

Algorithm for file updates in Python

Project description

This project demonstrates my ability to import, read, write, and manipulate files within Python.

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:
```

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)
```

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)
```

Iterate through the remove 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()

# Build iterative statement
# Name Loop variable `element`
# Loop through `ip_addresses`
for element in ip_addresses:

    # Display `element` in every iteration
    print(element)
```

Remove IP addresses that are on the remove 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()

# 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)

# Display `ip_addresses`
print(ip_addresses)
```

Update the file with the revised list of IP addresses

```
# 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()

# 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 = " ".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)
```

Summary

The algorithm above shows every step I performed to build this Python function to first import the file in question and store its contents into a string variable. Once the string variable was created, I used a `split()` function to convert the string into a list variable to easily be able to remove contents of the list of IP addresses. Using a list of IP addresses to remove, I created IF conditional within a FOR loop to compare each element in the `remove_list` to the full list of IP addresses. Once all the necessary IP addresses have been removed from the main list, we reconverted the list variable back into a string, and overwritten the imported file with the new list.