Algorithm for file updates in Python

Project description

I work at a health care company and need to create an algorithm that will parse through a list of IP address to ensure that only allowed users are able to get access to certain data. I have a file that contains IP addresses that need parsing. I also have a list of IP's that need to be removed. I've created an algorithm that can handle this.

Open the file that contains the allow list

To open a file we'll be using the "with" command in conjunction with the "open" command to ensure that the file gets closed after getting the needed data.

```
# import a file with IP addresses and compare it with an allowed list
with open(import_file, "r") as file:

```

Read the file contents

To read the data, I make use of a method on the file to read the contents. I then pass the result into a variable.

```
# import a file with IP addresses and compare it with an allowed list
with open(import_file, "r") as file:
    ip_addresses = file.read().split()
```

Convert the string into a list

I will then, by stringing the split() command after the read() method to convert the string into a list. In this instance, not passing in any parameters so it defaults to white space.

```
# import a file with IP addresses and compare it with an allowed list
with open(import_file, "r") as file:
    ip_addresses = file.read().split()
```

Iterate through the remove list

Next, I’ll use a for loop to work through the remove item list.

```
"""
    Iterate through the list and compare each element in the remove list
    to what's in the list of IP allowed list
"""

for element in remove_list:

```

Remove IP addresses that are on the remove list

For each element in the remove list, I’ll compare it to the IP address list with an if statement. If it matches one in the list, then I’ll use the remove() method to delete the item.

```
"""
    Iterate through the list and compare each element in the remove list
    to what's in the list of IP allowed list
"""

for element in remove_list:

    # check if the element is in the list and if so, remove it.
    if element in ip_addresses:
        ip_addresses.remove(element)
```

Update the file with the revised list of IP addresses

Once complete, I’ll update the initial file using the “with” command again, this time using the “w” argument and the write() method to update the file.

```
# convert the list back to a string.
ip_addresses = ip_addresses.join()

#write the updated list back to the file.
with open(import_file, "w") as file:
    file.write("\n".join(ip_addresses))
```

Summary

I created a function that would handle parsing through the data. I also created a variable outside the function that will contain the file name that would be used in the function for easier updates down the line.

```
# Update a file through a python algorithm example


# var to contain the file name for the allowed list of IP address
allow_list_file = "allow_list.txt"

# create a function that handles the comparisons.
def update_file(import_file, remove_list):

    # import a file with IP addresses and compare it with an allowed list
    with open(import_file, "r") as file:
        ip_addresses = file.read().split()

    """
        Iterate through the list and compare each element in the remove list
        to what's in the list of IP allowed list
    """

    for element in remove_list:

        # check if the element is in the list and if so, remove it.
        if element in ip_addresses:
            ip_addresses.remove(element)

    # convert the list back to a string.
    ip_addresses = ip_addresses.join()

    #write the updated list back to the file.
    with open(import_file, "w") as file:
        file.write("\n".join(ip_addresses))

```