### Elegir direcciones

Utilizaremos el método fill\_walk() para determinar la secuencia completa de puntos del recorrido. Añade este método a *random\_walk.py*:

**random\_walk.py**

def fill\_walk(self):  
 """Calculate all the points in the walk."""  
  
 # Keep taking steps until the walk reaches the desired length.  
❶ while len(self.x\_values) < self.num\_points:  
  
 # Decide which direction to go, and how far to go.  
❷ x\_direction = choice([1, -1])  
 x\_distance = choice([0, 1, 2, 3, 4])  
❸ x\_step = x\_direction \* x\_distance  
  
 y\_direction = choice([1, -1])  
 y\_distance = choice([0, 1, 2, 3, 4])  
❹ y\_step = y\_direction \* y\_distance  
  
 # Reject moves that go nowhere.  
❺ if x\_step == 0 and y\_step == 0:  
 continue  
  
 # Calculate the new position.  
❻ x = self.x\_values[-1] + x\_step  
 y = self.y\_values[-1] + y\_step  
  
 self.x\_values.append(x)  
 self.y\_values.append(y)

Primero establecemos un bucle que se ejecuta hasta que el paseo se llena con el número correcto de puntos ❶. La parte principal de fill\_walk() le dice a Python cómo simular cuatro decisiones aleatorias: ¿Irá el paseo hacia la derecha o hacia la izquierda? ¿Qué distancia recorrerá en esa dirección? ¿Irá hacia arriba o hacia abajo? ¿Qué distancia recorrerá en esa dirección?

Utilizamos choice([1, -1]) para elegir un valor para x\_direction, que devuelve 1 para el movimiento hacia la derecha o -1 para el movimiento hacia la izquierda ❷. A continuación, choice([0, 1, 2, 3, 4]) selecciona aleatoriamente una distancia para moverse en esa dirección. Asignamos este valor a x\_distance. La inclusión de un 0 permite la posibilidad de pasos con movimiento a lo largo de un solo eje.

Determinamos la longitud de cada paso en las direcciones *x-* y *y-*multiplicando la dirección del movimiento por la distancia elegida ❸❹. Un resultado positivo para x\_step significa moverse hacia la derecha, un resultado negativo significa moverse hacia la izquierda, y 0 significa moverse verticalmente. Un resultado positivo para y\_step significa moverse hacia arriba, negativo significa moverse hacia abajo y 0 significa moverse horizontalmente. Si los valores tanto de x\_step como de y\_step son 0, el recorrido no va a ninguna parte; cuando esto ocurre, continuamos el bucle ❺.

Para obtener el siguiente valor de *x* para el recorrido, sumamos el valor de x\_step al último valor almacenado en x\_values ❻ y hacemos lo mismo con los valores de *y*. Cuando tengamos las coordenadas del nuevo punto, las añadimos a x\_values y y\_values.

[anterior](c15_17.html)[Subtema 18 de 35: (Ver todo)](c15.html)[siguiente](c15_19.html)