### Comprobar que una lista no está vacía

Hemos hecho una suposición sencilla sobre todas las listas con las que hemos trabajado hasta ahora: hemos supuesto que cada lista tiene al menos un elemento. Pronto dejaremos que los usuarios proporcionen la información que se almacena en una lista, por lo que no podremos suponer que una lista tiene algún elemento cada vez que se ejecute un bucle. En esta situación, es útil comprobar si una lista está vacía antes de ejecutar un bucle for.

Como ejemplo, comprobemos si la lista de ingredientes solicitados está vacía antes de construir la pizza. Si la lista está vacía, preguntaremos al usuario y nos aseguraremos de que quiere una pizza normal. Si la lista no está vacía, construiremos la pizza igual que en los ejemplos anteriores:

requested\_toppings = []  
  
if requested\_toppings:  
 for requested\_topping in requested\_toppings:  
 print(f"Adding {requested\_topping}.")  
 print("\nFinished making your pizza!")  
else:  
 print("Are you sure you want a plain pizza?")

Esta vez empezamos con una lista vacía de ingredientes solicitados. En lugar de saltar directamente a un bucle for, primero hacemos una comprobación rápida. Cuando se utiliza el nombre de una lista en una sentencia if, Python devuelve True si la lista contiene al menos un elemento; una lista vacía se evalúa como False. Si requested\_toppings supera la prueba condicional, ejecutamos el mismo bucle for que utilizamos en el ejemplo anterior . Si la prueba condicional falla, imprimimos un mensaje preguntando al cliente si realmente quiere una pizza normal sin ingredientes.

La lista está vacía en este caso, por lo que la salida pregunta si el usuario realmente quiere una pizza normal:

Are you sure you want a plain pizza?

Si la lista no está vacía, la salida mostrará cada uno de los ingredientes solicitados añadidos a la pizza.

[anterior](c05_22.html)[Subtema 23 de 28: (Ver todo)](c05.html)[siguiente](c05_24.html)