Capítulo 4
Documentar y probar sus scripts en Python

1. Introducción

A partir de ahora, escribir pruebas unitarias es inevitable en la elaboración de un programa informático. A este respecto, Python se entrega con módulos opcionales que responden a las expectativas de los desarrolladores más ambiciosos. Python también le ofrece la posibilidad de inspeccionar su código de manera interactiva con REPL y comprobar instantáneamente el contenido de un objeto, su tipo y los métodos que ofrece. Para ayudar al desarrollador en las tareas relacionadas con la documentación, el análisis de rendimiento y la resolución de problemas en lo que respecta al código, la gama de módulos ofrecidos por el lenguaje es muy amplia. Por ejemplo, cuando el tamaño de un proyecto se hace crítico, el uso de pruebas unitarias permite implementar más rápidamente nuevas funcionalidades y detectar las regresiones de código desde el inicio de la implementación. Esto hace que se gane tiempo y productividad. La búsqueda de un buen rendimiento va a ayudar al desarrollador a identificar las funciones que consumen muchos recursos durante la ejecución y aquellas que consumen menos, con el objetivo de refactorizar y/o reescribir el código. En caso de escribir scripts destinados a la Raspberry Pi, donde los recursos son restringidos, auditar y hacer procesos de benchmarking sobre su código puede mejorar el tiempo de ejecución de un programa. De nuevo, aunque el objetivo de este libro no es mostrar la gestión de proyectos, sepa que es importante conocer las herramientas que ofrece Python a este respecto, con el objetivo de mejorar significativamente el ecosistema en su conjunto.

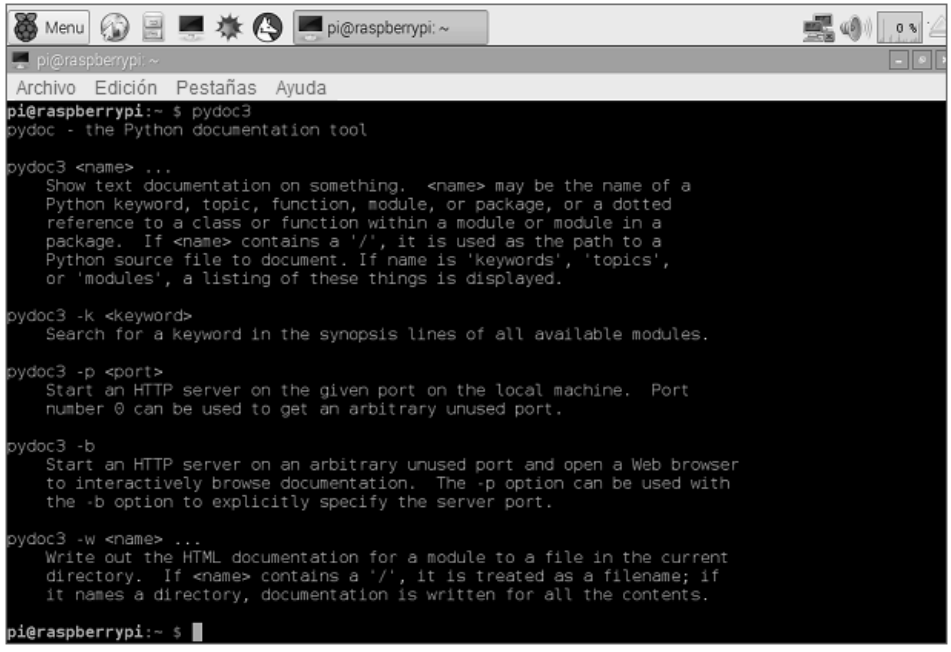
2. Consultar la documentación con pydoc3

En primer lugar, la herramienta más habitual que permite buscar en la documentación de los módulos se llama `pydoc3`. El uso de `pydoc3` interviene cuando queremos mostrar la documentación de un módulo, clase o palabra clave. Por ejemplo, la documentación de todas las palabras clave revisadas en los capítulos anteriores (`with`, `def`, `lambda`, etc.) se puede consultar con `pydoc3`. Un aspecto importante a observar antes de ir más lejos: la mayoría de la documentación instalada con el sistema por defecto de Python está escrita en inglés.

Esta herramienta se utiliza exclusivamente en línea de comandos. Abra una consola pulsando **Menú** - **Accesorios** - **LXTerminal**, como se explicó en el capítulo Entorno de programación. A continuación, escriba el siguiente comando:

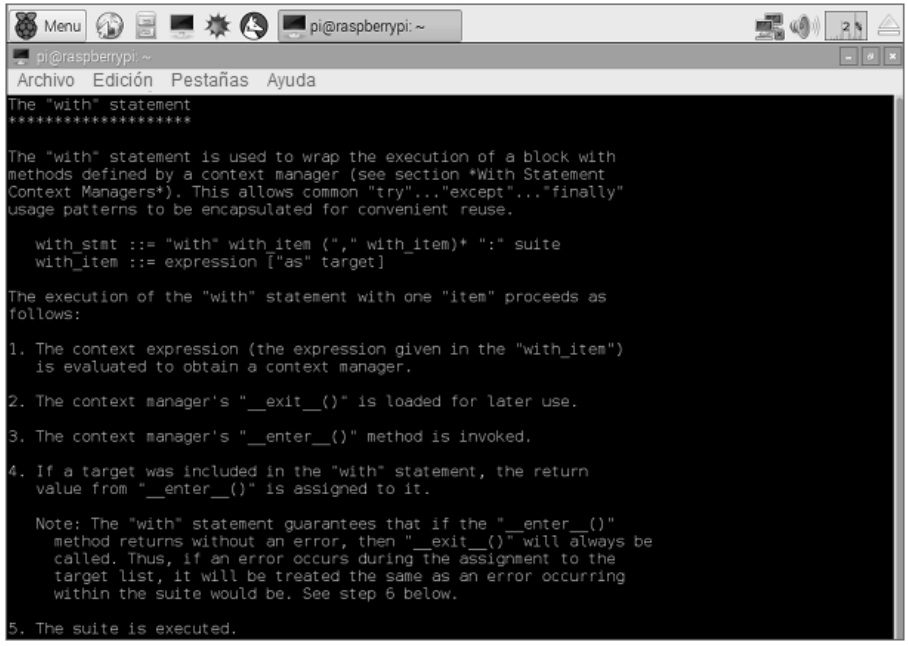
```
pi@raspberrypi:~ $ pydoc3
```

Debería mostrar el siguiente resultado:



Esto solo es la ayuda de `pydoc3`. Sin embargo, preste atención porque cada versión de Python instalada en el sistema se entrega con su propia versión de `pydoc`. Como se explica en el capítulo Entorno de programación, diferentes versiones del compilador Python deben convivir en la Raspberry Pi. Ocurre igual para `pydoc`, que se entrega en sus versiones 2.7 y 3.4. Por defecto, el binario `pydoc` apunta a `/usr/bin/pydoc2.7`. Por lo tanto, piense siempre en utilizar `pydoc3` para leer la documentación de la versión 3 de Python.

De este modo, `pydoc3` permite buscar y mostrar la documentación de numerosos *topics* o temas. El tema buscado se corresponde con el término que se ha pasado como argumento del comando. En este ejemplo, se busca el término `with` y se muestra la documentación que tiene asociada:

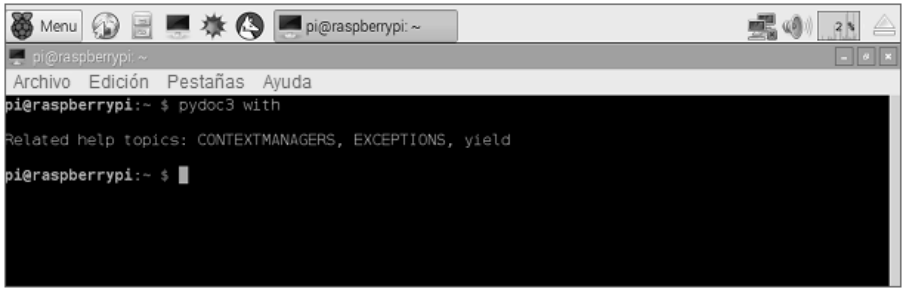


Por defecto, la lectura de la documentación se basa en el uso de un programa llamado `less`, que es el buscador por defecto de la mayor parte de distribuciones Linux actuales. La navegación en `less` puede resultar difícil para los principiantes que se inician en la línea de comandos.

A continuación se muestra una tabla que recoge los comandos básicos:

Mover hacia arriba un carácter	[Flecha hacia arriba]
Mover hacia abajo un carácter	[Flecha hacia abajo]
Mover hacia arriba una página	[Page Up]
Mover hacia abajo una página	[Page Down]
Buscar un término	Barra oblicua (slash) / seguida del término
Siguiente ocurrencia del término	Letra n en minúscula
Anterior ocurrencia del término	Letra N en mayúscula
Cerrar (salir) la documentación	Letra q en minúscula o Q en mayúscula

Navegar en `less` es relativamente fácil cuando se asimilan estos accesos directos. Una vez `less` se cierra, `pydoc3` sugiere los temas relacionados susceptibles de interesar al desarrollador:

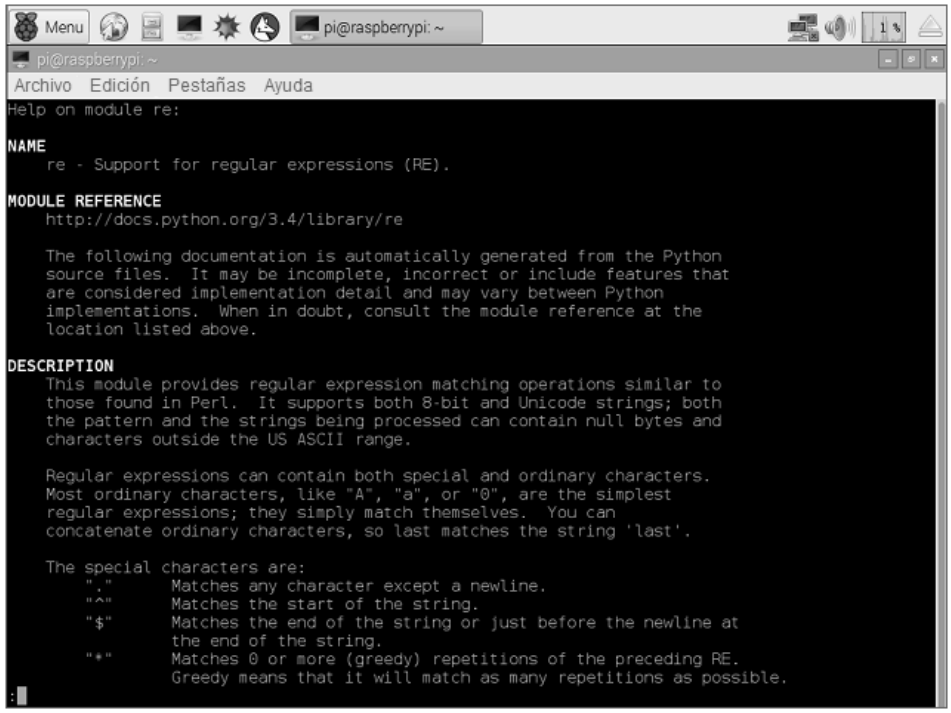


Además de buscar la documentación para prácticamente cualquier tema relativo al lenguaje, `pydoc3` busca también la documentación asociada a los módulos de la librería estándar, así como los módulos instalados con `pip3`, evidentemente si el autor del módulo en cuestión ha tenido la precaución de escribirla.

Para lograrlo, basta con pasar como argumento de `pydoc3` el nombre del módulo del que queremos consultar la documentación:

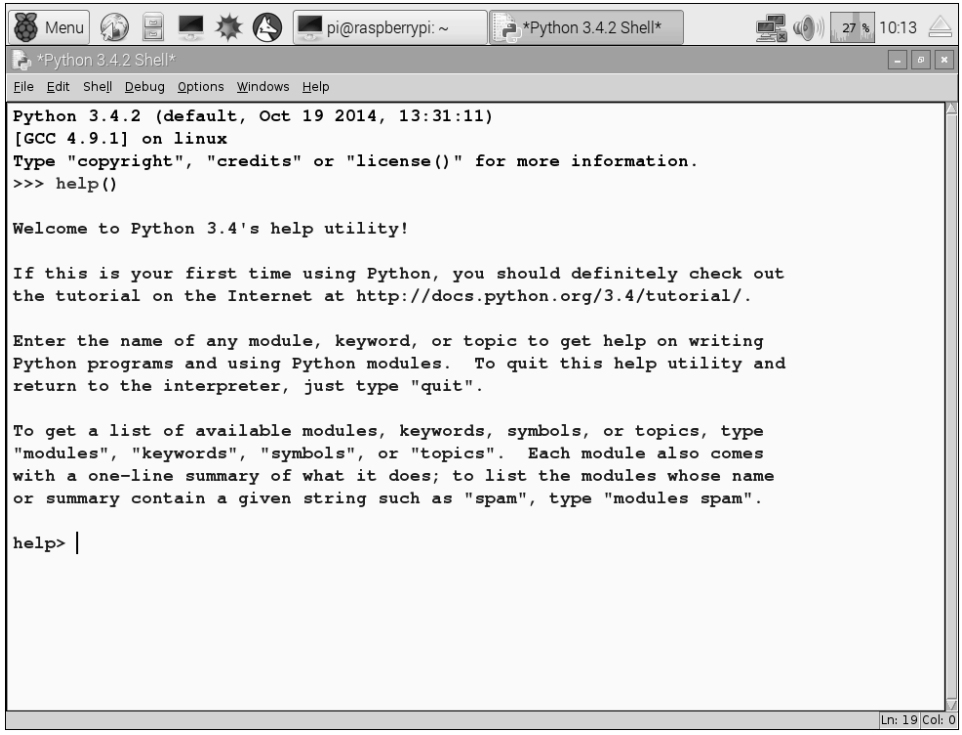
```
pi@raspberrypi:~ $ pydoc3 re
```

El resultado obtenido para la documentación asociada al módulo `re` es:



`pydoc3` también está disponible desde REPL de IDLE, gracias a la función `help()`. En efecto, esta función llama indirectamente a `pydoc3` para buscar documentación. `help()` se utiliza de dos maneras: la primera consiste en llamar a la función una vez en REPL, sin pasarle ningún argumento. El usuario cambia entonces a una segunda consola interactiva especializada en la búsqueda de documentación.

Escriba un término y deje que la consola realice la búsqueda para presentar la ayuda asociada al término:



La segunda consiste en pasar como argumento un objeto. Entonces `help()` va a analizar el objeto en cuestión y a mostrar la documentación asociada. En este caso, solo puede pasar los objetos ya instanciados o importados en el contexto actual.