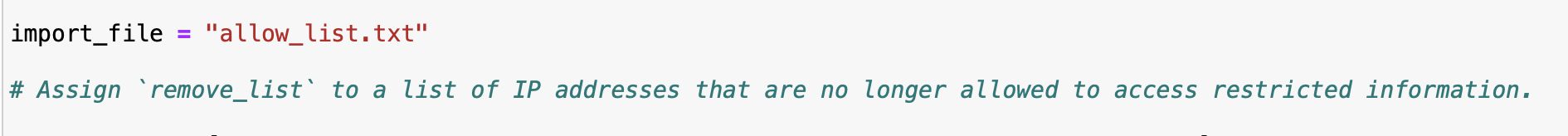
# Algorithm for file updates in Python

## Project description

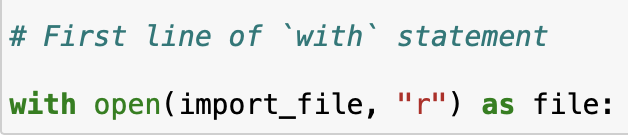
At my organization, specific restricted content is only accessible by IP addresses included on an allow list. The “allow\_list.txt” file contains a list of the approved IP addresses. A remove list was created to list IP addresses that no longer have access to the restricted content and need to be removed. I was tasked with creating an algorithm to automate updates to the “allow\_list.txt” file and remove the IP addresses that do not have access.

## Open the file that contains the allow list

To begin, I opened the text file “allow\_list.txt” using the import\_file variable. This was done by assigning the “allow\_list.txt” file to the import\_file variable:



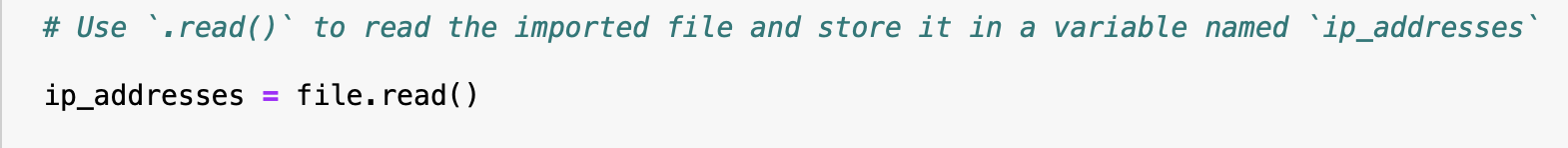
Then, the with keyword and .open( ) function were used to open the file:



The with statement is used with the .open( ) function to open files and read them. I utilized the with statement and .open( ) function to open the allow list and read it. The .open( ) function takes two parameters. The first parameter is the file that will be imported and the second parameter specifies what I want to do with the file. The function with open(import\_file, “r”) as file: uses “r” as the second parameter, which specifies I want to read the file.

## Read the file contents

I used the .read( ) function to read the imported file’s contents. I converted the imported file into a string and stored it in the variable ip\_addresses:



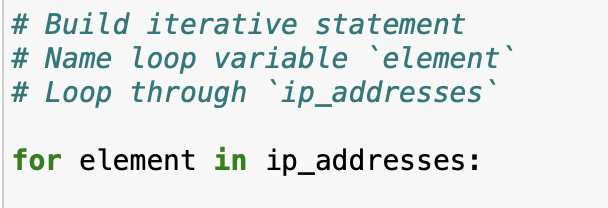
## Convert the string into a list

Then I used the .split( ) method to convert the ip\_addresses variable from a string into a list. This allowed me to remove the IP addresses from the allow list individually:



## Iterate through the remove list

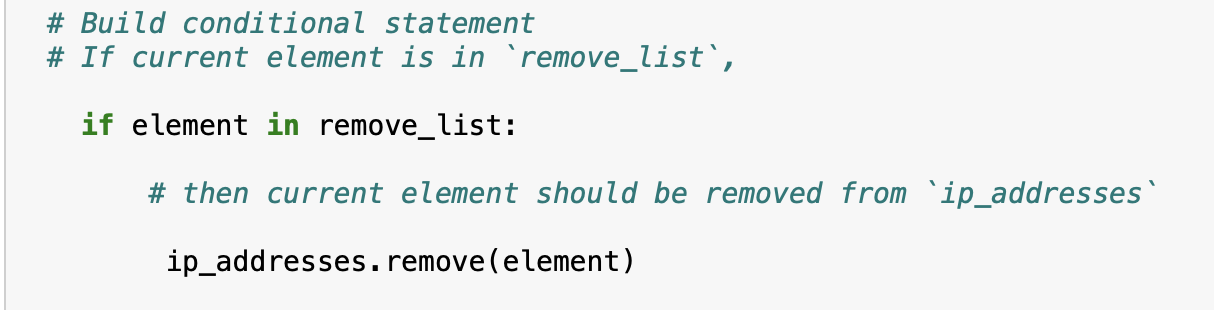
In order to remove the elements of the remove\_list from the ip\_addresses, I created a for loop:



In order to remove IP addresses, an iterative statement was needed to repeat code for a specified sequence. In this case, I used a for loop to loop through the ip\_addresses list. The for word initiates the loop and element is the loop variable. The in operator appears before the sequence and specifies to run the loop for every item in the sequence. The list ip\_addresses is the sequence and as the loop runs, each element from ip\_addresses will be stored temporarily in the element variable.

## Remove IP addresses that are on the remove list

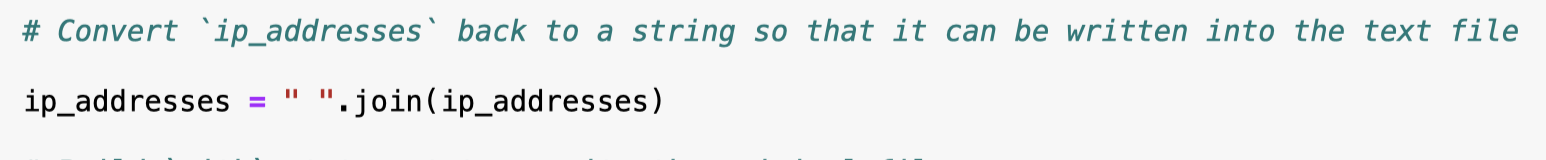
To determine if an element in ip\_addresses list is also in remove\_list, I created a conditional statement:



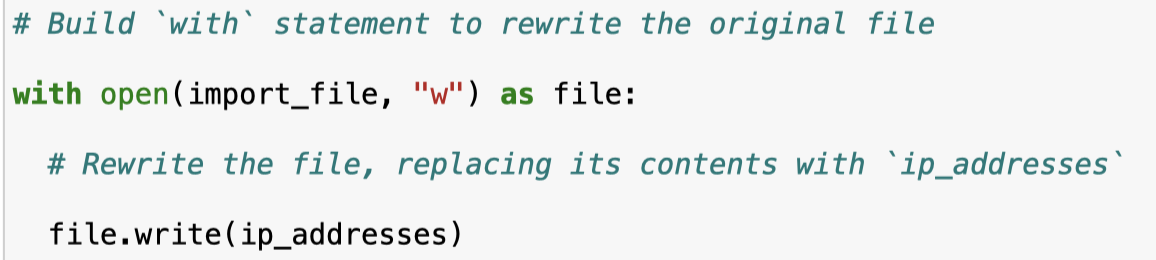
The conditional statement uses the “if” keyword, which executes a block of code only when the conditions are met. Furthermore, if the “element” was found in the remove\_list then it would be removed from ip\_addresses using the .remove( ) operator. This would result in the IP addresses that are on both the remove\_list and ip\_addresses being removed.

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

To complete my algorithm, the allow file needed to be updated with the revised ip\_addresses file. The .join( ) method was used to update the file:



In order to update the file, ip\_addresses must be in string format. Therefore, I used the .join( ) method which takes an iterable such as a list and concatenates every element of it into a string. The .join( ) method is applied to a string consisting of the characters that will be used to separate every element in the iterable once it's converted into a string. The method is applied to the string, “ “, which contains just a space character. The argument of the .join( ) method is ip\_addresses. The method converted the list ip\_addresses into a string, so that it could be written into the text file to be used with the with statement.



The with statement was used with .open( ) to open the text file and rewrite the original file using the “w” parameter. The parameter “w” indicates the old file contents will be deleted and replaced with the new file content. The .write( ) method will be used in conjunction with the with statement to rewrite the file. The .write( ) method takes ip\_addresses as its argument and file as the object. Considering this, I was able to update the allow list and remove IP addresses that do not have access to the restricted content.

## Summary

The algorithm I created removed IP addresses that did not have access to restricted content from the file “allow\_list.txt”. The algorithm required me to open the file, read the file contents, and convert it from string format into a list stored into the variable ip\_addresses. Then, I looped through ip\_addresses in remove\_list and displayed each element. Each iteration was considered to determine if the element should be removed from ip\_addresses using the .remove( ) method. To complete the algorithm, the file was updated with the revised file by converting the ip\_addresses list back into a string using the .join( ) method. This enabled the contents of the “allow\_list.txt” file to be written over with the updated IP address list.