

Algorithm for file updates in Python

Project description

Este proyecto busca crear un control de acceso a cierta información creando un programa que verifique las ip que se conectan. Para ello pasa por fases de creación de los bucles para luego integrarlo todo en una clase, con lo que luego poder reutilizar cuando sea necesario en alguna parte del proyecto.

Open the file that contains the allow list

```
import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
```

En “import_file” se tiene el string o ruta donde encontrar el archivo, o ruta relativa.

Read the file contents

```
with open(import_file, "r") as file:
    ip_addresses = file.read()
```

En este trozo de código primero “with” ayuda a gestionar el abrir archivos y los cierra de manera automática, para no tener que controlar ese flujo. Luego con “open(file, operación)” busca el archivo con el primer parámetro y el segundo ejecuta si leerlo o escribirlo (crearlo). Para finalmente llamar a “.read()” para obtener la información dentro del archivo.

Convert the string into a list

```
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
```

Luego de llamar a “open()” y obtener la información dentro del archivo vendrá en un bloque de texto, “split()” se divide el texto en marcadores o espacios.

Iterate through the remove list

```
for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
```

Con el bucle for recorreremos los elementos en la lista ip , y luego recorreremos en la lista remove_list para comparar si está presente en esta lista.

Remove IP addresses that are on the remove list

```
for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
```

Con “.remove(element)” busca el elemento en la lista y cuando coincide lo elimina.

Update the file with the revised list of IP addresses

```
ip_addresses = " ".join(ip_addresses)
with open(import_file, "w") as file:
    file.write(ip_addresses)
```

Se abre el archivo en escritura y se agrega la lista actualizada para tener guardados los últimos cambios.

Summary

En esta práctica pudimos seguir el proceso de creación de una clase, con todos los apartados a tener en cuenta para controlar el acceso de los usuarios a determinados datos. Se utilizó lectura y escritura además de cortar(split) y quitar(remove), con interacciones con archivos y listas.