

COURS POUR DÉBUTANT

Apprendre Linux

Tout ce qu'il faut savoir pour comprendre, installer et utiliser Linux au quotidien — sans connaissances préalables.



```
utilisateur@linux:~$ whoami
un futur utilisateur de Linux
utilisateur@linux:~$ echo "Commençons !"
Commençons !
utilisateur@linux:~$ _
```

Sommaire

— À qui s'adresse ce cours & comment l'utiliser

1 Découvrir Linux : c'est quoi, et pourquoi ?

2 Choisir et essayer une distribution

3 Premiers pas avec le bureau

4 Le terminal : la ligne de commande

5 Naviguer dans le système de fichiers

6 Manipuler fichiers et dossiers

7 Utilisateurs, droits et permissions

8 Installer et gérer des logiciels

9 Processus, services et état du système

10 Éditer du texte dans le terminal

11 Réseau et connexions de base

12 Bonnes pratiques et sécurité

A Aide-mémoire des commandes essentielles

B Glossaire des termes clés

C Pour aller plus loin

À qui s'adresse ce cours

Ce cours part de zéro. Vous n'avez besoin d'aucune connaissance technique : si vous savez allumer un ordinateur et utiliser un navigateur, vous avez tout ce qu'il faut pour commencer.

Ce que vous saurez faire à la fin

- Comprendre **ce qu'est Linux** et pourquoi il est utilisé partout (téléphones, serveurs, voitures, supercalculateurs).
- **Choisir une distribution** adaptée à vos besoins et l'essayer sans risque.
- Vous repérer dans le **bureau** et dans le **système de fichiers**.
- Utiliser le **terminal** avec aisance pour les tâches courantes.
- **Installer des logiciels**, gérer les **permissions**, surveiller le **système** et rester **en sécurité**.

Comment lire ce guide

Chaque module est progressif : il s'appuie sur le précédent. Le mieux est de le suivre dans l'ordre, en gardant un ordinateur (ou une machine virtuelle) sous Linux à côté de vous pour tester. Vous rencontrerez régulièrement ces encadrés :

Définition

Explique un mot ou un concept important en langage simple.

À retenir

L'essentiel à mémoriser absolument dans la section.

Astuce

Un raccourci ou une bonne pratique qui fait gagner du temps.

Attention

Un piège fréquent ou une manipulation à risque.

Exercice

Une petite mise en pratique à faire vous-même.

Convention typographique. Les commandes à taper apparaissent en *police machine à écrire* . Dans les blocs noirs façon terminal, le symbole `$` représente l'invite : *ne le tapez pas*, il est déjà affiché par le système. Les touches du clavier sont notées ainsi : `Entrée`, `Ctrl` + `C`.

Découvrir Linux : c'est quoi, et pourquoi ?

Avant de toucher au clavier, comprenons ce qu'est réellement Linux. Cette base rend tout le reste beaucoup plus clair.

Un système d'exploitation, comme Windows ou macOS

Un **système d'exploitation** (en anglais *operating system*, ou OS) est le logiciel principal de votre ordinateur. C'est lui qui fait le lien entre vous, les programmes et le matériel (écran, clavier, disque dur, réseau...). Windows et macOS sont des systèmes d'exploitation. **Linux en est un autre** — gratuit, libre, et utilisable par tout le monde.

■ Définition — Le « noyau » Linux

Au sens strict, **Linux** est seulement le *noyau* (en anglais *kernel*) : le cœur du système qui parle directement au matériel. Créé en 1991 par Linus Torvalds, ce noyau est aujourd'hui développé par des milliers de contributeurs dans le monde. Autour de ce noyau, on ajoute des programmes pour obtenir un système complet et utilisable : c'est ce qu'on appelle une **distribution**.

Linux est « libre » et « open source »

La grande particularité de Linux est d'être un **logiciel libre** et **open source**. Concrètement :

- **Open source** : le code source (les « plans » du logiciel) est public. N'importe qui peut le lire, l'étudier, le corriger.
- **Libre** : vous avez le droit de l'utiliser, de le copier, de le modifier et de le partager, le plus souvent **gratuitement**.

C'est l'inverse d'un logiciel « propriétaire » comme Windows, dont le code est secret et l'usage payant et encadré par une licence.

Pourquoi Linux est-il si répandu ?

Même si Linux est minoritaire sur les ordinateurs de bureau, il est **omniprésent** ailleurs :

Domaine	Présence de Linux
Téléphones	Android est construit sur le noyau Linux
Serveurs & sites web	La très grande majorité des serveurs d'Internet tournent sous Linux
Cloud	Les centres de données (Amazon, Google, etc.) reposent massivement sur Linux
Supercalculateurs	La quasi-totalité des plus puissants au monde utilisent Linux
Objets connectés	Box internet, téléviseurs, voitures, distributeurs...

Les avantages concrets pour vous

- **Gratuit** : aucune licence à acheter.
- **Léger** : redonne vie à des ordinateurs anciens ou modestes.
- **Sûr** : peu ciblé par les virus, et conçu autour d'un solide système de permissions.
- **Stable** : peut fonctionner des mois sans redémarrer.
- **Personnalisable** : vous choisissez l'apparence et le fonctionnement, dans le moindre détail.
- **Respectueux** : pas de publicité ni de collecte de données imposée.

⚠ Les limites à connaître

Certains logiciels (la suite Adobe complète, quelques jeux avec anti-triche, certains outils métier) ne fonctionnent pas nativement sous Linux. Il existe souvent des alternatives ou des solutions de contournement, mais c'est le principal frein selon votre usage. Vérifiez vos logiciels indispensables avant de basculer entièrement.

🎯 À retenir

- **Linux** = le noyau, le cœur du système.
- Une **distribution** = le noyau + des programmes = un système complet prêt à l'emploi.
- Linux est **libre, gratuit et open source**, et fait tourner une immense partie d'Internet.

Choisir et essayer une distribution

Il n'existe pas « un » Linux mais des centaines de **distributions**. Pas de panique : pour débiter, quelques choix sûrs suffisent.

Définition — Distribution (ou « distro »)

Une **distribution** est un assemblage tout prêt : le noyau Linux, un environnement de bureau, un navigateur, une logithèque, des outils... le tout configuré pour fonctionner ensemble. Choisir une distribution, c'est un peu comme choisir une « marque » de voiture : le moteur (le noyau) est commun, mais l'habillage et le confort varient.

Les meilleures distributions pour débiter (2026)

Distribution	Pour qui ?	Points forts
Linux Mint	Le débutant venant de Windows	Interface très familière, simple, stable, idéale pour une première fois
Ubuntu	La valeur sûre, prépa carrière IT	Énorme communauté, documentation immense, versions « LTS » supportées 5 ans
Zorin OS	Transition en douceur depuis Windows/macOS	Apparence configurable pour ressembler à ce que vous connaissez déjà
Pop!_OS	Productivité & jeu vidéo	Excellente gestion des cartes graphiques NVIDIA, organisation des fenêtres

Astuce — Quel choix pour un vrai débutant ?

Si vous hésitez : commencez par **Linux Mint** (édition « Cinnamon ») pour le confort, ou **Ubuntu** si vous voulez la plus grande communauté d'entraide en français. Vous pourrez toujours changer plus tard, ce n'est pas un engagement définitif.

Définition — Version « LTS »

LTS signifie *Long Term Support* (support à long terme). Une version LTS reçoit des mises à jour de sécurité pendant environ 5 ans. Pour débiter, choisissez toujours une version LTS : plus stable, moins de surprises.

Trois façons d'essayer Linux sans risque

1. La clé USB « live » — pour tester sans rien installer

Vous copiez la distribution sur une clé USB et vous démarrez l'ordinateur dessus. Linux fonctionne **directement depuis la clé**, sans toucher à votre disque dur ni à Windows. C'est parfait pour découvrir.

1. Téléchargez le fichier **.iso** sur le site officiel de la distribution.
2. Avec un logiciel comme **balenaEtcher** ou **Rufus**, écrivez cette image sur une clé USB (8 Go minimum).
3. Redémarrez l'ordinateur en demandant le démarrage sur la clé (touche **F12**, **F2** ou **Échap** selon les machines).
4. Choisissez « Essayer » (et non « Installer »).

2. La machine virtuelle — Linux dans une fenêtre

Avec un logiciel comme **VirtualBox** (gratuit), vous faites tourner Linux **dans une fenêtre** sur votre Windows ou macOS actuel, comme un programme. Idéal pour apprendre tranquillement sans rien modifier sur votre vrai système.

3. Le double démarrage (« dual boot »)

Vous installez Linux **à côté** de Windows. Au démarrage, un menu vous laisse choisir lequel lancer. Plus engageant, à réserver une fois que vous êtes à l'aise.

Avant toute installation

Sauvegardez vos documents importants. Modifier le disque dur comporte toujours un petit risque. Une copie de vos fichiers sur un disque externe ou dans le cloud vous met à l'abri.

Exercice

Rendez-vous sur le site officiel de Linux Mint ou d'Ubuntu, repérez le bouton de téléchargement, et identifiez la version **LTS**. Vous n'êtes pas obligé de télécharger : l'objectif est juste de savoir où trouver l'information fiable (toujours le *site officiel*, jamais un site tiers).

À retenir

Une **distribution** = une « saveur » de Linux. Pour débiter : **Mint** ou **Ubuntu**, en version **LTS**. Testez d'abord en **live USB** ou en **machine virtuelle** avant d'installer pour de bon.

Premiers pas avec le bureau

Bonne nouvelle : Linux moderne se pilote à la souris, comme Windows ou macOS. Le terminal viendra plus tard et restera optionnel pour bien des tâches.

Définition — Environnement de bureau

L'**environnement de bureau** (en anglais *desktop environment*) est la partie graphique : le fond d'écran, le menu, la barre des tâches, les fenêtres. Il en existe plusieurs (GNOME, Cinnamon, KDE Plasma, XFCE...). C'est souvent lui qui change l'« allure » d'une distribution.

Les éléments que vous retrouverez partout

- **Le menu des applications** : l'équivalent du menu Démarrer, pour lancer vos logiciels.
- **La barre des tâches / le dock** : affiche les fenêtres ouvertes et les programmes favoris.
- **La zone de notification** : heure, réseau (Wi-Fi), son, batterie.
- **Le gestionnaire de fichiers** : pour parcourir vos dossiers à la souris (l'équivalent de l'Explorateur Windows ou du Finder).
- **Les Paramètres** : pour le réseau, l'affichage, le son, les comptes, etc.

Les premières choses à faire après l'installation

1. **Se connecter au Wi-Fi** via l'icône réseau.
2. **Lancer les mises à jour** : un outil (« Gestionnaire de mises à jour ») vous propose les dernières versions et correctifs de sécurité. Faites-le dès le début.
3. **Régler la langue et le clavier** dans les Paramètres si nécessaire.
4. **Explorer la logithèque** pour installer vos premiers programmes (voir Module 8).

Astuce — Vos logiciels habituels ont des équivalents

Sous Windows / macOS	Équivalent libre sous Linux
Microsoft Office	LibreOffice (compatible avec les fichiers .docx, .xlsx)
Photoshop	GIMP (retouche) ou Krita (dessin)
Outlook	Thunderbird
Edge / Safari	Firefox, Chromium
Lecteur multimédia	VLC

🎯 À retenir

Le bureau Linux s'utilise à la souris, intuitivement. Vos trois réflexes après installation : **Wi-Fi, mises à jour, logithèque**. La plupart de vos logiciels ont un équivalent libre.

Le terminal : la ligne de commande

Le terminal effraie au début, mais c'est juste une façon d'écrire des ordres à l'ordinateur au lieu de cliquer. C'est rapide, précis, et c'est la grande force de Linux.

■ Définition — Terminal & Shell

Le **terminal** est la fenêtre noire où vous tapez du texte. Le **shell** est le programme qui *interprète* ce que vous tapez et exécute vos ordres. Le shell le plus courant s'appelle **Bash**. En résumé : vous écrivez une commande dans le terminal, le shell la comprend et la lance.

Pourquoi apprendre le terminal ?

- **Rapidité** : une commande peut remplacer dix clics.
- **Précision** : vous dites exactement ce que vous voulez.
- **Universalité** : les mêmes commandes marchent sur presque toutes les distributions et sur les serveurs.
- **Aide en ligne** : la quasi-totalité des tutoriels donnent des commandes à copier-coller.

Ouvrir un terminal

Cherchez « Terminal » dans le menu des applications, ou utilisez le raccourci **Ctrl** + **Alt** + **T** (très répandu).

Anatomie de l'invite de commande

À l'ouverture, vous voyez quelque chose comme ceci :

```
marie@portable:~$
```

Décortiquons cette ligne, qu'on appelle l'**invite** (en anglais *prompt*) :

Élément	Signification
marie	Le nom de l'utilisateur connecté
@	Séparateur « chez »
portable	Le nom de la machine
~	Le dossier où vous vous trouvez (~ = votre dossier personnel)
\$	Vous êtes un utilisateur normal (un # indiquerait l'administrateur)

Structure d'une commande

Une commande suit presque toujours le même schéma :

```
commande -options arguments
```

Par exemple :

```
$ ls -l Documents
```

- `ls` est la **commande** (lister le contenu).
- `-l` est une **option** (ici : affichage détaillé, en liste). Les options commencent par un tiret.
- `Documents` est l'**argument** (sur quoi la commande agit).

Astuces qui changent la vie

- **Auto-complétion** : tapez les premières lettres puis appuyez sur `Tab`. Le shell complète tout seul les noms de commandes et de fichiers. *Le réflexe n°1 à prendre.*
- **Historique** : les flèches `↑` et `↓` rappellent les commandes précédentes.
- **Annuler** : `Ctrl` + `C` interrompt une commande en cours.
- **Nettoyer l'écran** : la commande `clear` (ou `Ctrl` + `L`).

La commande la plus importante : l'aide

Vous ne saurez jamais tout par cœur, et ce n'est pas le but. Apprenez plutôt à **trouver l'aide** :

```
# Affiche le manuel complet d'une commande (q pour quitter)
$ man ls

# Affiche une aide rapide directement intégrée
$ ls --help
```

Le terminal ne pardonne pas les fautes de frappe

Une espace, une majuscule ou un caractère oublié, et la commande échoue (ou pire, agit sur autre chose). Linux **distingue les majuscules des minuscules** : `Documents` et `documents` sont deux noms différents. Relisez avant d'appuyer sur `Entrée`.

Exercice

Ouvrez un terminal et tapez successivement : `whoami` (qui suis-je ?), `date` (date et heure), puis `clear`. Enfin, tapez `ec` puis appuyez sur `Tab` pour voir l'auto-complétion proposer `echo`, et lancez `echo Bonjour`.

🎯 À retenir

Le terminal exécute des commandes structurées en **commande + options + arguments**. Vos meilleurs amis : la touche **Tab** (complétion), la flèche **↑** (historique) et **man** (aide). Linux est **sensible à la casse**.

Naviguer dans le système de fichiers

Sous Linux, l'organisation des fichiers diffère de Windows : pas de « lecteur C: ». Tout part d'un point unique. Comprendre cette carte, c'est ne plus jamais être perdu.

Une seule racine : l'arborescence

Tout commence à la **racine**, notée `/` (une simple barre oblique). À partir d'elle, les dossiers se ramifient comme les branches d'un arbre. Il n'y a pas de C:, D:, etc. : même une clé USB apparaît comme un dossier quelque part sous `/`.

```

/
├─ bin      # programmes essentiels
├─ boot    # fichiers de démarrage
├─ dev     # périphériques (disques, clés USB...)
├─ etc     # fichiers de configuration du système
├─ home    # les dossiers personnels des utilisateurs
│   └─ marie # VOTRE espace : documents, images...
├─ root    # dossier personnel de l'administrateur
├─ tmp     # fichiers temporaires
├─ usr     # applications et leurs ressources
└─ var     # données variables (journaux, etc.)

```

■ Définition — Votre dossier personnel

Le dossier `/home/votre_nom` est **votre maison** : il contient vos documents, images, téléchargements, et vos réglages personnels. Le raccourci `~` (tilde) le désigne toujours. C'est là que vous travaillez 99 % du temps.

Les trois commandes pour se déplacer

Commande	Rôle	Exemple
<code>pwd</code>	« Où suis-je ? » — affiche le dossier courant	<code>pwd</code>
<code>ls</code>	Lister le contenu du dossier	<code>ls</code>
<code>cd</code>	Changer de dossier (<i>change directory</i>)	<code>cd Documents</code>

```
$ pwd
/home/marie
$ ls
Documents Images Musique Téléchargements
$ cd Documents
$ pwd
/home/marie/Documents
```

Quelques déplacements bien utiles

```
$ cd ..          # remonter d'un niveau (dossier parent)
$ cd ~          # revenir à mon dossier personnel
$ cd /          # aller à la racine
$ cd -          # revenir au dossier précédent
```

Chemin absolu ou chemin relatif ?

■ Définition — Absolu vs relatif

Un **chemin absolu** part toujours de la racine `/` : il est complet et fonctionne d'où que vous soyez.

Exemple : `/home/marie/Documents`.

Un **chemin relatif** part de l'endroit où vous êtes *actuellement*. Exemple : si vous êtes dans `/home/marie`, écrire `Documents` suffit.

Voir plus de détails avec les options

```
$ ls -l          # liste détaillée (droits, taille, date)
$ ls -a          # montre aussi les fichiers cachés (qui commencent par un point)
$ ls -lh         # tailles lisibles (Ko, Mo, Go)
$ ls -la         # on combine les options !
```

💡 Astuce — Les fichiers cachés

Un fichier dont le nom commence par un point (ex. `.bashrc`) est **caché** par défaut. Ce ne sont généralement pas des secrets, mais des fichiers de configuration qu'on évite d'afficher pour ne pas encombrer. `ls -a` les révèle.

Exercice

Depuis votre dossier personnel : 1) affichez où vous êtes avec `pwd` ; 2) listez le contenu détaillé avec `ls -lh` ; 3) entrez dans `Téléchargements` ; 4) revenez en arrière avec `cd ..` ; 5) retournez directement à la maison avec `cd ~`.

À retenir

Tout part de la racine `/`. Votre espace de travail est `~` (= `/home/votre_nom`). Trois commandes pour vous repérer : `pwd` (où ?), `ls` (quoi ?), `cd` (aller).

Manipuler fichiers et dossiers

Créer, copier, déplacer, renommer, supprimer, lire : voici la boîte à outils du quotidien. Quelques commandes suffisent pour tout faire.

Créer

```
$ mkdir projet           # crée un dossier "projet"
$ mkdir -p a/b/c         # crée toute une arborescence d'un coup
$ touch notes.txt       # crée un fichier vide "notes.txt"
```

Copier, déplacer, renommer

```
$ cp notes.txt sauvegarde.txt # copier un fichier
$ cp -r projet projet_copie   # copier un dossier (-r = récursif)
$ mv notes.txt Documents/     # déplacer dans un autre dossier
$ mv notes.txt memo.txt       # renommer (déplacer = renommer !)
```

Astuce

Sous Linux, **déplacer et renommer sont la même opération** : la commande `mv` (*move*). Pour renommer, vous « déplacez » le fichier vers un nouveau nom dans le même dossier.

Supprimer — avec prudence

```
$ rm fichier.txt        # supprime un fichier
$ rm -r dossier         # supprime un dossier et son contenu
$ rmdir dossier_vide   # supprime un dossier (seulement s'il est vide)
```

Danger absolu : pas de corbeille !

Dans le terminal, `rm` supprime **définitivement et sans confirmation**. Il n'y a pas de corbeille. Vérifiez toujours *deux fois* ce que vous effacez. Et ne tapez **jamais** `rm -rf /` : cela tenterait d'effacer tout le système. En cas de doute, ajoutez l'option `-i` (`rm -i fichier`) pour qu'une confirmation soit demandée.

Lire et afficher le contenu d'un fichier texte

Commande	Usage
<code>cat fichier</code>	Affiche tout le contenu d'un coup (pour les petits fichiers)
<code>less fichier</code>	Affiche page par page (flèches pour défiler, <code>q</code> pour quitter)
<code>head fichier</code>	Affiche les 10 premières lignes
<code>tail fichier</code>	Affiche les 10 dernières lignes
<code>wc -l fichier</code>	Compte le nombre de lignes

Retrouver un fichier

```
$ find . -name "*.txt" # cherche tous les .txt à partir d'ici
$ find ~ -name "rapport*" # cherche dans la maison ce qui commence par "rapport"
```

Définition — Le caractère « * » (joker)

L'astérisque `*` remplace « n'importe quelle suite de caractères ». Ainsi `*.txt` désigne **tous** les fichiers se terminant par `.txt`. C'est très pratique pour agir sur plusieurs fichiers à la fois : `rm *.tmp` supprime tous les fichiers temporaires.

Exercice complet

1) Créez un dossier `atelier` et entrez-y. 2) Créez trois fichiers : `touch a.txt b.txt c.txt`. 3) Listez-les avec `ls`. 4) Copiez `a.txt` en `a_copie.txt`. 5) Renommez `b.txt` en `notes.txt`. 6) Supprimez `c.txt` avec `rm -i c.txt` (confirmez par `y`). 7) Vérifiez le résultat avec `ls`.

À retenir

`mkdir` / `touch` (créer), `cp` (copier), `mv` (déplacer/renommer), `rm` (supprimer — **sans corbeille !**), `cat` / `less` (lire), `find` (chercher). L'option `-r` sert pour les dossiers, `-i` pour demander confirmation.

Utilisateurs, droits et permissions

La sécurité de Linux repose sur une idée simple : chaque fichier a un propriétaire et des droits précis. Comprendre ce système, c'est comprendre pourquoi Linux est robuste.

L'utilisateur tout-puissant : root

■ Définition — root & sudo

root est le super-administrateur : il a *tous* les droits, y compris celui de tout casser. On évite donc de l'utiliser directement.

sudo (« *substitute user do* ») permet d'exécuter *une seule commande* avec les droits d'administrateur, en confirmant par votre mot de passe. C'est la méthode sûre et recommandée.

```
$ sudo apt update          # exécute la mise à jour en tant qu'admin
[sudo] Mot de passe de marie : # tapez votre mot de passe
```

⚠ Le mot de passe « invisible »

Quand vous tapez votre mot de passe après `sudo`, **rien ne s'affiche** : ni étoiles, ni points. C'est normal et voulu, par sécurité. Tapez votre mot de passe « à l'aveugle » puis appuyez sur **Entrée**.

Lire les permissions d'un fichier

La commande `ls -l` affiche les droits sous une forme codée. Décortiquons :

```
$ ls -l
-rw-r--r-- 1 marie marie 2048 12 mai rapport.txt
```

Le premier bloc, `-rw-r--r--`, se lit en quatre parties :

Partie	Exemple	Signification
Type	-	- = fichier, d = dossier (<i>directory</i>)
Propriétaire	rw-	Droits de l'utilisateur propriétaire
Groupe	r--	Droits des membres du groupe
Autres	r--	Droits de tous les autres utilisateurs

■ Les trois droits : r, w, x

- **r** = **read** (lire) — consulter le fichier.
 - **w** = **write** (écrire) — modifier ou supprimer.
 - **x** = **execute** (exécuter) — lancer le fichier (s'il s'agit d'un programme), ou entrer dans un dossier.
- Un tiret **-** à la place d'une lettre signifie « droit absent ».

Dans notre exemple `-rw-r--r--` : le propriétaire peut lire et écrire (`rw-`), tandis que le groupe et les autres peuvent seulement lire (`r--`).

Modifier les permissions et le propriétaire

```
$ chmod +x script.sh      # rend un fichier exécutable
$ chmod -w fichier.txt    # retire le droit d'écriture
$ sudo chown marie fichier # change le propriétaire (nécessite sudo)
```

💡 Astuce — La notation chiffrée

On voit souvent `chmod 755` ou `chmod 644`. Chaque chiffre est une somme : lecture = 4, écriture = 2, exécution = 1. Ainsi `7` = 4+2+1 (tous les droits), `6` = 4+2 (lire + écrire), `5` = 4+1 (lire + exécuter). Les trois chiffres correspondent à propriétaire / groupe / autres. Donc `644` = propriétaire lit+écrit, les autres lisent seulement.

🎯 À retenir

On n'utilise pas **root** directement : on préfixe avec **sudo** (mot de passe invisible, c'est normal). Chaque fichier a des droits **r** (lire), **w** (écrire), **x** (exécuter) pour trois catégories : propriétaire, groupe, autres. On les modifie avec `chmod`.

Installer et gérer des logiciels

Oubliez la chasse aux fichiers d'installation sur des sites douteux. Sous Linux, on installe des logiciels depuis des **dépôts** sûrs, en une commande ou un clic.

■ Définition — Paquet, dépôt, gestionnaire de paquets

- Un **paquet** est un logiciel prêt à installer, avec tout ce dont il a besoin.
- Un **dépôt** (en anglais *repository*) est une bibliothèque officielle en ligne, contenant des milliers de paquets vérifiés.
- Le **gestionnaire de paquets** est l'outil qui télécharge, installe et met à jour ces paquets pour vous, en gérant automatiquement les dépendances.

💡 C'est comme un magasin d'applications

Pensez au gestionnaire de paquets comme à un « App Store » officiel et gratuit, intégré au système. C'est **plus sûr** que de télécharger un installateur au hasard sur le web : les paquets des dépôts officiels sont vérifiés.

La commande dépend de votre distribution

Famille **Debian / Ubuntu / Mint** → l'outil s'appelle `apt`. C'est de loin le plus courant pour les débutants, donc concentrons-nous dessus :

```
$ sudo apt update           # rafraîchit la liste des logiciels disponibles
$ sudo apt upgrade         # met à jour tous les logiciels installés
$ sudo apt install vlc     # installe le lecteur VLC
$ sudo apt remove vlc     # désinstalle VLC
$ apt search navigateur    # cherche un paquet par mot-clé
```

⚠️ Toujours « update » avant « install »

Prenez l'habitude de lancer `sudo apt update` avant d'installer quelque chose. Cette commande ne met rien à jour : elle rafraîchit seulement le *catalogue*, pour que le système connaisse les dernières versions disponibles. C'est `upgrade` qui met réellement à jour les logiciels.

Pour information, les autres familles utilisent d'autres commandes équivalentes :

Famille de distribution	Gestionnaire	Installer un paquet
Debian, Ubuntu, Mint	apt	<code>sudo apt install nom</code>
Fedora, Red Hat	dnf	<code>sudo dnf install nom</code>
Arch, Manjaro	pacman	<code>sudo pacman -S nom</code>

Les formats universels : Flatpak et Snap

En complément, des formats modernes fonctionnent **sur toutes les distributions** : **Flatpak** et **Snap**. Ils embarquent tout le nécessaire dans le paquet, ce qui simplifie l'installation des applications récentes. Beaucoup de distributions proposent une **logithèque graphique** (« Logiciels », « Software ») où vous installez tout cela d'un clic, sans terminal.

Exercice

Sur une distribution de la famille Ubuntu/Mint : 1) rafraîchissez le catalogue avec `sudo apt update` ; 2) cherchez un éditeur de texte avec `apt search gedit` ; 3) (optionnel) installez-le avec `sudo apt install gedit`. Observez comment le système gère tout seul les composants nécessaires.

À retenir

On installe depuis des **dépôts officiels** via un **gestionnaire de paquets**. Sur Ubuntu/Mint : `apt`. Le trio gagnant : `sudo apt update` (catalogue), `sudo apt upgrade` (mises à jour), `sudo apt install nom` (installer). Ou bien la logithèque, en un clic.

Processus, services et état du système

Comment savoir ce qui tourne, ce qui consomme de la mémoire, ou comment fermer un programme bloqué ? Voici comment garder un œil sur la santé de votre machine.

Définition — Processus

Un **processus** est simplement un programme *en cours d'exécution*. Chaque processus reçoit un numéro unique, le **PID** (*Process ID*), qui permet de l'identifier et d'agir sur lui.

Voir ce qui tourne

```
$ top           # tableau de bord en temps réel (q pour quitter)
$ htop         # version plus lisible et colorée (à installer)
$ ps aux      # liste de tous les processus en cours
```

`top` est l'équivalent du « Gestionnaire des tâches » de Windows : il montre l'usage du processeur (CPU) et de la mémoire (RAM), processus par processus.

Arrêter un programme bloqué

```
$ kill 1234      # demande poliment au processus n°1234 de s'arrêter
$ kill -9 1234  # force l'arrêt (en dernier recours)
```

Astuce

Pour trouver le PID d'un programme par son nom, utilisez `pgrep firefox`. Ou plus direct : `killall firefox` ferme toutes les instances de Firefox d'un coup.

L'espace disque et la mémoire

```
$ df -h        # espace disque libre/occupé, par partition
$ du -sh ~     # taille totale de votre dossier personnel
$ free -h      # mémoire vive (RAM) utilisée et disponible
```

L'option `-h` (*human-readable*) affiche les tailles en Mo/Go plutôt qu'en octets bruts.

Les services (systemd)

Définition — Service & systemd

Un **service** (ou *daemon*) est un programme qui tourne en arrière-plan : impression, réseau, serveur web... Sur la plupart des distributions modernes, c'est **systemd** qui les pilote, via la commande `systemctl`.

```
$ systemctl status bluetooth      # état d'un service
$ sudo systemctl start bluetooth  # démarrer un service
$ sudo systemctl stop bluetooth   # arrêter un service
```

À retenir

Un programme en cours = un **processus**, identifié par son **PID**. Pour observer : `top`. Pour arrêter : `kill` ou `killall`. Pour l'état du matériel : `df -h` (disque), `free -h` (mémoire). Les programmes d'arrière-plan (services) se gèrent avec `systemctl`.

Éditer du texte dans le terminal

Tôt ou tard, vous devrez modifier un fichier de configuration depuis le terminal. Bonne nouvelle : un éditeur simple suffit largement pour débiter.

Nano : l'éditeur simple et recommandé

Nano est l'éditeur idéal du débutant : on tape directement, et les raccourcis sont affichés en bas de l'écran.

```
$ nano monfichier.txt
```

Une fois dans Nano, le symbole `^` affiché en bas signifie la touche `Ctrl`. Les commandes essentielles :

Raccourci	Action
<code>Ctrl + O</code>	Enregistrer (puis <code>Entrée</code> pour confirmer le nom)
<code>Ctrl + X</code>	Quitter
<code>Ctrl + W</code>	Rechercher un mot
<code>Ctrl + K</code>	Couper la ligne
<code>Ctrl + U</code>	Coller la ligne

Vim : puissant mais déroutant

Vous entendrez parler de **Vim**, un éditeur très puissant mais à la prise en main difficile. La seule chose vraiment indispensable au début est de savoir **en sortir** si vous y tombez par accident :

```
# Dans Vim : appuyez d'abord sur Échap, puis tapez :
:q! # quitter sans enregistrer, puis Entrée
```

Astuce

Pour débiter, restez sur **Nano**. Il fait tout ce dont vous avez besoin. Vous explorerez Vim (ou son cousin VS Code en mode graphique) quand vous serez plus à l'aise.

Exercice

Tapez `nano essai.txt`, écrivez quelques lignes de texte, enregistrez avec `Ctrl` + `O` puis `Entrée`, quittez avec `Ctrl` + `X`, et vérifiez le résultat avec `cat essai.txt`.

À retenir

Pour éditer un fichier dans le terminal, utilisez **Nano** : `nano fichier`. Retenez `Ctrl` + `O` (enregistrer) et `Ctrl` + `X` (quitter). Et si vous tombez dans Vim : `Échap` puis `:q!`.

Réseau et connexions de base

Quelques commandes simples pour vérifier votre connexion, diagnostiquer un souci, et découvrir comment on se connecte à distance à une autre machine Linux.

Vérifier la connexion

```
$ ping google.com      # teste si Internet répond (Ctrl+C pour arrêter)
$ ip address           # affiche vos adresses réseau (IP)
$ ip a                 # version abrégée de la commande ci-dessus
```

Définition — ping

ping envoie de petits messages à un serveur et mesure le temps de réponse. Si vous recevez des réponses, votre connexion fonctionne. Si « tout est perdu », il y a un problème de réseau. C'est le premier réflexe de diagnostic.

Télécharger un fichier depuis le terminal

```
$ wget https://exemple.com/fichier.zip  # télécharge un fichier
$ curl -O https://exemple.com/page.html # alternative très utilisée
```

Se connecter à distance avec SSH

Définition — SSH

SSH (*Secure Shell*) permet de contrôler un autre ordinateur Linux à distance, par un terminal sécurisé et chiffré. C'est la méthode standard pour administrer des serveurs. Vous l'utiliserez beaucoup si vous gérez un jour un serveur ou un mini-ordinateur comme un Raspberry Pi.

```
$ ssh marie@192.168.1.50  # se connecter à la machine distante
```

Après avoir saisi le mot de passe, votre terminal pilote la machine distante comme si vous étiez devant elle. La commande **exit** ferme la connexion.

🎯 À retenir

Pour diagnostiquer : `ping` (la connexion répond-elle ?) et `ip a` (mes adresses). Pour télécharger : `wget` ou `curl`. Pour piloter une machine à distance de façon sécurisée : `ssh utilisateur@adresse`.

Bonnes pratiques et sécurité

Linux est réputé sûr, mais la sécurité dépend aussi de vos habitudes. Voici les règles simples qui vous garderont tranquille.

Les réflexes essentiels

- **Mettez à jour régulièrement.** Les mises à jour corrigent les failles de sécurité. Lancez-les via la logithèque ou `sudo apt update && sudo apt upgrade`.
- **N'installez que depuis des sources fiables** : dépôts officiels, logithèque, sites officiels des éditeurs. Méfiez-vous des commandes copiées-collées sans comprendre ce qu'elles font.
- **Réfléchissez avant chaque `sudo`**. Une commande administrateur peut tout modifier. Ne lancez en `sudo` que ce que vous comprenez.
- **Utilisez des mots de passe solides** et différents pour chaque service.
- **Sauvegardez vos données importantes** régulièrement (disque externe, cloud). Aucun système n'est à l'abri d'une panne matérielle.

⚠ Le piège du copier-coller

On trouve en ligne des commandes « miracles » à coller dans le terminal. Certaines, surtout celles contenant `sudo`, `rm` ou des téléchargements suivis d'exécution, peuvent endommager votre système ou compromettre votre sécurité. **Règle d'or : ne collez jamais une commande que vous ne comprenez pas.** Dans le doute, cherchez ce que fait chaque partie.

Le pare-feu

La plupart des distributions de bureau sont sûres par défaut. Pour aller plus loin, l'outil **UFW** (*Uncomplicated Firewall*) gère simplement le pare-feu :

```
$ sudo ufw enable      # active le pare-feu
$ sudo ufw status      # vérifie son état
```

Pour terminer ce module : la philosophie « comprendre avant d'agir »

La plus grande qualité d'un utilisateur de Linux n'est pas de connaître mille commandes par cœur, mais de prendre l'habitude de **comprendre ce qu'il fait**. Lisez les messages d'erreur (ils sont souvent explicites), utilisez `man` et `--help`, et prenez votre temps. C'est ainsi que l'on progresse durablement.

🎯 À retenir

Sécurité = bonnes habitudes : **mettre à jour**, **installer depuis des sources fiables**, **réfléchir avant chaque `sudo`**, **sauvegarder**, et surtout **ne jamais exécuter une commande qu'on ne comprend pas**.

Aide-mémoire des commandes essentielles

À imprimer ou garder sous la main : les commandes du quotidien, regroupées par usage.

Se repérer et naviguer

Commande	Action
<code>pwd</code>	Afficher le dossier courant
<code>ls</code> / <code>ls -lh</code> / <code>ls -a</code>	Lister / détaillé / avec fichiers cachés
<code>cd dossier</code> / <code>cd ..</code> / <code>cd ~</code>	Entrer / remonter / revenir à la maison
<code>clear</code>	Nettoyer l'écran

Fichiers et dossiers

Commande	Action
<code>mkdir nom</code>	Créer un dossier
<code>touch nom</code>	Créer un fichier vide
<code>cp source dest</code>	Copier (<code>-r</code> pour un dossier)
<code>mv source dest</code>	Déplacer ou renommer
<code>rm fichier</code>	Supprimer (<code>-r</code> dossier, <code>-i</code> confirmer)
<code>cat</code> / <code>less</code> / <code>head</code> / <code>tail</code>	Afficher le contenu d'un fichier
<code>find . -name "*.txt"</code>	Rechercher des fichiers

Système, processus, logiciels

Commande	Action
<code>sudo commande</code>	Exécuter en administrateur
<code>top</code> / <code>htop</code>	Voir les processus en temps réel
<code>kill PID</code> / <code>killall nom</code>	Arrêter un processus
<code>df -h</code> / <code>free -h</code>	Espace disque / mémoire vive
<code>sudo apt update</code>	Rafraîchir le catalogue de logiciels
<code>sudo apt upgrade</code>	Mettre à jour les logiciels
<code>sudo apt install nom</code>	Installer un logiciel
<code>chmod</code> / <code>chown</code>	Modifier droits / propriétaire

Aide et réseau

Commande	Action
<code>man commande</code>	Manuel complet (q pour quitter)
<code>commande --help</code>	Aide rapide
<code>ping site</code>	Tester la connexion
<code>ip a</code>	Afficher les adresses réseau
<code>ssh user@adresse</code>	Se connecter à distance

Raccourcis clavier du terminal

Raccourci	Action
<code>Tab</code>	Auto-compléter
<code>↑</code> / <code>↓</code>	Historique des commandes
<code>Ctrl</code> + <code>C</code>	Interrompre la commande en cours
<code>Ctrl</code> + <code>L</code>	Nettoyer l'écran
<code>Ctrl</code> + <code>Alt</code> + <code>T</code>	Ouvrir un terminal

Glossaire des termes clés

Terme	Définition simple
Noyau (kernel)	Le cœur du système qui communique avec le matériel. « Linux » au sens strict.
Distribution	Système complet basé sur le noyau Linux (ex. Ubuntu, Mint).
Open source	Logiciel dont le code est public et modifiable par tous.
Terminal	La fenêtre où l'on tape des commandes texte.
Shell / Bash	Le programme qui interprète et exécute les commandes.
Invite (prompt)	La ligne d'attente du terminal (ex. <code>marie@pc:~\$</code>).
Racine (/)	Le point de départ unique de toute l'arborescence des fichiers.
Dossier personnel (~)	Votre espace dans <code>/home/votre_nom</code> .
Chemin absolu / relatif	Adresse complète depuis <code>/</code> / depuis l'endroit où l'on est.
root	L'utilisateur administrateur tout-puissant.
sudo	Exécuter une commande avec les droits d'administrateur.
Permissions (rwx)	Droits de lecture, écriture, exécution sur un fichier.
Paquet	Un logiciel prêt à installer.
Dépôt (repository)	Bibliothèque officielle de paquets vérifiés.
Gestionnaire de paquets	Outil qui installe/met à jour les logiciels (ex. apt).
Processus / PID	Programme en cours d'exécution / son numéro d'identification.
Service (daemon)	Programme tournant en arrière-plan, géré par systemd.
LTS	Version à support long (≈ 5 ans), recommandée pour débuter.
SSH	Connexion sécurisée à distance vers une autre machine.
ISO	Le fichier image d'une distribution, à graver sur clé USB.

Pour aller plus loin

Vous avez les bases. Voici comment poursuivre et où trouver de l'aide quand vous serez bloqué.

Continuer à apprendre

- **Pratiquez tous les jours.** Utilisez le terminal même pour de petites choses : c'est la répétition qui ancre les commandes.
- **Tenez un carnet personnel** de commandes : notez celles que vous découvrez et leur usage.
- **Donnez-vous des petits projets** : organiser des fichiers en ligne de commande, automatiser une tâche répétitive, installer un serveur web local, configurer un Raspberry Pi...

Où trouver de l'aide fiable

- **La documentation officielle** de votre distribution (Ubuntu, Mint, etc.) : la source la plus sûre.
- **Le wiki Arch Linux** : une référence technique de grande qualité, utile même si vous n'utilisez pas Arch.
- **Les forums et communautés francophones** dédiés à votre distribution : entraide bienveillante pour les débutants.
- **Les commandes** `man` et `--help` : votre documentation est déjà sur votre machine, hors-ligne.

Les prochaines étapes naturelles

Pour qui	Sujets à explorer ensuite
Utilisateur curieux	Personnaliser son bureau, raccourcis avancés, gestionnaires de fichiers
Futur développeur	Git, environnements de programmation, éditeur VS Code
Apprenti administrateur	Scripts Bash, planification de tâches (cron), serveurs, Docker
Bricoleur	Raspberry Pi, domotique, auto-hébergement

Le mot de la fin

Apprendre Linux est un marathon, pas un sprint. Chaque commande comprise est une brique de plus. Soyez patient, curieux, et n'ayez pas peur de l'erreur : sur une machine de test, casser quelque chose est souvent la meilleure façon d'apprendre. **Bonne route dans le monde du libre !**

