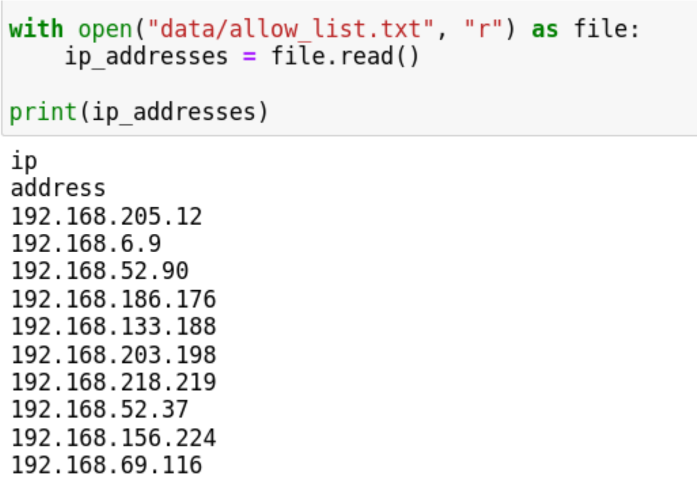
Algorithm for file updates in Python

Project description

Developing an algorithm that parses a file (“allow_list.txt”) containing IP addresses that are allowed access to restricted content. The algorithm will remove addresses that no longer are allowed access that will be stored in the variable remove_list. I will then update the file with the new list of allowed IP addresses.

Open the file that contains the allow list and read file contents

The following opens the file using “with” so that it automatically opens and closes the file after the file is read using the .read() method and stores the file contents in the variable ip_addresses.
It then prints the contents of the variable ip_addresses.



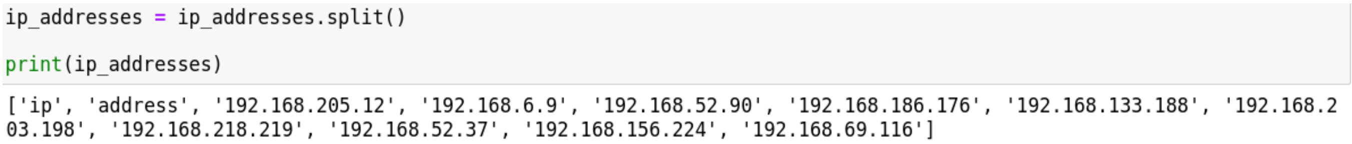
```
with open("data/allow_list.txt", "r") as file:
    ip_addresses = file.read()

print(ip_addresses)
```

```
ip
address
192.168.205.12
192.168.6.9
192.168.52.90
192.168.186.176
192.168.133.188
192.168.203.198
192.168.218.219
192.168.52.37
192.168.156.224
192.168.69.116
```

The addresses printed are those that are allowed access to restricted content.

Convert the string into a list

I use the .split() method to create the list, splitting the IP addresses by line of text. See the difference in the printout.

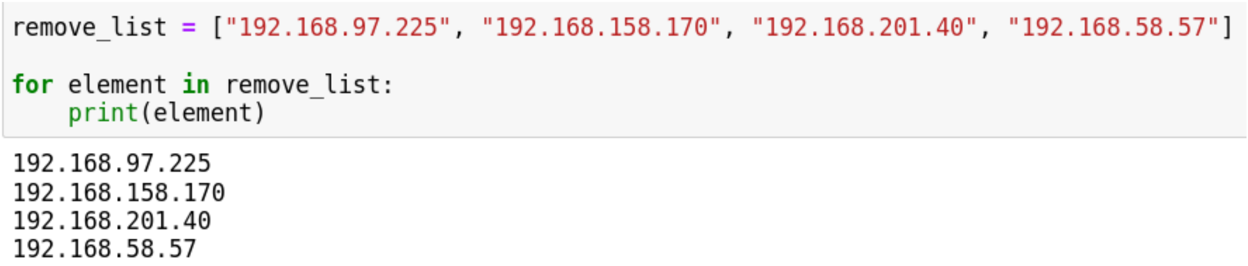


```
ip_addresses = ip_addresses.split()

print(ip_addresses)
```

```
['ip', 'address', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.186.176', '192.168.133.188', '192.168.2
03.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.69.116']
```

Iterate through the remove list

I use a for loop to iterate through the list of IP addresses that are to be removed (stored in the variable remove_list) from ip_addresses and then print them. The loop works by iterating through each element of the list (each IP address is an element of the list) and acting on the print() function for each element.

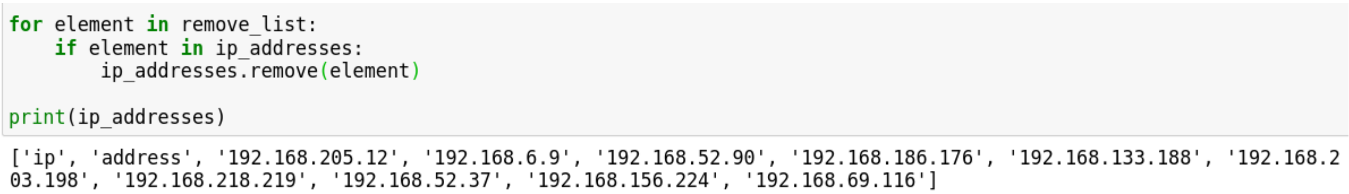


```
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

for element in remove_list:
    print(element)
```

```
192.168.97.225
192.168.158.170
192.168.201.40
192.168.58.57
```

Remove IP addresses that are on the remove list

I then use that for loop as a base to create a for loop that checks a condition. As the loop iterates it will check if the IP address in remove_list is in ip_addresses. If the element is in the list, it will act on the .remove() method to remove the IP address from ip_addresses. That IP address will no longer have access to restricted content.

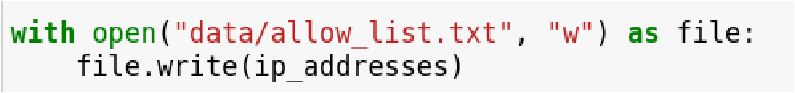


```
for element in remove_list:
    if element in ip_addresses:
        ip_addresses.remove(element)

print(ip_addresses)
```

```
['ip', 'address', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.186.176', '192.168.133.188', '192.168.2
03.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.69.116']
```

Update the file with the revised list of IP addresses

I then open the original file in “w”, write mode to replace the existing file using the .write() method with the updated list of IP addresses that have access to restricted content.



```
with open("data/allow_list.txt", "w") as file:
    file.write(ip_addresses)
```

Summary

The text file allow_list.txt held IP addresses that should no longer have access to restricted content. In order to remove those IP addresses, I used python to import the existing list of IP addresses and use a for loop to remove the IP addresses (stored in a variable) that were no longer allowed access to restricted content. I then updated the file with the new list of approved IP addresses.