Algorithm for File Updates in Python

Project description

I am developing a Python script to automate the management of an access list based on the list of IP addresses found in `"allow_list.txt"`. There is a separate list identifying the IP addresses that need to be removed from the allow list, so I am writing an algorithm to delete those IP addresses from the allow list and update the document.

Open the file that contains the allow list

First, to open the file, I assigned the variable `import_file` to the file `"allow_list.txt"`

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"
```

I then built a `'with'` statement to open `import_file` as the new variable `file`, and used the `"r"` parameter to tell Python to open the file with the purpose of reading it.

```
# Build 'with' statement to read the contents of the file
with open(import_file, "r") as file:
```

Read the file contents

To read the file, I used the method `.read()` and assigned its output to the new variable `ip_addresses`.

```
with open(import_file, "r") as file:
  # Using `.read()` to read the imported file and storing it in variable `ip_addresses`
  ip_addresses = file.read()
```

Convert the string into a list

In order for me to be able to iterate through the items of the file, I first had to separate the string into a list, and I used the `.split()` method for this.

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

ip_addresses = ip_addresses.split()
```

Iterate through the IP address list

To be able to repeat the instruction through each element of the list, I created a `for` loop where the loop variable is `element` for it to iterate through the `ip_addresses` list.

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

for element in ip_addresses:
```

Remove IP addresses that are on the remove list

To remove the IP addresses from the `ip_addresses` list, I wrote an `if` statement, where if the variable `element` is `in` the list `remove_list`, Python will run the `.remove( )` method through the `ip_addresses` list, with `element` as the argument of the method.

```
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)
```

Update the file with the revised list of IP addresses

In order to update the .txt file, I first had to convert the `ip_addresses` list back into a string, and I used the `.join()` method for this, with the instruction **"\n"** to indicate to Python to separate each element in new lines.

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

ip_addresses = "\n".join(ip_addresses)
```

Then, I wrote a `'with'` statement to open the file `import_file` with the `"w"` instruction to edit the file, and then stored the output in the variable `file`. To edit the file I used the `.write()` on the variable `file` and then used the `ip_addresses` list as the argument to be added into the file.

```
# 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)
```

Summary

I created an algorithm to remove IP addresses in the list `remove_list` from the `"allow_list.txt"` file. This algorithm converts the string output from the file into a list, iterates through the list with the instruction to remove any addresses from the `"allow_list.txt"` file also found in the `remove_list` list, converts the output back into a string, and updates the `"allow_list.txt"` file with the output from the algorithm. I then defined the function `def update_file(import_file, remove_list):` to be able to call the algorithm and automate the task. At the very end of the code, I added the `print(text)` function, with the variable `text` being the file with the `.read()` method to be able to double check my work once the algorithm runs.

```
def update_file(import_file, remove_list):

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

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

    # Using `.read()` to read the imported file and storing it in variable `ip_addresses`

    ip_addresses = file.read()

  # Using `.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 = "\n".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)

# Call `update_file()` and pass in "allow_list.txt" and a list of IP addresses to be removed

update_file("allow_list.txt", ip_addresses)

# Build `with` statement to read in the updated file

with open("allow_list.txt", "r") as file:

  # Read in the updated file and store the contents in `text`

  text = file.read()

# Display the contents of `text`

print(text)
```