Aprende a programar con Minecraft

### **Devolver valores con métodos**

Al igual que las funciones, los métodos también pueden devolver valores, o los atributos de un objeto, utilizando la palabra clave return. Por ejemplo, supongamos que queremos convertir el peso del gato Fluff de kilogramos a gramos. Un kilogramo equivale a 1000 gramos, así que para hacer la conversión, multiplica el atributo peso por 1000 y devuélvelo. Añade el siguiente método getPesoEnGramos() a la clase Gato en *catClass.py*:

*catClass.py*

class Cat(object):  
def \_\_init\_\_(self, name, weight):  
self.name = name  
self.weight = weight  
  
def getWeightInGrams(self):  
return self.weight \* 1000

Para obtener el valor devuelto por el método, creas un objeto y llamas al método. En el código siguiente, se utiliza el objeto pelusa, y se llama al método dentro de una función print( ) para obtener el peso del gato en gramos:

*catClass.py*

fluff = Cat("Fluff", 4.5)  
print(fluff.getWeightInGrams())

Ahora, cuando ejecutes el archivo, saldrá lo siguiente:

4500

En la próxima misión, ampliaremos el programa de la casa fantasma para incluir un método que devuelva información sobre el edificio.

#### **Misión nº 70: Castillo fantasma**

Tengo en mente todo tipo de nombres para los distintos lugares que he construido en mi mundo Minecraft: la casa de la playa, la granja de plantas, la granja de animales, el almacén, el palacio, el palacio submarino, el palacio subterráneo, y un montón más. ¡El problema es que los nombres sólo existen en mi cabeza!

Con las clases, puedes crear atributos como ubicación y tamaño para las cosas que construyas, como viste en la Misión 69[(página 267](ch12.xhtml#page_267)). ¡También puedes incluir nombres!

Pongamos un nombre a la casa fantasma y hagamos que Python lo recuerde por nosotros. Actualizaremos la clase Edificio de la Misión nº 69 para añadir un método extra que devuelva el nombre del edificio. Copia [el Listado 12-3](ch12.xhtml#ch12ex3) en un nuevo archivo llamado *castillofantasma.py* en la carpeta de *clases*.

*ghostCastle.py*

from mcpi.minecraft import Minecraft  
mc = Minecraft.create()  
import time  
  
➊ class NamedBuilding(object):  
➋ def \_\_init\_\_(self, x, y, z, width, height, depth, name):  
self.x = x  
self.y = y  
self.z = z  
  
self.width = width  
self.height = height  
self.depth = depth  
  
➌ self.name = name  
  
def build(self):  
mc.setBlocks(self.x, self.y, self.z,  
self.x + self.width, self.y + self.height,  
self.z + self.depth, 4)  
  
mc.setBlocks(self.x + 1, self.y + 1, self.z + 1,  
self.x + self.width - 1, self.y + self.height - 1,  
self.z + self.depth - 1, 0)  
  
def clear(self):  
mc.setBlocks(self.x, self.y, self.z,  
self.x + self.width, self.y + self.height,  
self.z + self.depth, 0)  
  
➍ def getInfo():  
# Add the body of the getInfo() method here  
  
pos = mc.player.getTilePos()  
x = pos.x  
y = pos.y  
z = pos.z  
ghostCastle = NamedBuilding(x, y, z, 10, 16, 16, "Ghost Castle")  
ghostCastle.build()  
➎ mc.postToChat(ghostCastle.getInfo())  
  
time.sleep(30)  
  
ghostCastle.clear()

*Listado 12-3:* EdificioNombrado *es muy similar a la* *clase* Edificio *, salvo que tiene un atributo extra llamado* nombre *y un método extra que devuelve una descripción del edificio.*

En primer lugar, he cambiado el nombre de la clase a EdificioNombrado para que no la confundamos con la clase Edificio de la misión anterior ➊. He añadido un argumento y un atributo extra al constructor llamado nombre ➋. El argumento te permite dar un nombre al edificio, y el constructor asigna el nombre al atributo nombre ➌.

Tu misión es añadir un método llamado getInfo() a la nueva clase EdificioNombrado que devuelva el nombre y la posición del edificio. Te he añadido el inicio del método getInfo () en ➍. Sólo tienes que añadir el cuerpo. El método getInfo () se invoca en el objeto castillofantasma en ➎ para que envíe la cadena devuelta por el método al chat de Minecraft. Por ejemplo, si el castillo fantasma está situado en x = -310, y = 64, z = 1081, el método getInfo ( ) debe devolver la cadena "La ubicación del castillo fantasma está en -310, 64, 1081".

[La Figura 12-3](ch12.xhtml#ch12fig3) muestra mi programa en funcionamiento. Aunque el castillo fantasma es más alto, se parece a la casa de la Misión nº 69. Esto se debe a que los métodos build( ) son los mismos para ambos, pero siéntete libre de cambiar tu versión del código para que tu edificio se parezca más a un castillo.



*Figura 12-3: Aparece la descripción del castillo fantasma.*

**OBJETIVO EXTRA: UNA CÁLIDA BIENVENIDA**

¿No sería genial que el nombre de cualquier edificio al que entraras apareciera automáticamente en el chat? Bueno, es posible, pero es un poco difícil. Si quieres intentarlo, puedes utilizar el programa *ducha.py* de la Misión 32[(página 124](ch06.xhtml#page_124)) como punto de partida. El archivo debe estar en tu carpeta *ifDeclaraciones*. Puedes utilizar el programa para detectar las coordenadas del jugador y, si está dentro del edificio, llamar al método getInfo() del objeto edificio.

[anterior](ch12_4.html)[Subtema 5 de 10: (Ver todo)](ch12.html)[siguiente](ch12_6.html)