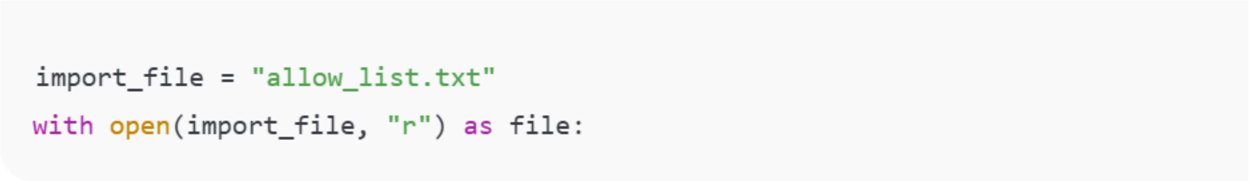
Algorithm for file updates in Python

Project description

In this project, I created a Python algorithm that updates an allow list of employee IP addresses at a healthcare company. The allow list controls access to a restricted network that stores sensitive patient data. A separate remove list contains the IPs of employees who should no longer have access.

My task was to identify and remove any matching IPs in both lists and update the file accordingly. This helps maintain secure access control to private data and supports compliance with security policy.

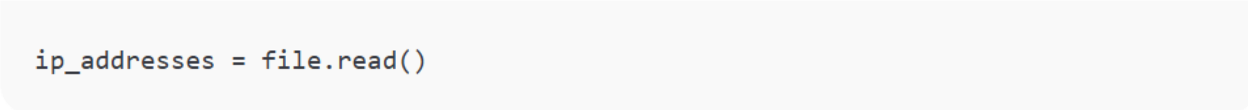
Open the file that contains the allow list



```
import_file = "allow_list.txt"
with open(import_file, "r") as file:
```

The `open()` function opens the file named `allow_list.txt` in read mode (`"r"`). The `with` statement ensures the file is properly closed afterward. The `file` variable holds the reference to the open file object.

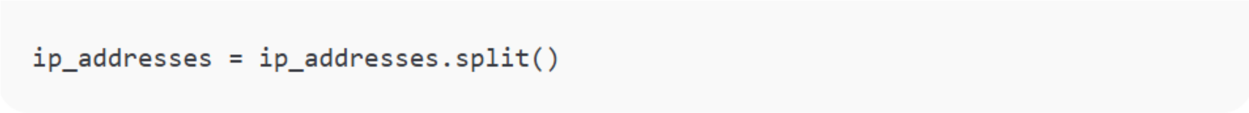
Read the file contents



```
ip_addresses = file.read()
```

The `.read()` method reads the entire contents of the file and stores it as a string in the `ip_addresses` variable.

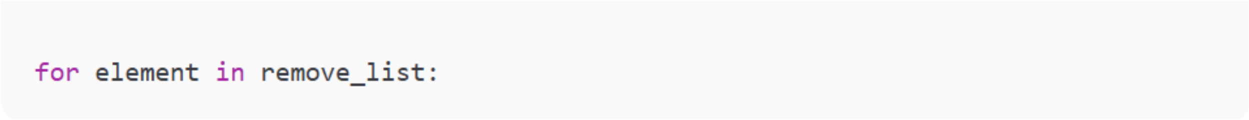
Convert the string into a list



```
ip_addresses = ip_addresses.split()
```

The `.split()` method converts the string into a list by separating the text wherever it finds whitespace (by default), allowing us to handle each IP address individually

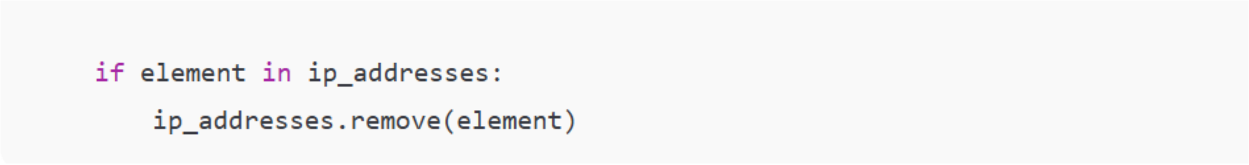
Iterate through the remove list



```
for element in remove_list:
```

This `for` loop goes through each item (`element`) in the `remove_list`. The loop variable `element` represents one IP address per iteration.

Remove IP addresses that are on the remove list

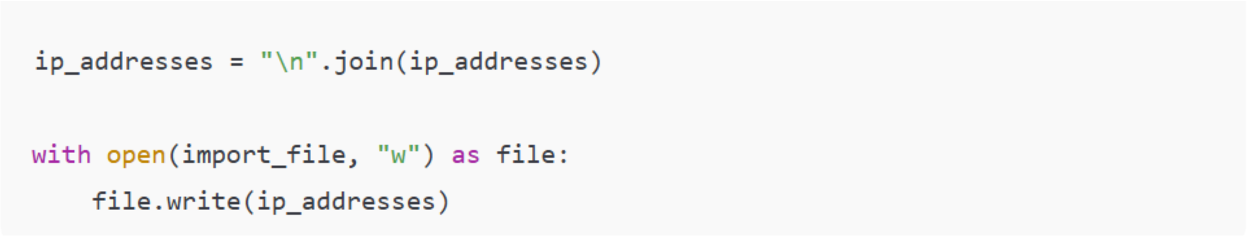


```
    if element in ip_addresses:
        ip_addresses.remove(element)
```

This conditional checks if the current IP (`element`) exists in the `ip_addresses` list. If so, it removes it using `.remove()`.

This method works correctly here because the IP addresses in the allow list do not contain duplicates.

Update the file with the revised list of IP addresses



```
ip_addresses = "\n".join(ip_addresses)

with open(import_file, "w") as file:
    file.write(ip_addresses)
```

The `.join()` method converts the list back into a string, separating each IP address with a newline (`\n`).

The `open()` function is now used with mode `"w"` to overwrite the file with the updated list using

Summary

The Python algorithm I created helps automate the secure maintenance of an allow list for IP-based access control. It opens a file containing allowed IP addresses and reads its contents. Then, it converts the data into a list and compares it to another list containing IPs that should be removed.

Using a `for` loop and conditional logic, the algorithm removes all matching entries. After cleaning the list, it converts it back to a string and overwrites the original file with the updated list. This ensures only authorized IPs remain and keeps access controls up to date in a secure, automated way.