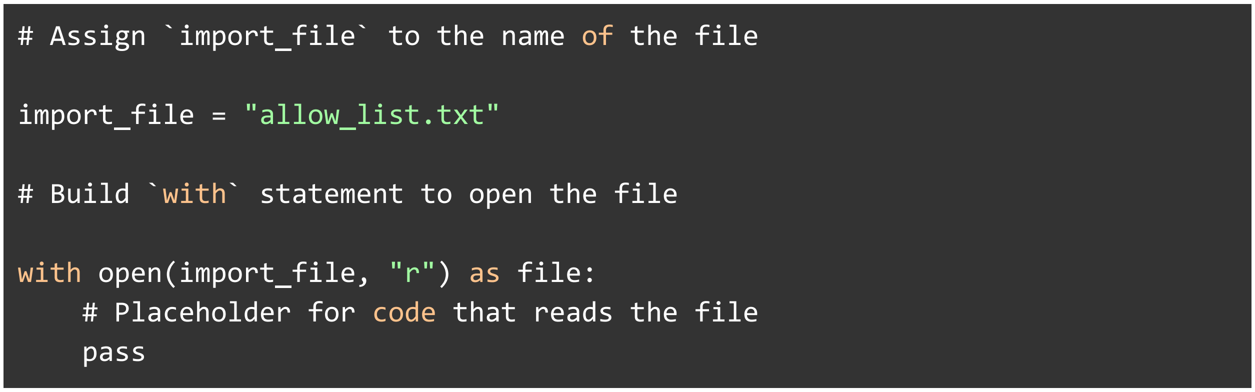
Algorithm for file updates in Python

Project description

The scenario involves managing an allow list file containing IP addresses that are allowed to access restricted information. I'm using Python to automate the process of updating this file by removing specified IP addresses.

Open the file that contains the allow list



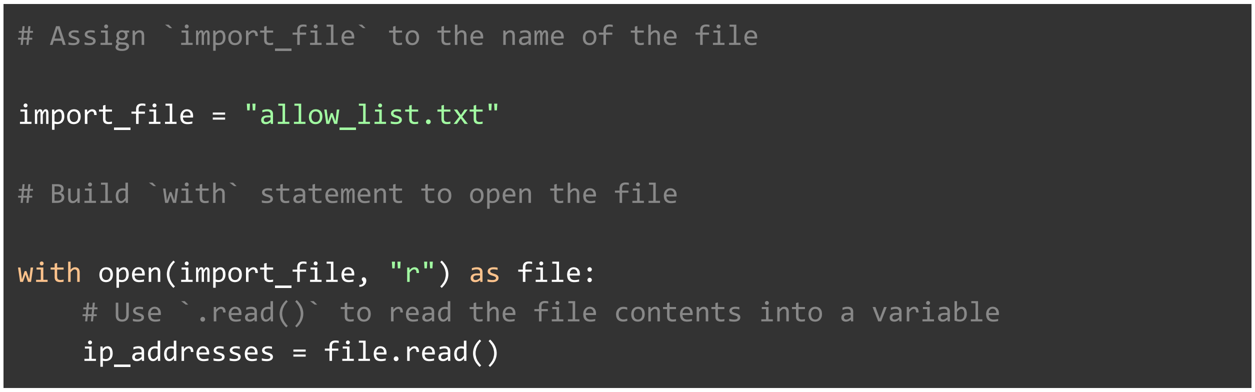
```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Build `with` statement to open the file

with open(import_file, "r") as file:
    # Placeholder for code that reads the file
    pass
```

Read the file contents



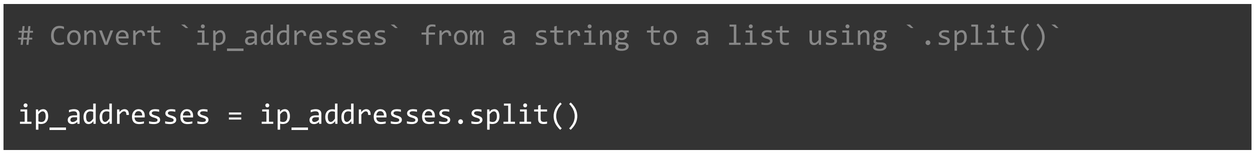
```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Build `with` statement to open the file

with open(import_file, "r") as file:
    # Use `.read()` to read the file contents into a variable
    ip_addresses = file.read()
```

Convert the string into a list



```
# Convert `ip_addresses` from a string to a list using `.split()`

ip_addresses = ip_addresses.split()
```

Iterate through the remove list

```
# Assign `remove_list` to a list of IP addresses to be removed

remove_list = ["192.168.25.60", "192.168.140.81", "192.168.203.198"]

# Iterate through `remove_list` using a for loop

for element in remove_list:
    # Placeholder for code that iterates through the remove list
    pass
```

Remove IP addresses that are on the remove list

```
# Iterate through `remove_list` using a for loop

for element in remove_list:
    # Check if the element is in `ip_addresses` and remove it if found
    if element in ip_addresses:
        ip_addresses.remove(element)
```

Update the file with the revised list of IP addresses

```
# Convert `ip_addresses` back to a string using `.join()`

ip_addresses_str = " ".join(ip_addresses)

# Build `with` statement to rewrite the file

with open(import_file, "w") as file:
    # Use `.write()` to update the file with the revised list
    file.write(ip_addresses_str)
```

Summary

The Python code automates the process of updating an allow list file by reading its contents, converting them into a list, removing specified IP addresses, and then updating the file with the revised list. This automation saves time and effort in managing the allow list.

Full code

```
# Define a function named `update_file` that takes in two parameters:
`import_file` and `remove_list`
# and combines the steps you've written in this lab leading up to this

def update_file(import_file, remove_list):

  # 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)

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

update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])

# 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)
```