

Algorithm for file updates in Python

Project description

In this project I was tasked with writing algorithms in python that would allow for file updates. The organization described in the scenario had restricted content that gave authorization through the “allow_list.txt” file. In this project I created an algorithm that removed the IP addresses that should not have access to the restricted content.

Open the file that contains the allow list

To start I had to open the “allow_list.txt” file. I gave this file name as a string to the variable “import_file”:

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

After this I used the “with” command to open the file:

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

In this algorithm the “with” statement is combined with the `.open()` function in read mode which will allow the file to be opened for the sole purpose of reading it. The “r” indicates that I want to open the Import_file in read mode. The “as file:” at the end tells the algorithm to assign a variable named “file”.

Read the file contents

In order to actually read the contents of the file I used the command `.read()` to convert it into a string:

```
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 code above allows me to read the contents of the file in a string and I will be able to use this to extract data using Python later on.

Convert the string into a list

The next step was to convert `ip_addresses` into a list format using the `.split()` command:

```
# Use '.split()' to convert 'ip_addresses' from a string to a list  
  
ip_addresses = ip_addresses.split()
```

The `.split()` function above converts the `ip_addresses` from a string into a list. This list gets stored by reassigning it back to the variable `ip_addresses`.

Iterate through the remove list

Iterating through the `remove_list` requires me to incorporate a `for` loop in my algorithm as shown below:

```
# Build iterative statement  
# Name loop variable 'element'  
# Loop through 'remove_list'  
  
for element in remove_list:
```

In python the `for` loop is used to repeat code in a sequence. `Element` in the statement above is the loop variable and the keyword `in` is used to iterate through the `ip_addresses` and assign each value to the `element` variable.

Remove IP addresses that are on the remove list

Removing Ip addresses from the remove list was also a task I had to include in my python algorithm. To do this I used the following code:

```

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
        # elements from `ip_addresses`

        ip_addresses.remove(element)

```

This code creates a conditional statement that evaluates if the element is in `ip_addresses`. I then used the `.remove()` method to remove elements from `ip_addresses`.

Update the file with the revised list of IP addresses

In the last step of my algorithm I had to update the file with the revised list of IP addresses. I did so using the code shown below:

```

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

ip_addresses = "\n".join(ip_addresses)

```

I used the `.join()` method to combine all the items in the iterable into a string. I then used a `with` statement and `.write()` method to update 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)

```

I then used the `with open(import_file, "w") as file` to open the file in write mode and write string data to a file and replace the existing content.

The `file.write(ip_addresses)` allowed me to rewrite the file and replace its contents with `ip_addresses`.

Summary

I enjoyed the project a lot! Creating an algorithm to update file contents using python was tricky but rewarding. In my algorithm I opened the `allow_list.txt` file and first converted it to a string. After converting it to a string I then converted the string to a list stored in the

variable `ip_addresses`. I then did some iteration through the IP addresses in the `remove_list`. I then used the `.remove()` command to remove unwanted elements from `ip_addresses`. I then used the `.join()` method to turn the `ip_addresses` back into a string. I then wrote over the contents of the `allow_list.txt` with the new and updated version of the `ip_addresses`.