# **Algorithm for file updates in Python**

## Project description

A health care organization needs me to regularly update a file of employees who can access restricted access level content. The file includes employee IP addresses that currently have access to personal patient rcords, and I have been given a list of IP addresses to remove. In this project, I create a Python algorithm that checks whether the allow list contains any of the IP addresses to remove, and remove them.

## Open the file that contains the allow list

The file path to "allow\_list.txt" is stored for easy access.

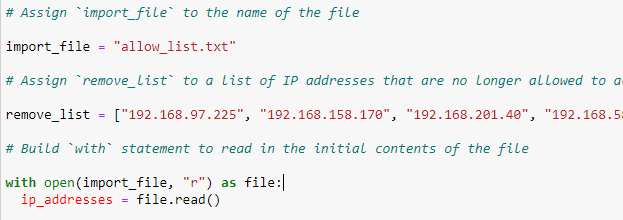


The "with open()" construct grants access to the file's contents.



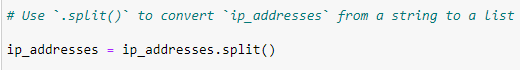
## Read the file contents

The .read() method transforms the file's contents into a Python string.



## Convert the string into a list

To facilitate IP address manipulation, the string is converted into a list using .split().



## Iterate through the remove list

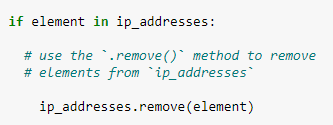
With the available IP addresses in list, the code iterates through the IP addresses designated for removal.





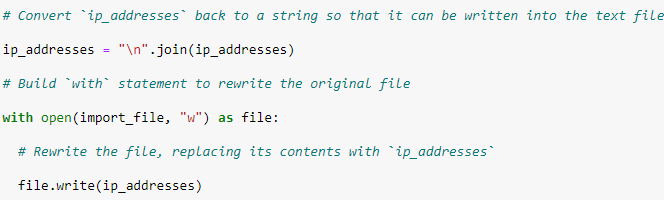
## Remove IP addresses that are on the remove list

Iterating IP addresses to be removed, a conditional check was created to identify and remove any IP addresses present on both the allow list and the removal list.



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

The modified list of allowed IP addresses is transformed back into a string, with each IP address occupying a separate line. The "allow\_list.txt" file is opened in write mode ("w") to overwrite its contents, preventing duplicates. The updated string of allowed IP addresses is written to the file.



## Summary

This Python algorithm effectively refines an allow list of IP addresses using a separate list of IPs to be removed. It achieves this through strategic file handling, list manipulation, and conditional logic. Key Python tools employed include "with open()", .read(), .remove(), .split(), and string formatting.