Step-By-Step Instructions

Follow the instructions and fill in the sections to complete the activity.

Step 1: Access the template

To use the template for this course item, click the link and select *Use Template*. (In this step, you will just open the template. More instructions for how to use the template will be included in later steps.)

Access the template (Algorithm for the updates in Python)

Step 2: Access supporting materials

The following supporting material will help you complete this activity. The document Instructions for including Python code provides instructions and best practices for including samples of Python code in your portfolio activity. Keep it open as you proceed to the next steps.

Navigate to supporting material(Instructions for including Python code)

Step 3: Open the file that contain the allow list

The file that you want to open is called **`"allow_list.txt"`**. Assign a string containing this file name to the **`import_file`** variable. Then, use a **`with`** statement to open it. Use the variable **`file`** to store the file while you work with it inside the **`with`** statement.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Open the file that contains the allow** list section of the Algorithm for file updates in Python template. In the **Task 2** section of **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 4: Read the file contents

Next, use the `.read()` method to convert the contents of the allow list file into a string so that you can read them. Store this string in a variable called `ip_addresses`.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Read the file contents** section of the **Algorithm for file updates in Python** template. In the **Task 3** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 5: Convert the string into a list

In order to remove individual IP addresses from the allow list, the IP addresses need to be in a list format. Therefore, use the `.split()` method to convert the `ip_addresses` string into a list.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Convert the string into a list** section of the **Algorithm for file updates in Python** template. In the **Task 4** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 6: Iterate through the remove list

A second list called `remove_list` contains all of the IP addresses that should be removed from the `ip_addresses` list. Set up the header of a `for` loop that will iterate through the `remove_list`. Use `element` as the loop variable.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Iterate through the remove list** section of the **Algorithm for file updates in Python** template. In the **Task 5** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 7: Remove IP addresses that are on the remove list

In the body of your iterative statement, add code that will remove all the IP addresses from the allow list that are also on the remove list. First, create a conditional that evaluates if the loop variable `element` is part of the `ip_addresses` list. Then, within that conditional, apply the `.remove()` method to the `ip_addresses` list and remove the IP addresses identified in the loop variable `element`.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Remove IP addresses that are on the remove list** section of the **Algorithm for file updates in Python** template. In the **Task 6** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

In addition, include a sentence that explains that applying the `.remove()` method in this way is possible because there are no duplicates in the `ip_addresses` list.

Step 8: Update the file with the revised list of IP addresses

Now that you have removed these IP addresses from the `ip_address` variable, you can complete the algorithm by updating the file with this revised list. To do this, you must first convert the `ip_addresses` list back into a string using the `.join()` method. Apply `.join()` to the string `"\n"` in order to separate the elements in the file by placing them on a new line.

Then, use another **`with`** statement and the `.write()` method to write over the file assigned to the `import_file` variable.

Describe the Python syntax, functions, and keywords you need to accomplish this in the **Update the file with the revised list of IP addresses** section of the **Algorithm for file updates in Python** template. In the **Task 7** section of the **Create another algorithm** lab, take a screenshot of this portion of your code. Or, type this code directly into the template.

Step 9: Finalize your document

To finalize the document and make its purpose clear to potential employers, be sure to complete the **Project description** and **Summary** sections of the **Algorithm for file updates in Python** template.

In the Project description section, give a general overview of the scenario and what you accomplished in Python. Write three to five sentences.

In the Summary section, provide a short summary of the algorithm by highlighting its main components. Write four to six sentences.

What to Include in Your Response

Be sure to include the following in your completed activity:

- Screenshots of your Python code or typed versions of the code
- Explanations of the syntax, functions, and keywords in the code
- A project description at the beginning
- A summary at the end
- Details on using a `with` statement and the `open()` function in your algorithm
- Details on using the `.read()` and `.write()` methods in your algorithm
- Details on using the `.split()` method in your algorithm
- Details on using a `for` loop in your algorithm
- Details on using the `.remove()` method in your algorithm