

# Algorithme de mises à jour de fichiers Python

Google Cybersecurity Certificate

## Description du projet

Dans mon entreprise, l'accès à un contenu restreint est contrôlé par une liste d'adresses IP autorisées. Le fichier `"allow_list.txt"` identifie ces adresses IP. Une liste de suppression distincte identifie les adresses IP qui ne doivent plus avoir accès à ce contenu. J'ai donc créé un algorithme pour automatiser la mise à jour du fichier `"allow_list.txt"` et supprimer ces adresses IP qui ne doivent plus y avoir accès.

## Ouvrez le fichier contenant la liste d'autorisations

Le fichier que je veux ouvrir est appelé `"allow_list.txt"`. Pour commencer, j'ai créé une variable appelée `import_file` et je lui ai assigné le nom du fichier sous la forme d'une chaîne de caractères :

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

Ensuite j'ai utilisé une instruction `with` pour ouvrir le fichier et gérer les ressources :

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

L'instruction `with` est très utile car elle garantit que le fichier est correctement fermé après son utilisation, même si des erreurs surviennent. L'argument `"r"` indique que je veux ouvrir le fichier en mode lecture.

J'ai utilisé le mot-clé `as` pour créer une variable appelée `file` qui représente le fichier ouvert. Je peux utiliser cette variable pour travailler avec le fichier pendant que je suis à l'intérieur de l'instruction `with`.

## Lire le contenu du fichier

Pour lire le contenu du fichier, j'ai utilisé la méthode `.read()` pour le convertir en chaîne. Lorsque j'utilise une fonction `open()` qui inclut l'argument `"r"` pour `"read"`, je peux appeler la fonction `.read()` dans le corps de l'instruction `with`. La méthode `.read()` convertit le fichier en chaîne et me permet de le lire. J'ai appliqué la méthode `.read()` à la variable `file` identifiée dans l'instruction `with`. Ensuite, j'ai attribué la sortie de chaîne de cette méthode à la variable `ip_addresses`.

```
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()
```

En résumé, ce code lit le contenu du fichier `"allow_list.txt"` dans un format de chaîne qui me permet d'utiliser ultérieurement la chaîne pour organiser et extraire des données dans mon programme Python.

## Convertir la chaîne en liste

Pour supprimer des adresses IP individuelles de la liste d'autorisations, j'avais besoin qu'elles soient présentées sous la forme d'une liste. J'ai ensuite utilisé la méthode `.split()` pour convertir la chaîne `ip_addresses` en liste :

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

ip_addresses = ip_addresses.split()
```

J'ai appelé la fonction `.split()` en l'ajoutant à une variable chaîne. Cela fonctionne en convertissant le contenu d'une chaîne en une liste. Le but de diviser `ip_addresses` en une liste est de faciliter la suppression des adresses IP de la liste d'autorisations. La fonction `.split()` divise le texte par des espaces en éléments de liste. Pour cet algorithme, la fonction `.split()` prend les données stockées dans la variable

`ip_addresses`, qui est une chaîne d'adresses IP séparées par un espace, et qui convertit cette chaîne en une liste d'adresses IP. Afin de stocker cette liste, je l'ai réattribuée à la variable `ip_addresses`.

## Itérer dans la liste de suppression

Un élément clé de mon algorithme consiste à parcourir les adresses IP qui sont des éléments de `remove_list`. J'ai donc incorporé une boucle `for` :

```
#Build iterative statement
#Name loop variable `element`
#Loop through `remove_list`

for element in remove_list:
    # Code pour supprimer les adresses IP
```

La boucle `for` en Python répète le code pour une séquence spécifiée. L'objectif général de la boucle `for` dans un algorithme Python comme celui-ci est d'appliquer des instructions de code spécifiques à tous les éléments d'une séquence. Afin de démarrer une boucle `for`, on utilise le mot-clé `for`. Il est ensuite suivi de la variable de la boucle `element` et du mot-clé `in`. Le mot-clé `in` indique qu'il faut itérer dans la séquence `ip_addresses` et attribuer chaque valeur à la variable de boucle `element`.

## Supprimer les adresses IP figurant sur la liste de suppression

Mon algorithme nécessite de supprimer toute adresse IP de la liste d'autorisations, `ip_addresses`, qui est également contenue dans `remove_list`. Étant donné que tous les éléments de `remove_list` figurent également dans la liste `ip_addresses` et que la liste `ip_addresses` ne contient pas de doublons, j'ai pu incorporer la méthode `.remove()` dans le corps de ma boucle `for` de la manière suivante :

```
for element in remove_list:

    #use the `.remove()` method to remove
    #elements from `ip_addresses`

    ip_addresses.remove(element)
```

Étant donné que les adresses IP de `remove_list` doivent être supprimées de la liste `ip_addresses`, j'ai appliqué la méthode `.remove()` à `ip_addresses`. J'ai passé la variable de boucle `element` comme argument afin que chaque adresse IP qui se trouvait dans `remove_list` soit supprimée de `ip_addresses`.

## Mettre à jour le fichier avec la liste révisée d'adresses IP

Comme dernière étape, j'ai dû mettre à jour le fichier de liste d'autorisations avec la liste révisée d'adresses IP. Pour ce faire, j'ai d'abord dû convertir la liste en une chaîne. J'ai utilisé la méthode `.join()` pour cela :

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

ip_addresses = " ".join(ip_addresses)
```

La méthode `.join()` combine tous les éléments d'un itérable en une chaîne. Ici elle est appliquée à une chaîne contenant des caractères qui sépareront les éléments de l'itérable une fois qu'ils sont joints en une chaîne. J'ai utilisé la méthode `.join()` pour créer une chaîne à partir de la liste `ip_addresses` afin de pouvoir l'indiquer comme argument à la méthode `.write()` lors de l'écriture dans le fichier `"allow_list.txt"`. J'ai utilisé un seul espace `(" ")` comme séparateur.

Ensuite, j'ai utilisé une autre instruction `with` et la méthode `.write()` pour mettre à jour le fichier :

```
#Build `with` statement to rewrite the original file

with open(import_file, "w") as file:

    # Rewrite the file, replacing its content with `ip_addresses`

    file.write(ip_addresses)
```

Cette fois, j'ai utilisé un deuxième argument de `"w"` avec la fonction `open()` dans mon instruction `with`. Cet argument indique que je veux ouvrir un fichier pour écrire dans son contenu. Lorsque j'utilise cet argument `"w"`, je peux appeler la fonction `.write()` dans le corps de l'instruction `with`. La fonction `.write()` écrit des données de chaîne dans un fichier spécifié et remplace tout contenu de fichier existant.

Dans ce cas, je voulais écrire la liste d'autorisations mise à jour sous forme de chaîne dans le fichier `"allow_list.txt"`. De cette façon, le contenu restreint ne sera plus accessible aux adresses IP qui ont été supprimées de la liste d'autorisations. Pour réécrire le fichier, j'ai ajouté la fonction `.write()` au fichier objet file que j'ai identifié dans l'instruction `with`. J'ai passé la variable `ip_addresses` comme argument pour spécifier que le contenu du fichier spécifié dans l'instruction `with` doit être remplacé par les données de cette variable.

## Synthèse

J'ai créé un algorithme qui supprime les adresses IP identifiées dans une variable `remove_list` du fichier `"allow_list.txt"` des adresses IP approuvées. Cet algorithme implique d'ouvrir le fichier, de le convertir en une chaîne à lire, puis de convertir cette chaîne en une liste stockée dans la variable `ip_addresses`. J'ai ensuite itéré les adresses IP de `remove_list` et supprimé chacune de ces adresses IP de la liste `ip_addresses` avec la méthode `.remove()`. Après cela, j'ai utilisé la méthode `.join()` pour convertir `ip_addresses` en une chaîne afin de pouvoir écrire dans le contenu du fichier `"allow_list.txt"` avec la liste révisée des adresses IP.