](c06_15.html)[Subtema 16 de 28: (Ver todo)](c06.html)[siguiente](c06_17.html)