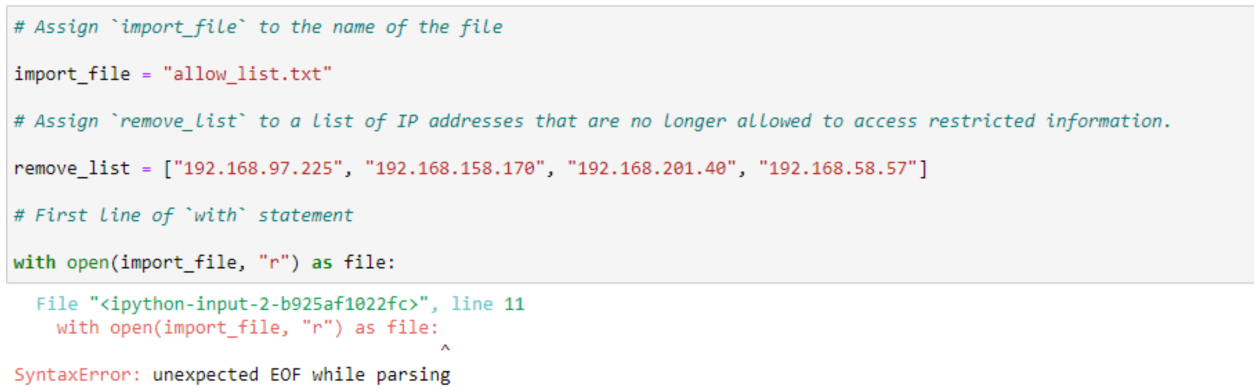
Algorithm for file updates in Python

Project description

I created an algorithm that both parses a file and updates it to remove IP addresses that are no longer permitted access.

Open the file that contains the allow list

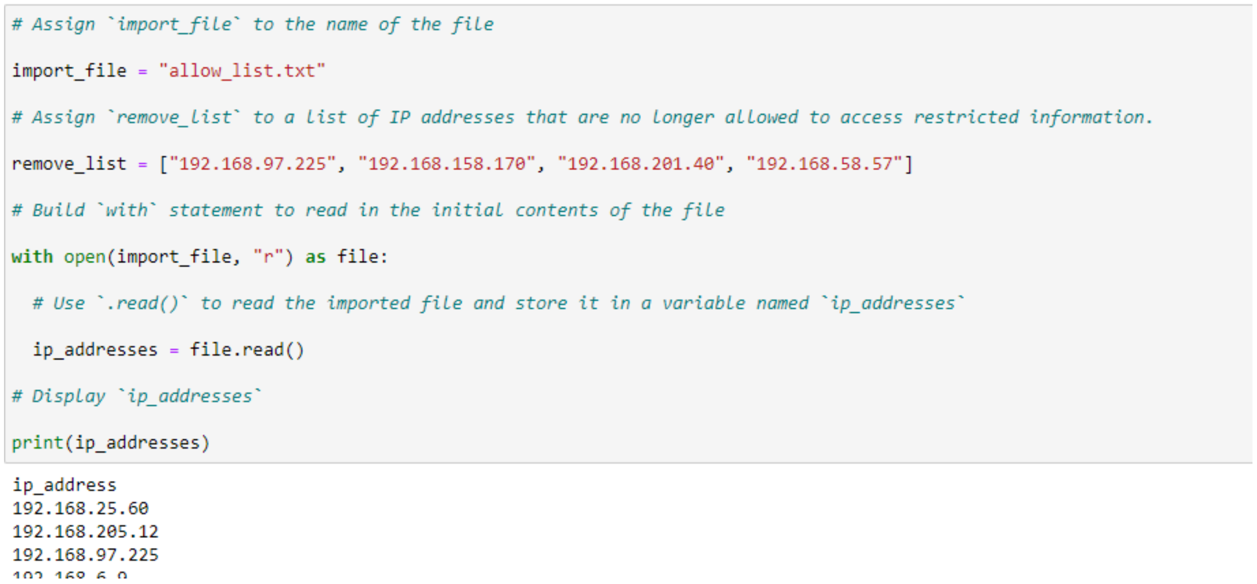


```
# Assign `import_file` to the name of the file

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:
```

```
  File "<ipython-input-2-b925af1022fc>", line 11
    with open(import_file, "r") as file:
                                        ^
SyntaxError: unexpected EOF while parsing
```

Here I begin writing code to open the file “allow_list.txt.” The file is stored in the “import_file” variable. I opened the file using the open() function in a “with” statement, and passed the argument read(r). This code turns the file into a string to be read.

Read the file contents

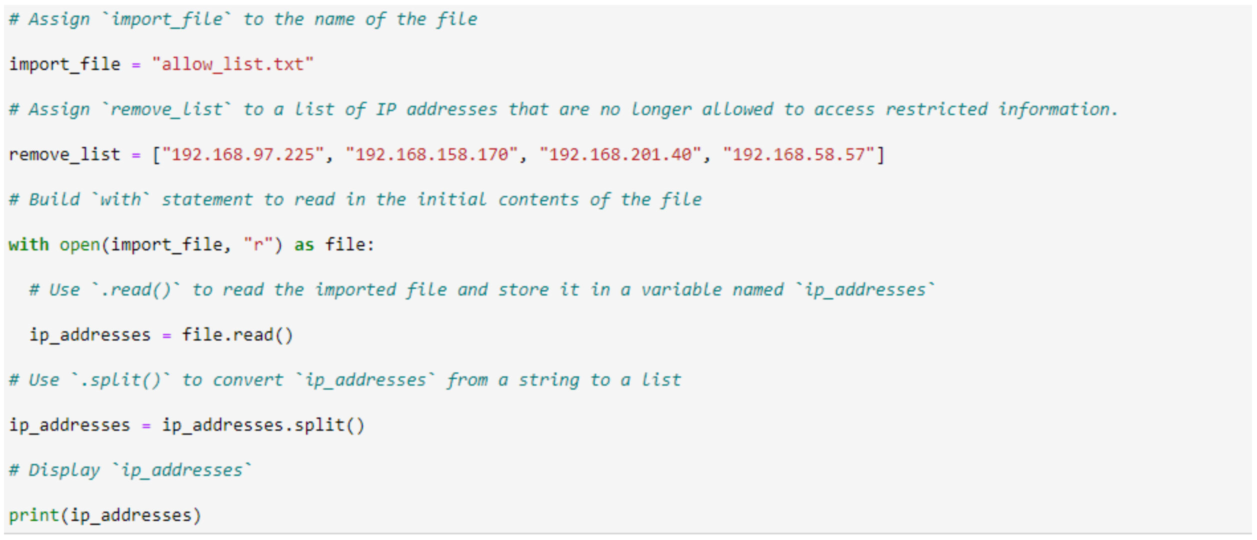


```
# Assign `import_file` to the name of the file

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)
```

```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
```

Here I complete the code by reading the file using .read() to call the “r” argument, and store this in the variable “ip_addresses.” I then use the print() function to display the contents of “ip_addresses.”

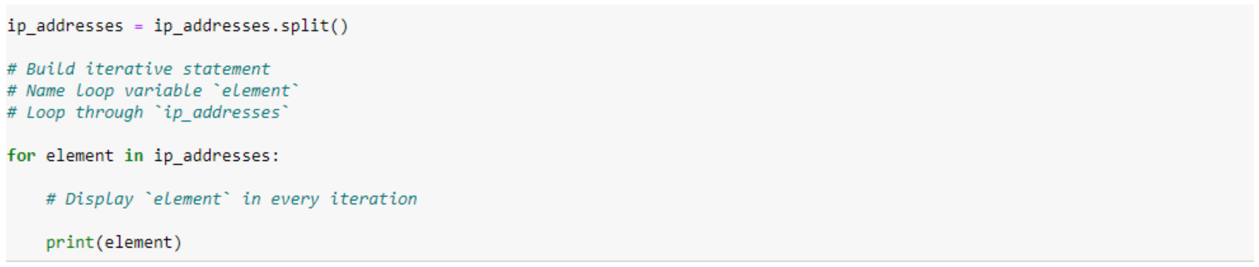
Convert the string into a list



```
# 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)
```

Here I update the ip_addresses variable using the .split() function on ip_addresses to convert the string into a list. The .split() function separates elements of a string by the white spaces into list elements separated by commas.

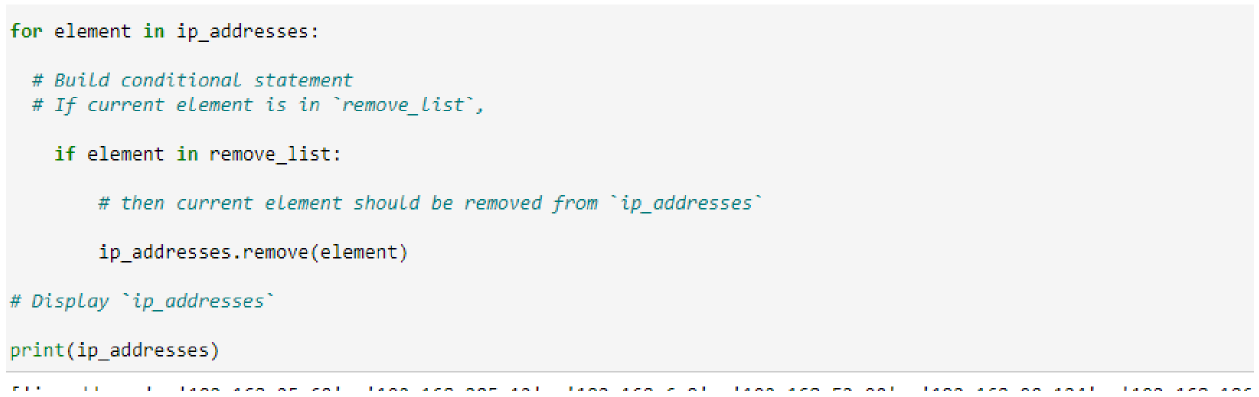
Iterate through the remove list



```
ip_addresses = ip_addresses.split()

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

for element in ip_addresses:

    # Display `element` in every iteration

    print(element)
```

Now I created a “for” loop that uses the loop variable “element” to iterate through ip_addresses via “in” and assign each value to “element.”

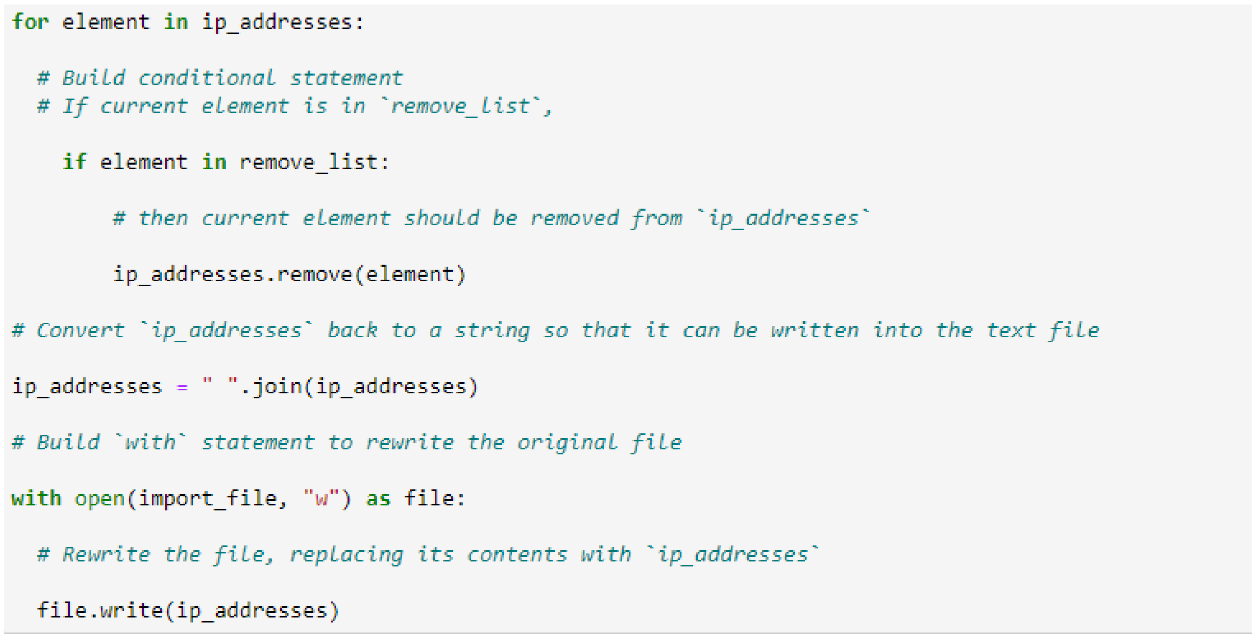
Remove IP addresses that are on the remove list



```
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)
```

Here I included an “if” statement to check if any of the values of remove_list are found in ip_addresses. The loop variable “element” is passed into the .remove() function to remove values that match from ip_addresses.

Update the file with the revised list of 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)
```

Here I converted the ip_addresses list back into a string by using the .join() function that will use the blank space " " to separate values. This is stored in the updated ip_addresses variable. I then open the original file again, but this time pass in the write(w) argument. I call the "w" argument with .write() to replace the file with the new string that has all values of remove_list removed.

Summary

I opened a file and read its contents. I then converted the string into a list. After that I used a "for" loop to to go through and remove any values that matched the remove_list variable from the ip_addresses variable. Finally, I converted the list back into a string by using .join() and updated the original file with the new string using .write().