Algorithm for file updates in Python

Project description

The "allow_list.txt" is used by my organisation to identify ip addresses and control restricted content.ip_addresses that should no longer have access are removed by the algorithm to automate updating."allow_list.txt"

Open the file that contains the allow list

To open the file that contains the "allow_list.txt" I assigned as a string to the variable import file.

```
# Assign 'import_file' to the name of the file
 'Import_file' = "allow_list.txt"
To open the file the i used with statement
# build 'with' statement to read in the initial content of the file
With open (Import_file "r") as file:
```

The with function is used with .open() for the purpose of reading the file and gives access to the ip addresses stored in the allow list file. The .open() function identifies the file. The "r" indicates that i can read the file.The as keyword assigns a variable.

Read the file contents)

To read a file content , i will use the method that converts it into strings .read( )

```
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 ( )
```

The .read() method converts the file into a string and allows me to read it.this code reads the contents of the "allow_list . txt" file into a string format

that allows me to later use the string to organize and extract data in my Python program

Convert the string into a list

The `.split()` method converts the ip_addresses string into a list

```
# use '.split( )' to convert ip_addresses  to a list
Ip_addresses = ip_addresses . split( )
```

The `.split()` function is called by appending it to a string variable.The purpose of splitting `ip_addresses` into a list is to make it easier to remove IP addresses from the allowed list.The `.split( )` function converts this string into a list of IP addresses. To store this list, I reassigned it back to the variable `ip_addresses.`

Iterate through the remove list

iterating through the `IP_addresses` that are elements in the `remove_list`. To do this a for loop is incorporated before the remove_list

```
# build iterative statement
# name loop variable 'element'
# loop through 'remove_list'

for element in remove_list:
```

The `for` keyword apply specific code statements to all elements in a sequence.The `for` keyword starts the loop followed by the loop variable element
The keyword `in` indicates to iterate through the sequence and assign each value to the loop variable element.

Remove IP addresses that are on the remove list

I will be using the following code to remove ip_addresses on the `remove_list`

```
for  element in remove_list:
```

```
# create conditional statement to evaluate if element is in
ip_addresses
If element in ip_addresses:
# use the remove () method to remove
# element from ip_addresses
ip_addresses.remove (element)
```

To achieve this i created a conditional statement to evaluate if or not the loop variable
element was found in the ip_addresses list. I applied `.remove()` to ip_addresses which would remove `ip_addresses` that are listed in the `remove_list`.

Update the file with the revised list of IP addresses

To update the file of the revised list of ip_addresses I converted the list back into a string and used the `.join( )` method .

```
# convert ip_addresses back to string so that it can be written into
the text file
# ip_addresses = " \n" . join (ip_addresses)
```

The `.join( )` method combines all items in an iterable into a string. The string `("\n")` serves as the separator to instruct Python to place each element on a new line.

To update the file the `with` statement and the `.write ( )` method is used .

```
# build with statement to rewrite the original file
with open ( import_file, "w" ) as file :
# rewrite the file, replacing its content with ip_addresses
File.write (ip_addresses)
```

I used the `with` statement and the `.write()` method to write over the file assigned to the `import_file` variable. The `"w"` with the `open()` function indicates that I want to open a file to write over its contents. To rewrite the file, I appended the `.write()` function to the file object file that I identified in the `with` statement.

Summary

The algorithm I created opens the file that contains the `"allow_list.txt"` I assigned as a string to the variable `import_file`. It opens the file that contains the "allow_list.txt", to read the file I used `.read( )` method that converts it back to strings so that I can read them. I used the `.split()` method to convert the ip_addresses string into a list.I setup a `for loop` header to iterate through the remove_list that contains all the ip_addresses that should be removed.I applied the `.remove()` method to the `ip_addresses` list and remove the IP addresses identified in the loop variable `element`.i updated the file with the revised list of ip_addresses by converting the `ip_addresses` list back into a string using the `.join()` method. I applied the `.join()` method to combine all items in an iterable into a string. the string `"\n"` separate the elements in the file by placing them on a new line