# Python Algorithm: Update file

## Project Description

In my organization, access to restricted content is controlled by a list of allowed IP addresses. The "allow\_list.txt" file identifies these IP addresses.

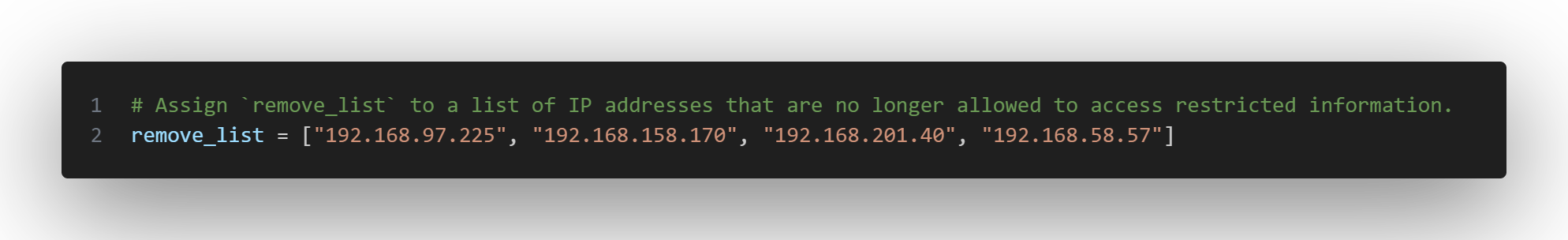
A separate allow list identifies IP addresses that should no longer have access to this content. I created an algorithm to automate updating the "allow\_list.txt" file and remove these IP addresses that should no longer have access.

## Open the file containing the allow list

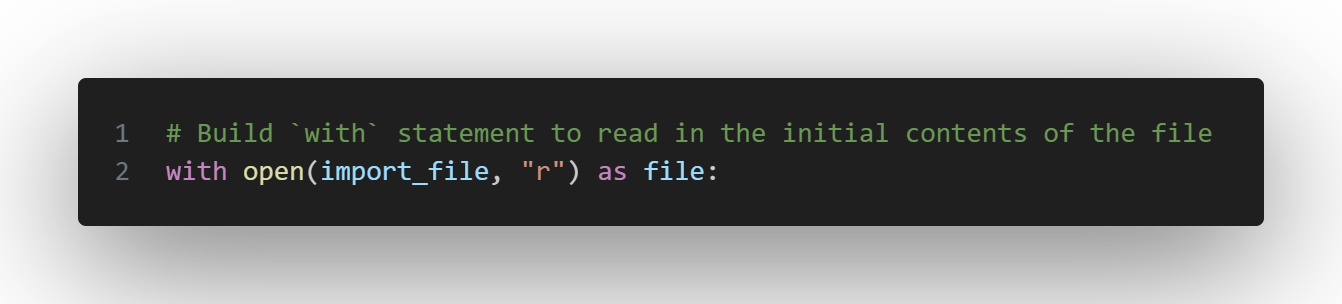
For the first part of the algorithm, I opened the file "allow\_list.txt" . First, I assigned this file name as a string to the variable import\_file :



Then assign the “remove\_list” file to a list of IP addresses that are no longer allowed to access restricted information:



Then I used a with statement to open the file:



In my algorithm, the with statement is used with the .open() function in read mode to open the allowlist file for reading purposes. The purpose of opening the file is to allow me to access the IP addresses stored in the allowlist file. The with keyword will help in managing the resources by closing the file after exiting the with statement .

In the code with open(import\_file, "r") as file: , the open() function has two parameters. The first identifies the file to import and then the second indicates what I want to do with the file. In this case, "r" indicates that I want to read it. The code also uses the as keyword to assign a variable called file ; file stores the output of the .open() function as I work inside the with statement .

## Read the contents of the file

To read the contents of the file, I used the .read() method to convert it to a string:



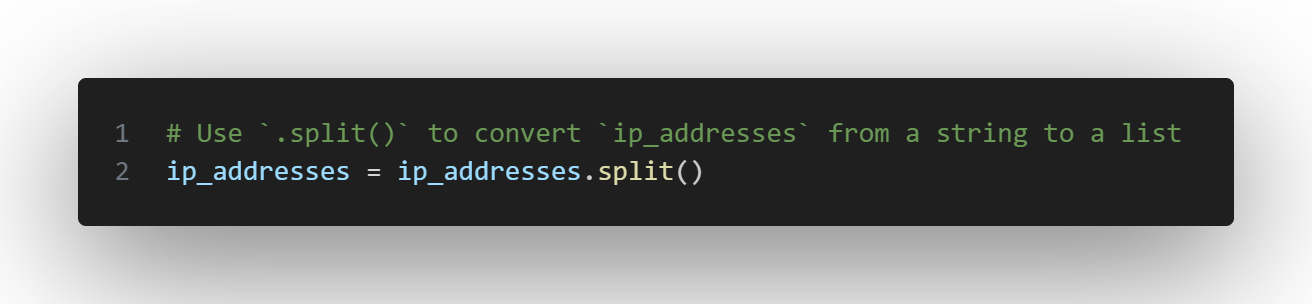
By using an .open() function that includes the "r" argument for "read", I can call the .read() function in the body of the with statement. The .read() method converts the file to a string and allows me to read it. I applied the .read() method to the file variable identified in the with statement .

Then, I assigned the string output of this method to the ip\_addresses variable .

In short, this code reads the contents of the "allow\_list.txt" file into a string format which allows me to then use the string to organize and extract data in my Python program.

## Convert string to list

To remove individual IP addresses from the allow list, I needed them to be in list format. So, next I used the .split() method to convert the string ip\_addresses into a list:

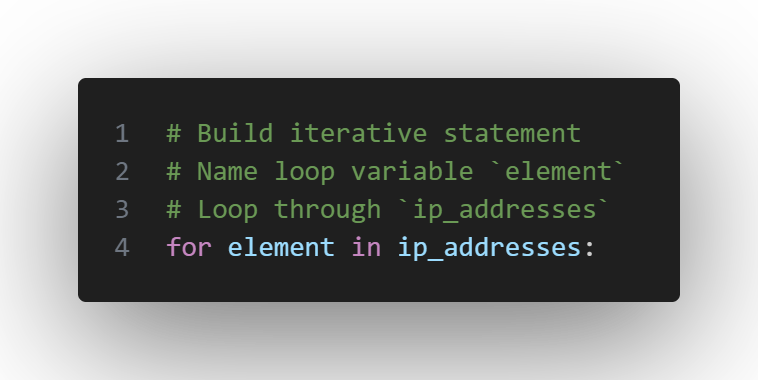


.split() function is called by adding a string variable to it. It works by converting the contents of a string into a list. The purpose of splitting ip\_addresses into a list is to make it easier to remove IP addresses from the allow list. By default, the .split() function splits text by whitespace into list items.

.split() function takes the data stored in the variable ip\_addresses , which is a string of IP addresses separated by whitespace, and converts this string into a list of IP addresses. To store this list, I've again reassigned it to the variable ip\_addresses .

## Iterate through the delete list

A key part of my algorithm involves iterating through the IP addresses that are elements in the remove\_list . To do this, I incorporated a for loop :



The for loop in Python repeats the code for a specific sequence. The general purpose of the for loop in a Python algorithm like this is to apply specific code statements to all elements of a sequence.

for keyword starts the for loop. It is followed by the element loop variable and the in keyword . The in keyword instructs to iterate through the ip\_addresses sequence and assign each value to the element loop variable .

## Delete IP addresses that are on the delete list

My algorithm requires removing any IP addresses from the allow list, ip\_addresses , that are also contained in remove\_list . Because there were no duplicates in ip\_addresses , I was able to use the following code to do this:



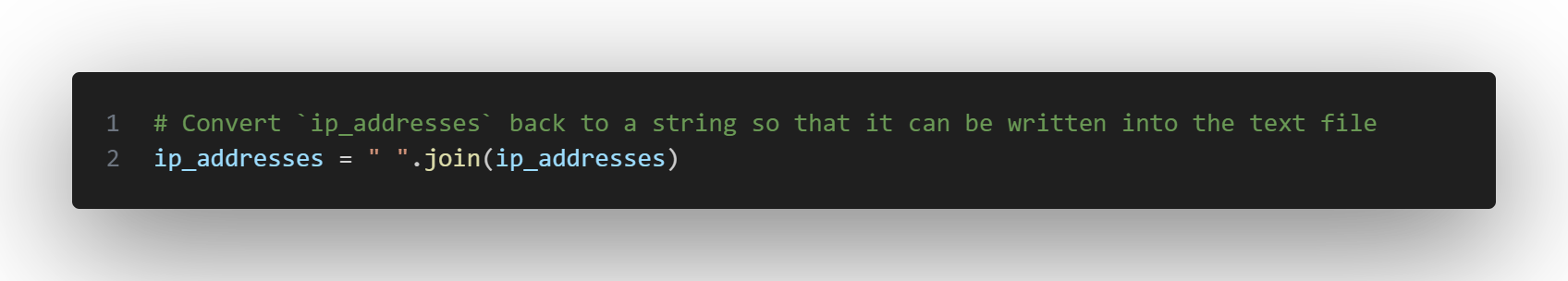
First, inside my for loop, I created a condition that evaluated whether the element loop variable was found in the ip\_addresses list . I did this because applying .remove() to elements that were not found in ip\_addresses could result in an error.

So inside that conditional, I applied .remove() to ip\_addresses . I passed the loop variable element as an argument so that every IP address that was in the remove\_list would be removed from

ip\_addresses .

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

As the next step of my algorithm, I needed to update the allowlist file with the revised list of IP addresses. To do this, I first had to convert the list back into a string. Use the .join() method for this:

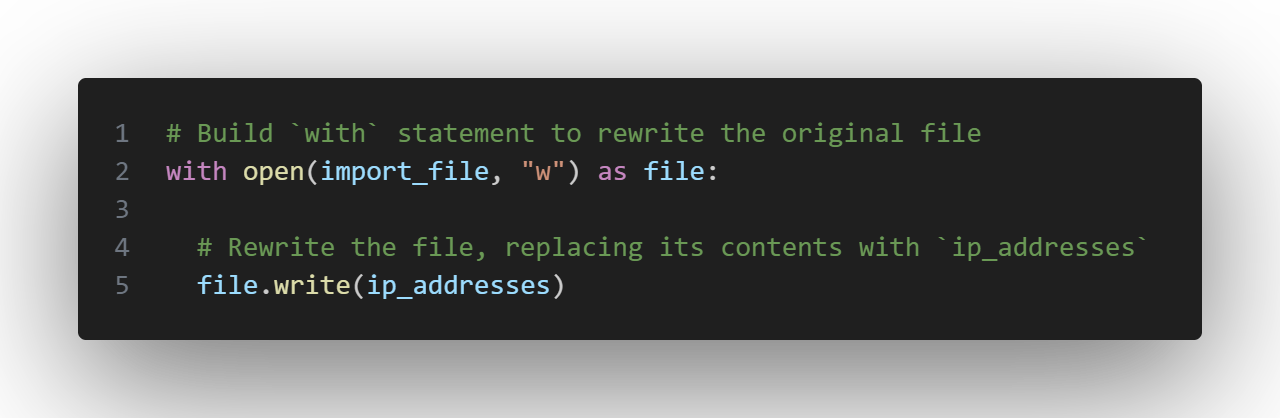


.join() method combines all the elements of an iterable into a string. The .join() method is applied to a string containing characters that will separate the elements of the iterable once they are joined into a string.

In this algorithm, use the .join() method to create a string from the ip\_addresses list so that it can be passed as an argument to the .write() method when writing to the "allow\_list.txt" file .

Use the string (" ") (empty character) as a separator to tell Python to put each item on a new line.

Then I used another with statement and the .write() method to update the file:



This time, I used a second argument of "w" with the open() function in my with statement . This argument indicates that I want to open a file to write over its contents. By using this "w" argument , I can call the .write() function in the body of the with statement . The .write() function writes string data to a specified file and replaces any existing file contents.

In this case, I wanted to write the updated allow list as a string to the "allow\_list.txt" file . This way, the restricted content will no longer be accessible to any IP addresses that have been removed from the allow list.

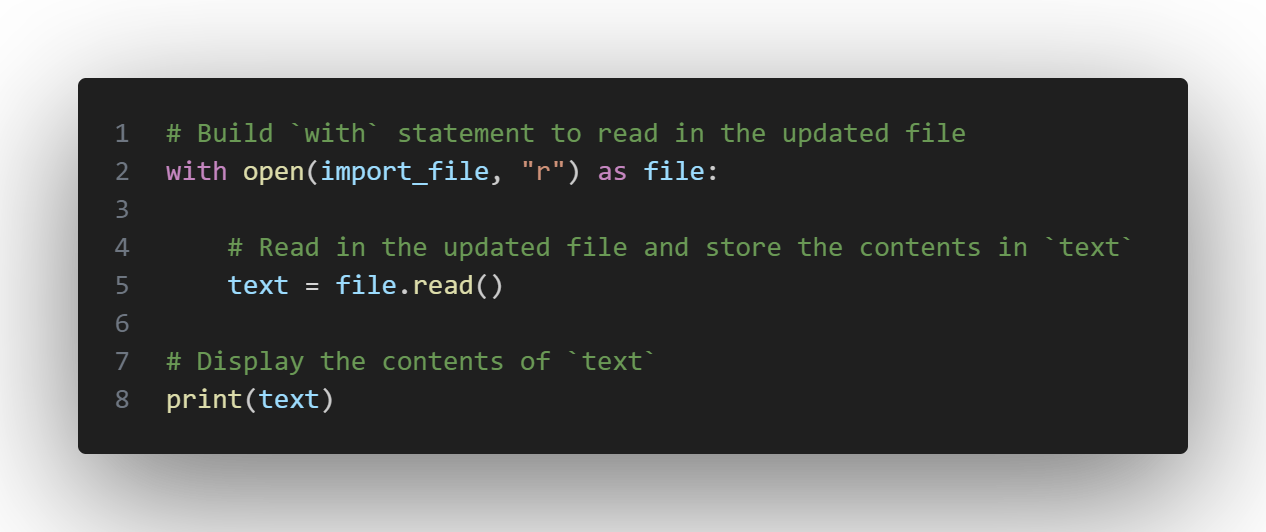
To rewrite the file, I added the .write() function to the file object file I identified in the with statement. I passed the ip\_addresses variable as an argument to specify that the contents of the file specified in the with statement should be replaced with the data in this variable.

## 

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

As a final step in my algorithm, I needed to read the updated allowlist file with the revised list of IP addresses.

with statement again , the open() function , the "r" argument , the code also uses the as keyword to assign a variable called file ; with the function with open(import\_file, "r") as file:



.read() method to the file variable identified in the with statement . Then, I assigned the string output of this method to the text variable .

print() function and assigning the text variable we can print the contents of the revised import\_file file on the screen in string format.

## Summary

I created an algorithm that removes IP addresses identified in a remove\_list variable from the "allow\_list.txt" file of approved IP addresses.

This algorithm involved opening the file, converting it into a string to read, and then converting this string into a list stored in the variable ip\_addresses .

Next, I iterated through the IP addresses in remove\_list . With each iteration, I evaluated whether the item was part of the ip\_addresses list . If it was, I applied the .remove() method to it to remove the item from ip\_addresses .

After this, I used the .join() method to convert ip\_addresses back into a string so I could overwrite the contents of the "allow\_list.txt" file with the revised list of IP addresses.

## 

## Final Algorithm



## 

## 