

✓ ¡Felicitaciones! ¡Aprobaste!

Calificación recibida 100 % Para Aprobar 75 % o más

Ir al siguiente elemento

1. Quieres abrir el archivo "logs.txt" y almacenarlo en la variable `file` con el fin de leerlo. También quieres asegurarte de que se liberen todos los recursos y que el archivo se cierre después de leerlo. ¿Cuál es la línea de código correcta para hacer esto?

1 / 1 punto

- ☐ `with open("r", "logs.txt") as file:`
- ☐ `with file.open("logs.txt", "r"):`
- ☒ `with open("logs.txt", "r") as file:`
- ☐ `file = open("logs.txt", "r"):`

✓ Correcto

El código `with open("logs.txt", "r") as file:` es la línea de código correcta para hacer esto. La palabra clave `with` garantiza que todos los recursos se liberen al abrir y leer el archivo. Esto incluye asegurarse de que el archivo se cierre después de salir de la sentencia `with`. Luego, llamar a la función `open()` con el archivo "logs.txt" y "r" como argumentos indica leer el archivo "logs.txt". Finalmente, `as file` especifica que hay que almacenar el objeto de archivo en la variable `file`.

2. Después de abrir un archivo de registro como `login_file`, ¿qué línea de código puedes usar para leer el archivo y almacenarlo en una variable llamada `login_attempts`?

1 / 1 punto

- ☒ `login_attempts = login_file.read()`
- ☐ `login_file.read() as login_attempts`
- ☐ `login_attempts = login_file.reader()`
- ☐ `login_attempts = read(login_file)`

✓ Correcto

El código `login_attempts = login_file.read()` lee el archivo de registro y lo almacena en una variable llamada `login_attempts`. El método `.read()` convierte los archivos en cadenas. El código asigna la cadena que crea a otra variable llamada `login_attempts`.

3. Acabas de leer un archivo de registro en una variable llamada `file`. La variable `file` contiene una cadena de varias direcciones IP que están separadas por un espacio en blanco. ¿Qué línea de código separa cada dirección IP y la almacena como una lista en una variable llamada `ip_addresses`?

1 / 1 punto

- ☒ `ip_addresses = file.split()`
- ☐ `ip_addresses = split(file)`
- ☐ `split(file, ip_addresses)`
- ☐ `ip_addresses.split(file)`

✓ Correcto

El código `ip_addresses = file.split()` separa cada dirección IP en la variable `file` y, luego, las almacena como una lista en una variable llamada `ip_addresses`. El método `.split()` convierte una cadena en una lista. Separa la cadena en función de un carácter indicado a la función como argumento. Si no se pasa un carácter, separará la cadena cada vez que encuentre un espacio en blanco.

4. Necesitas saber si hay actividad de inicio de sesión inusual. Específicamente, tienes que revisar una lista de marcas de tiempo de inicio de sesión para determinar si alguno de los inicios de sesión ocurrió en un horario inusual. Si quisieras automatizar este proceso con Python, ¿qué deberías incluir en el código? Selecciona dos respuestas.

1 / 1 punto

- ☐ Una variable de contador que realice un seguimiento de la cantidad de intentos de inicio de sesión fallidos.
- ☒ Un bucle `for` que itere a través de la lista de marcas de tiempo.

✓ Correcto

El código debe incluir un bucle `for` que itere a través de la lista de marcas de tiempo y una sentencia `if` que verifique si la marca de tiempo de inicio de sesión se produjo en un horario inusual.

- ☐ Una sentencia `if` que compruebe si un usuario específico tiene varias marcas de tiempo de inicio de sesión en un horario inusual.

✓ Una sentencia `if` que verifica si la marca de tiempo de inicio de sesión se produjo en un horario inusual.

✓ **Correcto**

El código debe incluir un bucle `for` que itere a través de la lista de marcas de tiempo y una sentencia `if` que verifique si la marca de tiempo de inicio de sesión se produjo en un horario inusual.