# Algorithm for file updates in Python

## Project description

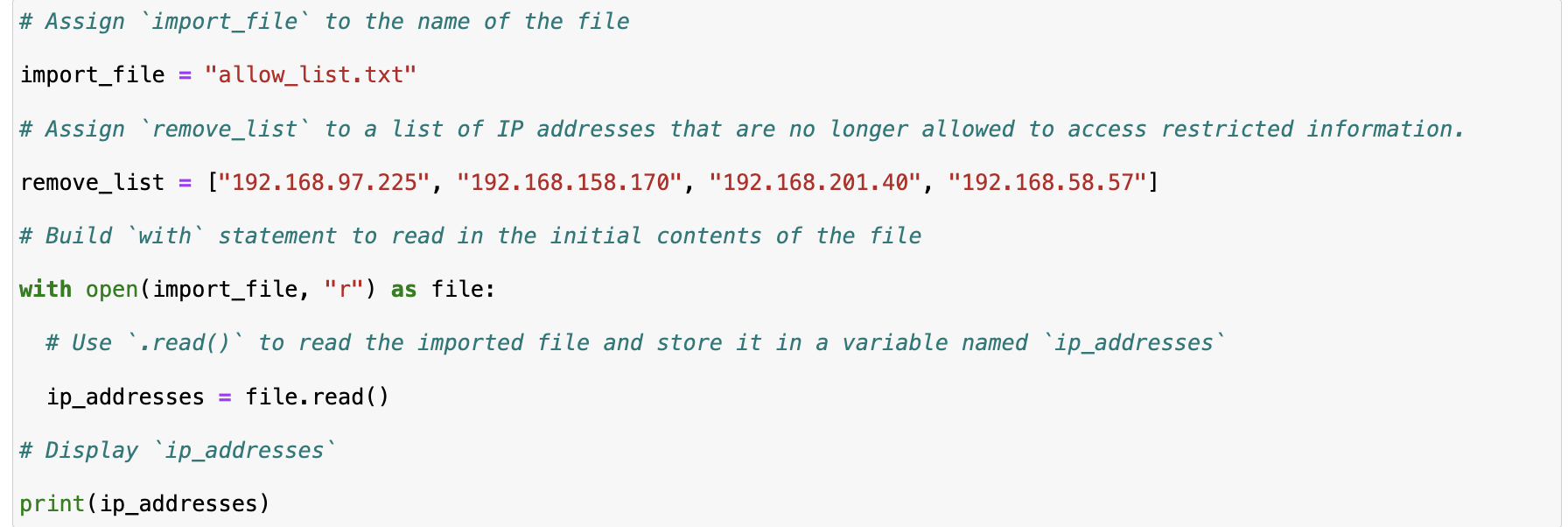
Updating the allow list file at my company, which contains the ip addresses of computers that are allow to view certain content and removing ip address that shouldn’t.

## Open the file that contains the allow list

## For the first part of the algorithm, I opened the “allow\_list.txt” file and assigned this file name as a string to the “import\_file” variable:

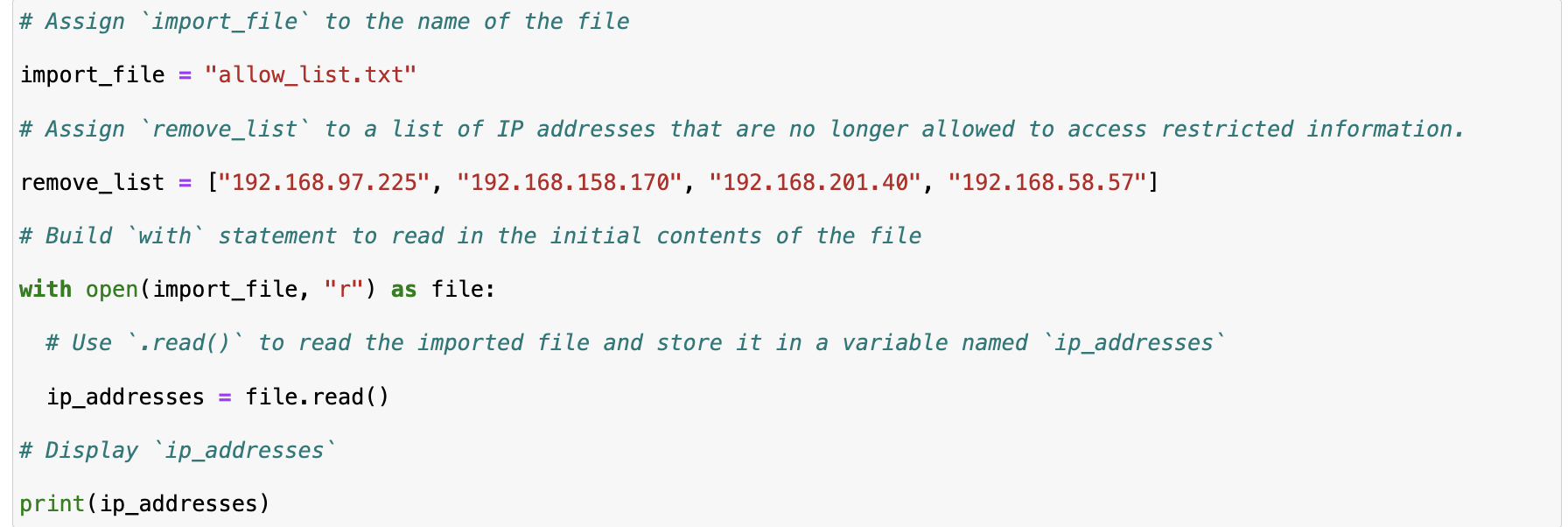


In the code with open(import\_file, "r") as file:, the open() function has two parameters. The first identifies the file to import, and then the second indicates what I want to do with the file. "r" indicates that I want to read it. The code also uses the as keyword to assign a variable named file; file stores the output of the .open() function while I work within the with statement. The “with” is used to closed the file after exiting the statement.



## Read the file contents

When using an .open() function that includes the argument "r" for “read,” I can call the .read() function in the body of the with statement. The .read() method converts the file into a string and allows me to read it. I applied the .read() method to the file variable identified in the with statement. Then, I assigned the string output of this method to the variable ip\_addresses.



## Convert the string into a list

The .split() function is called by appending it to a string variable. It works by converting the contents of a string to a list.



## 

## Iterate through the remove list

A key part of my algorithm involves iterating through the IP addresses that are elements in the remove\_list. To do this, I incorporated a for loop to apply specific code to all the elements in the sequence. Afterwards the result will be stored in the loop variable’ “elements.”



## Remove IP addresses that are on the remove list

First, within my for loop, I created a conditional that evaluated whether or not the loop variable element was found in the ip\_addresses list. I did this because applying .remove() to elements that were not found in ip\_addresses would result in an error.

Then, within that conditional, I applied .remove() to ip\_addresses. I passed in the loop variable element as the argument so that each IP address that was in the remove\_list would be removed from ip\_addresses.



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

The .join() method combines all items in an iterable into a string. The .join() method is applied to a string containing characters that will separate the elements in the iterable once joined into a string. In this algorithm, I used the .join() method to create a string from the list ip\_addresses so that I could pass it in as an argument to the .write() method when writing to the file "allow\_list.txt". I used the string (" ") as the separator to instruct Python to place each element on a new line.

Then, I used another with statement and the .write() method to update the file:

I used a second argument of "w" with the open() function in my with statement. This argument indicates that I want to open a file to write over its contents. When using this argument "w", I can call the .write() function in the body of the with statement. The .write() function writes string data to a specified file and replaces any existing file content.

In this case I wanted to write the updated allow list as a string to the file "allow\_list.txt". This way, the restricted content will no longer be accessible to any IP addresses that were removed from the allow list. To rewrite the file, I appended the .write() function to the file object file that I identified in the with statement. I passed in the ip\_addresses variable as the argument to specify that the contents of the file specified in the with statement should be replaced with the data in this variable

