Une image contenant texte, Police, capture d’écran, Bleu électrique

Description générée automatiquement

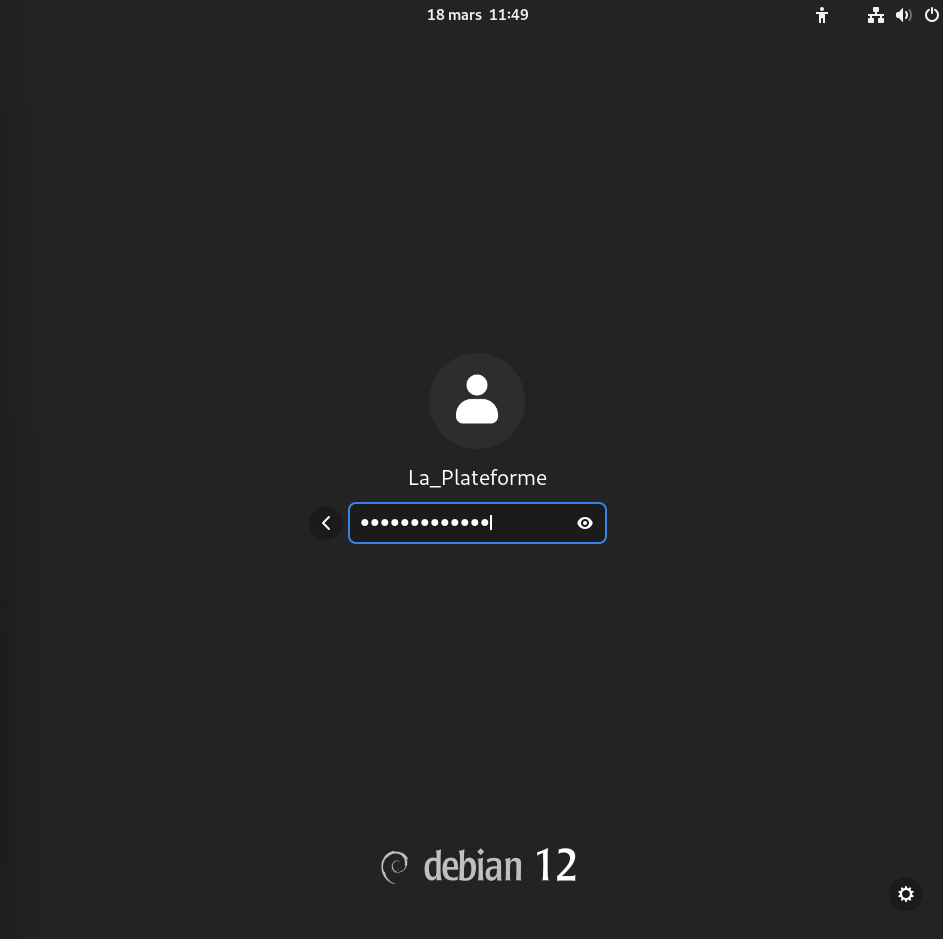
**Création d’une VM Debian :**

VMware : [Téléchargez VMware Workstation Player - VMware Customer Connect](https://customerconnect.vmware.com/fr/downloads/details?downloadGroup=WKST-PLAYER-1751&productId=1377&rPId=117013)  
  
Mansour : ARM64  
Youcef : ADM64  
Rija : AMD64

Installation de VMware:

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement



**Commandes de recherche avancée :**  
  
Commande de création du fichier « mon\_texte.txt » :  
  
Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement  
  
**Méthode 1 :**   
création d’un fichier « mon\_texte.txt » et insertion du texte « Que la force soit avec toi » :  
  
  
Déplacer le fichier dans le dossier « Bureau » :  


**Méthode 2 :**  
création d’un fichier avec le texte intégré directement dans le fichier de destination :  


**Recherche des noms de fichiers contenant le mot « force » :**  
 **Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement**

**Ajout de la commande qui permet d’isoler le fichier « mon\_texte.txt » :  
Une image contenant texte, Police, capture d’écran

Description générée automatiquement**

**Compression et décompression de fichiers :**

**Créer un repertoire nommée « Plateforme » dans le dossier « Documents » :  
Une image contenant texte, Police, capture d’écran

Description générée automatiquement**

**Déplacer le fichier « mon\_texte.txt » dans le répertoire « Plateforme » :  
**

**Dupliquer les fichiers 4 fois dans le même répertoire :  
**

**Archiver le repertoire « Plateforme » en .tar et compresser en .gzip :  
Une image contenant texte, Police, capture d’écran

Description générée automatiquement**

**Les différentes options de compression de cette étape :  
Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquement**Source :[Linux : compresser et décompresser des fichiers avec tar (gzip, bzip2, xz) – Le Crabe Info](https://lecrabeinfo.net/linux-compresser-decompresser-fichiers-dossiers-avec-tar-gzip-bzip2-xz.html#compresser-un-fichier-ou-un-dossier)

**Décompresser l’archive :  
Une image contenant texte, Police, capture d’écran

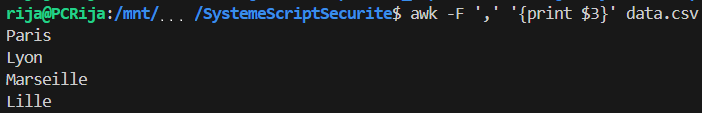
Description générée automatiquement**

**Les différentes options de compression de cette étape :**Il s’agit des mêmes commandes, sauf que l’on remplace l’option -c par -x qui indique à tar d’extraire une archive au lieu d’en créer une.

**Manipulation de texte :**

**Script Python :  
Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement**

**Commande "awk" afin d'extraire des informations spécifiques d'un fichier CSV :  
**

* **-F  ‘ , ‘ :** spécifie que le délimiteur de champ est la virgule
* **{print $3} :** spécifie à awk d’imprimer la troisième colonne de chaque ligne

Source : <https://www.malekal.com/comment-utiliser-la-commande-awk-avec-des-exemples/>

**Avant de créer mon script python je créer un environnement de travail qui me permettra d’éviter les conflits par exemple au niveau des versions de fichier, faut savoir que nous pouvons insérer du python dans un script shell.**

**A quoi sert de créer un nouvel environnement sur linux :**

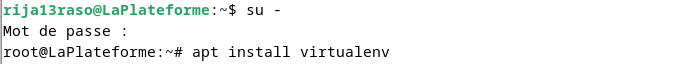
Isolation des dépendances : En créant un nouvel environnement, vous pouvez isoler les dépendances spécifiques à un projet ou une application, évitant ainsi les conflits entre différentes versions de bibliothèques ou de paquets logiciels. Cela garantit que votre application fonctionne de manière cohérente, quel que soit l'état du système global.

Développement et test : Les environnements isolés sont utiles pour le développement et le test de logiciels. Vous pouvez créer un environnement spécifique pour chaque projet, ce qui facilite la gestion des dépendances et permet une configuration cohérente pour tous les membres de l'équipe travaillant sur le projet.

Versionnement : En utilisant des environnements virtuels, vous pouvez gérer les versions des packages et des bibliothèques utilisées dans votre application. Cela signifie que vous pouvez travailler sur plusieurs projets nécessitant différentes versions de dépendances sans conflits.

Sécurité : Les environnements isolés peuvent aider à renforcer la sécurité en limitant l'accès des applications à certaines ressources système. Cela peut être particulièrement utile pour exécuter des applications tierces dont vous n'avez pas entièrement confiance.

Nettoyage et organisation : En ayant des environnements distincts pour différents projets ou applications, vous pouvez garder votre système plus propre et mieux organisé. Cela rend également plus facile la désinstallation des dépendances spécifiques à un projet lorsque vous n'en avez plus besoin.

Les commandes :  
Si l’utilisateur ne fait pas partie des utilisateur « sudoers » on ne pourras pas installer le paquet, une des manières de faire est de passer par l’utilisateur root **(attention, le fait de donner un mot de passe à root est une mauvaise manière en termes de sécurité, il aurait donc fallu que je ne définisse pas de mdp à root, cela met root en sécurité car je ne pourrais pas l’utiliser et les droits serais donné à l’utilisateur normal)**   


**Pour créer un environnement avec virtualenv, il faut taper la commande « virtualenv ‘nom de l’environnement’ » :**

**Pour vérifier si l’environnement est bien un répertoire :**

**Pour travailler dans l’environnement créé :**

**J’installe ensuite l’éditeur de texte « vim ». « vim » est un éditeur de texte puissant qui est utilisable directement dans mon terminal.**Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

**Dans mon environnement j’installe psutil :**psutil est une bibliothèque Python qui permet d'interagir avec les processus et d'obtenir des informations sur les ressources système telles que la mémoire, le CPU, le disque et le réseau. Elle est utile pour surveiller et gérer les processus en cours d'exécution, diagnostiquer les problèmes de performance, détecter les goulets d'étranglement, optimiser l'utilisation des ressources et surveiller le trafic réseau. En résumé, psutil est un outil précieux pour les administrateurs système, les développeurs d'applications et les ingénieurs de performance.

**Pour ouvrir vim, écrire « vim ‘nom du fichier’ » :  
**Cheatsheet Vim : [Vim cheatsheet (devhints.io)](https://devhints.io/vim)

**L’éditeur vim est ouvert je peux maintenant écrire mon script :  
Une image contenant texte, capture d’écran, Police, menu

Description générée automatiquement**

Ce script me permet d’écrire un script Shell avec un code en python intégré, de l’exécuter afin de créer mon fichier .csv

Ce script va créer un fichier nommé « PythonFile.py » le lire et exécuter les lignes de codes de ce fichier, le code Python a pour mission de créer mon fichier .csv

**#!/usr/bin/env python3 :** pour pouvoir exécuter le script python, nous marquons ceci dans l’entête afin que le système d’exploitation puisse déterminer quel interpréteur utiliser.

**$PY** : $PY est une variable qui stocke le nom du fichier Python à créer et à exécuter. Vous l'avez définie au début de votre script avec la ligne suivante :

**EOF :** EOF est un acronyme qui signifie "End Of File" (Fin de Fichier). Dans le contexte de scripts Shell, notamment avec des constructions telles que cat > fichier << EOF, EOF est utilisé comme un délimiteur pour indiquer la fin d'un bloc de texte ou de commandes à inclure dans un fichier ou à traiter par une commande donnée.  
Ce raccourci est utile lorsque vous avez besoin de créer rapidement un script ou un fichier avec un contenu prédéfini sans avoir à ouvrir un éditeur de texte

**Chmod +x :** « chmod » est l'abréviation de "change mode", qui est la commande principale pour modifier les permissions des fichiers sous Unix/Linux.  
+x est un argument passé à chmod pour ajouter le droit d'exécution au fichier spécifié. Le + indique l'ajout de la permission, et x représente le droit d'exécution.

**csv.DitcWrite :** csv.DictWriter est une classe fournie par le module CSV de Python qui permet d'écrire des dictionnaires dans des fichiers CSV. Elle est utilisée pour écrire des données structurées sous forme de dictionnaires dans des fichiers CSV, où chaque dictionnaire représente une ligne de données.

**writeheader :** La méthode writeheader() est utilisée avec un objet csv.DictWriter en Python pour écrire une ligne d'en-tête dans un fichier CSV. Cette ligne d'en-tête contient les noms des champs que vous avez spécifiés lors de la création de l'objet csv.DictWriter.

**writerows :** La méthode writerows() est utilisée avec un objet csv.DictWriter en Python pour écrire plusieurs lignes de données dans un fichier CSV en une seule fois. Cette méthode prend en entrée une liste de dictionnaires où chaque dictionnaire représente une ligne de données à écrire dans le fichier CSV.

**awk -F :** L'option -F dans la commande awk est utilisée pour spécifier le séparateur de champ (Field Separator). Cela permet à awk de diviser chaque ligne d'entrée en champs distincts en fonction du caractère spécifié en tant que séparateur. Par défaut, le séparateur de champ est l'espace.

Par exemple, si vous avez un fichier CSV où les champs sont séparés par des virgules, vous pouvez utiliser awk -F',' pour indiquer à awk de diviser chaque ligne en champs en utilisant la virgule comme séparateur.

**Gestion des processus**

**Explorer les processus, commencez par recenser tous ceux qui sont actifs sur votre système :  
Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquement  
Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement**

Source : [Comment Vérifier les Processus Linux en Cours d'Exécution à l'aide des Commandes ps, top, htop et atop ? - Hostinger Tutoriels](https://www.hostinger.fr/tutoriels/gerer-processus-linux-ligne-commande#Comment_verifier_les_processus_en_cours_dexecution_sous_Linux)

**Arrêter un processus :  
  
**

**Forcer l’arrêt d’un processus :  
  
**

**Arrêter un processus avec le nom exact :**Killall <nom du processus>  
Exemple : si on ecrit « killall Firefox.txt » la commande va tuer tous les processus Firefox.txt

**Arrêter un processus avec le processus contenant le nom :**pkill <nom du processus>   
Exemple :su on écrit « pkill Firefox » la commande va tuer les processus contenant le mot « Firefox » ex : Firefox.exe et Firefox.txt

Source : [TUTOS.EU : Tuer un processus sous Linux](https://www.tutos.eu/5728#:~:text=La%20commande%20kill%20permet%20d,en%20fonction%20de%20son%20pid.&text=Si%20un%20arr%C3%AAt%20propre%20ne,faut%20alors%20envoyer%20un%20SIGKILL.)

**Surveillance des ressources système :**

**Mettre en place une surveillance en temps réel de l’utilisation du CPU, de la mémoire et d’autres ressources système :**

Monitoring de l’utilisation du CPU, mémoire et autres ressources système avec « top »:  
Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Enregistrement des infos visualisé :****

**Top :** est une commande utilisée dans les systèmes Unix/Linux pour afficher des informations en temps réel sur les processus en cours d'exécution, la charge du système, l'utilisation de la mémoire, etc.

**-b :** est utilisée pour exécuter top en mode batch, c'est-à-dire pour qu'il produise une sortie unique qui peut être redirigée vers un fichier ou un autre processus.

**-n 1 :** L'option -n 1 spécifie que top doit s'exécuter une seule fois, puis se terminer.

**Scripting avancé :**

Commande dans mon script qui me permet d’archiver + compresser mon répertoire « Plateforme », ajoutant la date et l’heure sur le nom du dossier et le déplace dans un dossier nommé « Save » :Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Source de l’inspiration : [[résolu] copie du meme fichier avec date et heure / Terminal, scripts et ligne de commande / Forum Ubuntu-fr.org](https://forum.ubuntu-fr.org/viewtopic.php?id=2021517)

**Activer cron :**  


**Commande Cron qui active mon script toutes les minutes et enregistre l’historique des actions effectué dans un fichier « sauvegarde.log », je l’ajoute tout en bas de l’affichage cron :**  
  
J’ai mis 5 étoiles pour qu’il m’active la commande toutes les minutes

\*= minutes \*= heures \*= jour du mois \*= mois de l’année \*= année  
  
**Afin que cron prend en compte les modifications, faire un restart :**  


**Pour arrêter cron :**  
  
  
Sources :  
-[Lancer un script bash avec crontab - Communauté Fedora-Fr](https://forums.fedora-fr.org/d/43303-lancer-un-script-bash-avec-crontab/6)  
-[Cron et crontab : le planificateur de tâches ! - Wiki - Wiki (linuxtricks.fr)](https://www.linuxtricks.fr/wiki/cron-et-crontab-le-planificateur-de-taches)

**Automatisation des mises à jour logicielles :**

**Créez un script automatisant la recherche de mise à jour des logiciels existants sur le système.**

**Mon script :  
Une image contenant texte, capture d’écran, Police

Description générée automatiquement**Source : [Confirmation prompt yes/no in Bash | how.wtf](https://how.wtf/confirmation-prompt-yes/no-in-bash.html)

**read :** est utilisé pour lire une entrée de l’utilisateur depuis le clavier

**-r :** est utilisée pour empêcher l'interprétation des séquences d'échappement dans l'entrée de l'utilisateur. Par défaut, certaines séquences spéciales telles que ‘\’ sont interprétées par le Shell.  
Exemple de séquence d’échappement avec ‘\n’, si je marque une phrase avec dedans un ‘\n’ et j’utilise read <sans -r> la séquence d’échappement ‘\n’ sera interprété comme un saut de ligne. En revanche, si je veux qu’elle ne soit pas interprétée, là j’ajoute read -r.

**-p :** est utilisée pour afficher un message dans l’invite de commande de l'utilisateur avant de lire son entrée.

**-n 1 :** L'option -n suivie d'un nombre dans la commande read spécifie le nombre de caractères à lire à partir de l'entrée standard. Dans le cas de -n 1, cela signifie que read lit un seul caractère de l'entrée standard.

**Gestion des dépendances logicielles :  
  
Mon script :  
Une image contenant texte, capture d’écran, document, menu

Description générée automatiquement**

**Dans mon fichier sudoers, je modifie mon utilisateur afin de lui attribuer les droits nécessaires à l’executer du script :**

**Attention la commande « curl » ne fonctionnera pas s’il n’est pas installé :**

**Pour executer phpMyAdmin choisir apache2 pour la configuration automatique :Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

**Je défini un mot de pass à l’administrateur « root » :  
Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement**

**Création de Stockage de la configuration de phpMyAdmin en utilisant dbconfig-common :  
Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

**Je laisse la configuration par défaut :  
Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

**Je choisis un greffon(plugin) pour l’authentification avec MySQL, pour plus de protection, je choisi le greffon sha256\_password :  
Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement**

**Je définie le nom de ma base de donnée MySQL (LAPlateforme\_ pour ma part) :  
Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

**Je définie le nom d’identifiant de ma base de données :  
Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement**

**Je définie un Mdp pour ma base de données (rija13LAP pour moi):  
Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement**

**Je laisse le nom de l’administrateur de la base de données à « root » :  
Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

**Je définie le Mdp de l’administrateur de la base de donnée :**à ce niveau, une erreur « ERROR 2002 (HY000): Can't connect to local server through socket

│ '/run/mysqld/mysqld.sock' » est affiché, à régler plus tard car j’ai une erreur.

**Vérifier le statut de apache2 :  
Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement**

**Vérifier le statut de mariadb :  
Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

Source installation de apache2 : [Installing Apache on Debian 12: A Step-by-Step Guide | Reintech media](https://reintech.io/blog/installing-apache-on-debian-12-step-by-step-guide)

Source installation de Nginx : [Setup Nginx on Debian 12 and Host a Simple Static Website | by May Rain | Medium](https://medium.com/@mayrain1923/setup-nginx-on-debian-12-and-host-a-simple-static-website-f832e6761e63)

Source installation phpMyAdmin : [Comment installer phpMyAdmin sur Ubuntu ou Debian - malekal.com](https://www.malekal.com/installer-phpmyadmin-ubuntu-debian/)

Source installation de MariaDB (Dans Debian, MySQL existe sous la forme de MariaDB en version 12) : [Installer MySQL (MariaDB) sur Debian 12 - malekal.com](https://www.malekal.com/installer-mysql-mariadb-debian/)

Source installation de Node.js et NPM : [Installing Node.js and NPM on Debian 12 | Reintech media](https://reintech.io/blog/installing-nodejs-npm-debian-12)

Source installation de git : [How to Install Git on Debian 12 Step-by-Step (linuxbuzz.com)](https://www.linuxbuzz.com/how-to-install-git-on-debian/)

**Sécuriser ses scripts :**

1. **Eviter autant que possible de mettre des « -y » dans son script, qui permet de toujours accepter les propositions.**
2. **Ajouter des autorisations spécifiques à l’utilisateur comme fait précédemment, l’utilisateur, dans cet exemple j’autorise l’utilisateur à exécuter les commandes sudo nécessaire à l’exécution du script :**
3. **Les commandes du script nécessitent des privilèges sudo, donc un utilisateur non sudo ne pourras pas utiliser le script.**
4. **Dans mon script, Le script demande à l'utilisateur de sélectionner un serveur Web en entrant un numéro, mais il ne valide pas cette entrée. Cela pourrait permettre à un utilisateur malveillant d'entrer une valeur non prévue et provoquer un comportement non souhaité du script.  
   Voici mon script modifié :  
   Une image contenant texte, capture d’écran, menu, Police

   Description générée automatiquement**
5. **L’utilisation de la commande curl pour installer Node.js est risqué car le script téléchargé pourrait être malveillant.  
   L’alternative c’est de d’abord de télécharger le script et le vérifier visuellement :  
     
   Si le script est safe alors on peut l’exécuter :  
   **
6. **J’ai ajouté l’option « set -e » qui permet de fermer le script si une erreur est détecté, l’utilisateur malveillant ne pourras pas continuer son action**

**Utilisation d’API Web dans un script :**

**Script qui me permet d’afficher les noms d’utilisateur disponible sur JsonPlaceHolder et me propose de les extraires sous fichier json. Il enregistre les différentes requêtes vers l’API ainsi que les réponses envoyées par l’API, capture ce qu’écrit l’utilisateur au clavier et l’enregistre dans un fichier api.log :**

**Une image contenant texte, capture d’écran

Description générée automatiquement**

**set -e :** avec l’option -e,le script va se fermer lorsqu’une erreur est détectée (-errexit).  
Source : [set -e, -u, -o, -x pipefail explanation (github.com)](https://gist.github.com/vncsna/64825d5609c146e80de8b1fd623011ca)

**cURL (curl):** est conçue pour fonctionner comme un moyen de vérifier la connectivité aux URL et comme un excellent outil pour transférer des données.  
Source : [Comment Utiliser la Commande Curl Sous Linux ? (hostinger.fr)](https://www.hostinger.fr/tutoriels/comment-utiliser-la-commande-curl-sous-linux#:~:text=La%20commande%20cURL%20est%20disponible,outil%20pour%20transf%C3%A9rer%20des%20donn%C3%A9es.)

**tee -a : l**a commande tee est utilisée pour lire depuis l'entrée standard et écrire à la fois sur la sortie standard et dans des fichiers spécifiés. L'option -a dans tee signifie "append" (ajouter). Cela signifie que tee ajoute les données à la fin du fichier plutôt que de les écraser.  
Source 1 : [Outils de base de traitement du texte - Linux Administration (goffinet.org)](https://linux.goffinet.org/administration/traitement-du-texte/outils-de-base-traitement-du-texte/)  
Source 2 : [How to use the Linux Tee Command (vegastack.com)](https://vegastack.com/tutorials/how-to-use-the-linux-tee-command/)

**-z :** vérifie si ma chaîne de caractère est vide.  
Source : [[Bash] Test chaîne vide ou non – Unicoda](https://www.unicoda.com/?p=3758#:~:text=Pour%20v%C3%A9rifier%20qu'une%20variable,cha%C3%AEne%20n'est%20pas%20z%C3%A9ro.)

**jq -r :** jq permet d’extraire des informations d’un flux JSON. Il est également capable de générer un flux JSON en sortie. L’option -r signifie "raw output" (sortie brute). Lorsqu'elle est utilisée avec jq, elle indique à jq de produire une sortie sans les délimiteurs de chaîne de caractères. Cela signifie que les guillemets autour des chaînes de caractères sont supprimés et les caractères d'échappement sont interprétés littéralement.   
Source 1 : [jq - manipuler du JSON en shell - Le weblogue de SeB (lecacheur.com)](https://blog.lecacheur.com/2016/02/16/jq-manipuler-du-json-en-shell/)  
Source 2 : [How To Transform JSON Data with jq | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-transform-json-data-with-jq)

**2>> :** il était possible de faire 2>> pour rediriger les erreurs à la fin d'un fichier d'erreurs.  
Source : [Linux/Shell: 2>, 2>> et 2>&1 : rediriger les erreurs | by Boris Rose | Medium](https://medium.com/@boris.alexandre.rose/linux-shell-2-2-et-2-1-rediriger-les-erreurs-1db204f0d2c1)

**.[].name :** .[] permet de sélectionner tous les objets contenus dans le tableau JSON. En ajoutant .name après .[], j’extrait la valeur de la clé "name" de chaque objet dans le tableau. Cela signifie que pour chaque objet du tableau, jq extraira la valeur associée à la clé "name".  
Source : [How To Transform JSON Data with jq | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-transform-json-data-with-jq)

**Mon script qui me permet d’effectuer des recherches sur youtube à partir d’une phrase, la recherche me retournera en résultat un titre avec le lien de la vidéo :**

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**