### Acceder a las líneas de un archivo

Cuando trabajes con un archivo, a menudo querrás examinar cada línea del mismo. Puede que busques cierta información en el archivo, o puede que quieras modificar el texto del archivo de alguna manera. Por ejemplo, puede que quieras leer un archivo de datos meteorológicos y trabajar con cualquier línea que incluya la palabra *sunny* en la descripción del tiempo de ese día. En un informe de noticias, podrías buscar cualquier línea con la etiqueta y reescribir esa línea con un tipo de formato específico.

Puedes utilizar el método splitlines() para convertir una cadena larga en un conjunto de líneas, y luego utilizar un bucle for para examinar cada línea de un archivo, de una en una:

**file\_reader.py**

from pathlib import Path  
  
path = Path('pi\_digits.txt')  
❶ contents = path.read\_text()  
  
❷ lines = contents.splitlines()  
for line in lines:  
 print(line)

Comenzamos leyendo todo el contenido del archivo, como hicimos anteriormente ❶. Si piensas trabajar con las líneas individuales de un archivo, no es necesario que elimines ningún espacio en blanco al leer el archivo. El método splitlines() devuelve una lista de todas las líneas del archivo, y asignamos esta lista a la variable lines ❷. A continuación, hacemos un bucle sobre estas líneas e imprimimos cada una de ellas:

3.1415926535  
 8979323846  
 2643383279

Como no hemos modificado ninguna de las líneas, la salida coincide exactamente con el fichero de texto original.

[anterior](c10_6.html)[Subtema 7 de 37: (Ver todo)](c10.html)[siguiente](c10_8.html)