Desconocido

### Ya que hablamos de bucles . . .

Un bucle for no es el único tipo de bucle que puedes hacer en Python. También existe el bucle while . Mientras que un bucle for tiene una longitud específica, un bucle while no la tiene. Se utiliza cuando no sabes con antelación cuándo debe detenerse el bucle.

Imagina una escalera con 20 peldaños. La escalera está en el interior, y sabes que puedes subir fácilmente 20 peldaños. Un bucle for es así:

>>> for step in range(0, 20):  
 print(step)

Ahora imagina una escalera que sube por la ladera de una montaña. La montaña es muy alta, y puede que te quedes sin energía antes de llegar a la cima. O puede que el tiempo empeore y te obligue a detenerte. Así es un bucle while :

step = 0  
while step < 10000:  
 print(step)  
 if tired == True:  
 break  
 elif badweather == True:  
 break  
 else:  
 step = step + 1

Si intentas ejecutar este código, obtendrás un error, porque no hemos creado las variables tired y badweather . Aunque aquí no hay suficiente código para hacer un programa que funcione, demuestra un bucle while sencillo.



Empezamos creando la variable step con step = 0 . A continuación, creamos un bucle while que comprueba si el valor de step es menor que 10.000 ( step < 10000 ), que es el número de pasos que hay desde la base de la montaña hasta la cima. Mientras step sea menor que 10.000, Python ejecutará el resto del código.

Con print(step) , imprimimos el valor de step y luego comprobamos si el valor de la variable tired es True con la condición if tired == True . Si lo es, utilizamos break para salir del bucle. La palabra clave break es una forma de saltar fuera (o detener) un bucle inmediatamente, y funciona tanto con los bucles while como con los for .

En este ejemplo, break hace que Python salte del bloque y pase a cualquier comando que apareciera después de la línea step = step + 1 .

La línea elif badweather == True comprueba si badweather está en True ; si es así, break sale del bucle. Si ni tired ni badweather son True (visto en else ), añadimos 1 a la variable step con step = step + 1 , y el bucle continúa.

Los pasos de un bucle while son los siguientes:

1. Comprueba la condición.
2. Ejecuta el código del bloque.
3. Repetir.

Lo más habitual es crear un bucle while con varias condiciones, en lugar de sólo una, como en este caso:

>>> x = 45  
>>> y = 80  
>>> while x < 50 and y < 100:  
 x = x + 1  
 y = y + 1  
 print(x, y)

Aquí, creamos una variable x con el valor 45 y una variable y con el valor 80. El bucle comprueba dos condiciones: si x es menor que 50 y si y es menor que 100. Si ambas condiciones son verdaderas, se ejecutan las líneas siguientes, que suman 1 a ambas variables y luego las imprimen. La salida de este código es la siguiente:

46 81  
47 82  
48 83  
49 84  
50 85

¿Puedes averiguar cómo funciona esto?

Empezamos a contar en 45 para x y en 80 para y , y luego *incrementamos* (añadimos 1 a cada variable) cada vez que se ejecuta el código del bucle. El bucle se ejecutará mientras x sea menor que 50 y y sea menor que 100. Después de ejecutar el bucle cinco veces, el valor de x alcanza 50. Ahora la primera condición ( x < 50 ) ya no es cierta, por lo que Python detiene el bucle.

También podemos utilizar un bucle while para crear un *bucle semi-eterno* que podría durar eternamente, pero que continúa hasta que ocurre algo en el código para salir de él.

Aquí tienes un ejemplo:

while True:  
 lots of code here  
 lots of code here  
 lots of code here  
 if some\_value == True:  
 break

La condición para el bucle while es simplemente True , que siempre es verdadera, por lo que el código del bloque siempre se ejecutará (por tanto, el bucle es eterno). Python saldrá del bucle sólo si la variable some\_value es verdadera.

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