Aprende a programar con Minecraft

### **Convertir cadenas en enteros con int()**

Al igual que la función str (), que convierte tipos de datos que no son cadenas en cadenas, la función int( ) convierte tipos de datos que no son enteros en enteros.

La función int () es útil cuando se utiliza con la función input (). La función input () devuelve la entrada del usuario como una cadena, pero a menudo querrás utilizar esta entrada en operaciones matemáticas. Para ello, primero tendrás que convertir la entrada a un tipo entero utilizando int( ).

Funciona así. Supongamos que ya hemos asignado un valor entero a una variable llamada latasDeAtúnPorGato, y queremos un programa que nos diga cuánto atún se come, en función del número de gatos que tenga el usuario. Aquí tienes un ejemplo de programa que podríamos escribir:

cansOfTunaPerCat = 4  
cats = input("How many cats do you have? ")  
cats = int(cats)  
dailyTunaEaten = cats \* cansOfTunaPerCat

Puedes hacer lo mismo en una sola línea poniendo una función dentro de la otra:

cats = int(input("How many cats do you have? "))  
dailyTunaEaten = cats \* cansOfTunaPerCat

Ahora que sabes cómo convertir la entrada en un número entero, puedes utilizarlo para introducir tipos de bloque en los programas de Minecraft.

#### **Misión nº 14: Crear un bloque con entrada**

Hay montones de tipos de bloques en Minecraft. Aunque puedes elegir muchos bloques en el modo creativo, muchos otros no se pueden utilizar. Sin embargo, la API de Python para Minecraft te da acceso a todos los tipos de bloque y te permite establecerlos mediante la función setBlocks().

Ya has utilizado antes la función setBlocks (), pero tenías que codificar el tipo de bloque en tu programa. Esto significa que no podías cambiarlo mientras se ejecutaba el programa. Ahora puedes utilizar la función input(). Si escribes un programa que acepte entradas, cada vez que ejecutes el programa podrás elegir el tipo de bloque que quieres crear. Podrías crear un bloque de lana la primera vez que ejecutes el programa, y luego crear mineral de hierro la segunda vez.

En esta misión, escribirás un programa que permita al usuario decidir qué tipo de bloque quiere crear. Copia [el Listado 4-3](ch04.xhtml#ch4ex3) en un nuevo archivo y guárdalo como *blockInput.py* en tu carpeta de *cadenas*.

*blockInput.py*

from mcpi.minecraft import Minecraft  
mc = Minecraft.create()  
➊ blockType = # Add input() function here  
pos = mc.player.getTilePos()  
x = pos.x  
y = pos.y  
z = pos.z  
mc.setBlock(x, y, z, blockType)

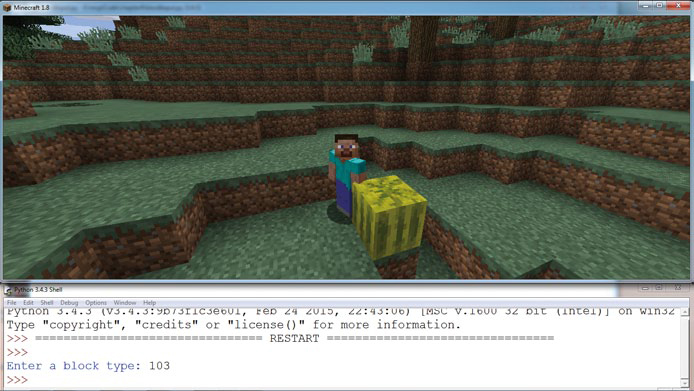
*Listado 4-3: Código para establecer un bloque en la posición del jugador*

Este programa establece un bloque en la posición actual del jugador. Cámbialo para que la variable blockType se establezca mediante la función input () ➊. Te sugiero que incluyas una pregunta u otro aviso de texto para que el usuario sepa que debe escribir un número de bloque, y no otro tipo de entrada. Si no incluyes una pregunta, IDLE se limitará a esperar en una línea en blanco hasta que el usuario introduzca algo, y tú quieres dejar claro que el programa necesita un número del usuario.

Recuerda que input() devuelve la entrada como una cadena, y para que introduzca el valor como un número entero, tienes que utilizar la función int(). La expresión para recoger la entrada para el tipo de bloque debería tener este aspecto:

blockType = int(input("Enter a block type: "))

Guarda tu programa modificado, ejecútalo e introduce el número de bloque que quieras. [La Figura 4-4](ch04.xhtml#ch4fig4) muestra el resultado del programa.



*Figura 4-4: ¡Ahora puedo crear el bloque que quiera!*

**OBJETIVO EXTRA: JUEGOS MÁS INTERACTIVOS**

Puedes utilizar tantas entradas en el programa como quieras. De momento, el programa crea el bloque en la posición actual del jugador. Piensa cómo establecerías las variables x, y y z utilizando entradas. Si te sientes realmente aventurero, intenta teletransportar al jugador a unas coordenadas concretas utilizando entradas.

[anterior](ch04_5.html)[Subtema 6 de 8: (Ver todo)](ch04.html)[siguiente](ch04_7.html)