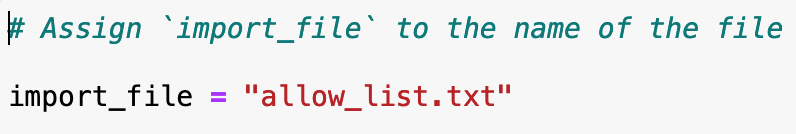
# Algorithm for file updates in Python

## Project description

My company has tasked me with handling Identity and Access Management (IAM) security and regularly updating a file that identifies the employees who can access restricted personal patient records. Access is restricted by IP addresses on an “allow list”, however there is a “remove list” for any IP addresses that should no longer have authorization to the file. I have created an algorithm to automate this process of updating and removing IP addresses from this file.

## Open the file that contains the allow list

First I assign the file name to a variable.



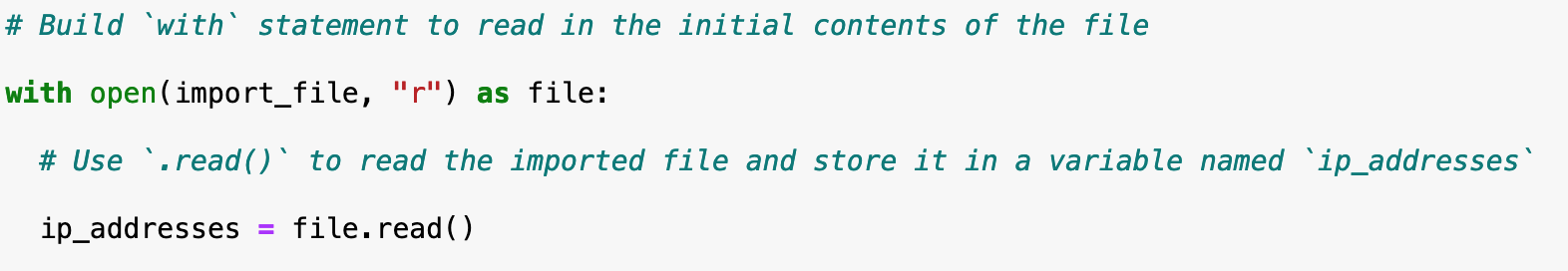
Next I open this file using a with statement to ensure the file only remains open while inside this statement for my use.



The parameters of the open() function indicates the file to import as well as what I want to do with this file, in this case the “r” allows me to read the file.

## Read the file contents

To read the file contents I use the read() method on the variable file created by the open function. This method reads the file contents and converts the output into a string which I have saved into a variable named ip\_addresses.



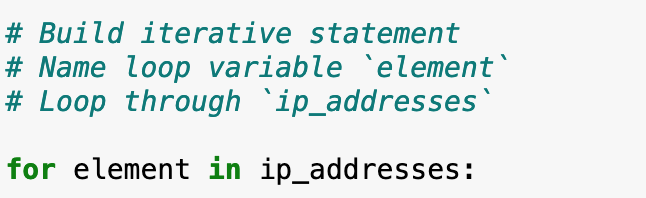
## Convert the string into a list

To make it easier to work with the file content I converted this string into a list using the split() method and saved it as the same variable of ip\_addresses.



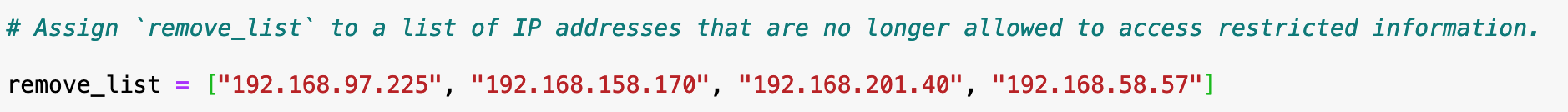
## Iterate through the remove list

I need to iterate through the list I have created to work with each item separately. To do this I use a for loop which iterates through each item in the ip\_addresses list and saves it to the element variable while in the for loop.

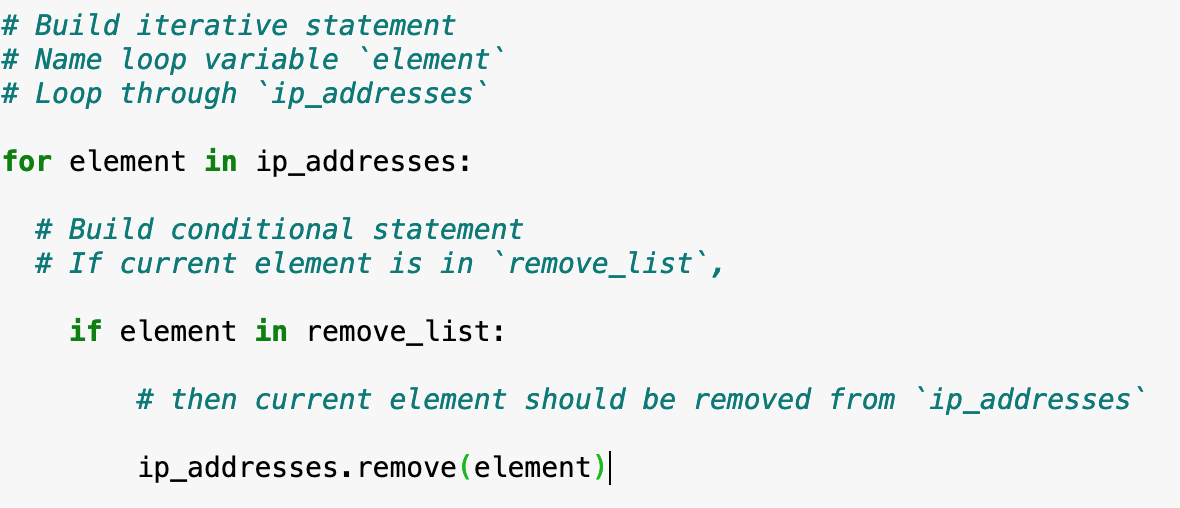


## Remove IP addresses that are on the remove list

Any IP addresses on the remove\_list then need to be removed from the ip\_addresses list which was created from the “allow list”

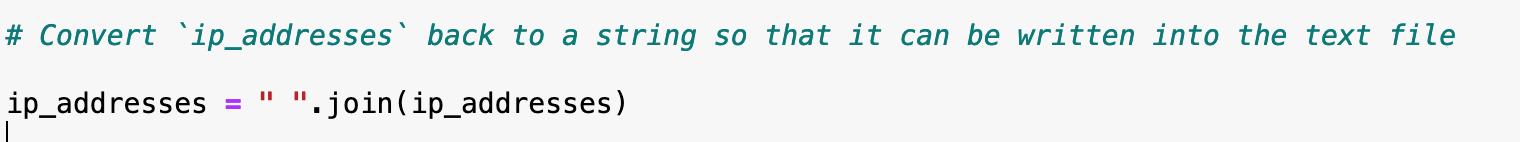


To do this I use the for loop to iterate through each element in the ip\_addresses list and using a conditional if statement check if any IP addresses are also in the remove\_list. If they do appear in the remove\_list they are taken out using the remove() method.

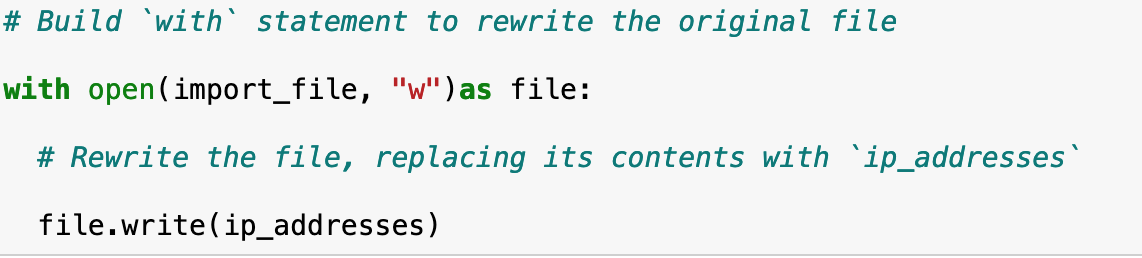


## Update the file with the revised list of IP addresses

Finally the “allow list” needs to be updated after being revised. The list first needs to be converted back to a string using the join() method. This method takes all items from an iterable and joins them into one string. I used “” to specify the items should be separated by a space.



Now with the revised ip\_addresses back as a string it can be written back into the original file replacing the content. Using a with statement I again open() the file this time using the “w” argument to write into the file as oppos ed to reading it.



Using the write() method with the ip\_addresses argument the revised content is written into the original file without the IP addresses that no longer have access.

## Summary

I was able to create an algorithm automating the process of removing IP addresses that no longer have access to a restricted file. I opened the “allow\_list” file and read the contents. Then converted that content from a string into a list so it could be iterated through in a for loop. Using the given remove\_list which contained the IP addresses that no longer have access to the file I created a conditional if statement to check for any IP addresses in the list that matched ones from the remove\_list. After removing these, the file was updated with the revised content by using the join() method to convert the list back to a string and the write() method to replace the contents of the “allow\_list” file.