# Algorithm for file updates in Python

## Project description

As a security analyst in a healthcare company, I am tasked with updating a document that identifies company employees that can access patients Personal Identifiable Information.

The employees must be able to connect through a predetermined list of IP addresses stored in an allow list. Any unauthorised IP addresses must be removed from the list.

My aim in this project is to create a Python algorithm that iterates through the allow list to check for unauthorised IP addresses and deletes them if found. Let’s get right to it!

NB: there is also a remove list which contains the IP addresses of employees who are no longer allowed access to patients’ data or any other unauthorised IP address detected.

## Open the file that contains the allow list

To open the file, I will use the “with” and “open” keywords. The with keyword closes the file right after it is read or written into. It also handles errors when used with other functions.

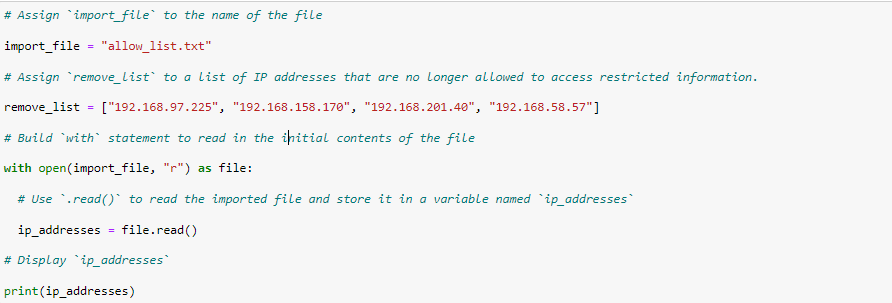
The open function opens files in Python. The open function takes two parameters – the file to be opened and the second indicates what is to be done to the file. “r” means the file will be read, “w” stands for write while “a” stands for append.

The following is the IDE screenshot



## Read the file contents

To read the contents of the import\_file variable, I will use the .read() method. In Python, the .read() method is used to convert the contents of a file into a string which can be stored in another variable.

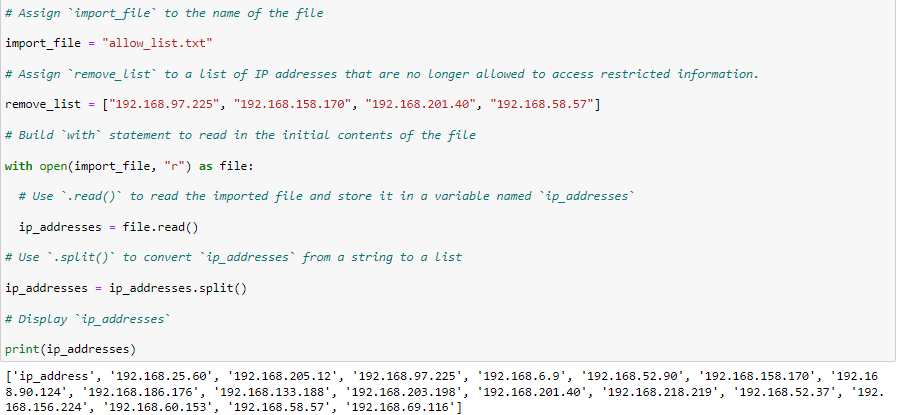


## Convert the string into a list

Converting the contents of ip\_addresses variable into a list is important in order to allow us to easily iterate through the list and look for unauthorised IP addresses, instead of sorting through a large string merged together.

To accomplish this, I will use the .split() method.

The IDE screenshot is provided below

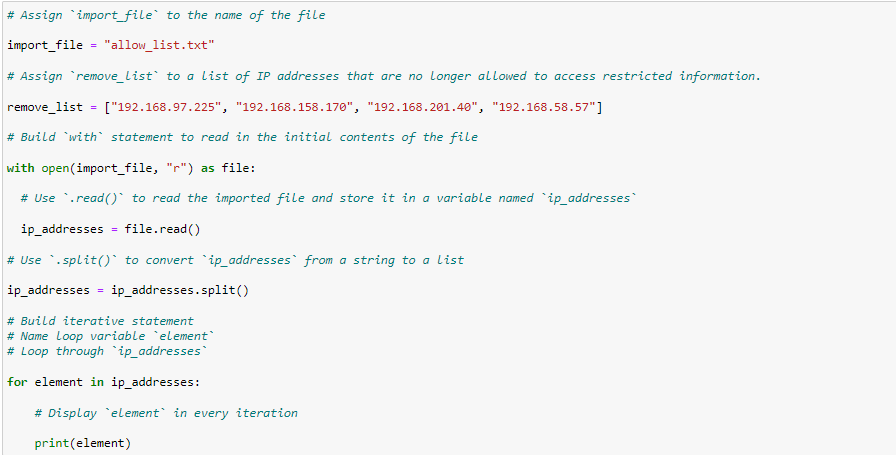


## Iterate through the remove list

In Python, iterating means going through the contents of a list one sequentially. To accomplish this, I will use the for loop. The for loop has a header which contains a loop variable.

In this case, I will use the loop variable element, to search through the remove\_list.

The IDE screenshot is provided below.



## Remove IP addresses that are on the remove list

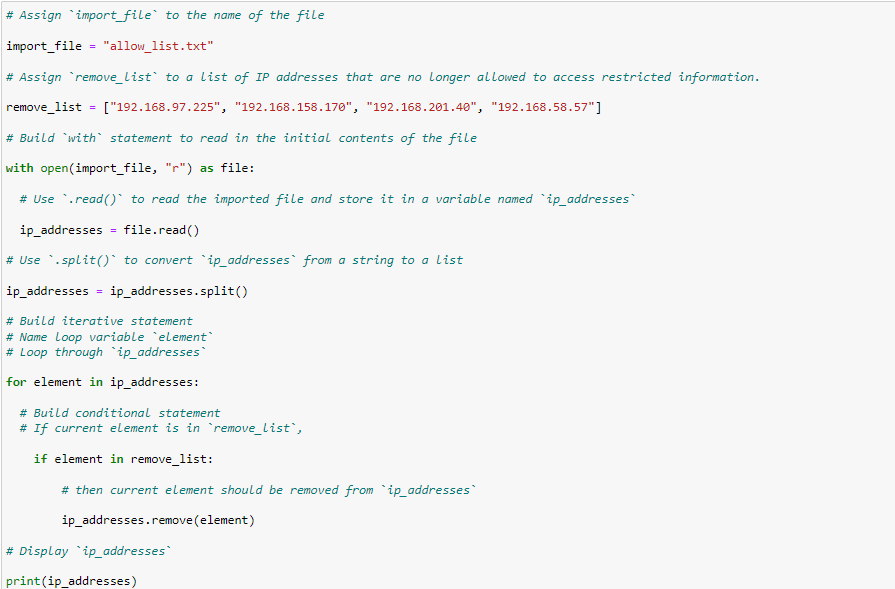
To remove the unauthorised IP addresses would require a conditional statement. A conditional statement evaluates code to determine if it meets a specified set of conditions.

In this case, after iterating through the remove\_list, we need to determine whether any of the IP addresses found in the remove list matchs those in the allow list, hence the need for the conditional statement you can see in the screenshot.

To remove the unauthorised IP addresses, I used the .remove() method. The remove method is used to remove the first occurrence of an element in a list.

In our case, using this method is justified because there are no repeated elements in the list.

Examine the code below:

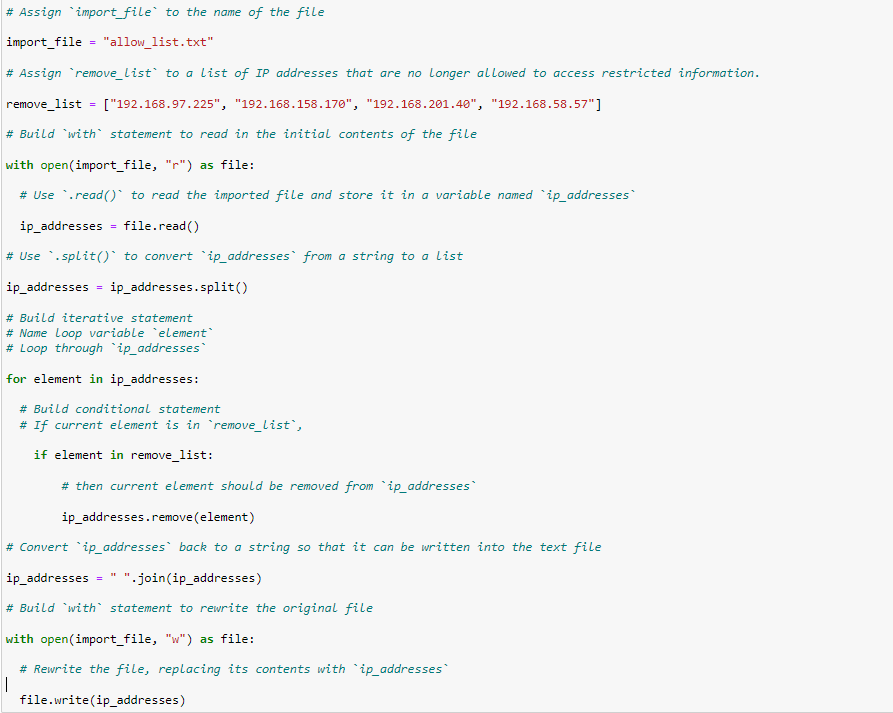


## Update the file with the revised list of IP addresses

It is important to remember that the value the ip\_addresses variable is what’s i’ve updated in the previous code I ran containing the conditional. Hence, the original file which is import\_file needs to be updated.

To accomplish this, I have to open the import\_file and replace it’s contents with the value of the ip\_addresses variable.

Remember the with keyword and the open()function? I will use these to open and overwrite the file.



If you wish to read the contents of the file to verify the updated contents, kindly add the following code:

with open(import\_file, “r”) as file:

text = file.read()

This reads the updated contents of the file and stores it in a new variable named text.

## Summary

My goal was to create an algorithm that automates the process of promptly detecting and preventing unauthorised employees or anyone else from accessing patients’ records.

The algorithm comprised opening, reading and writing to a file in Python. To accomplish this, I used the open(), .read(), and .write()function and methods.

The .split() was used to convert the string variable to a list which made it easier to iterate by means of a loop and a conditional statement.

Deletion was made possible using the .remove() method while the .join() method helped me to convert the list back to a string before updating the original document.

Thank you for reviewing this project! If you have any questions or suggestions, please feel free to contact me via LinkedIn.