# Algoritmo Python: actualizar un archivo

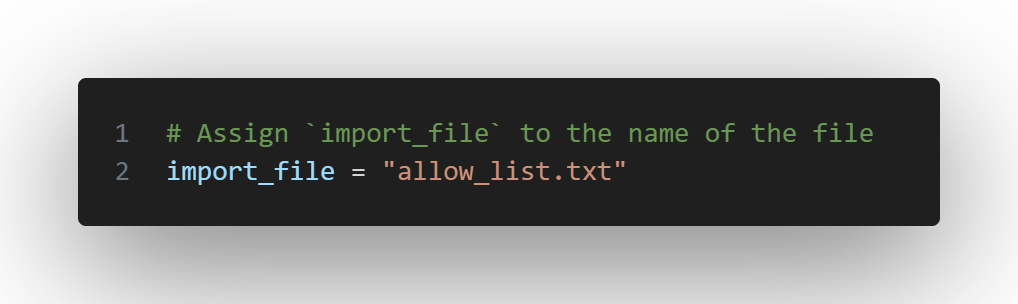
## Descripción del Proyecto

En mi organización, el acceso a contenido restringido se controla con una lista de direcciones IP permitidas. El archivo "allow\_list.txt" identifica estas direcciones IP.

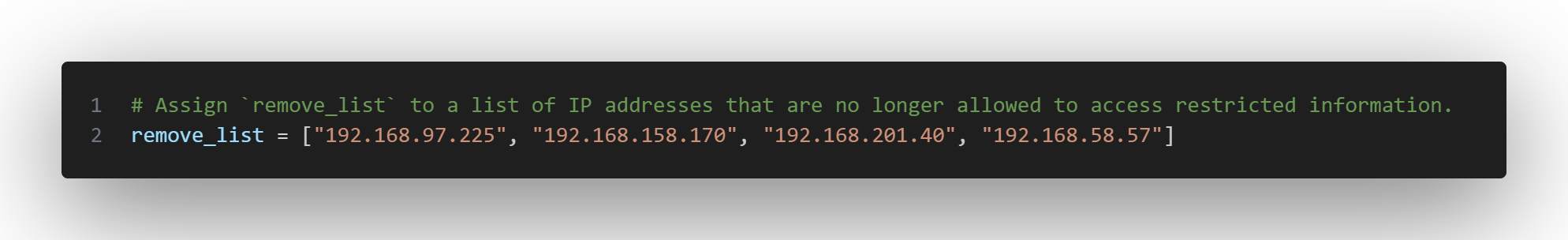
Una lista de eliminación independiente identifica las direcciones IP que ya no deberían tener acceso a este contenido. Creé un algoritmo para automatizar la actualización del archivo "allow\_list.txt" y eliminar estas direcciones IP que ya no deberían tener acceso.

## Abra el archivo que contiene la lista de permitidos

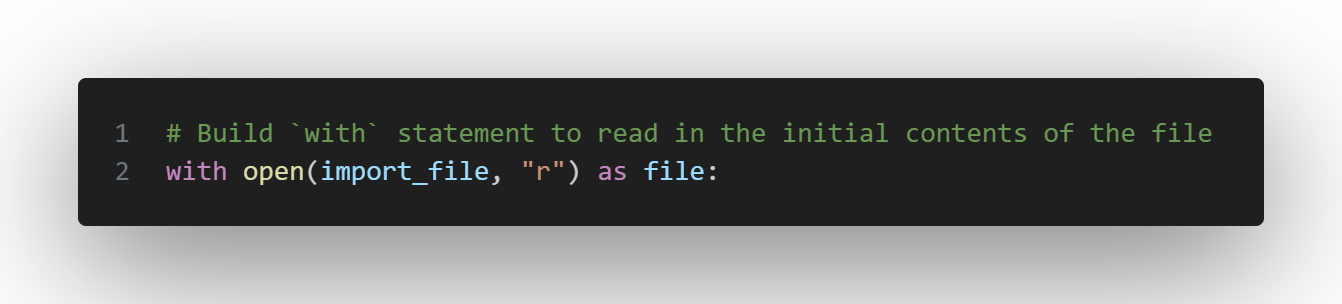
Para la primera parte del algoritmo, abrí el archivo "allow\_list.txt". Primero, asigné este nombre de archivo como una cadena a la variable import\_file:



Luego asigne el archivo ”remove\_list” a una lista de direcciones IP que ya no tienen permitido acceder a información restringida:



Luego, utilicé una declaración with para abrir el archivo:



En mi algoritmo, la declaración with se utiliza con la función .open() en modo de lectura para abrir el archivo de lista de permitidos con el fin de leerlo. El propósito de abrir el archivo es permitirme acceder a las direcciones IP almacenadas en el archivo de lista de permitidos. La palabra clave with ayudará a administrar los recursos cerrando el archivo después de salir de la declaración with.

En el código with open(import\_file, "r") as file:, la función open() tiene dos parámetros. El primero identifica el archivo a importar y luego el segundo indica lo que quiero hacer con el archivo. En este caso,"r" indica que quiero leerlo. El código también utiliza la palabra clave as para asignar una variable llamada file; file almacena la salida de la función .open() mientras trabajo dentro de la declaración with.

## Leer el contenido del archivo

Para leer el contenido del archivo, utilicé el método .read() para convertirlo en cadena:



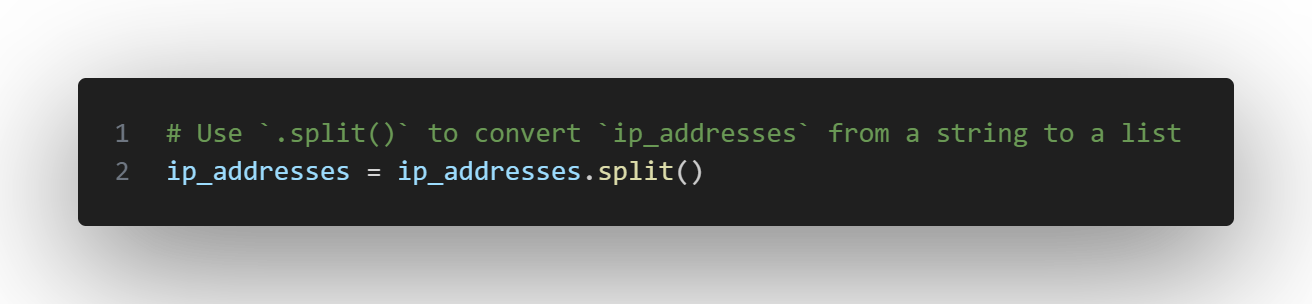
Al utilizar una función .open() que incluye el argumento"r" para "leer", puedo llamar a la función.read() en el cuerpo de la declaración with . El método.read() convierte el archivo en una cadena y me permite leerlo. Apliqué el método .read() a la variable de file identificada en la declaración with .

Luego, asigné la salida de cadena de este método a la variable ip\_addresses.

En resumen, este código lee el contenido del "allow\_list.txt" archivo en un formato de cadena que me permite luego usar la cadena para organizar y extraer datos en mi programa Python.

## Convertir la cadena en una lista

Para eliminar direcciones IP individuales de la lista de permitidos, necesitaba que estuvieran en formato de lista. Por lo tanto, a continuación utilicé el método .split() para convertir el ip\_addresses de cadena en una lista:

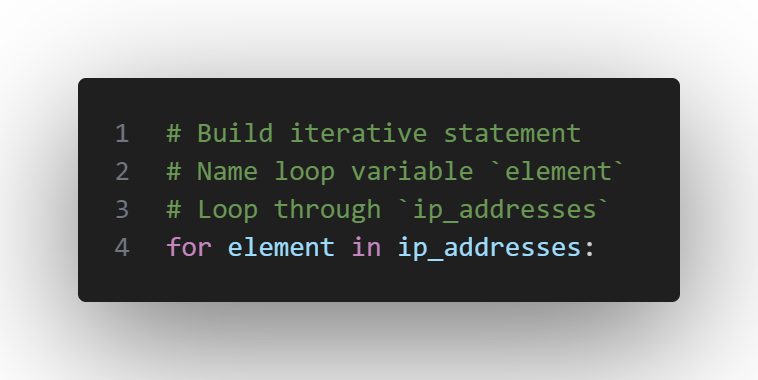


La función .split() se llama agregándole una variable de cadena. Funciona convirtiendo el contenido de una cadena en una lista. El propósito de dividir ip\_addresses en una lista es para facilitar la eliminación de direcciones IP de la lista de permitidos. De forma predeterminada, la función .split() divide el texto por espacios en blanco en elementos de lista.

En este algoritmo, la función .split() toma los datos almacenados en la variable ip\_addresses, que es una cadena de direcciones IP separadas por un espacio en blanco, y convierte esta cadena en una lista de direcciones IP. Para almacenar esta lista, la reasigné nuevamente a la variable ip\_addresses.

## Iterar a través de la lista de eliminación

Una parte clave de mi algoritmo implica iterar a través de las direcciones IP que son elementos en el remove\_list. Para ello, incorporé un loop for :



El loop for en Python, se repite el código de una secuencia específica. El propósito general de el loop for en un algoritmo de Python como este, se trata de aplicar declaraciones de código específicas a todos los elementos de una secuencia.

La palabra clave for inicia el loop for . Le sigue la variable del loop element y la palabra clave in. La palabra clave in indica iterar a través de la secuencia ip\_addresses y asignar cada valor a la variable del loop element.

## Eliminar direcciones IP que están en la lista de eliminación

Mi algoritmo requiere eliminar cualquier dirección IP de la lista de permitidos, ip\_addresses, que también esté contenido en remove\_list. Porque no había ningún duplicado en ip\_addresses, Pude usar el siguiente código para hacer esto:



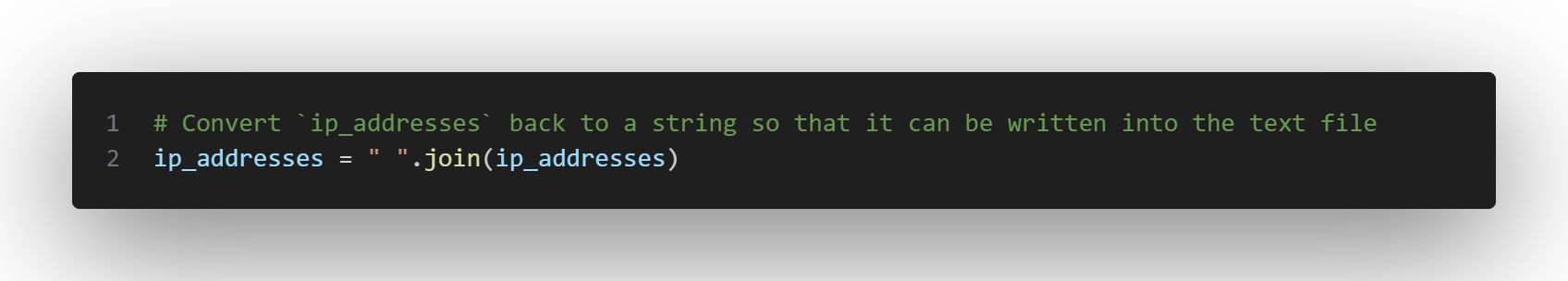
Primero, dentro de mi loop for , creé una condición que evaluaba si la variable de bucle element fue encontrada en la lista ip\_addresses . Hice esto porque al aplicar .remove() a elementos que no se encontraron en ip\_addresses esto podría provocar un error.

Entonces, dentro de ese condicional, apliqué .remove() a ip\_addresses. Pasé la variable de bucle element como argumento para que cada dirección IP que estaba en el remove\_list sería eliminado de

ip\_addresses.

## Actualice el archivo con la lista revisada de direcciones IP

Como siguiente paso de mi algoritmo, necesitaba actualizar el archivo de lista de permitidos con la lista revisada de direcciones IP. Para ello, primero tuve que convertir la lista nuevamente en una cadena. Use el método .join() para esto:

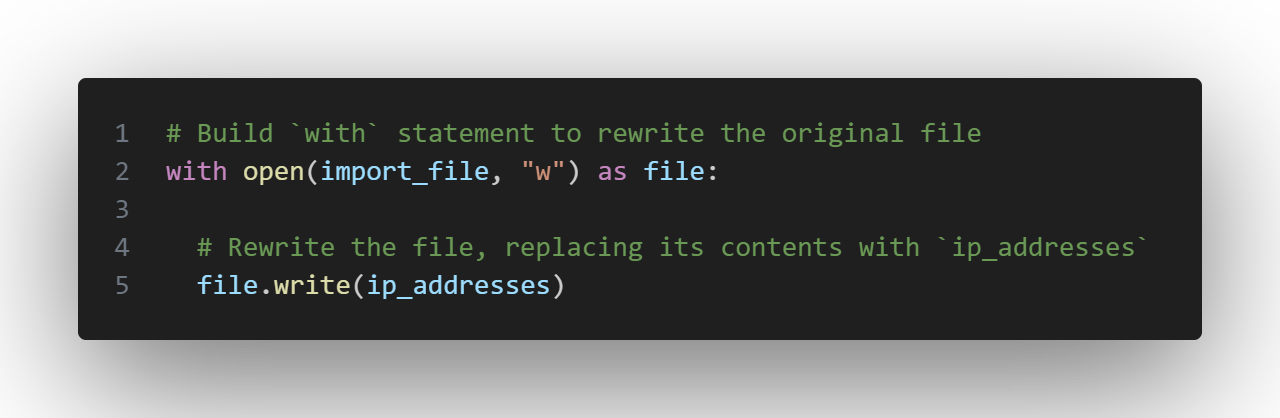


El método .join() combina todos los elementos de un iterable en una cadena. El método .join() se aplica a una cadena que contiene caracteres que separarán los elementos del iterable una vez unidos en una cadena.

En este algoritmo, use el método .join() para crear una cadena a partir de la lista ip\_addresses para poder pasarlo como argumento al método .write() al escribir en el archivo "allow\_list.txt".

Use el string (" ") (carácter vacío) como separador para indicar a Python que coloque cada elemento en una nueva línea.

Luego, utilicé otra declaración with y el método .write() para actualizar el archivo :



Esta vez, usé un segundo argumento de "w" con la función open() en mi declaración with . Este argumento indica que quiero abrir un archivo para escribir sobre su contenido. Al usar este argumento "w", puedo llamar a la función .write() en el cuerpo de la declaración with La función .write() escribe datos de cadena en un archivo especificado y reemplaza cualquier contenido de archivo existente.

En este caso, quería escribir la lista de permitidos actualizada como una cadena en el archivo "allow\_list.txt". De esta manera, el contenido restringido ya no será accesible para ninguna dirección IP que se haya eliminado de la lista de permitidos.

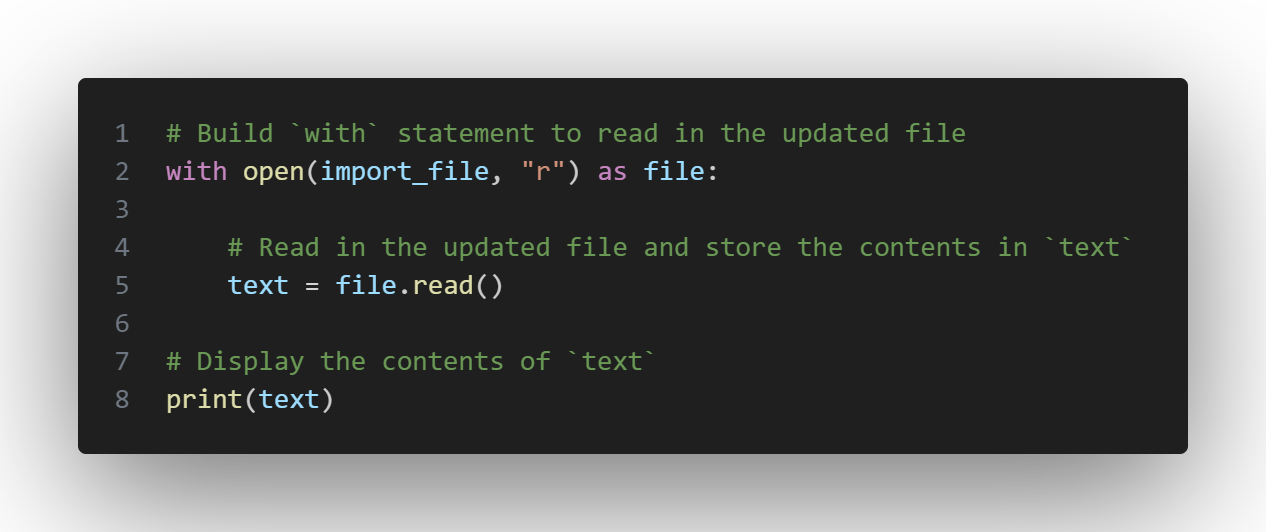
Para reescribir el archivo, agregué la función .write() al archivo de objeto file que identifiqué en la declaración with statement. Pasé la variable ip\_addresses como argumento para especificar que el contenido del archivo especificado en la declaración with debe reemplazarse con los datos en esta variable.

## 

## Leer el archivo con la lista revisada de direcciones IP

Como paso final de mi algoritmo, necesitaba leer el archivo actualizado de lista de permitidos con la lista revisada de direcciones IP.

Utilicé nuevamente la declaración with, la función open() , el argumento"r", el código también utiliza la palabra clave as para asignar una variable llamada file; con la función with open(import\_file, "r") as file:



Apliqué el método .read() a la variable de file identificada en la declaración with . Luego, asigné la salida de cadena de este método a la variable text.

Utilizando la función print() y asignándole la variable text podemos imprimir en pantalla el contenido del archivo import\_file revisado en formato de cadena.

## Resumen

Creé un algoritmo que elimina las direcciones IP identificadas en una variable remove\_list del archivo "allow\_list.txt" de direcciones IP aprobadas.

Este algoritmo implicaba abrir el archivo, convertirlo en una cadena para leer y luego convertir esta cadena en una lista almacenada en la variable ip\_addresses.

Luego, iteré a través de las direcciones IP en remove\_list. Con cada iteración, evalué si el elemento era parte de la lista ip\_addresses. Si lo era, le apliqué el método .remove() para eliminar el elemento de ip\_addresses.

Después de esto, usé el método .join() para convertir ip\_addresses nuevamente en una cadena para poder escribir sobre el contenido del archivo "allow\_list.txt" con la lista revisada de direcciones IP.

## 

## Algoritmo Final



## 

## 