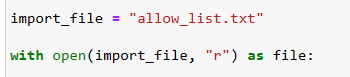
# Algorithm for file updates in Python

## Project description

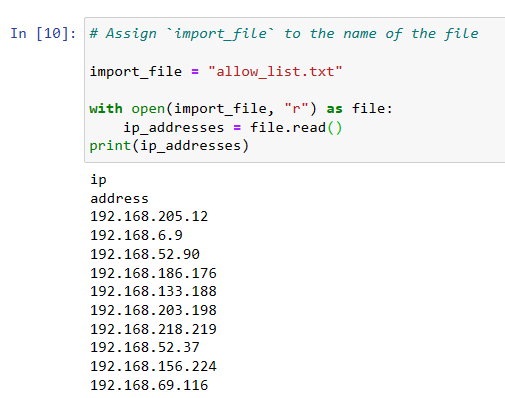
The project consists of helping a health care company update a *"allow\_list.txt"* file that identifies which employees have access to restricted content, based on their IP address. We will do this by creating an algorithm that uses Python code to check whether the allowed list contains any IP addresses that are flagged by the remove list, and if that is the case, the algorithm should remove it from the allowed IP address list.

## Open the file that contains the allow list

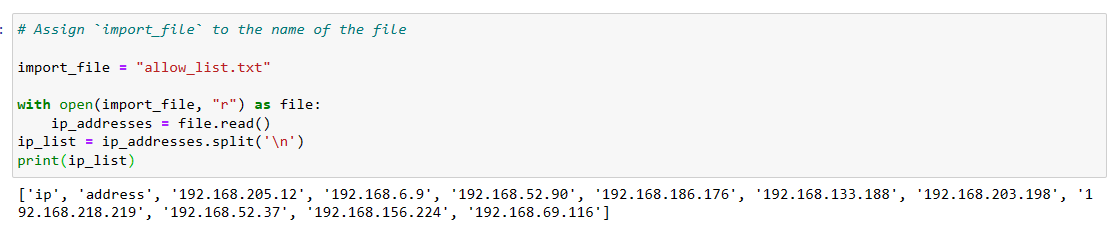


Using the *open()* command to open the file in read mode, in order to just get the content without actually modifying the file, and wrapping it on a with statement to handle errors and manage external resources later on.

## Read the file contents This code converts the content of the *"allow\_list.txt"* file into a string format. This string can be used in my Python program to organize and extract data later on.

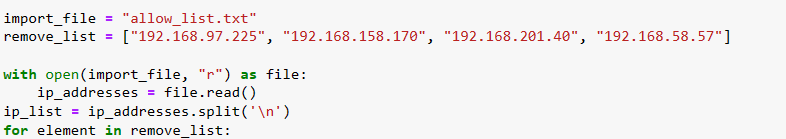


## Convert the string into a list To convert a string into a list, the code appends the *.split()* function to a string variable. This function is used to separate the contents of the string into individual elements in the list. The reason for splitting the ip\_addresses variable into a list is to facilitate the removal of IP addresses from the allow list. By default, the *.split()* function divides the text based on whitespace and creates list elements accordingly. In this algorithm, the *.split()* function is applied to the data stored in the ip\_addresses variable, which contains a string of IP addresses separated by whitespace. This process transforms the string into a list of IP addresses, which is then reassigned to the ip\_addresses variable for storage.



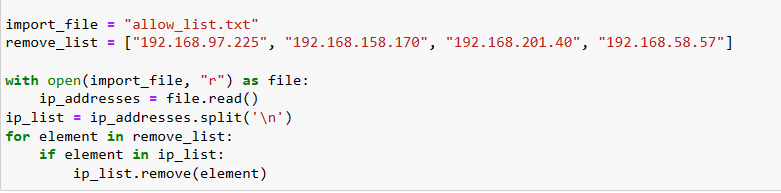
## Iterate through the remove list

The Python for loop repeats code for a given sequence. Its purpose in this algorithm is to execute specific code statements for each element in a sequence. The loop starts with the "for" keyword, followed by the loop variable "element" and the "in" keyword. "In" instructs the loop to iterate through the "ip\_addresses" sequence and assign each value to the loop variable "element".

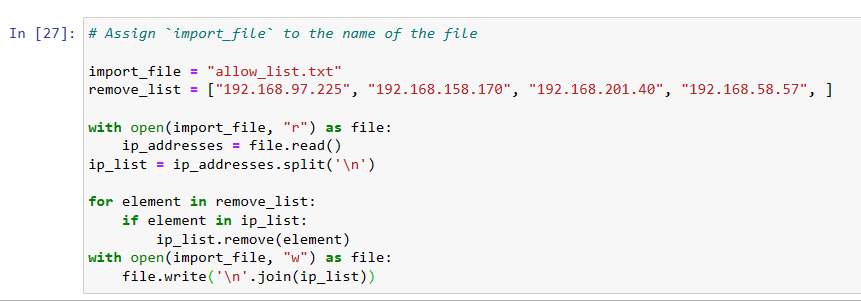


## Remove IP addresses that are on the remove list

## In the for loop, I checked if the loop variable "element" existed in the "ip\_addresses" list to avoid errors. If it was found, I used the *.remove()* function on "ip\_addresses" with "element" as the argument. This ensured that each IP address in the "remove\_list" was successfully removed from "ip\_addresses".



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



## Summary

I made a program that takes out IP addresses listed in a *"remove\_list"* from a file called *"allow\_list.txt"* which contains approved IP addresses. To do this, I first opened the file and turned its content into a readable string. Then, I converted that string into a list and stored it in a variable called "ip\_addresses". Next, I went through each IP address in the *"remove\_list"*. For each address, I checked if it existed in the ip\_addresses list. If it did, I used the *.remove()* function to eliminate it from the ip\_addresses list. After that, I used the *.join()* function to convert the ip\_addresses list back into a string. Finally, I replaced the contents of the *"allow\_list.txt"* file with the updated list of IP addresses.