Aprende a programar con Minecraft

### **Diccionarios**

Los diccionarios son un tipo de lista que utiliza un enfoque diferente. En lugar de utilizar un índice para identificar los elementos, los diccionarios identifican los elementos utilizando un conjunto de claves definidas por el programador.

Por ejemplo, este diccionario raceTimes almacena los nombres de las personas que corrieron en una carrera y sus tiempos de carrera:

raceTimes = {'Katy': 26,  
'Alex': 30,  
'Richard': 19}

La clave identifica de forma única cada valor del diccionario. En este ejemplo, la clave es el nombre de la persona. La clave "Katy " tiene un valor asociado de 26.

Al igual que las listas, los diccionarios son mutables; su contenido puede modificarse.

#### **Definir un diccionario**

Para definir un diccionario, utiliza un par de llaves alrededor de un conjunto de pares clave-valor. Por ejemplo, puedes utilizar un diccionario para describir a una persona. Puedes utilizar claves como "nombre" y "animal favorito " para almacenar información sobre la persona, de esta forma

person = {'name': 'David',  
'age': 42,  
'favoriteAnimal': 'Snake',  
'favoritePlace': 'Inside a cardboard box'}

En este ejemplo, cada clave es una cadena. Cada clave se empareja con un valor mediante dos puntos. Por ejemplo, "edad" es una clave y 42 es su valor correspondiente. A continuación, los elementos del diccionario se separan mediante comas.

Te habrás dado cuenta de que utilizar diccionarios facilita al programador la comprensión de lo que representa cada elemento de la lista; por ejemplo, es fácil entender que la clave "nombre " almacena un nombre, no un número u otra información aleatoria.

También puedes utilizar enteros y flotantes como claves del diccionario. Utilizar flotantes o enteros en los diccionarios es muy útil cuando las claves que quieres emparejar con valores no siguen una secuencia estricta.

El siguiente ejemplo crea un diccionario de horas de tren. La hora del tren (que es un flotante) se almacena como clave, y el destino del tren se almacena como valor:

trainTimes = {1.00: 'Castle Town',  
2.30: 'Sheep Farm',  
3.15: 'Lake City',  
3.45: 'Castle Town',  
3.55: 'Storage Land'  
}

Como los diccionarios pueden almacenar dos datos que van juntos como un par, son ideales para una situación como ésta. Si utilizara una lista de destinos de trenes en lugar de un diccionario, no podría emparejar los tiempos con los destinos. Sólo podría utilizar las posiciones de índice de la lista, que serían 0, 1, 2, 3, 4, etc., en lugar de las horas.

#### **Acceder a los elementos de los diccionarios**

Para acceder al valor de un elemento de un diccionario, utiliza corchetes y una clave en lugar de un índice. La clave suele ser una cadena o un número entero. Cuando crees un diccionario que utilice cadenas como claves, asegúrate de entrecomillarlas.

Por ejemplo, para acceder al valor de la clave "nombre " en el diccionario de personas creado anteriormente, utilizarías esta sintaxis

person = {'name': 'David',  
'age': 42,  
'favoriteAnimal': 'Snake',  
'favoritePlace': 'Inside a cardboard box'}  
  
agentName = person['name']

La variable NombreAgente contendrá el valor "David " porque accede al valor de la clave " nombre ". Del mismo modo, si quisieras acceder a la edad del agente, utilizarías la clave ' edad ':

agentAge = person['age']

Esto almacenaría el valor 42 en la variable agentAge.

En el ejemplo de trainTimes, puedes acceder a los valores del diccionario (los destinos) utilizando sus valores clave (las horas de los trenes), que son flotantes:

trainTimes = {1.00: 'Castle Town',  
2.30: 'Sheep Farm',  
3.15: 'Lake City',  
3.45: 'Castle Town',  
3.55: 'Storage Land'  
}  
  
myTrain = trainTimes[3.15]

Al acceder a la clave 3.15 del diccionario trainTimes, la variable myTrain se establece en 'Lake City'.

#### **Misión nº 53: Guía turística**

Cuando utilizas diccionarios, puedes almacenar cualquier tipo de datos como valor, incluso listas y tuplas. Por ejemplo, puedes almacenar una tupla que contenga los valores de x, y y z. He aquí un ejemplo de código que hace exactamente eso:

places = {'Living room': (76, 1, -61), 'Bedroom': (61, 9, -61)}

El diccionario de lugares almacena dos elementos. La clave del diccionario es el nombre de una ubicación en mi juego Minecraft (como mi salón o mi dormitorio), y el valor es una tupla de las coordenadas. Si quisiera acceder a las coordenadas de mi salón, utilizaría el siguiente código:

location = places['Living room']  
x, y, z = location[0], location[1], location[2]

Tu misión es crear un programa que utilice un diccionario para almacenar las ubicaciones de los distintos lugares de tu juego Minecraft, de forma que puedas teletransportarte a ellos por su nombre. Incluye tantas localizaciones en el diccionario como quieras. Para teletransportarte a esos lugares, tienes que acceder a la tupla de coordenadas almacenada en el diccionario y luego establecer x, y y z a los valores almacenados en la tupla. Los comentarios del código muestran dónde hacerlo.

Copia [el Listado 9-7](ch09.xhtml#ch9ex7) en el editor de texto IDLE y guárdalo en la carpeta de *listas* como *guiaviajes.py*.

*sightseeingGuide.py*

from mcpi.minecraft import Minecraft  
mc = Minecraft.create()  
  
# Add locations to the dictionary  
places = {}  
  
choice = ""  
while choice != "exit":  
➊ choice = input("Enter a location ('exit' to close): ")  
➋ if choice in places:  
# Store the dictionary item's value using its key (choice)  
location =  
# Store the values stored in the tuple in the x, y, and z variables  
x, y, z =  
mc.player.setTilePos(x, y, z)

*Listado 9-7: Código para teletransportarse a distintos lugares*

He incluido una sentencia que te pide que introduzcas el nombre del lugar al que quieres ir. Esta información se almacena en la variable de elección ➊. A continuación, el programa utiliza una sentencia if para comprobar si el valor de elección está en el diccionario ➋. La última línea utiliza las variables x, y y z para teletransportar al jugador a la posición almacenada en el diccionario.

Cuando se ejecute el programa, introduce el nombre del lugar al que quieres ir. [La Figura 9-8](ch09.xhtml#ch9fig8) muestra mi versión del programa teletransportándome a distintos lugares de mi juego.



*Figura 9-8: Me teletransporté a mi salón (arriba) y a mi dormitorio (abajo).*

#### **Cambiar o añadir un elemento en un diccionario**

No cuesta mucho trabajo cambiar el valor de un elemento de un diccionario. Utiliza corchetes con una tecla para acceder al elemento y establécelo como harías con una variable normal (con un signo igual). También puedes añadir un nuevo elemento utilizando este método.

Cambiemos el valor del elemento edad del diccionario persona de 42 a 43:

person['age'] = 43

Añadamos también un nuevo elemento llamado ubicación con el valor "USS Discovery":

person['location'] = 'USS Discovery'

Después de ejecutar este código, el diccionario tendrá una nueva clave llamada ubicación con el valor "USS Discovery".

#### **Borrar elementos de los diccionarios**

A veces querrás borrar un elemento de un diccionario. Al igual que con una lista, utiliza la palabra clave del para hacerlo. Por ejemplo, para borrar el ítem AnimalFavorito del diccionario Persona, harías lo siguiente:

del person['favoriteAnimal']

Como puedes ver, funciona igual que borrar elementos de una lista.

#### **Misión nº 54: Bloquear la puntuación de los aciertos**

En la Misión nº 50[(página 184](ch09.xhtml#page_184)), escribiste un programa que cuenta el número de veces que el jugador golpea un bloque con su espada en 60 segundos. Por muy divertido que sea el programa, sería aún más genial si pudieras registrar las puntuaciones de todos los que jugaron.

Para añadir un marcador al juego, utilizarás un diccionario. El diccionario almacenará el nombre del jugador y su puntuación, que luego podrá mostrarse junto a las puntuaciones de todos los demás.

Para empezar, abre *swordHits.py* y guárdalo como *swordHitsScore.py* en la carpeta de *listas*. Actualiza el código para que coincida con [el Listado 9-8](ch09.xhtml#ch9ex8), donde he hecho algunos cambios en el programa para que se repita, pregunte al jugador su nombre y luego imprima todas las puntuaciones. (También he incluido las soluciones al código que falta en *swordHits.py*.) Las secciones antiguas están en gris. (Recuerda sangrar todo dentro del bucle).

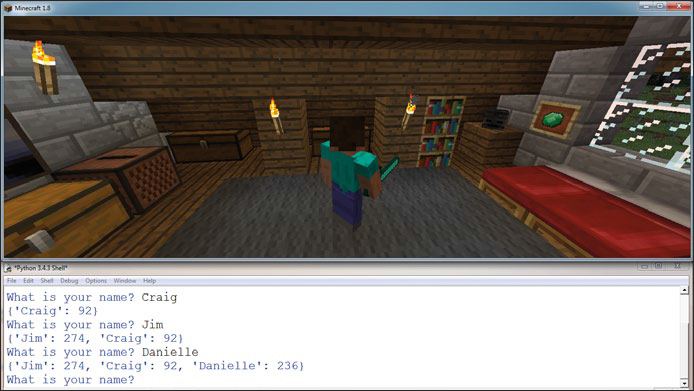
*swordHitsScore.py*

# Connect to the Minecraft game  
from mcpi.minecraft import Minecraft  
mc = Minecraft.create()  
  
import time  
  
name = ""  
scoreboard = {}  
  
while True:  
# Get the player's name  
name = input("What is your name? ")  
# Break loop if name is exit  
if name == "exit":  
break  
mc.postToChat("Go!")  
  
# Wait 60 seconds  
time.sleep(60)  
  
# Get the list of block hits  
blockHits = mc.events.pollBlockHits()  
  
# Display the length of the block hits list to chat  
blockHitsLength = len(blockHits)  
mc.postToChat("Your score is " + str(blockHitsLength))  
  
➊ # Add the player to the scoreboard  
  
# Display the scoreboard  
print(scoreboard)

*Listado 9-8: Cuando el código esté completo, añadirá un marcador al juego de golpes de bloque.*

Para terminar el programa, necesitas almacenar el nombre y la puntuación de cada jugador que participe en el juego. Hazlo añadiendo un nuevo elemento de diccionario utilizando los datos del código en ➊. El diccionario se llama marcador, y el nombre del jugador se almacena en la variable nombre.

[La Figura 9-9](ch09.xhtml#ch9fig9) muestra la salida de mi marcador.



*Figura 9-9: Mis amigos y yo jugamos una partida, y Jim es el ganador con 274 aciertos de bloque.*

**NOTA**

*Te habrás dado cuenta de que cuando* se *imprime el diccionario del marcador, no es fácil de leer. Aprenderás a solucionarlo en la Misión 59 (*[*página 209*](ch10.xhtml#page_209)*).*

**OBJETIVO EXTRA: MEJOR PUNTUACIÓN**

Actualmente, si alguien juega al juego *swordHitsScore.* py dos o más veces (e introduce el mismo nombre de usuario), el programa sólo registrará su puntuación más reciente. ¿Puedes averiguar cómo utilizar una sentencia if para comprobar si el jugador ya ha introducido una puntuación y almacenar la nueva puntuación sólo si es mayor que la anterior? Aquí tienes el principio del código para ayudarte. Comprueba si el nombre de alguien ya está en el diccionario del marcador:

if name in scoreboard:

[anterior](ch09_6.html)[Subtema 7 de 8: (Ver todo)](ch09.html)[siguiente](ch09_8.html)