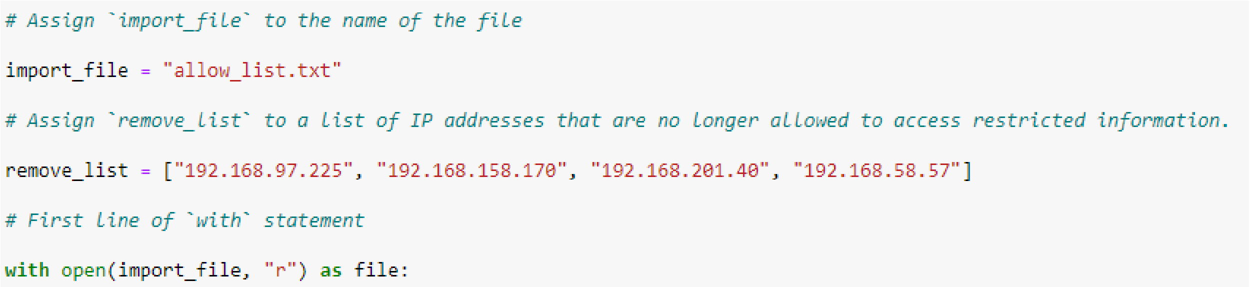
Algorithm for file updates in Python

Project description

This project's goal is to create an algorithm that can parse through a list of IP addresses that's allowed to access personal patient records, compare each IP addresses to a list of IP addresses that should be removed to restrict employees access to the records

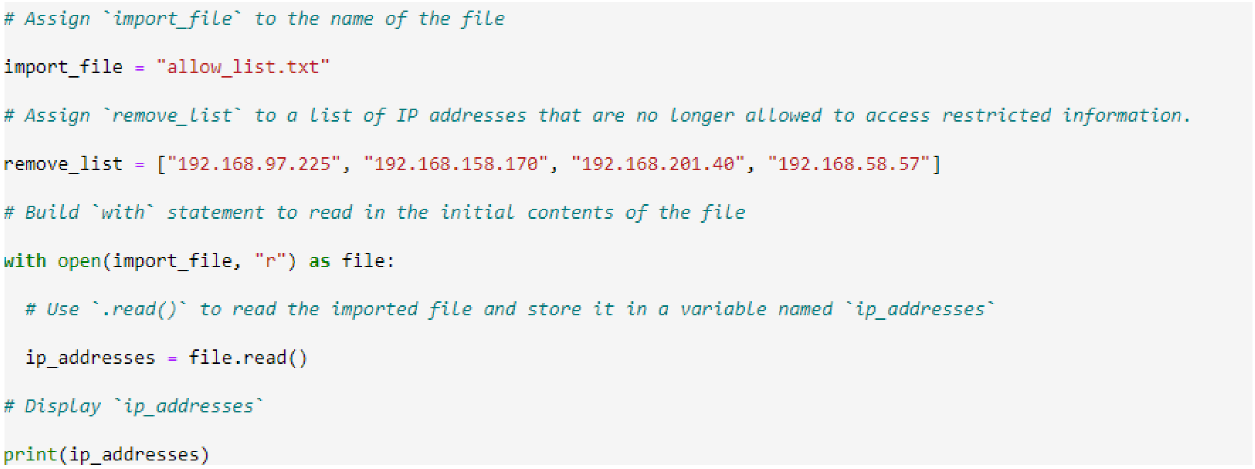
Open the file that contains the allow list



```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# First line of `with` statement

with open(import_file, "r") as file:
```

In here the list of IP addresses mention above is stored in the "allow_list.txt" file, we assigned it to a variable named import_file
The next line is storing the list of IP addresses that need to be removed into a variable called remove_list.
Then the next line is to open the import file. Here we use "with" to handle opening file errors and to make sure that the file is open and closed immediately after use to preserve the data.
The "open" function takes 2 arguments, one is the name of the file that we need to open, here it is "import_file" and the second argument is what action we are going to do to the file, here it is "r" for reading. After that we assign the opened file to an object name file.

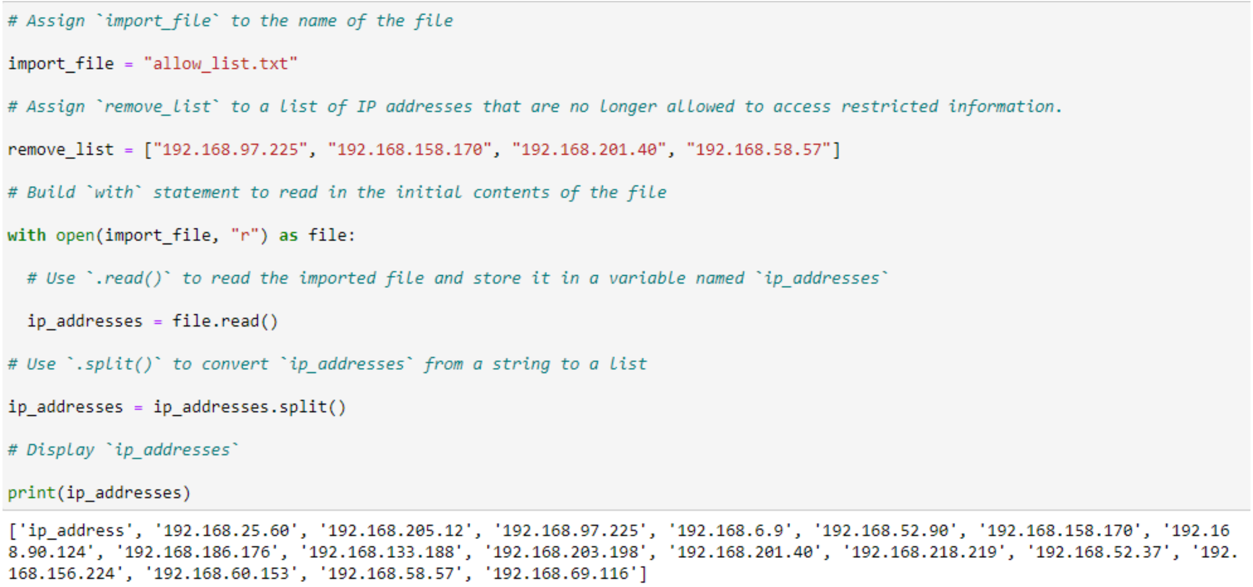
Read the file contents



```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Display `ip_addresses`

print(ip_addresses)
```

The next step is to read the file. Here we use the .read() method from the file object to read the contents of the import file and store them in the ip_addresses variable. To test if the command works we printed it using the print function.

Convert the string into a list

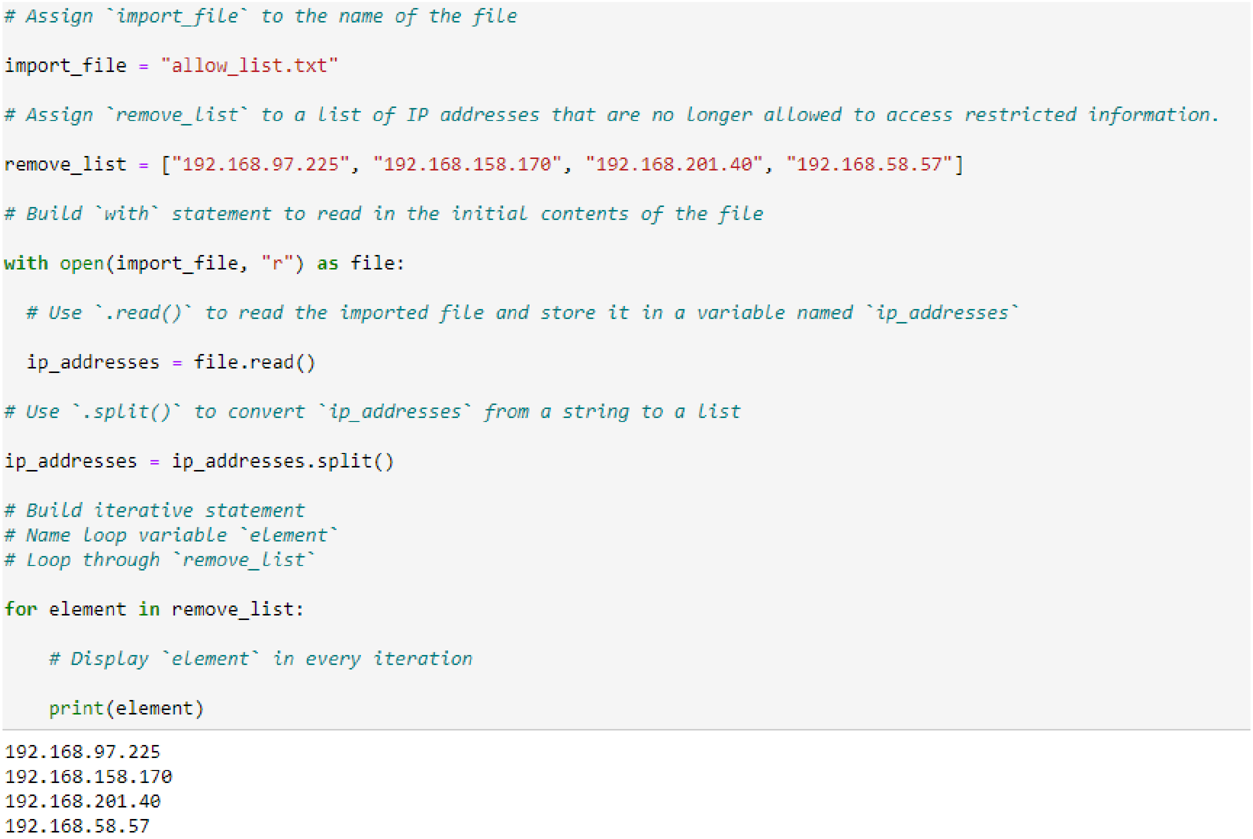


```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.16
8.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.
168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

Our next step is to convert the string that is stored in the ip_addresses variable into a list that we will store in the same variable. To do that we use the split() function from the string class. The result is shown in the bottom of the screenshot

Iterate through the remove list

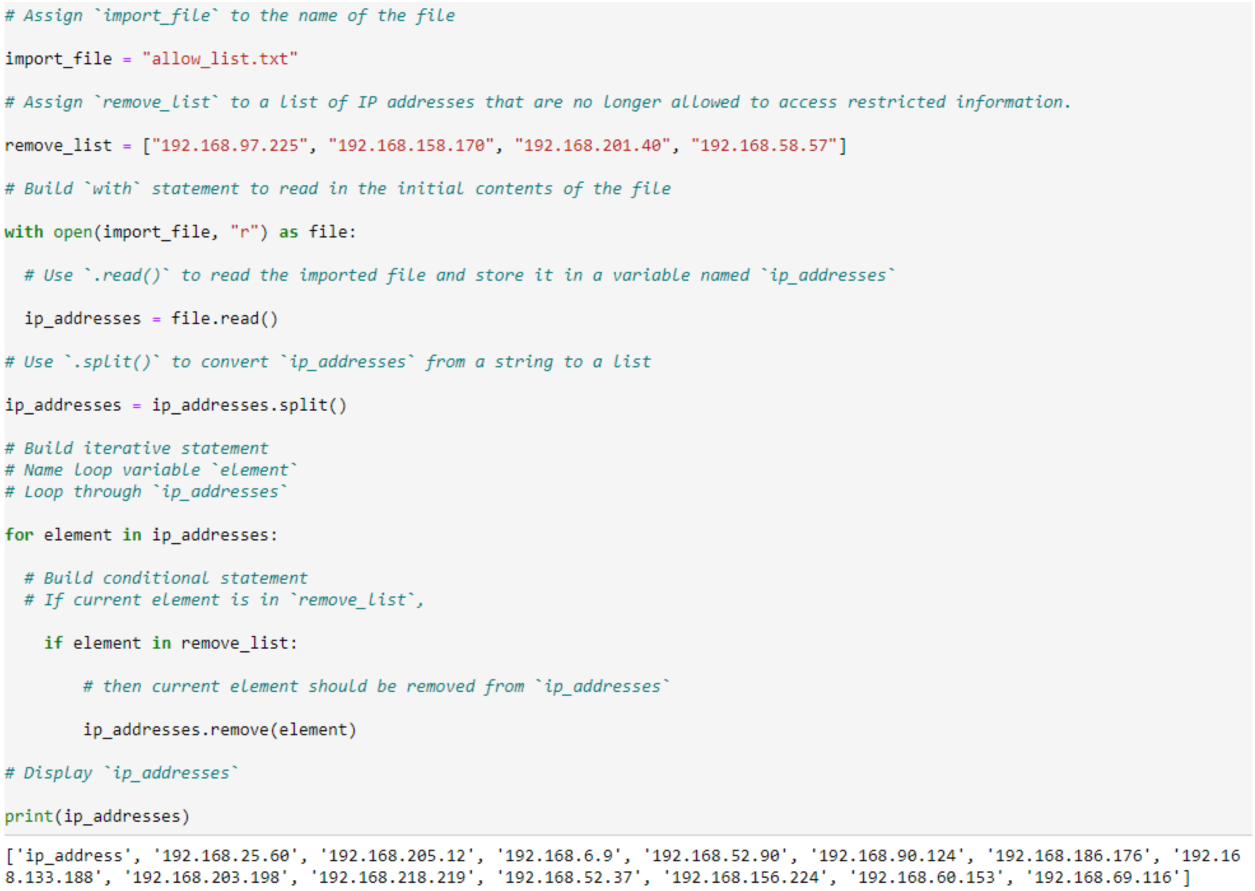


```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `remove_list`

for element in remove_list:

    # Display `element` in every iteration

    print(element)
```

```
192.168.97.225
192.168.158.170
192.168.201.40
192.168.58.57
```

We also need to iterate through the remove_list, to do that we use the for loop, we name the iterator as element, and we use the “in” syntax to tell Python to use the iterator to iterate through the remove_list.

Remove IP addresses that are on the remove list

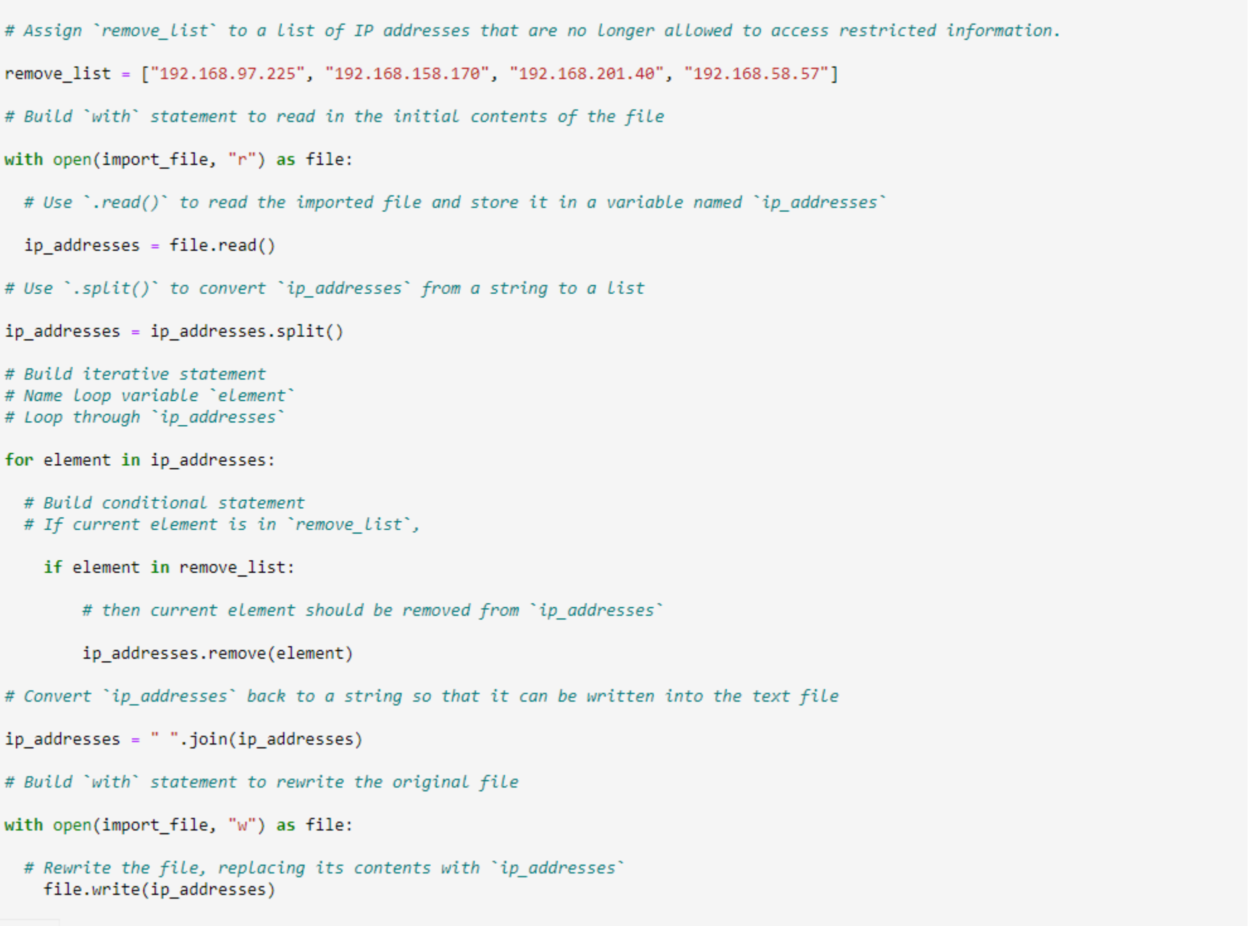


```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

for element in ip_addresses:

  # Build conditional statement
  # If current element is in `remove_list`,

    if element in remove_list:

        # then current element should be removed from `ip_addresses`

        ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.90.124', '192.168.186.176', '192.16
8.133.188', '192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

The next step we need to do is to iterate through every element in the ip_addresses, compare them to every element in the remove_list and then remove the matching elements in the ip_addresses list. To do this task we use a for loop to iterate through the ip_addresses. We use the if statement to check if an element in the ip_addresses list exists in the remove_list. And lastly we use the .remove() method to form the string class to remove the matching ip_addresses.

Update the file with the revised list of IP addresses



```
# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

for element in ip_addresses:

  # Build conditional statement
  # If current element is in `remove_list`,

    if element in remove_list:

        # then current element should be removed from `ip_addresses`

        ip_addresses.remove(element)

# Convert `ip_addresses` back to a string so that it can be written into the text file

ip_addresses = " ".join(ip_addresses)

# Build `with` statement to rewrite the original file

with open(import_file, "w") as file:

  # Rewrite the file, replacing its contents with `ip_addresses`
    file.write(ip_addresses)
```

And the last step of the algorithm is to update the original file with the new list of ip_addresses. We use the .join() function to convert back the list of ip_addresses into a string. The syntax for that is " ".join(ip_addresses). Here we join them with the ""(space) white space that means every element in the ip_addresses will be separated with a space. And the .join() function takes one argument and that is the list. And then we do something similar to opening and reading a file, only this time instead of "r" for reading we use "w" for rewriting the file. And then we use the write() method from the file object to write the updated list into the original file.

Summary

The algorithm takes several steps to complete its goal. First we opened and stored the file in a variable called import_list, and we stored the supposed to be removed ip addresses in the remove_list object. THen we read the file and convert it to a list of ip addresses and store them in ip_addresses list. Next we use a for loop, if statement, and the remove() method to iterate through the list and remove the matching ip addresses. And lastly we rewrite the original file with the updated list.