### Refactorización de \_create\_fleet()

Si el código que hemos escrito hasta ahora fuera todo lo que necesitamos para crear una flota, probablemente dejaríamos \_create\_fleet() tal cual. Pero tenemos más trabajo que hacer, así que vamos a limpiar un poco el método. Añadiremos un nuevo método auxiliar, \_create\_alien(), y lo llamaremos desde \_create\_fleet():

**alien\_invasion.py**

def \_create\_fleet(self):  
 --snip--  
 while current\_x < (self.settings.screen\_width - 2 \* alien\_width):  
 self.\_create\_alien(current\_x)  
 current\_x += 2 \* alien\_width  
  
❶ def \_create\_alien(self, x\_position):  
 """Create an alien and place it in the row."""  
 new\_alien = Alien(self)  
 new\_alien.x = x\_position  
 new\_alien.rect.x = x\_position  
 self.aliens.add(new\_alien)

El método \_create\_alien() requiere un parámetro además de self: el -valor *x* que especifica dónde debe colocarse el ❶ alienígena . El código del cuerpo de \_create\_alien() es el mismo código que había en \_create\_fleet(), salvo que utilizamos el nombre de parámetro x\_position en lugar de current\_x. Esta refactorización facilitará la adición de nuevas filas y la creación de una flota completa.

[anterior](c13_8.html)[Subtema 9 de 33: (Ver todo)](c13.html)[siguiente](c13_10.html)