# Mise à jour d'un fichier avec un algorithme Python

## 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 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

Pour la première partie de l'algorithme, j'ai ouvert le fichier "allow\_list.txt". Tout d'abord, j'ai attribué ce nom de fichier en tant que chaîne à la variable import\_file :



Ensuite, j'ai utilisé une instruction with pour ouvrir le fichier :



Dans mon algorithme, l'instruction with est utilisée avec la fonction .open() en mode lecture pour ouvrir le fichier de liste d'autorisation dans le but de le lire. Le but de l'ouverture du fichier est de me permettre d'accéder aux adresses IP stockées dans le fichier de liste d'autorisations. Le mot-clé with permet de gérer les ressources en fermant le fichier après avoir quitté l'instruction with. Dans le code with open(import\_file, "r") as file:, la fonction open() a deux paramètres. Le premier identifie le fichier à importer, puis le second indique ce que je veux faire avec le fichier. Dans ce cas, "r" indique que je veux le lire. Le code utilise également le mot-clé as pour attribuer une variable nommée file ; file stocke la sortie de la fonction .open() pendant que je travaille dans l'instruction with.

## Lisez le contenu du fichier

Afin de 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.

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.

## Convertissez 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. Par conséquent, j'ai ensuite utilisé la méthode .split() pour convertir la chaîne ip\_addresses en liste :



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. Par défaut, la fonction .split() divise le texte par des espaces en éléments de liste. Dans 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. Pour stocker cette liste, je l'ai réattribuée à la variable ip\_addresses.

## Itération dans la liste de suppression

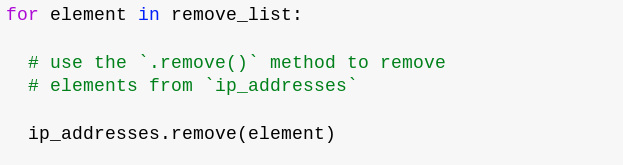
Un élément clé de mon algorithme consiste à parcourir les adresses IP qui sont des éléments de remove\_list. Pour ce faire, j'ai incorporé une boucle for :



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. Le mot-clé for démarre la boucle for. Il est suivi de la variable de boucle element et du mot-clé in. Le mot-clé in indique d'itérer dans la séquence ip\_addresses et d'attribuer chaque valeur à la variable de boucle element.

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

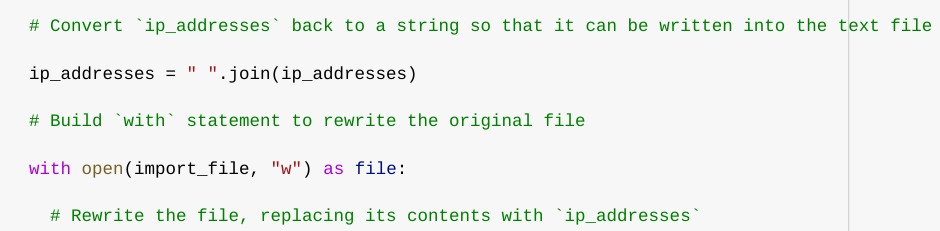
Mon algorithme nécessite de supprimer toute adresse IP de la liste d'autorisations, ip\_addresses, qui est également contenue dans remove\_list. Étant donné que tous les éléments de remove\_list figurent également dans la liste ip\_addresses et que la liste ip\_addresses ne contient pas de doublons,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



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.

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

Comme dernière étape de mon algorithme, 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 :



La méthode .join() combine tous les éléments d'un itérable en une chaîne. La méthode .join() est appliquée à une chaîne contenant des caractères qui sépareront les éléments de l'itérable une fois qu'ils sont joints en une chaîne. Dans cet algorithme, 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 :



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 impliquait 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.