# Algorithm for file updates in Python

#### **Project description** In this project, you are tasked with updating a file that contains a list of IP addresses authorized to access restricted content in a healthcare company. The goal is to remove any IP addresses from the authorization list that are specified in a removal list. The algorithm reads the authorization list from a file, processes it to exclude the specified IP addresses, and writes the updated list back to the file.

#### **Open the file that contains the allow list**

python

import\_file = "allow\_list.txt"

with open(import\_file, 'r') as file:

# File is now open and can be read

##### *Explanation:* The with open(import\_file, 'r') as file statement opens the file in read mode ('r') and ensures that the file is properly closed after its suite finishes, even if an exception is raised. The file variable is used to access the opened file.

#### **Read the file contents**

python

ip\_addresses = file.read()

##### *Explanation:* The .read() method reads the entire content of the file into a string, which is stored in the ip\_addresses variable. This method allows you to process the entire file content at once.

#### **Convert the string into a list**

python

ip\_addresses = ip\_addresses.split('\n')

##### *Explanation:* The .split('\n') method splits the string at each newline character, converting it into a list of IP addresses. Each line in the file becomes an element in the list.

#### **Iterate through the remove list**

python

remove\_list = ["192.168.140.81", "192.168.109.50", "192.168.243.140"]

for element in remove\_list:

# Process each IP in the remove\_list

##### *Explanation:* The for element in remove\_list loop iterates through each IP address in the remove\_list, allowing you to check and remove each IP address from the authorization list.

#### **Remove IP addresses that are on the remove list**

python

if element in ip\_addresses:

ip\_addresses.remove(element)

##### *Explanation:* The if element in ip\_addresses condition checks if the current IP address from the remove\_list is present in the ip\_addresses list. If it is, the .remove(element) method removes it from the list.

#### **Update the file with the revised list of IP addresses**

python

with open(import\_file, 'w') as file:

file.write('\n'.join(ip\_addresses))

##### *Explanation:* The with open(import\_file, 'w') as file statement opens the file in write mode ('w'). The .write('\n'.join(ip\_addresses)) method writes the updated list of IP addresses back to the file, joining each IP address with a newline character to format the file correctly.

**Summary**This algorithm demonstrates how to manage and update a list of IP addresses stored in a file. It involves opening a file, reading and processing its contents, and then writing the updated information back to the file. Key operations include using with open() for file handling, .read() to retrieve file contents, .split() to convert strings into lists, and .remove() to eliminate specific entries. This approach ensures that only authorized IP addresses remain in the list, maintaining the security and integrity of restricted content access.

# Algoritmo para la actualización de archivos en Python

#### **Descripción del proyecto** En este proyecto, se te solicita actualizar un archivo que contiene una lista de direcciones IP autorizadas para acceder a contenido restringido en una empresa de atención médica. El objetivo es eliminar cualquier dirección IP de la lista de autorizaciones que esté especificada en una lista de eliminación. El algoritmo lee la lista de autorizaciones desde un archivo, procesa la lista para excluir las direcciones IP especificadas, y escribe la lista actualizada de nuevo en el archivo.

#### **Abrir el archivo que contiene la lista de autorizaciones**

python

import\_file = "allow\_list.txt"

with open(import\_file, 'r') as file:

# El archivo está abierto y se puede leer

##### *Explicación:* La sentencia with open(import\_file, 'r') as file abre el archivo en modo lectura ('r') y asegura que el archivo se cierre adecuadamente después de que se termine de trabajar con él, incluso si se produce una excepción. La variable file se usa para acceder al archivo abierto.

#### **Leer el contenido del archivo**

python

ip\_addresses = file.read()

##### *Explicación:* El método .read() lee todo el contenido del archivo en una cadena, que se almacena en la variable ip\_addresses. Este método permite procesar todo el contenido del archivo de una vez.

#### **Convertir la cadena en una lista**

python

ip\_addresses = ip\_addresses.split('\n')

##### *Explicación:* El método .split('\n') divide la cadena en cada carácter de nueva línea, convirtiéndola en una lista de direcciones IP. Cada línea en el archivo se convierte en un elemento de la lista.

#### **Iterar a través de la lista de eliminación**

python

remove\_list = ["192.168.140.81", "192.168.109.50", "192.168.243.140"]

for element in remove\_list:

# Procesar cada IP en la remove\_list

##### *Explicación:* El bucle for element in remove\_list itera a través de cada dirección IP en la remove\_list, permitiéndote verificar y eliminar cada IP de la lista de autorizaciones.

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

python

if element in ip\_addresses:

ip\_addresses.remove(element)

##### *Explicación:* La condición if element in ip\_addresses verifica si la dirección IP actual de la remove\_list está presente en la lista ip\_addresses. Si lo está, el método .remove(element) la elimina de la lista.

#### **Actualizar el archivo con la lista revisada de direcciones IP**

python

with open(import\_file, 'w') as file:

file.write('\n'.join(ip\_addresses))

##### *Explicación:* La sentencia with open(import\_file, 'w') as file abre el archivo en modo escritura ('w'). El método .write('\n'.join(ip\_addresses)) escribe la lista actualizada de direcciones IP de nuevo en el archivo, uniendo cada dirección IP con un carácter de nueva línea para formatear el archivo correctamente.

#### **Resumen**

Este algoritmo demuestra cómo gestionar y actualizar una lista de direcciones IP almacenadas en un archivo. Implica abrir un archivo, leer y procesar su contenido, y luego escribir la información actualizada de nuevo en el archivo. Las operaciones clave incluyen el uso de with open() para manejar archivos, .read() para recuperar el contenido del archivo, .split() para convertir cadenas en listas, y .remove() para eliminar entradas específicas. Este enfoque asegura que solo queden en la lista las direcciones IP autorizadas, manteniendo la seguridad e integridad del acceso al contenido restringido.