

# Algorithm for file updates in Python

## Project description

In this lab, I'm working as a security analyst and am responsible for developing an algorithm that parses a file containing IP addresses that are allowed to access restricted content and removes addresses that no longer have access. My eventual goal is to develop an algorithm that parses a series of IP addresses that can access restricted information and removes the addresses that are no longer allowed.

## Open the file that contains the allow list

```
In [ ]: # 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:
```

Using:

- Import\_file - variable that allows Python to access a file
- With - keyword manages that external resource
- open() - opens the file in Python
- "r" - parameter that tells Python that I want to read the contents of the file

## Read the file contents

```
In [ ]: # 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)
```

Using:

- .read() - method that converts files into strings

## Convert the string into a list

```
In [ ]: # 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)
```

Using:

- `.split()` - convert a string into a list

## Iterate through the remove list

```
In [ ]: # 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)
```

Using:

- Iterative statement - used to execute a block of code repeatedly as long as a specified condition is met

- Conditional statement - allows you to execute code based on certain conditions. The primary conditional statements are if, elif, and else.
- for loop - used for iterating over a sequence

## Remove IP addresses that are on the remove list

```
In [ ]: # 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)
```

Using:

- .read() - method that converts files into strings
- .split() - convert a string into a list
- Iterative statement - used to execute a block of code repeatedly as long as a specified condition is met
- .remove() - removes the first occurrence of a specific element in a list

## Update the file with the revised list of IP addresses

```
In [ ]: # 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)
```

Using:

- `.join()` - converts a list into a string
- `.write()` - writes a specified text to a file

## Summary

I created an algorithm that removes IP addresses identified in a `remove_list` variable from the "allow\_list.txt" file of approved IP addresses. This algorithm involved opening the file, converting it to a string to be read, and then converting this string to a list stored in the variable `ip_addresses`. I then iterated through the IP addresses in `remove_list`. With each iteration, I evaluated if the element was part of the `ip_addresses` list. If it was, I applied the `.remove()` method to it to remove the element from `ip_addresses`. After this, I used the `.join()` method to

convert the ip\_addresses back into a string so that I could write over the contents of the "allow\_list.txt" file with the revised list of IP addresses.