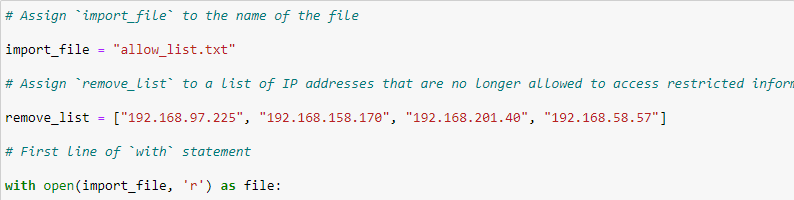
# Algorithm for file updates in Python

## Project description

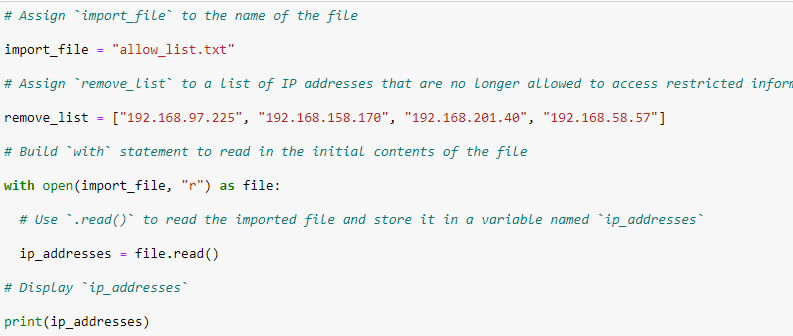
In this project we will create an algorithm that will allow us to update an ‘allow list’ of IP addresses by removing undesired IP’s using python. To accomplish this the file will be converted into a format that python can use. Then elements will be removed from the file and these changes will be written to the original file.

## Open the file that contains the allow list

Our first step will be to open the file using python. To do this we will need to first create variables for our allow list, “allow\_list.txt” and the IP’s we want to remove. The allow list we be called as import\_file and the denied as remove\_list. We will now use the import\_file variable to create the beginning of a with statement to read the allow list. Followed by with we type open() and pass the import\_file variable, as well as a second argument ‘r’(preceded by a ‘, ‘) which signifies we want to read the file, ending with the syntax as file:. This line imports the file, “allow\_list.txt”, to python.

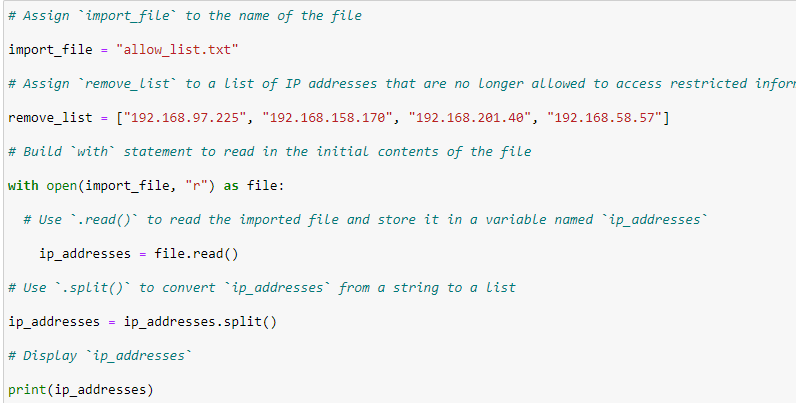
## Read the file contents

Next, we’ll use .read() to both tell python to convert the file into a string and create a new variable for it called, ip\_addresses. We’ll use this new variable for simplicity so we can distinguish import\_file from the edited file once we change it.



## Convert the string into a list

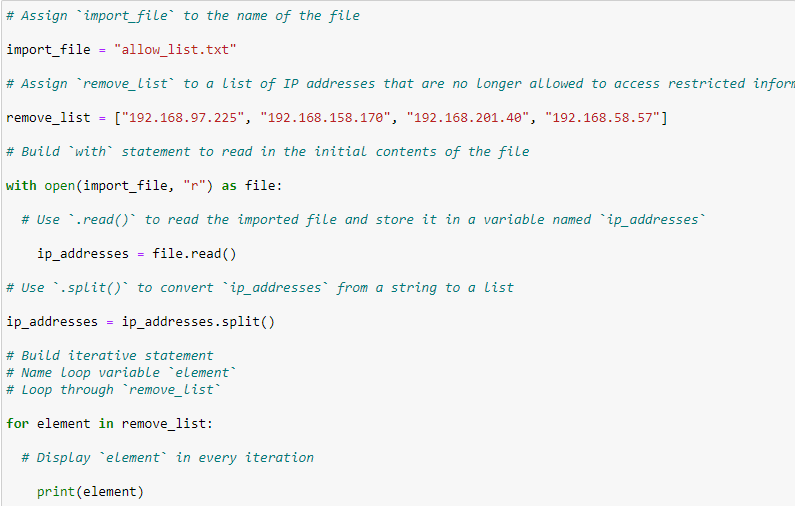
Now we’ll convert the string we created into a list using the .split() method. We convert into a list so we can remove elements from the list later.



## Iterate through the remove list

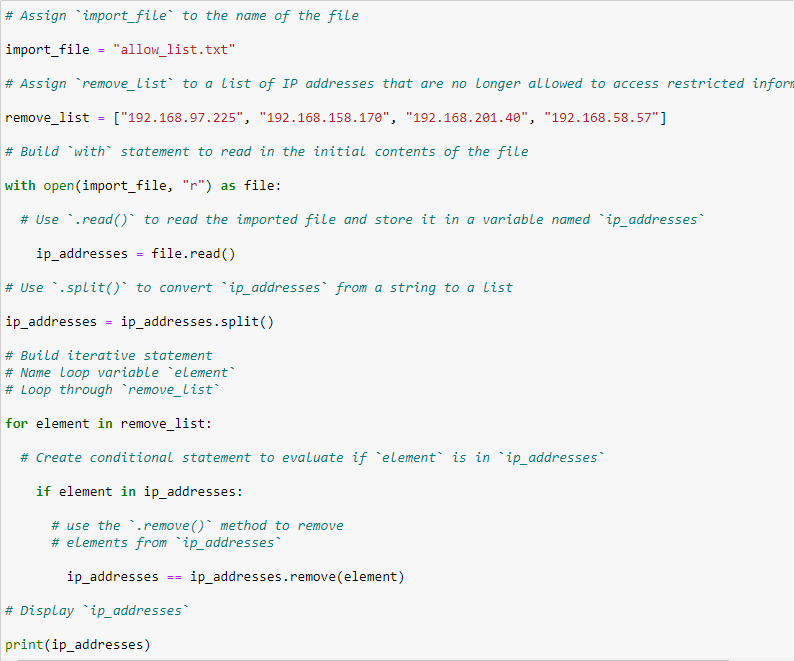
So, now that we have a list that we can work with, we want to find a way to remove unwanted IPs from the allow list. To do this we will create a for loop that will run through the addresses in the remove\_list. We’ll start with a header for the for loop containing the keyword for followed by element (for the loop variable

), then the in keyword ending with the remove\_list variable and ‘:’.

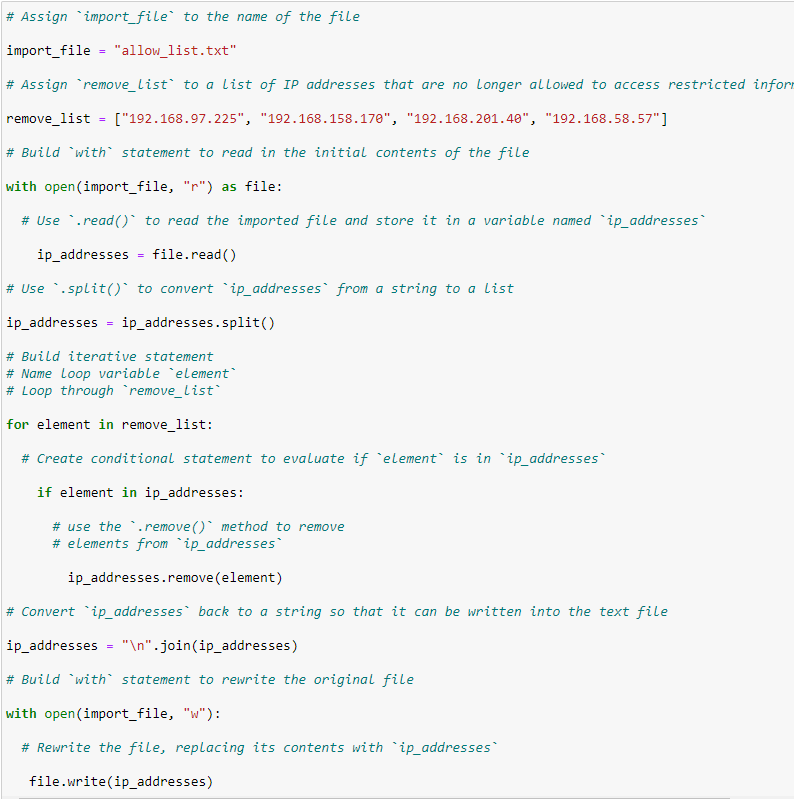


## Remove IP addresses that are on the remove list

Next, we’ll follow the for header with an if statement with the loop variable, element, and the variable ip\_addresses. This determines if any IPs in the loop variable are part of the variable ip\_addresses. After the if statement we’re going to remove any IPs from remove\_list from the allowed IPs using the .remove() method. To do this we will write ip\_addresses followed by .remove() with elements passed to it. Set this equal to ip\_addresses for conciseness. The .remove() method only works well in this case because .remove() only removes the first instance of an IP, from element, in ip\_addresses. If the variable contained multiples of any IPs, this would not be useful.



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

Finally, we’ll update the original file with our updated list of IPs. To do this we’ll need to convert the list back into a string using the .join() method. Again, set the syntax equal to the variable ip\_addresses with syntax as follows: "\n".join(ip\_addresses). With “\n” allowing each entry to populate a new line and ip\_addresses being passed to the .join() command. Next, we’ll write another with statement this time using the .write() method to write over the file. Pass the argments import \_file and, this time, “w” for write. The next line contains the syntax file.write() with ip\_addresses passed as the argment. Now your file allow\_list.txt has been updated with the correct IP addresses.

## Summary

So, to summarize we updated the file allow\_list.txt by opening the file in python and converting the resulting string into a list. From there a for loop was created to compare each element of the remove list to the allowed list of IPs. We then removed any IPs found in the remove list from the allow list. The list was then converted back into a string using the .join() method and written to the original file.