# Update a file through a Python algorithm

## Project description

In this project. I will show how python can be used to easily update a file with relevant information, such as updating a file containing allowed IP addresses in this case. I will build an algorithm to retrieve a file named “allow\_list.txt”, compare the content of the file to the content of a file containing the IP addresses to be removed, and then edit the content of the “allow\_list.txt” file accordingly.

## Open the file that contains the allow list

To begin, we will assign to a variable a string with the name of our file. In this case, we are assigning the string “allow\_list.txt” to the variable import\_file so that it can be used more easily later.

A close-up of a computer screen

Description automatically generated

To read the file, we start the header with the *with* and open() statements to tell python to open the file so it can be read. The arguments within the open() statement specific what file to open and that we want to read it. The output of this statement will be put into the variable *file* for further use.

A close-up of words

Description automatically generated

## Read the file contents

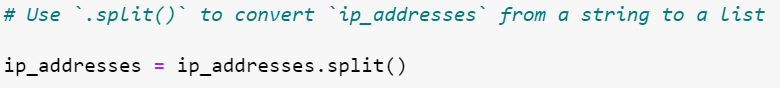
To read the file after opening, we will use the .read() method and convert the content of the file into a string that is then assigned to the variable ip\_addresses.

A close-up of a white background

Description automatically generated

## Convert the string into a list

With the content of our file now in a string format assigned to the variable ip\_addresses, we will want to change it from a string to a list to allow changes to be made more easily. This conversation can be done using the .split() method. .split() works by taking the data separated by whitespace and using that to create a single entry in the list. For our imported file, it will create a list where each entry is a separate IP address



## Iterate through the remove list

Next, we will create an iterative statement to go through each entry in our list of IP addresses to be removed. We can do this with a *for* loop. With the following code, it will use the variable *element* as each entry in remove\_list, which can be used in subsequent code. It will continue until it reaches the end of the remove\_list entries.

A close-up of a white background

Description automatically generated

## Remove IP addresses that are on the remove list

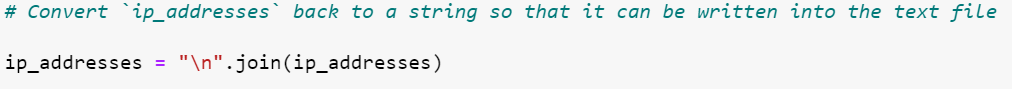
We now want to check each entry in remove\_list and if it exists in our current ip\_addresses, we want to remove it. Since each entry in remove\_list will be assigned to the variable *element* one at a time, we can use an if statement to check if *element* is within ip\_addresses. If this condition is true, it will perform the next steps where we will use the .remove() method. The .remove() method can be used to remove an entry from a list and the argument passed in it specifies which entry to remove, which in this case the entry stored within *element*.

A screenshot of a computer program

Description automatically generated

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

With the unwanted IP addresses removed, we have our new list stored in ip\_addresses but will need to write those changes back to the file. We’ll start this process by converting our ip\_addresses list back into a string by using the .join() method. Using the “\n” with .join() means that each entry in our list ip\_addresses will be entered on a separate line within our new string. We will assign this new string to our same ip\_addresses variable.



Now that our new string stored in ip\_addresses, we can write to our file by once again using the *with* and *open*() statements, this time using the “w” option to indicate that we want to write to the file. With our file open, we will use the .write() method to write into the file, passing in the ip\_addresses variable as an argument to represent the content we want to write into the file.

A close-up of a computer screen

Description automatically generated

We have now rewritten the content of our file “import\_file” so that the IP addresses within no longer include the IP addresses found within remove\_list.

## Summary

The goal of the algorithm shown here is to remove the unwanted IP addresses from our file named “import\_file.” To do this, we opened the file and read it into the variable ip\_addresses, which we then converted into a list. We then used an iterative statement to go through each entry in the list of unwanted IP addresses named “remove\_list.” If our iterative statement found a match between an entry in “remove\_list” and our list of IP addresses, it would remove that entry. Once completed, we rewrote the new list into our original “import\_file.”