Aprende a programar con Minecraft

### **Uso de sentencias if para comprobar un rango de valores**

Como aprendiste en [el Capítulo 5](ch05.xhtml#ch05), en Python puedes determinar si un valor se encuentra entre otros dos valores. Como una comprobación de rango se evalúa como Verdadero o Falso, puedes utilizar una comprobación de rango como condición en una sentencia if, igual que utilizarías una simple comparación menor que/mayor que o igual a/no igual a. Cualquier expresión que se evalúe como Verdadero o Falso puede ser una condición de una sentencia if.

Digamos que te has pasado todo el día reuniendo ingredientes para hornear unos deliciosos pasteles de Minecraft. Encuentras suficientes ingredientes para hornear 30 pasteles, y ahora quieres vender los pasteles. La persona que te compre pasteles debe comprar 1 pastel pero menos de 30; de lo contrario, no le venderás pasteles a esa persona. ¡No pueden acaparar todos los pasteles!

Este código representa la situación de los pasteles:

cakes = int(input("Enter the number of cakes to buy: "))  
➊ if 0 < cakes < 30:  
print("Here are your " + str(cakes) + " cakes.")  
➋ elif cakes == 0:  
print("Don't you want some delicious cake?")  
➌ else:  
print("That's too many cakes! Don't be selfish!")

Si la variable pasteles tiene un valor entre 0 y 30, como 15, imprimimos "Aquí tienes tus 15 pasteles". ➊. En caso contrario, imprimimos un mensaje diferente. Si pasteles tiene el valor 0, imprimimos "¿No quieres un delicioso pastel? ". ➋ y si es mayor que 30, imprimimos "¡Son demasiados past eles! No seas egoísta!" ➌.

Podemos probar una expresión más complicada añadiendo un operador booleano. Si fuera realmente raro y no quisiera que la gente comprara una cantidad de pan entre 20 y 30, podría hacerlo utilizando el operador no:

bread = int(input("Enter the amount of bread: "))  
if not 20 <= bread <= 30:  
print("Here are your " + bread + " breads.")  
else:  
print("I don't sell that amount of bread for some reason.")

Aquí utilizo un operador not y las comparaciones mayor que o igual a para comprobar un rango de valores como primera condición. La comprobación del rango determina si la cantidad de pan que la gente quiere comprar está entre 20 y 30. Entonces, el operador not pasa de Verdadero a Falso y de Falso a Verdadero. Por tanto, si el pan está dentro del intervalo, la expresión global se evalúa como Falso, y ejecutamos el código de la sentencia else. Si el pan no está en el rango entre 20 y 30 -digamos que es 40- la expresión global es Verdadera, e imprimimos "Aquí tienes tus 40 panes".

Si alguien intenta comprar 23 panes, no se lo permito. Pero 17 ó 32 está bien.

#### **Misión 31: Restringir las ubicaciones de teletransporte**

¿Recuerdas el programa de teletransporte que creaste en el [Capítulo 2](ch02.xhtml#ch02)? Se llamaba *teleport.py*. En esta misión, utilizarás comprobaciones de rango y sentencias if para limitar a dónde puede teletransportarse el jugador. Si utilizas Minecraft en la Raspberry Pi, hay lugares fuera del mundo del juego que no existen, pero tu programa te permitirá teletransportarte a ellos. Si estás utilizando la edición de escritorio de Minecraft, tu mundo es mucho más grande, por lo que no tienes las mismas restricciones que en la edición Pi del juego, pero este programa sigue siendo útil. Por ejemplo, podrías utilizarlo en un juego del escondite para limitar la zona donde pueden esconderse los jugadores.

[El listado 6-6](ch06.xhtml#ch6ex6) se supone que obtiene las coordenadas x, y y z de la entrada del usuario y las teletransporta a esa posición. Pero el programa no está completo.

*teleportLimit.py*

from mcpi.minecraft import Minecraft  
mc = Minecraft.create()  
valid = True  
  
x = int(input("Enter x: "))  
y = int(input("Enter y: "))  
z = int(input("Enter z: "))  
  
if not -127 < x < 127:  
valid = False  
  
# check if y is not between -63 and 63  
  
# check if z is not between -127 and 127  
  
if valid:  
mc.player.setPos(x, y, z)  
else:  
mc.postToChat("Please enter a valid location")

*Listado 6-6: Un programa para limitar los lugares a los que puede teletransportarse el jugador*

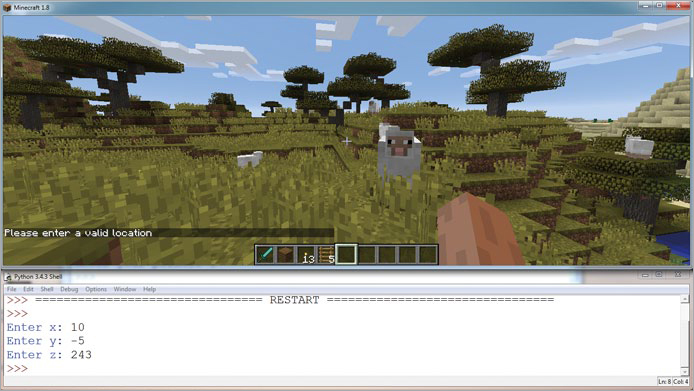
Para limitar a dónde puede teletransportarse el jugador, creamos una variable llamada válido. Esta variable almacenará un Verdadero o un Falso para representar si todas las coordenadas del destino son válidas o no. Pedimos al usuario que introduzca valores para x, y y z. A continuación, hacemos que una sentencia if compruebe si la variable x no está entre -127 y 127. Si no lo está, esta variable x no es válida. Si no lo está, esta coordenada x no es válida, y la variable válida se pone a Falso.

Cuando el programa llegue a la última sentencia if, sólo se llamará a setPos() si valid es True. Y válido será Verdadero sólo si se han cumplido las tres condiciones. De lo contrario, el jugador no podrá teletransportarse, y publicaremos un mensaje de chat indicando al usuario que introduzca una ubicación válida.

Crea un nuevo archivo en IDLE y copia en él [el Listado 6-6](ch06.xhtml#ch6ex6). Guarda el programa como *teleportLimit.py* en la carpeta *ifStatements*.

Completa el programa para que utilice sentencias if y comprobaciones de rango en las variables y y z y ponga valid en False si los valores no son válidos.

Cuando creas que has terminado el programa, ejecútalo. El programa debería teletransportarte cuando introduzcas valores que estén dentro del intervalo de -127 a 127 para las variables x y z y dentro del intervalo de -63 a 63 para la variable y. Cuando introduzcas un valor que no esté dentro de estos rangos, el programa no debería teletransportarte. [La Figura 6-6](ch06.xhtml#ch6fig6) muestra el aspecto que debería tener el juego cuando el usuario introduce un número no válido.



*Figura 6-6: La* variable z *era demasiado grande, por lo que no me teletransporté.*

**OBJETIVO EXTRA: MANTENTE SOBRE EL SUELO**

Uno de los problemas del programa de teletransporte es que puede teletransportarte bajo tierra, atrapándote allí. Puedes hacer cambios en el programa para impedir que el jugador se teletransporte bajo tierra. Compara la coordenada y que introduce el usuario con la función getHeight() para comprobar que el jugador se teletransporta por encima del suelo y detenerlo si se teletransporta bajo tierra.

[anterior](ch06_2.html)[Subtema 3 de 5: (Ver todo)](ch06.html)[siguiente](ch06_4.html)