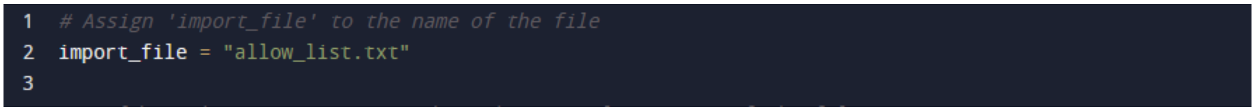
Algorithm for file updates in Python

Project description

This project is an algorithm to automate updating “allow_list.txt” by removing IP addresses that shouldn’t have access.

Open the file that contains the allow list

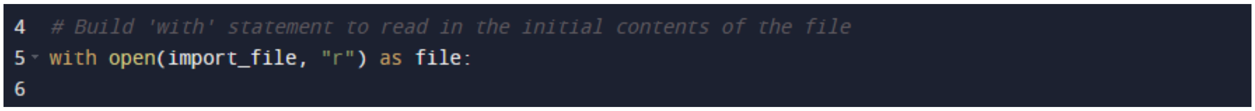
First we will open “allow_list.txt” and assign it to a variable import_file



```
# Assign 'import_file' to the name of the file
import_file = "allow_list.txt"

```

Then use with and open() to read the contents. The contents will be stored as the variable file.

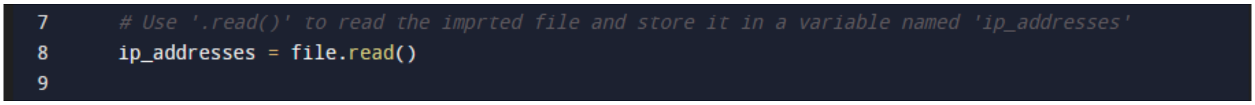


```
# Build 'with' statement to read in the initial contents of the file
with open(import_file, "r") as file:

```

Read the file contents

Using .read() we can convert the contents of the file to string and print the contents of it.

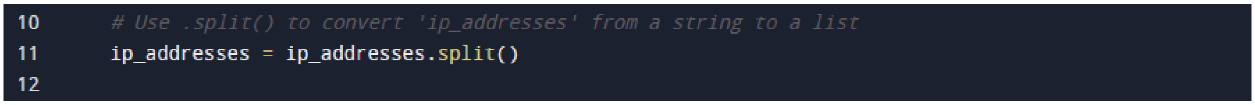


```
    # Use '.read()' to read the imprted file and store it in a variable named 'ip_addresses'
    ip_addresses = file.read()

```

Convert the string into a list

Using the .split() method we can convert the string into a list for easier removal of IP addresses

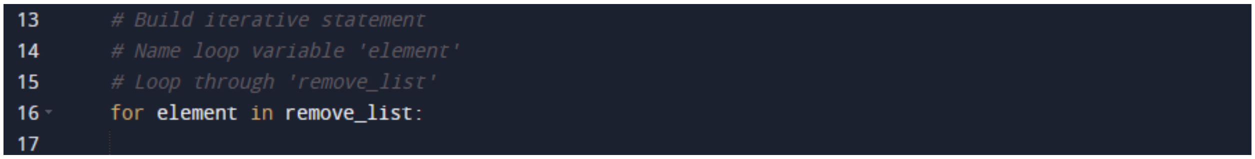


```
    # Use .split() to convert 'ip_addresses' from a string to a list
    ip_addresses = ip_addresses.split()

```

Iterate through the remove list

The algorithm will iterate through the IP addresses that are in the remove_list using a for loop.

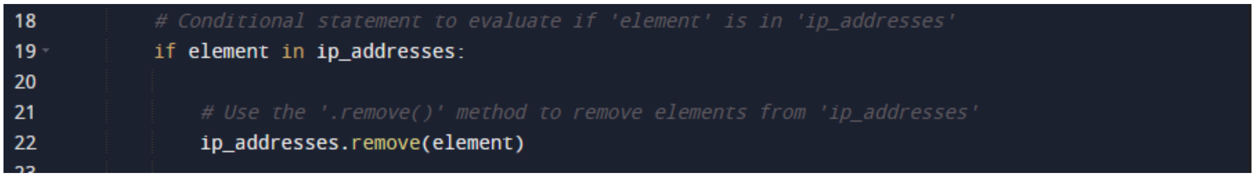


```
    # Build iterative statement
    # Name loop variable 'element'
    # Loop through 'remove_list'
    for element in remove_list:

```

Remove IP addresses that are on the remove list

Inside of the for loop we will compare if the element is in the list of IP addresses and if it is, remove it from the list.

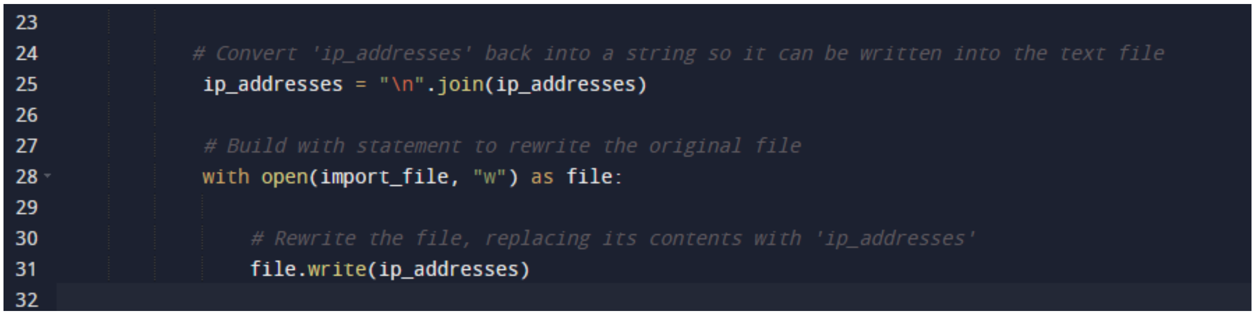


```
        # Conditional statement to evaluate if 'element' is in 'ip_addresses'
        if element in ip_addresses:

            # Use the '.remove()' method to remove elements from 'ip_addresses'
            ip_addresses.remove(element)
```

Update the file with the revised list of IP addresses

Lastly, the algorithm will use .join() to combine the remaining IP addresses back into a string and using .write() to update the contents of “allow_list.txt”, adding “\n” for more readability.



```

            # Convert 'ip_addresses' back into a string so it can be written into the text file
            ip_addresses = "\n".join(ip_addresses)

            # Build with statement to rewrite the original file
            with open(import_file, "w") as file:

                # Rewrite the file, replacing its contents with 'ip_addresses'
                file.write(ip_addresses)

```

Summary

The algorithm I made will remove IP addresses specified in the remove_list from the “allow_list.txt”. It does so by opening the file, converting it to a string to be read, then as a list to be stored in ip_addresses. Then it will iterate through the IP addresses in remove_list, comparing it to the values in ip_addresses, and removing it if it matches. Once this process is completed, it will convert ip_addresses back into a string to rewrite “allow_list.txt” with the updated information.