Algorithm for file updates in Python

Project description

The following algorithm helps us structure data in a way that's readable and can be work with. This is called "**parsing**". This activity exemplifies the steps to be taken when specific elements of a list need to be removed from a different list, like IP addresses.

Open the file that contains the allow list

We begin by using the "**with**" keyword so that we can handle files in Python. Then we use the "**open**( )" method to import the file. The first argument of this function is the file we want to import, and the second argument is either "**r**", "**w**", or "**a**" to read, write or append respectively. For now, all we have to do is open the file to read it:

```
# 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

We now need to convert the contents of the file into string data so that we can work with it throughout our code. The "**read**( )" method helps us do this. We create a new variable named "**ip_addresses**" to store the resulting string and we display its contents with "**print**( )":

```
# 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

Later we want to iterate through the elements of a list, but we don't have one yet. That's what the function "**split**( )" does. It converts a string into a list, and its elements are separated by whitespace if we don't input an argument in the function, like so:

```
# 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)
```

And this is the output:

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.16
8.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.
168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

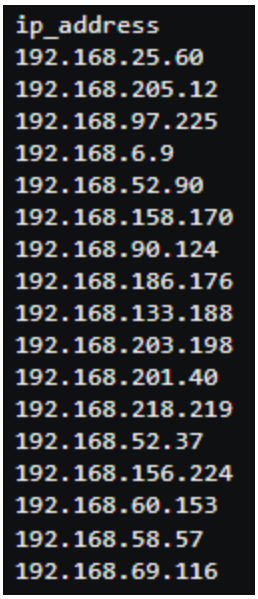
As expected, we now have a list from the original string.

Iterate through the remove list

The end goal is to remove IP addresses of a list from the approved list. The first step is to iterate through our original list and explore its contents. We do this by using a **for loop**. In this case, we are using the variable "**element**" to be used within the loop to iterate through the list and print each of its elements with the last instruction of this section of code, which is the function "**print**( )":

```
# 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)
```

And the output is the expected one:



Remove IP addresses that are on the remove list

We now complete the process by adding a conditional statement into the **for loop**. This condition checks if each IP address of the original list is also in the list of IPs to be removed. If it is, the "**remove**( )" method comes into action, deleting that element from the original list. The only argument we need to pass is the element itself, which in this case is the variable used to iterate through the original list. The last "**print**( )" function displays the updated list. All four IPs included in the "**remove_list**" variable have been deleted. For readability, I will leave the first lines of code out of this and subsequent screen captures.

```
# 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)
```

The output is as follows:

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.90.124', '192.168.186.176', '192.16
8.133.188', '192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

Update the file with the revised list of IP addresses

Finally, we need to convert the list back to a string, by using the "**join**( )" function. Once that's done, we open the original file and overwrite its contents with the updated information stored in the "**ip_addresses**" variable. To do this, we use the "**write**( )" method.

```
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

In this activity, we had to work with two different lists, and the task was to remove the elements of “removed_list” from the original list, which is a separate file we imported to Python: “allow_list.txt”. We put in practice the concepts needed to work with a file in Python, like opening it, reading it (converting the information the file contains to a string data type), writing on it to update the changes that were made with the program.