### Utilizar range() para hacer una lista de números

Si quieres hacer una lista de números, puedes convertir los resultados de range() directamente en una lista utilizando la función list(). Cuando envuelvas list() en una llamada a la función range(), la salida será una lista de números.

En el ejemplo de la sección anterior, simplemente imprimimos una serie de números. Podemos utilizar list() para convertir esa misma serie de números en una lista:

numbers = list(range(1, 6))  
print(numbers)

Este es el resultado:

[1, 2, 3, 4, 5]

También podemos utilizar la función range() para decirle a Python que se salte los números de un rango determinado. Si pasas un tercer argumento a range(), Python utiliza ese valor como tamaño de paso al generar los números.

Por ejemplo, aquí tienes cómo hacer una lista de los números pares entre el 1 y el 10:

**even\_numbers.py**

even\_numbers = list(range(2, 11, 2))  
print(even\_numbers)

En este ejemplo, la función range() comienza con el valor 2 y luego suma 2 a ese valor. Suma 2 repetidamente hasta que alcanza o sobrepasa el valor final, 11, y produce este resultado:

[2, 4, 6, 8, 10]

Puedes crear casi cualquier conjunto de números que desees utilizando la función range(). Por ejemplo, considera cómo podrías hacer una lista de los 10 primeros números cuadrados (es decir, el cuadrado de cada número entero del 1 al 10). En Python, dos asteriscos (\*\*) representan exponentes. He aquí cómo podrías poner los 10 primeros números cuadrados en una lista:

**square\_numbers.py**

squares = []  
for value in range(1, 11):  
❶ square = value \*\* 2  
❷ squares.append(square)  
  
print(squares)

Empezamos con una lista vacía llamada squares. A continuación, le decimos a Python que repase cada valor del 1 al 10 utilizando la función range(). Dentro del bucle, el valor actual se eleva a la segunda potencia y se asigna a la variable square ❶. A continuación, cada nuevo valor de square se añade a la lista squares ❷. Por último, cuando el bucle ha terminado de ejecutarse, se imprime la lista de cuadrados:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Para escribir este código de forma más concisa, omite la variable temporal square y añade cada nuevo valor directamente a la lista:

squares = []  
for value in range(1,11):  
 squares.append(value\*\*2)  
  
print(squares)

Esta línea hace el mismo trabajo que las líneas dentro del bucle for del listado anterior. Cada valor del bucle se eleva a la segunda potencia y se añade inmediatamente a la lista de cuadrados.

Puedes utilizar cualquiera de estos métodos cuando hagas listas más complejas. A veces, utilizar una variable temporal hace que tu código sea más fácil de leer; otras veces hace que el código sea innecesariamente largo. Céntrate primero en escribir código que entiendas claramente y que haga lo que quieres que haga. Luego busca enfoques más eficientes a medida que revises tu código.

[anterior](c04_14.html)[Subtema 15 de 41: (Ver todo)](c04.html)[siguiente](c04_16.html)