### Uso de argumentos arbitrarios de palabras clave

A veces querrás aceptar un número arbitrario de argumentos, pero no sabrás de antemano qué tipo de información se pasará a la función. En este caso, puedes escribir funciones que acepten tantos pares clave-valor como proporcione la sentencia de llamada. Un ejemplo tiene que ver con la creación de perfiles de usuario: sabes que obtendrás información sobre un usuario, pero no estás seguro de qué tipo de información recibirás. La función build\_profile() del siguiente ejemplo siempre recibe un nombre y un apellido, pero también acepta un número arbitrario de argumentos de palabra clave:

**user\_profile.py**

def build\_profile(first, last, \*\*user\_info):  
 """Build a dictionary containing everything we know about a user."""  
❶ user\_info['first\_name'] = first  
 user\_info['last\_name'] = last  
 return user\_info  
  
user\_profile = build\_profile('albert', 'einstein',  
 location='princeton',  
 field='physics')  
print(user\_profile)

La definición de build\_profile() espera un nombre y un apellido, y luego permite al usuario pasar tantos pares nombre-valor como desee. Los asteriscos dobles delante del parámetro \*\*user\_info hacen que Python cree un diccionario llamado user\_info que contiene todos los pares nombre-valor adicionales que recibe la función. Dentro de la función, puedes acceder a los pares clave-valor de user\_info como lo harías con cualquier diccionario.

En el cuerpo de build\_profile(), añadimos el nombre y los apellidos al diccionario user\_info porque siempre recibiremos estos dos datos del usuario ❶, y aún no se han colocado en el diccionario. Después devolvemos el diccionario user\_info a la línea de llamada a la función.

Llamamos a build\_profile(), pasándole el nombre 'albert', el apellido 'einstein', y los dos pares clave-valor location='princeton' y field='physics'. Asignamos el profile devuelto a user\_profile e imprimimos user\_profile:

{'location': 'princeton', 'field': 'physics',  
'first\_name': 'albert', 'last\_name': 'einstein'}

El diccionario devuelto contiene el nombre y apellidos del usuario y, en este caso, también la localidad y la especialidad. La función funcionará independientemente de cuántos pares clave-valor adicionales se proporcionen en la llamada a la función.

Puedes mezclar valores posicionales, de palabra clave y arbitrarios de muchas formas distintas al escribir tus propias funciones. Es útil saber que existen todos estos tipos de argumentos porque los verás a menudo cuando empieces a leer el código de otras personas. Utilizar correctamente los distintos tipos y saber cuándo utilizar cada uno de ellos requiere práctica. Por ahora, recuerda utilizar el método más sencillo que haga el trabajo. A medida que avances, aprenderás a utilizar cada vez el enfoque más eficaz.

[anterior](c08_28.html)[Subtema 29 de 40: (Ver todo)](c08.html)[siguiente](c08_30.html)