### Fallar en silencio

En el ejemplo anterior, informamos a nuestros usuarios de que uno de los archivos no estaba disponible. Pero no es necesario que informes de todas las excepciones que detectes. A veces, querrás que el programa falle silenciosamente cuando se produzca una excepción y continúe como si no hubiera pasado nada. Para hacer que un programa falle silenciosamente, escribe un bloque try como de costumbre, pero dile explícitamente a Python que no haga nada en el bloque except. Python tiene una sentencia pass que le dice que no haga nada en un bloque:

def count\_words(path):  
 """Count the approximate number of words in a file."""  
 try:  
 --snip--  
 except FileNotFoundError:  
 pass  
 else:  
 --snip--

La única diferencia entre este listado y el anterior es la sentencia pass en el bloque except. Ahora, cuando se lanza un FileNotFoundError, el código del bloque except se ejecuta, pero no ocurre nada. No se produce ningún rastreo, y no hay ninguna salida en respuesta al error que se ha producido. Los usuarios ven el recuento de palabras de cada archivo que existe, pero no ven ninguna indicación de que no se ha encontrado un archivo:

The file alice.txt has about 29594 words.  
The file moby\_dick.txt has about 215864 words.  
The file little\_women.txt has about 189142 words.

La declaración pass también actúa como marcador de posición. Es un recordatorio de que estás eligiendo no hacer nada en un punto concreto de la ejecución de tu programa y que puede que quieras hacer algo allí más adelante. Por ejemplo, en este programa podríamos decidir escribir cualquier nombre de archivo que falte en un archivo llamado *missing\_files.txt*. Nuestros usuarios no verían este archivo, pero nosotros podríamos leerlo y ocuparnos de cualquier texto que faltara.

[anterior](c10_27.html)[Subtema 28 de 37: (Ver todo)](c10.html)[siguiente](c10_29.html)