

Université Abderrahmane Mira - Bejaia

Faculté des Sciences de Technologie

Département de Technologie

Cours :

Logiciel Libre et Open Source

1ère Année ST/Ingénieur



Année universitaire 2025/2026

Table des matières

I - Chapitre 1 : Fondements du logiciel libre	4
1. Histoire du mouvement du logiciel libre et open source :.....	4
2. Différence entre "free software" et "open source" :.....	4
3. Philosophie de Richard Stallman et le projet GNU	5
4. Impact économique et social des logiciels libres en Algérie et dans le monde.....	5
II - Chapitre 2 : Cadre juridique et licences	6
1. Introduction au droit d'auteur appliqué aux logiciels	6
2. Licences libres principales : GPL, LGPL, BSD, MIT, Apache.....	6
2.1. licence GPL (General Public License)	6
2.2. licence LGPL (Lesser General Public License)	6
2.3. licences BSD	6
2.4. licence MIT	7
2.5. licence Apache.....	7
3. Compatibilité entre licences	7
III - Chapitre 3 : Systèmes d'exploitation libres	8
1. Présentation des distributions.....	8
2. Principes d'installation et configuration de base.....	10
2.1. Linux en ligne :	10
2.2. Linux en machine virtuelle.....	10
2.3. Installant sur une clé USB	12
2.4. Linux installé sur le disque dur.....	14
3. Commandes fondamentales.....	17
3.1. Commandes système essentielles.....	17
3.2. Navigation dans le système de fichiers.....	19
3.3. Manipulation de fichiers et dossiers	22
3.4. Consultation et édition de fichiers texte.....	25
3.5. Aide et documentation	26
IV - Chapitre 4 : Solutions bureautiques libres	28
1. Writer (traitement de texte).....	28
1.1. Interface Writer	29
1.2. Mise en page.....	29
1.3. Styles	30
1.4. En-tête / pied de page / numéros de page.....	31
1.5. Images (figures).....	32
1.6. Tableaux	34
1.7. Table des matières, Liste des figures et liste des tableaux(automatique)	35
2. Calc (tableur).....	35
2.1. Interface Calc.....	35
2.2. Notions fondamentales dans LibreOffice Calc.....	36
2.3. Mise en forme dans LibreOffice Calc	37

2.4. Formules et calculs.....	38
2.5. Fonctions conditionnelles dans LibreOffice Calc	39
2.6. Les graphiques dans LibreOffice Calc.....	39

I Chapitre 1 : Fondements du logiciel libre

1. Histoire du mouvement du logiciel libre et open source :

Le mouvement du logiciel libre et open source est né de la volonté de préserver la possibilité, pour les utilisateurs, d'étudier, modifier et partager les programmes qu'ils utilisent. À partir des années 1980, la généralisation des logiciels propriétaires et des restrictions de diffusion du code source a progressivement limité ces pratiques, transformant le logiciel en produit fermé. En réaction, des initiatives comme le projet GNU et la Free Software Foundation ont défendu l'idée que le logiciel devait garantir des libertés fondamentales (utiliser, comprendre, modifier et redistribuer). Plus tard, l'expression « open source » s'est imposée pour mettre davantage l'accent sur l'efficacité du développement collaboratif et la qualité technique des logiciels. Aujourd'hui, ces deux approches recouvrent souvent des logiciels similaires, mais avec des motivations différentes : l'une insiste sur les libertés et l'éthique du partage, l'autre sur la méthode de production et les bénéfices pratiques. Dans ce cours, nous retracerons cette évolution et nous verrons comment ces modèles ont façonné les systèmes, les licences et les usages actuels.

2. Différence entre "free software" et "open source" :

La différence entre le « free software » (logiciel libre) et « l'open source » repose principalement sur la philosophie de ces deux approches, même si, dans la pratique, elles reposent souvent sur des licences similaires. Le free software, défendu par Richard Stallman et la Free Software Foundation, met l'accent sur la liberté des utilisateurs et sur une vision éthique et sociale du logiciel. Il insiste sur les quatre libertés fondamentales :

- utiliser le programme
- étudier son fonctionnement
- le modifier
- le redistribuer.


À l'inverse, le terme open source a été introduit pour promouvoir le partage du code sous un angle plus économique, en mettant en avant des avantages comme la qualité du code, la sécurité, la fiabilité et la rapidité d'innovation, sans nécessairement adopter le discours idéologique du logiciel libre.

 Exemple


En open source : le code est ouvert mais la licence peut autoriser :

- qu'une entreprise prenne le code
- le modifie
- le referme
- et redistribue une version propriétaire

En résumé, un logiciel libre doit rester libre, tandis qu'un logiciel open source peut, dans certains cas, devenir fermé selon sa licence.

 Attention

le terme free software peut prêter à confusion, car il ne signifie pas « gratuit », mais libre, c'est-à-dire le respect des libertés d'utilisation, de modification et de partage du logiciel.

 Remarque

un logiciel free software est toujours open source, mais un logiciel open source n'est pas forcément free software, car il peut ne pas garantir toutes les libertés des utilisateurs.

3. Philosophie de Richard Stallman et le projet GNU

La philosophie de Richard Stallman repose sur l'idée que le logiciel doit respecter la liberté des utilisateurs. À l'origine, la programmation s'inscrivait dans une culture de partage où le code pouvait être étudié, modifié et redistribué librement. Cette culture a progressivement disparu avec la généralisation des logiciels propriétaires, qui interdisent la modification et le partage du code source à travers des accords de confidentialité. Stallman considère ce modèle comme contraire à l'éthique, car il empêche la coopération et rend les utilisateurs dépendants des éditeurs.

Face à cette situation, il lance en 1983 le projet GNU, dont l'objectif est de créer un système d'exploitation entièrement libre, compatible avec Unix, afin de permettre aux utilisateurs de reprendre le contrôle de leur informatique. Le projet GNU ne se limite pas au noyau du système, mais inclut l'ensemble des outils nécessaires à un système complet (compilateurs, éditeurs, bibliothèques, interpréteurs, etc.). Pour garantir durablement ces libertés, Stallman introduit le principe du copyleft, mis en œuvre notamment à travers la licence GNU GPL, qui impose que toute version modifiée d'un logiciel libre reste libre

4. Impact économique et social des logiciels libres en Algérie et dans le monde

Le logiciel libre a profondément transformé l'économie et la société à l'échelle mondiale en réduisant la dépendance aux licences propriétaires et en favorisant un modèle fondé sur les services, la formation et le développement. Il constitue aujourd'hui une infrastructure stratégique de l'économie numérique, notamment dans les domaines des serveurs, du cloud, de l'intelligence artificielle et des sciences des données. Sur le plan social, il a démocratisé l'accès aux outils numériques, renforcé l'apprentissage autonome et la reproductibilité scientifique. Toutefois, son impact réel dépend fortement du niveau de compétences des utilisateurs.

En Algérie, le logiciel libre représente un potentiel économique et social important, notamment en matière de réduction des coûts, de développement des compétences locales et de modernisation des usages numériques. Il est déjà présent dans plusieurs secteurs, en particulier dans l'éducation et l'administration. Toutefois, son impact structurant demeure encore limité, principalement en raison de défis liés à la formation, à l'accompagnement des utilisateurs et à l'intégration du logiciel libre dans des stratégies numériques à long terme. ce qui a motivé la mise en place d'une formation sur les logiciels libres et open source destinée aux étudiants de première année en sciences et technologies.

II Chapitre 2 : Cadre juridique et licences

1. Introduction

Le cadre juridique et les licences constituent un pilier essentiel du logiciel libre, car les libertés qu'il promet ne reposent pas uniquement sur des principes techniques ou philosophiques, mais sur des mécanismes légaux précis. Sans un encadrement juridique clair, le partage du code source, sa modification et sa redistribution resteraient vulnérables aux appropriations et aux restrictions imposées par le droit d'auteur classique. Ce chapitre vise donc à expliquer comment le droit a été détourné de sa fonction traditionnelle de protection exclusive pour devenir un outil garantissant les libertés des utilisateurs.

2. Introduction au droit d'auteur appliqué aux logiciels

Le droit d'auteur appliqué aux logiciels constitue le fondement juridique sur lequel reposent aussi bien les logiciels propriétaires que les logiciels libres. Un logiciel est juridiquement considéré comme une œuvre de l'esprit, ce qui signifie que, par défaut, son auteur dispose de droits exclusifs sur son utilisation, sa modification et sa diffusion. Sans autorisation explicite, toute copie, modification ou redistribution est interdite, même lorsque le code source est techniquement accessible. Cette logique protectrice, initialement conçue pour les œuvres littéraires et artistiques, a été étendue aux logiciels afin de sécuriser leur exploitation économique.

3. Licences libres principales : GPL, LGPL, BSD, MIT, Apache

Les licences libres sont des instruments juridiques essentiels qui définissent précisément les droits et obligations liés à l'utilisation, la modification et la redistribution des logiciels. Elles s'appuient sur le droit d'auteur pour garantir les libertés fondamentales du logiciel libre tout en encadrant les conditions de partage du code.

3.1. licence GPL (General Public License)

est la licence emblématique du logiciel libre et repose sur le principe du copyleft fort. Elle garantit que toute version modifiée ou redistribuée d'un logiciel sous GPL doit obligatoirement rester sous la même licence, assurant ainsi la pérennité des libertés initiales. Ce mécanisme empêche l'appropriation privée du code et favorise un écosystème entièrement libre, mais il peut être perçu comme contraignant par certaines entreprises, car il impose la publication du code source dérivé.

3.2. licence LGPL (Lesser General Public License)

est une variante plus souple de la GPL, principalement conçue pour les bibliothèques logicielles. Elle permet à des logiciels propriétaires ou sous d'autres licences de lier ces bibliothèques sans obligation de libérer l'ensemble du code de l'application. La LGPL cherche ainsi un compromis entre la protection des libertés du logiciel libre et la compatibilité avec des usages industriels plus larges, ce qui en fait une licence couramment utilisée dans des contextes mixtes.

3.3. licences BSD

appartiennent à la famille des licences dites permissives. Elles autorisent l'utilisation, la modification et la redistribution du code avec très peu de contraintes, y compris l'intégration dans des logiciels propriétaires. Leur objectif n'est pas de préserver les libertés par un mécanisme juridique fort, mais de maximiser la diffusion du code. Cette permissivité a favorisé leur adoption dans de nombreux projets académiques et industriels, au prix toutefois d'un risque d'appropriation sans retour vers la communauté.

3.4. licence MIT

est l'une des licences libres les plus simples et les plus permissives. Elle impose essentiellement la conservation de la notice de copyright et de la licence, tout en laissant une grande liberté d'exploitation, y compris commerciale. Très appréciée dans le développement web et les projets modernes, la licence MIT facilite la réutilisation rapide du code, mais offre peu de garanties quant au maintien des libertés dans les versions dérivées.

3.5. licence Apache

se situe également dans la catégorie des licences permissives, tout en intégrant des clauses supplémentaires, notamment sur la gestion des brevets. Elle accorde une protection explicite contre les litiges liés aux brevets logiciels, ce qui la rend particulièrement attractive pour les grandes entreprises et les projets collaboratifs à grande échelle. La licence Apache combine ainsi flexibilité juridique et sécurité industrielle, ce qui explique son adoption par de nombreux projets open source majeurs.

4. Compatibilité entre licences

La compatibilité entre licences désigne la capacité de combiner, dans un même logiciel ou projet, des composants soumis à des licences différentes sans violer leurs conditions juridiques. Cette question est centrale dans le développement logiciel moderne, où la réutilisation de bibliothèques et de codes tiers est la norme. Deux licences sont dites compatibles lorsque leurs obligations respectives peuvent être respectées simultanément lors de la redistribution du logiciel final. À l'inverse, une incompatibilité apparaît lorsque les exigences d'une licence imposent des conditions contradictoires ou impossibles à satisfaire avec celles d'une autre.

Exemple

- Un logiciel sous licence MIT peut être intégré dans un logiciel sous licence GPL. (La licence MIT est permissive : elle autorise la modification, la redistribution et l'intégration dans d'autres projets.)
- Un logiciel sous licence GPL ne peut pas être intégré dans un logiciel propriétaire. (La GPL impose que toute redistribution reste libre)

Az Définition

Une licence permissive (MIT, BSD) est compatible avec une licence libre stricte (GPL), mais l'inverse n'est pas possible.

III Chapitre 3 : Systèmes d'exploitation libres

1. Introduction

Un système d'exploitation libre est un système dont le code source est accessible et qui garantit aux utilisateurs les libertés fondamentales : utiliser le système, l'étudier, le modifier et le redistribuer, avec ou sans modifications. Grâce à cette possibilité de modification et de redistribution, de nombreuses versions de systèmes libres ont vu le jour. La plupart sont basées sur Linux, on parle alors de distributions, mais pas uniquement. Chaque communauté peut ainsi construire sa propre distributions, adapté à ses besoins spécifiques.

Aujourd'hui, les systèmes d'exploitation libres offrent une alternative solide aux systèmes propriétaires classiques comme Windows et macOS. Linux n'est plus réservé aux informaticiens ou aux experts : il est devenu un système d'exploitation accessible, avec une interface graphique conviviale. Des distributions comme Ubuntu ou Linux Mint sont particulièrement adaptées aux débutants, faciles à prendre en main et largement utilisées. Elles offrent de nombreux avantages, notamment un accès gratuit, une grande stabilité, et une longévité importante, avec des mises à jour régulières même après la sortie de nouvelles versions.

2. Présentation des distributions

existe un nombre important de distributions Linux. Différentes méthodes peuvent être utilisées pour les définir et les classer, notamment selon leur distribution mère (Debian, Red Hat, Arch...), selon leur niveau de difficulté (débutant, intermédiaire, avancé), ou encore selon d'autres critères techniques. Cependant, dans ce cours, la méthode de classement qui nous convient le mieux est le classement selon l'usage, car il permet de choisir plus facilement une distribution en fonction des besoins de l'utilisateur.

- **Distributions pour usage bureautique / personnel**

Ces distributions sont conçues pour un usage quotidien (navigation Internet, bureautique, multimédia).

 Exemple

Ubuntu, Linux Mint , Zorin OS



Les distributions présentées, de gauche à droite, sont : Ubuntu, Linux Mint et Zorin OS.

- **Distributions Linux pour l'éducation**

Ces distributions sont conçues pour l'enseignement, les écoles, les lycées et parfois les universités. les Logiciels éducatifs préinstallés.

Exemple

Edubuntu, Endless OS



Les distributions présentées, de gauche à droite, sont : Edubuntu, Endless OS

- **Distributions pour la sécurité et les réseaux**

Ces distributions sont spécialisées dans la cybersécurité et les tests réseaux.

Exemple

Kali Linux, Parrot OS



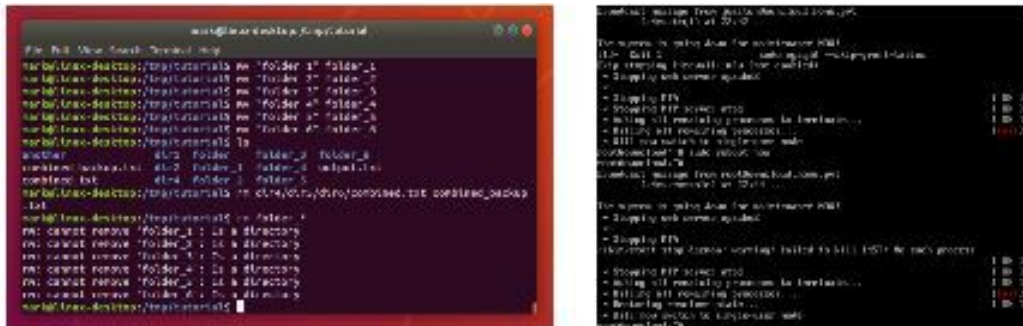
Les distributions présentées, de gauche à droite, sont : Kali linux, Parrot OS

- **Distributions pour serveurs**

Elles sont destinées à la gestion de serveurs (web, bases de données, réseaux).

Exemple

Debian, Ubuntu Server



Les distributions présentées, de gauche à droite, sont : Debian, Ubuntu Server

3. Principes d'installation et configuration de base

Il existe plusieurs façons d'utiliser Linux : en ligne via un navigateur, en l'installant sur une clé USB, sur le disque dur d'un ordinateur, ou en utilisant une machine virtuelle. Dans ce cours, nous allons nous concentrer sur ces quatre façons d'utiliser Linux, en présentant pour chacune les avantages et les inconvénients.

3.1. Linux en ligne :

Linux en ligne consiste à utiliser le système directement via un navigateur Internet, sans aucune installation. Cette méthode est accessible depuis n'importe quel ordinateur et convient particulièrement pour la découverte et les premiers tests de Linux. En revanche, elle dépend entièrement d'une connexion Internet et offre des performances ainsi que des fonctionnalités limitées par rapport aux autres modes d'utilisation.

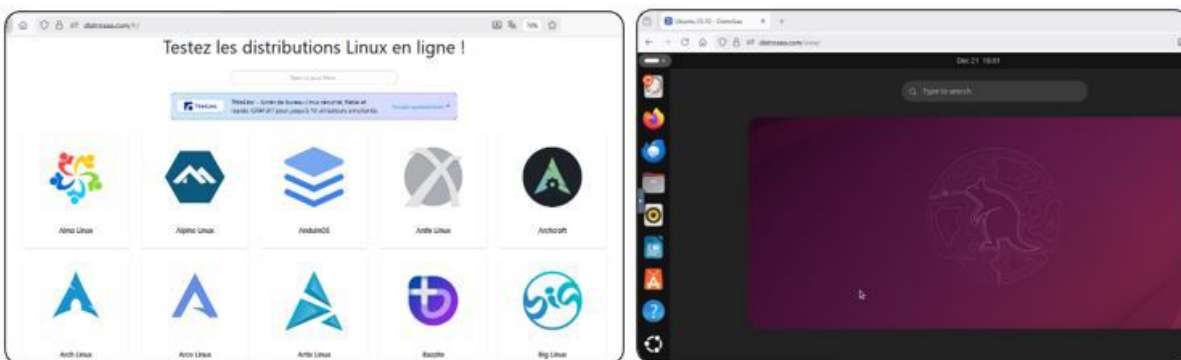
Plusieurs sites proposent de tester Linux en ligne. Il suffit d'accéder au site, de choisir la distribution et la version souhaitées, puis de lancer l'environnement Linux directement depuis le navigateur.

 Exemple

Exemple avec le site : <https://distrosea.com/fr/>

 Méthode

1. accéder au site
2. choisir la distribution et la version souhaitées et lancé Linux



de gauche à droite : interface du site, Linux via navigateur.

3.2. Linux en machine virtuelle

La machine virtuelle permet d'exécuter Linux à l'intérieur d'un autre système d'exploitation, comme Windows. Cette méthode est sécurisée, car elle ne modifie pas le système principal, et elle est particulièrement adaptée aux travaux pratiques (TP). En revanche, elle consomme davantage de ressources et offre des performances inférieures à celles d'une installation directe sur le disque dur.

Exemple

Exemple installation sur Oracle VirtualBox



interface de la machine virtuelle

⚠ Attention

il est fortement recommandé de toujours télécharger les fichiers d'installation de Linux uniquement depuis les sites officiels des distributions, afin d'éviter les fichiers corrompus, les versions modifiées ou les risques de sécurité.

🔧 Méthode

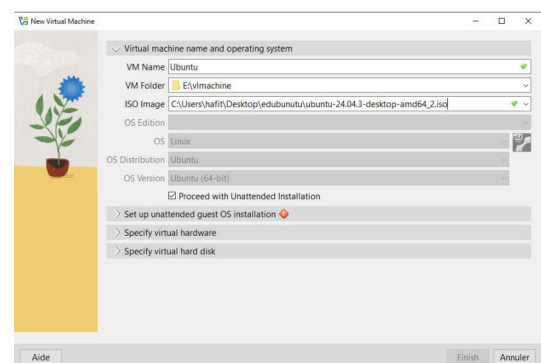
les étapes pour installer Linux sur une machine virtuelle sont les suivantes :

1. Télécharge le fichier ISO de la distribution Linux souhaitée (par exemple Ubuntu).

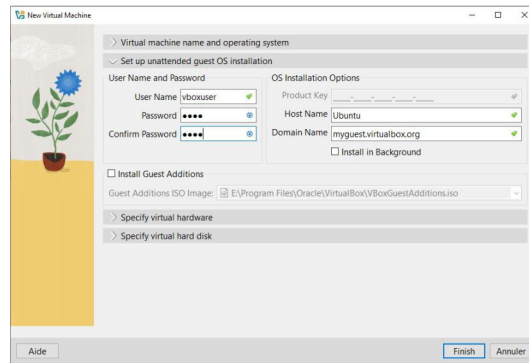
Exemple : <https://ubuntu.com/download/desktop>

2. Cliquez sur "Nouvelle machine" et configurez-la comme montré dans les illustrations suivantes.

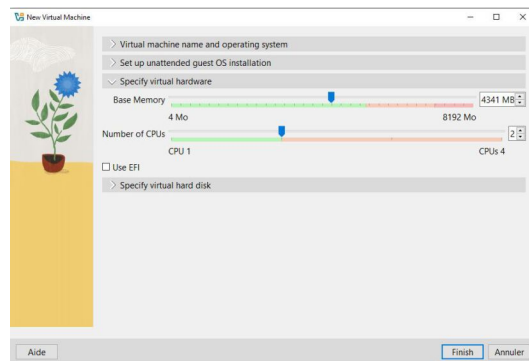
Attribuez un nom à la machine virtuelle, définissez son emplacement, puis sélectionnez le fichier ISO préalablement téléchargé.



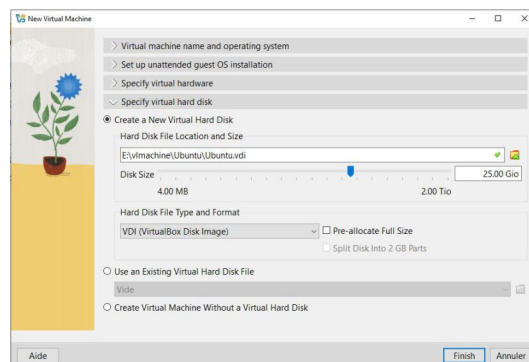
Définissez un nom d'utilisateur et un mot de passe pour la machine.



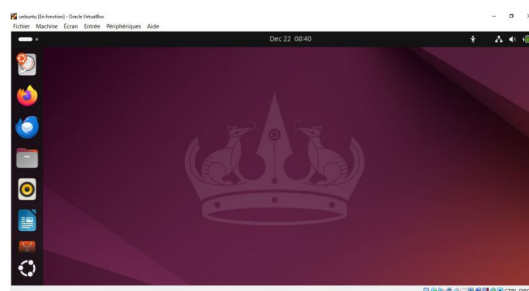
Définissez la quantité de mémoire vive (RAM) ainsi que le nombre de processeurs à attribuer à la machine virtuelle.



Définissez la capacité du disque dur à attribuer à la machine virtuelle.



Validez en cliquant sur Terminé (Finish), puis démarrez la machine virtuelle.



Linux via une machine virtuelle

3.3. Installant sur une clé USB

Linux sur clé USB consiste à démarrer le système directement depuis une clé USB, en mode live ou persistant. Cette méthode est portable, ne nécessite aucune installation sur l'ordinateur et est particulièrement utile pour le dépannage et l'apprentissage. En revanche, les performances sont limitées, l'espace de stockage est restreint et le fonctionnement dépend fortement de la vitesse de la clé USB.

Remarque

Deux modes sont possible :

- **Mode Live** → utilisation temporaire (aucune sauvegarde)
- **Mode Persistant** → les fichiers sont conservés sur la clé USB (si configuré)

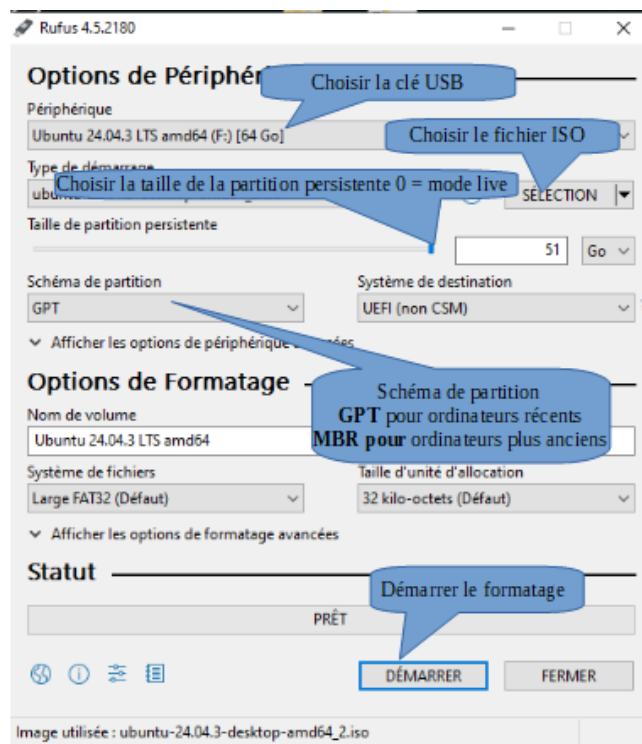
Méthode

les étapes pour installer Linux sur clé USB (bootable) sont les suivantes :

1. Préparer une clé USB de 8 Go minimum ;
2. Télécharge la distribution Linux souhaitée depuis son site officiel ;
3. Utilise un logiciel permettant d'écrire Linux sur la clé USB (ex. Rufus, BalenaEtcher).

Exemple

Exemple avec Rufus



Attention

cette opération va supprimer tous les fichiers présents sur la clé USB.

Une fois Rufus terminé, redémarrez votre PC, démarrez à partir de la clé USB et choisissez « Essayer » (Try) au lieu de « Installer ».

3.4. Linux installé sur le disque dur

Elle correspond à une installation complète du système directement sur l'ordinateur. Cette méthode offre les meilleures performances, permet une utilisation complète et stable du système, et convient parfaitement à un usage quotidien. En revanche, l'installation est plus complexe et comporte un risque de mauvaise manipulation, notamment pour les utilisateurs débutants.

⚠ Attention

l'installation de Linux sur le disque dur comporte des risques (suppression de données, erreur de partitionnement, problème de démarrage). Si vous n'êtes pas à l'aise avec les outils informatiques, il est conseillé d'utiliser l'une des méthodes précédentes.

💬 Remarque

Lors de l'installation, deux choix sont possibles : installer Linux seul ou installer Linux avec Windows (dual-boot).

- **Cas 1 : Linux seul (installation complète)**

Ce mode consiste à installer Linux comme seul système d'exploitation de l'ordinateur. Cette solution est simple à mettre en place et offre les meilleures performances, car tout le disque est dédié à Linux. En revanche, elle supprime définitivement tout le contenu du disque dur, y compris les autres systèmes et les données existantes.

- **Cas 2 : Linux + Windows (dual-boot)**

Ce mode permet d'installer Linux tout en conservant Windows sur le même ordinateur. Cette solution permet de garder Windows et d'avoir le choix du système d'exploitation au démarrage de l'ordinateur. En revanche, elle est plus délicate pour les débutants. L'installateur gère automatiquement le partitionnement du disque afin de partager l'espace entre les deux systèmes.

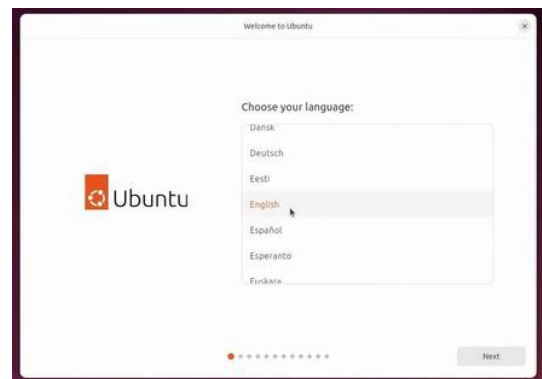
🔗 Méthode

Étapes d'installation de Linux sur le disque dur :

- Téléchargez Linux depuis le site officiel
- Préparez une clé USB bootable (vue dans la section précédente).
- Démarrer sur la clé USB
- Sélectionnez Installer ou Essayer Linux



Choisissez la langue et la disposition du clavier, puis cliquez sur Suivant jusqu'à arriver à la fenêtre de choix de l'installation.



Choisir Installer Linux (ici Ubuntu).



Sélectionnez l'installation interactive afin d'être guidé étape par étape.



Choisissez l'installation par défaut.



À cette étape, l'installateur propose l'installation de logiciels propriétaires indispensables au bon fonctionnement du matériel, notamment les pilotes graphiques.

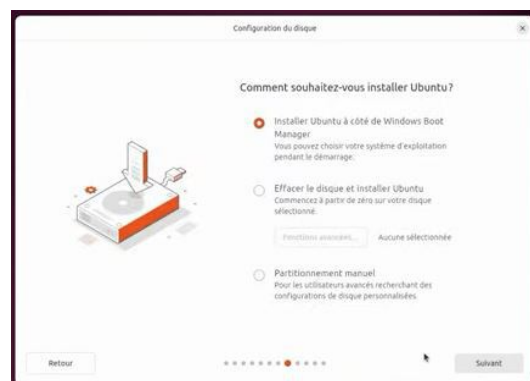
Cochez l'ensemble des options proposées afin d'assurer un fonctionnement optimal du système.



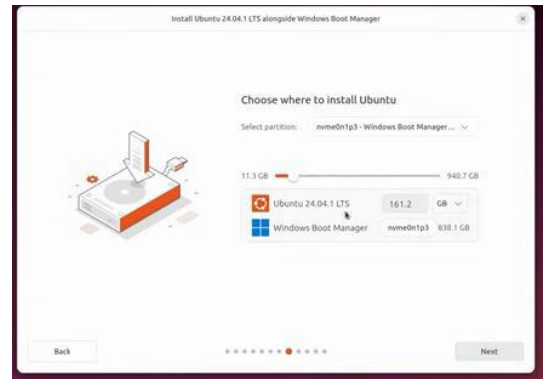
⚠ Attention

l'étape suivante concerne le partitionnement des disques, elle est une étape critique pouvant entraîner une perte de données. En cas de doute, il est fortement recommandé d'annuler l'installation et d'utiliser Linux par une autre méthode présentée précédemment.

Cette étape permet de choisir entre une installation de Linux à côté de Windows (dual-boot) ou une installation de Linux seule, entraînant la suppression complète de l'ancien système. Dans ce cours, nous opterons pour l'installation de Linux à côté de Windows.



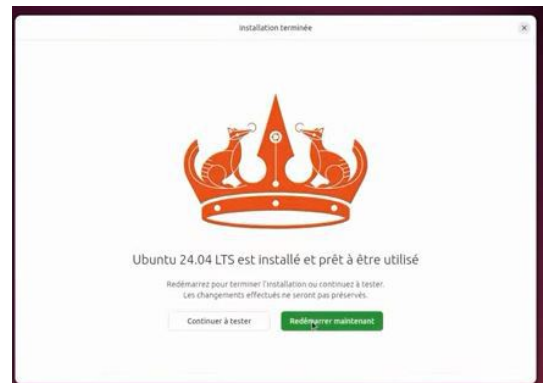
À l'aide du curseur, définissez l'espace disque à attribuer à Linux. L'installateur se chargera ensuite de créer automatiquement les partitions nécessaires.



Renseignez votre nom d'utilisateur, le nom de l'ordinateur ainsi que le mot de passe, puis cliquez sur Suivant afin de démarrer l'installation de Linux.



Patiencez jusqu'à la fin de l'installation. Une fois celle-ci terminée, le système Linux est prêt à être utilisé ; redémarrez alors l'ordinateur pour le démarrer.



4. Commandes fondamentales

Le terminal est un outil essentiel dans les systèmes Linux. Il permet à l'utilisateur de communiquer directement avec le système à l'aide de commandes textuelles afin d'exécuter des tâches telles que la gestion des fichiers, l'installation de logiciels ou l'administration du système.

Dans les systèmes Linux, de nombreux outils scientifiques, techniques et professionnels ne disposent pas d'interface graphique et s'utilisent exclusivement à travers le terminal. C'est notamment le cas de plusieurs logiciels utilisés en informatique, en bioinformatique, en réseaux ou en calcul scientifique. Le terminal permet alors d'exécuter ces programmes, de gérer les fichiers de données et d'automatiser des traitements complexes de manière efficace. Connaître les commandes de base du terminal est donc indispensable pour pouvoir utiliser ces outils, comprendre leur fonctionnement et travailler de façon autonome dans un environnement Linux.

4.1. Commandes système essentielles

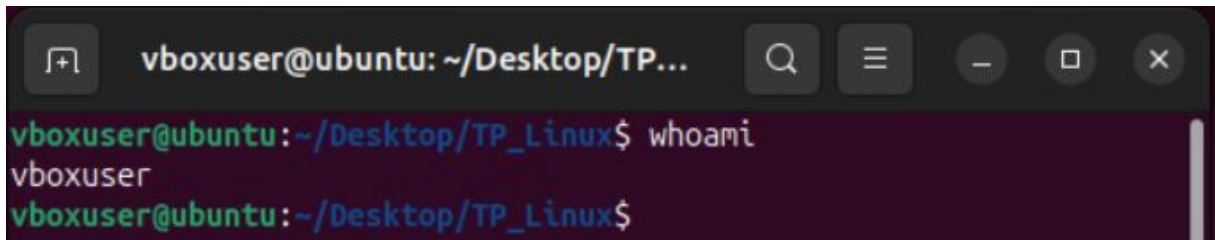
- **affiche le nom de l'utilisateur actuellement connecté au système**

Cette commande affiche le nom de l'utilisateur actuellement connecté au système.

Syntaxe

whoami

Exemple



```
vboxuser@ubuntu: ~/Desktop/TP...
vboxuser@ubuntu:~/Desktop/TP_Linux$ whoami
vboxuser
vboxuser@ubuntu:~/Desktop/TP_Linux$
```

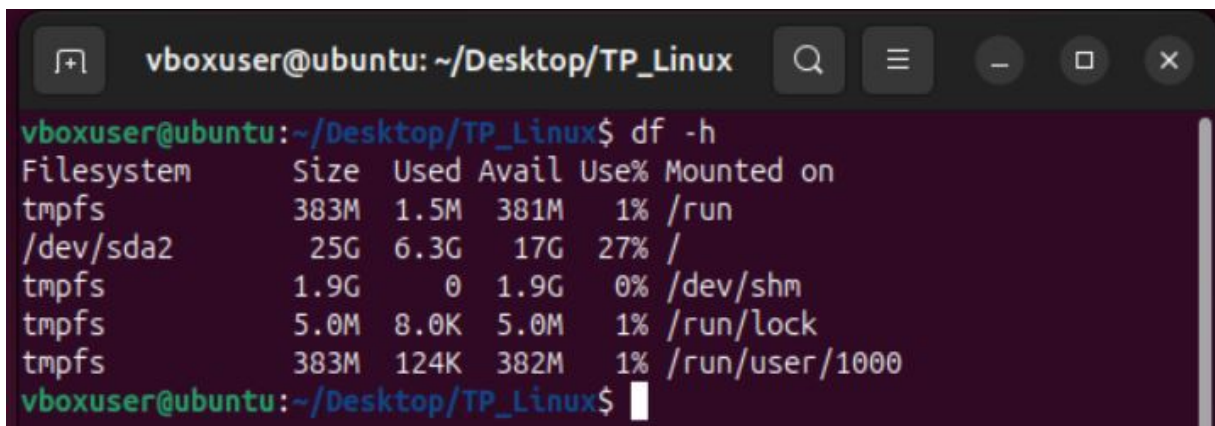
- **espace disque disponible**

Cette commande montre l'espace disque disponible et utilisé sur l'ordinateur. L'option `-h` permet d'afficher les tailles de manière lisible (en Go et Mo).

Syntaxe

df -h

Exemple



```
vboxuser@ubuntu: ~/Desktop/TP_Linux
vboxuser@ubuntu:~/Desktop/TP_Linux$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           383M  1.5M  381M   1% /run
/dev/sda2       25G   6.3G   17G  27% /
tmpfs           1.9G    0    1.9G   0% /dev/shm
tmpfs           5.0M   8.0K   5.0M   1% /run/lock
tmpfs           383M  124K   382M   1% /run/user/1000
vboxuser@ubuntu:~/Desktop/TP_Linux$
```

- **quantité de mémoire vive (RAM)**

Cette commande affiche la quantité de mémoire vive (RAM) utilisée et disponible. L'option `-h` rend l'affichage plus clair pour l'utilisateur.

Syntaxe

free -h

Exemple

```
vboxuser@ubuntu:~/Desktop/TP_Linux$ free -h
              total        used        free      shared  buff/cache   available
Mem:           3.7Gi         987Mi         878Mi         28Mi         2.2Gi         2.8Gi
Swap:          0B           0B           0B
```

4.2. Navigation dans le système de fichiers

Ces commandes permettent de se repérer dans le système de fichiers et de se déplacer entre les dossiers.

- **afficher le chemin du dossier courant**

Cette commande permet d'afficher le chemin du dossier courant.

Syntaxe

pwd

Exemple

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pwd
/home/ubuntu
ubuntu@ubuntu:~$
```

- **afficher la liste des fichiers et dossiers**

Cette commande permet d'afficher la liste des fichiers et dossierscourant.

Syntaxe

ls

Exemple

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  snap
Documents  Music      Public    Videos
```

- **afficher les fichiers avec des informations détaillées**

Cette commande permet d'afficher les fichiers avec des informations détaillées.

Syntaxe

ls -l

Exemple

```

ubuntu@ubuntu: ~
┌───┴───┐
ubuntu@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 60 Dec 28 19:51 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Public
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Dec 28 19:52 Videos
drwx----- 5 ubuntu ubuntu 100 Dec 28 19:53 snap
ubuntu@ubuntu:~$

```

- **afficher tous les fichiers, y compris les fichiers cachés**

Cette commande permet d'afficher tous les fichiers, y compris les fichiers cachés.

Syntaxe

ls -a

Exemple

```

ubuntu@ubuntu: ~
┌───┴───┐
ubuntu@ubuntu:~$ la -a
.          .bashrc  .gvfs    Desktop  Music    Templates
..         .cache   .local   Documents Pictures  Videos
.bash_logout .config  .profile Downloads Public    snap
ubuntu@ubuntu:~$

```

- **accéder à un dossier**

Cette commande permet **d'accéder à un dossier**.

Syntaxe

cd nom_dossier

Exemple

```
ubuntu@ubuntu: ~/Documents
ubuntu@ubuntu:~$ cd Documents
ubuntu@ubuntu:~/Documents$
```

- **revenir au dossier parent**

Cette commande permet de revenir au dossier parent.

Syntaxe

```
cd ..
```

Exemple

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~/Documents$ cd ..
ubuntu@ubuntu:~$
```

- **accéder au dossier personnel de l'utilisateur**

Cette commande permet d'accéder au dossier personnel de l'utilisateur.

Syntaxe

```
cd ~
```

Exemple

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~/Desktop/TP$ cd ~
ubuntu@ubuntu:~$
```

- **nettoyer l'écran du terminal**

Cette commande permet de nettoyer l'écran du terminal.

Syntaxe

```
clear
```

Exemple

```

ubuntu@ubuntu: ~
┌───┴───┐
ubuntu@ubuntu:~/Desktop/TP$ cd -
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  snap
Documents Music      Public    Videos
ubuntu@ubuntu:~$ clear

ubuntu@ubuntu: ~
┌───┴───┐
ubuntu@ubuntu:~$ █

```

4.3. Manipulation de fichiers et dossiers

- **créer un nouveau dossier**

Cette commande permet de créer un nouveau dossier.

Syntaxe

```
mkdir nom_dossier
```

Exemple

```

ubuntu@ubuntu: ~/Desktop
┌───┴───┐
ubuntu@ubuntu:~/Desktop$ mkdir TP_linux
ubuntu@ubuntu:~/Desktop$ █

```

- **créer un fichier vide**

Cette commande permet de créer un fichier texte vide.

Syntaxe

```
touch Nom_fichier.txt
```

Exemple


```

ubuntu@ubuntu: ~/Desktop/TP_linux
┌───┴───┐
ubuntu@ubuntu:~/Desktop/TP_linux$ touch fichier1.txt
ubuntu@ubuntu:~/Desktop/TP_linux$ █

```

- **copier un fichier**

Cette commande permet de copier un fichier.

 Syntaxe

```
cp fichier_a_copier fichier_copie
```


 Exemple



```
ubuntu@ubuntu: ~/Desktop/TP_linux
~/Desktop/TP_linux
ubuntu@ubuntu:~/Desktop/TP_linux$ cp fichier1.txt fichier2.txt
ubuntu@ubuntu:~/Desktop/TP_linux$
```

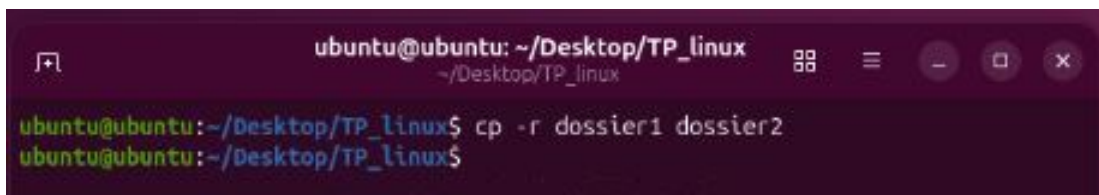
- **copier un dossier avec son contenu**

Cette commande permet de copier un dossier avec son contenu.

 Syntaxe

```
cp -r dossier_a_copier dossier_copie
```


 Exemple



```
ubuntu@ubuntu: ~/Desktop/TP_linux
~/Desktop/TP_linux
ubuntu@ubuntu:~/Desktop/TP_linux$ cp -r dossier1 dossier2
ubuntu@ubuntu:~/Desktop/TP_linux$
```

- **déplacer ou de renommer un fichier**

Cette commande permet de déplacer ou de renommer un fichier.

 Syntaxe

déplacer un fichier

```
mv fichier_a_deplacer dossier_destination
```

renommer un fichier

```
mv fichier_a_renommer nouveau_nom
```

Exemple

```
ubuntu@ubuntu: ~/Desktop/TP_linux
~/Desktop/TP_linux

ubuntu@ubuntu:~/Desktop/TP_linux$ mv fichier1.txt dossier1
ubuntu@ubuntu:~/Desktop/TP_linux$ █

vboxuser@ubuntu:~/Desktop/TP_linux$ mv fichier1.txt fichier2.txt
vboxuser@ubuntu:~/Desktop/TP_linux$ █
```

- **supprimer un fichier**

Cette commande permet de supprimer un fichier.

Syntaxe

```
rm fichier
```

Exemple

```
ubuntu@ubuntu: ~/Desktop/TP_linux
~/Desktop/TP_linux

ubuntu@ubuntu:~/Desktop/TP_linux$ rm fichier1.txt
ubuntu@ubuntu:~/Desktop/TP_linux$ █
```

- **supprimer un dossier vide**

Cette commande permet de supprimer un dossier vide.

Syntaxe

```
rmdir nom_dossier
```

Exemple

```
ubuntu@ubuntu: ~/Desktop
~/Desktop

ubuntu@ubuntu:~/Desktop$ rmdir TP_linux
ubuntu@ubuntu:~/Desktop$ █
```


- **supprimer un dossier et son contenu**

Cette commande permet de supprimer un dossier et son contenu

Syntaxe

`rm -r dossier`

Exemple



```
ubuntu@ubuntu: ~/Desktop/TP_linux
~/Desktop/TP_linux
ubuntu@ubuntu:~/Desktop/TP_linux$ rm -r dossier1
ubuntu@ubuntu:~/Desktop/TP_linux$
```

4.4. Consultation et édition de fichiers texte

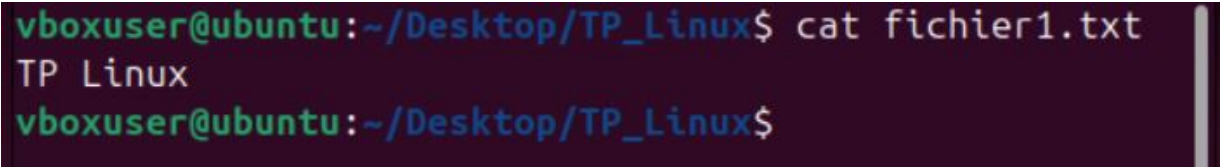
- **affiche le contenu d'un fichier**

Cette commande affiche tout le contenu du fichier.

Syntaxe

`cat fichier.txt`

Exemple



```
vboxuser@ubuntu:~/Desktop/TP_Linux$ cat fichier1.txt
TP Linux
vboxuser@ubuntu:~/Desktop/TP_Linux$
```

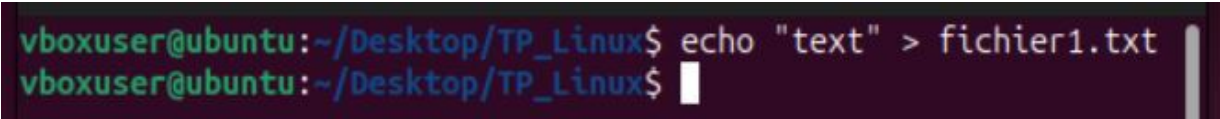
- **écrire du texte dans un fichier**

Cette commande permet d'écrire du texte dans un fichier.

Syntaxe

`echo "texte" > fichier.txt`

Exemple



```
vboxuser@ubuntu:~/Desktop/TP_Linux$ echo "text" > fichier1.txt
vboxuser@ubuntu:~/Desktop/TP_Linux$
```

⚠ Attention

Si le fichier existe, son contenu est remplacé.

Pour ajouter du texte à la fin d'un fichier existant, il suffit de remplacer le symbole > par >>.

📄 Syntaxe

```
echo "texte" >> fichier.txt
```

- ouvrir et modifier un fichier texte à l'aide de l'éditeur nano

📄 Syntaxe

```
nano fichier.txt
```

💬 Remarque

Pour enregistrer : **Ctrl + O**, pour quitter : **Ctrl + X**.

👁 Exemple

```
vboxuser@ubuntu:~/Desktop/TP_Linux$ nano fichier1.txt
```


```
GNU nano 7.2      fichier1.txt
text
```

```
[ Read 1 line ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File  ^_ Replace   ^U Paste     ^J Justify
```

4.5. Aide et documentation

- afficher le manuel d'une commande


Cette commande permet d'afficher le manuel d'une commande.

 Syntaxe

```
man commande
```

- **afficher une aide rapide**

Cette commande permet d'afficher une aide rapide.

 Syntaxe

```
commande --help
```

IV Chapitre 4 : Solutions bureautiques libres

1. Introduction

LibreOffice est une suite bureautique gratuite et open source qui peut remplacer Microsoft Office pour la majorité des usages courants : écrire des documents, faire des tableaux de calcul, créer des présentations ou même gérer des bases de données simples.

Elle est surtout constituée de ces modules :

- **Writer** : traitement de texte (documents, rapports, mémoires).
- **Calc** : tableur (tableaux, calculs, graphiques).
- **Impress** : présentations (diaporamas).
- **Draw** : dessin/diagrammes (schémas, organigrammes, affiches simples).
- **Base** : base de données (formulaires, tables, requêtes) – moins utilisé en L1.
- **Math** : éditeur de formules mathématiques (équations).

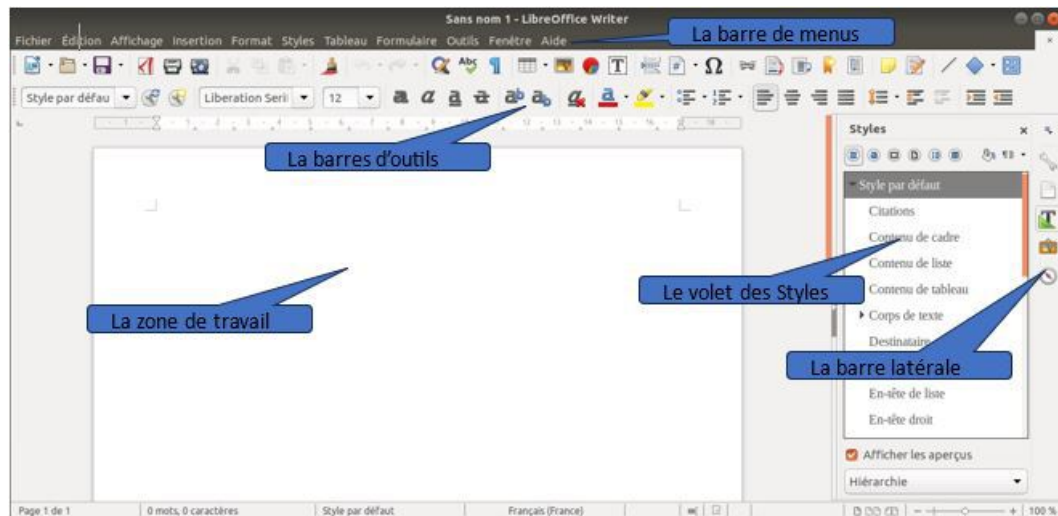
2. Writer (traitement de texte)

Writer est le module de traitement de texte de LibreOffice, comparable à Microsoft Word, et il couvre largement les besoins classiques de rédaction et de mise en forme. Il permet de produire des documents simples comme des lettres ou des comptes rendus, mais aussi des rapports plus longs avec une structure propre.

Le but de ce cours est de maîtriser LibreOffice Writer pour produire un document académique propre (rapport, mini-mémoire, mini-thèse) en évitant le bricolage et la mise en forme manuelle. Il vise à apprendre l'utilisation des **styles** et de la **structure automatique** afin d'obtenir une numérotation correcte des titres, une **table des matières** qui se met à jour, des **figures et tableaux** avec légendes et listes automatiques, une **pagination** propre, puis un export **PDF** cohérent. L'objectif final est d'avoir un document stable, facile à modifier, et conforme aux exigences universitaires.

2.1. Interface Writer

L'interface de LibreOffice Writer regroupe les menus et barres d'outils pour écrire et mettre en forme un document, ainsi que le volet **Styles** qui permet de structurer correctement le texte et d'automatiser la mise en page.



2.2. Mise en page

La **mise en page** dans Writer, c'est tout ce qui définit l'apparence "globale" du document avant même d'écrire : le format de la feuille, les marges, l'orientation. Si elle est bien réglée dès le début, le document reste propre et ne se casse pas quand on ajoute du contenu.

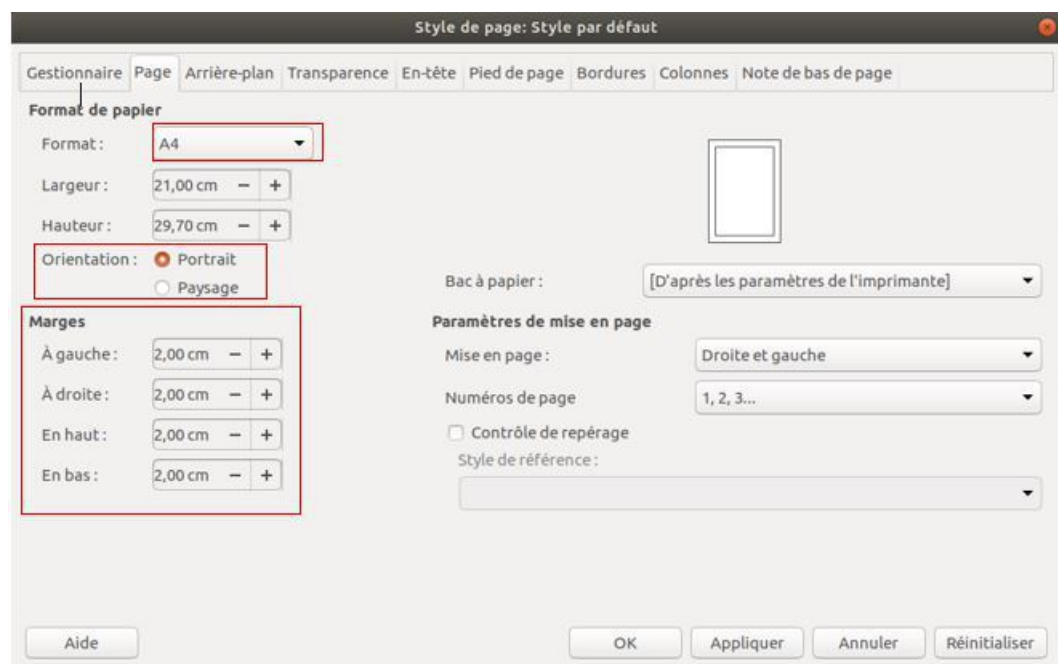
Format de page

- **A4** (le standard universitaire)
- **Portrait** (souvent), parfois Paysage pour de grands tableaux (mais seulement sur certaines pages)

Marges

- Les marges donnent de l'espace pour la lecture et l'impression.
 - Exemple simple et accepté : **2,5 cm** partout

Pour modifier la mise en page, le chemin est : **Format > page...** (ou **Format > styles de Page...** selon la version)



2.3. Styles

Les **styles** dans Writer, c'est la méthode correcte pour mettre en forme un document **sans bricolage**. Au lieu de changer la police, la taille, le gras et l'alignement à la main à chaque fois, on applique un **style** (Titre 1, Titre 2, Corps de texte, Légende...), et Writer garde une mise en forme **cohérente** sur tout le document.

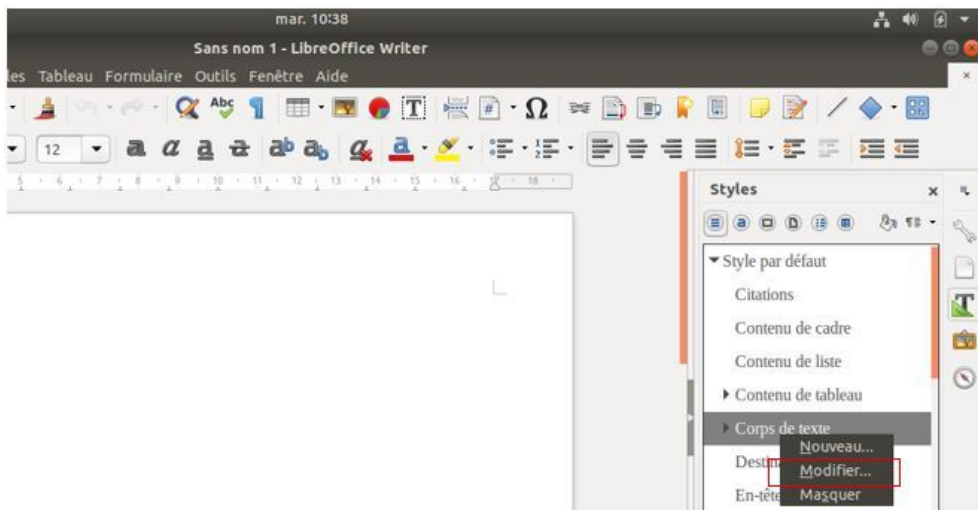
À quoi servent les styles ?

- **Uniformiser** : tous les titres ont le même aspect.
- **Automatiser** : table des matières, numérotation des titres, listes de figures/tableaux.
- **Gagner du temps** : si on modifie un style, tout le document se met à jour.
- **Éviter les erreurs** : plus besoin de bricoler avec des espaces et du gras.

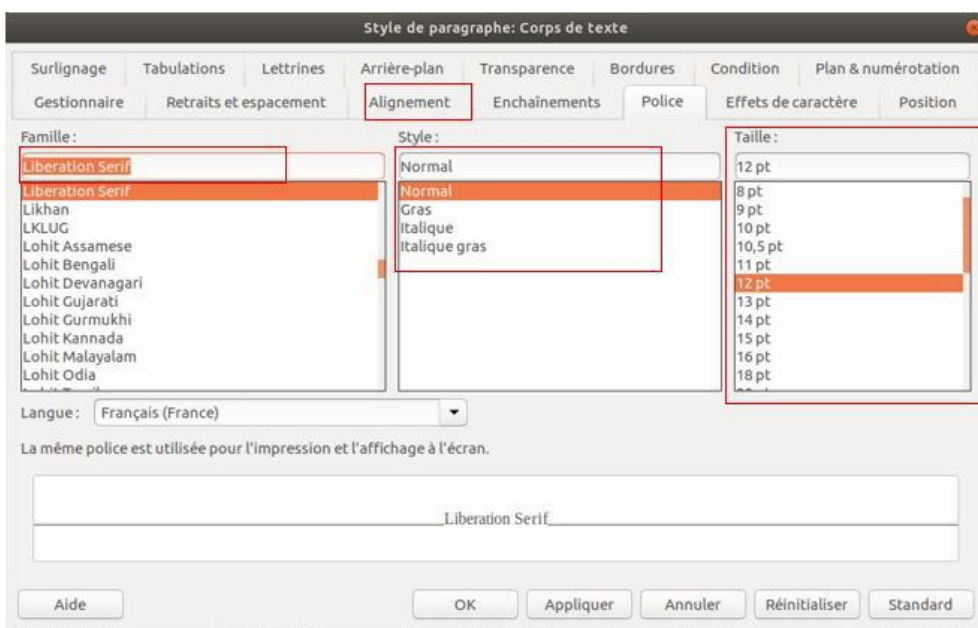
Les styles importants pour un rapport/mémoire

- **Titre 1** : grandes parties (Introduction, Chapitre 1...)
- **Titre 2** : sous-parties (1.1, 1.2...)
- **Corps de texte** : paragraphes normaux
- **Légende** : texte sous figure/tableau

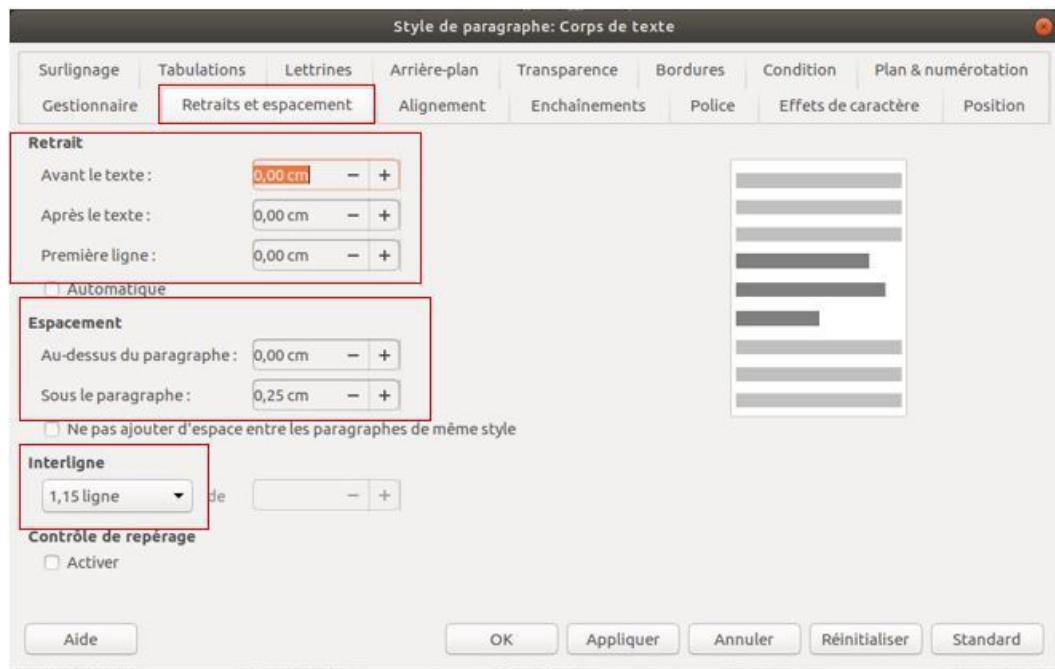
Pour modifier un style : **clic droit sur le style** dans le **volet Styles** → **Modifier**.



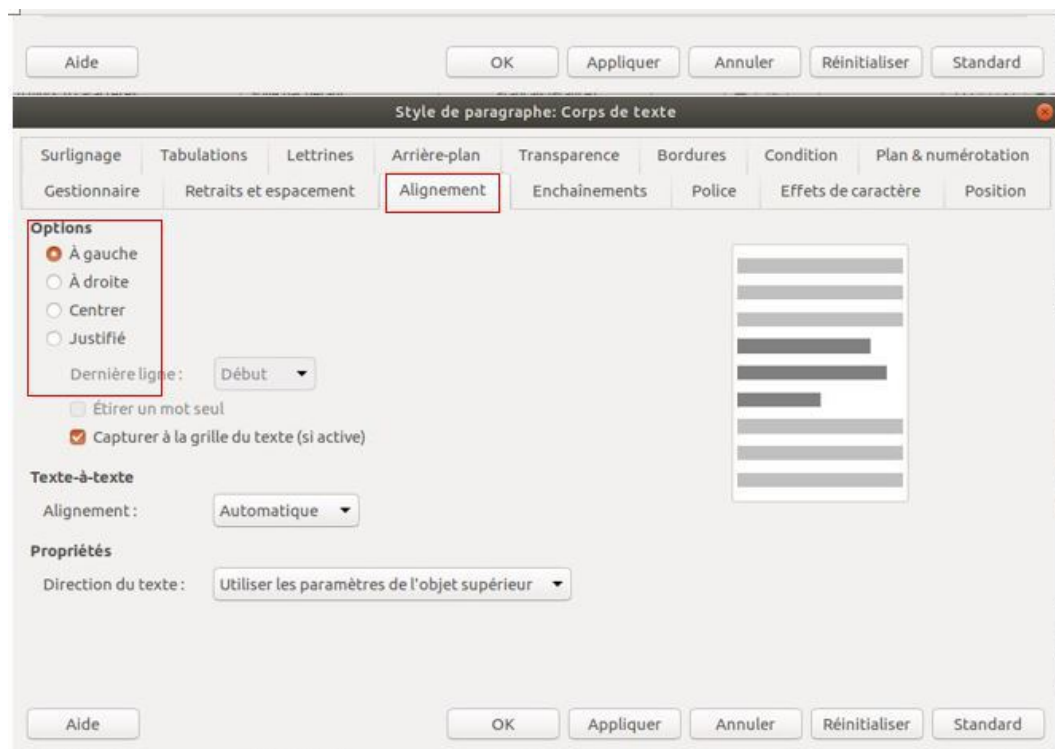
Dans la fenêtre de modification des styles, plusieurs paramètres peuvent être réglés, mais ceux qui nous intéressent sont : l'onglet **Police** pour choisir la police, le style (gras/italique) et la taille.



l'onglet **Retraits et espacement** pour régler l'espacement avant et après le texte ainsi que l'interligne.



l'onglet **Alignement** pour définir la position du texte (gauche, centré, droite ou justifié).



2.4. En-tête / pied de page / numéros de page

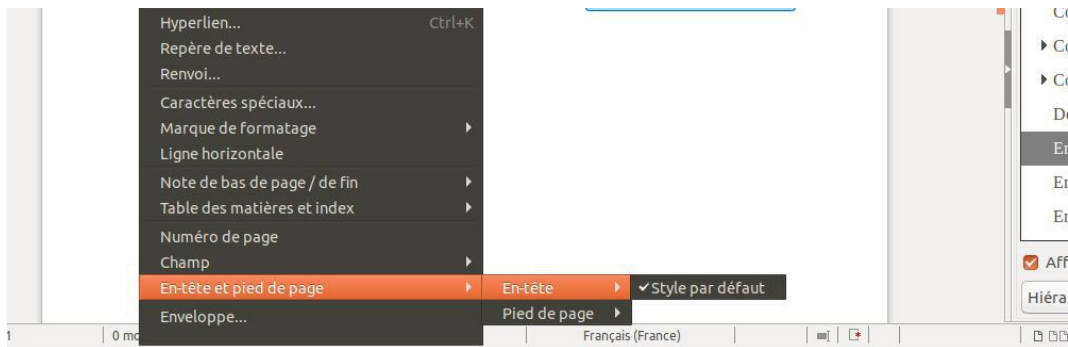
En-tête (Header)

L'en-tête est la zone en haut de chaque page. Il sert à afficher des informations répétées, par exemple :

- titre du document,
- nom du chapitre,
- nom/prénom, groupe, année universitaire (selon consignes).

Activer l'en-tête :

Insertion > En-tête et pied de page > En-tête (puis choisir le style de page concerné).



Pied de page (Footer)

Le pied de page est la zone en bas de chaque page. Il sert surtout pour :

- **numérotation des pages**,
- date, nom du document, etc.

Activer le pied de page :

Insertion > En-tête et pied de page > Pied de page (puis choisir le style de page concerné).

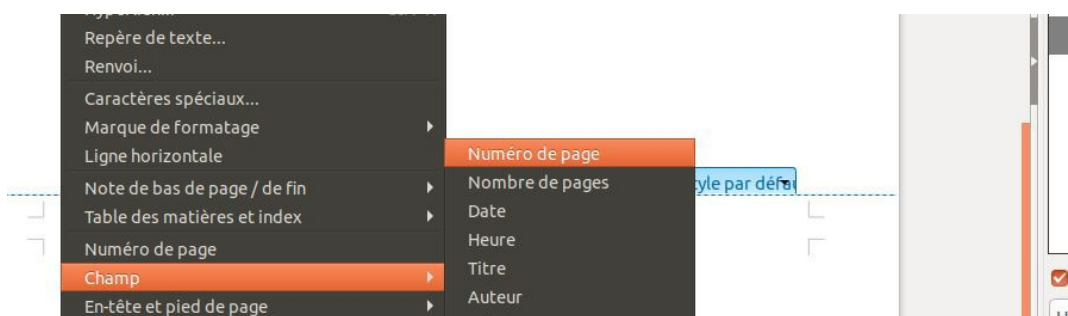


Numéros de page

Pour insérer un numéro de page, on utilise un **champ** (automatique), pas un numéro tapé au clavier.

Insérer le numéro :

- Cliquer dans le pied de page
- **Insertion > Champ > Numéro de page**



2.5. Images (figures)

L'objectif est d'insérer une image qui **ne casse pas la mise en page**, et de l'intégrer comme une vraie **figure académique** (numéro + titre), afin de pouvoir générer ensuite la **liste des figures**.

Insérer une image

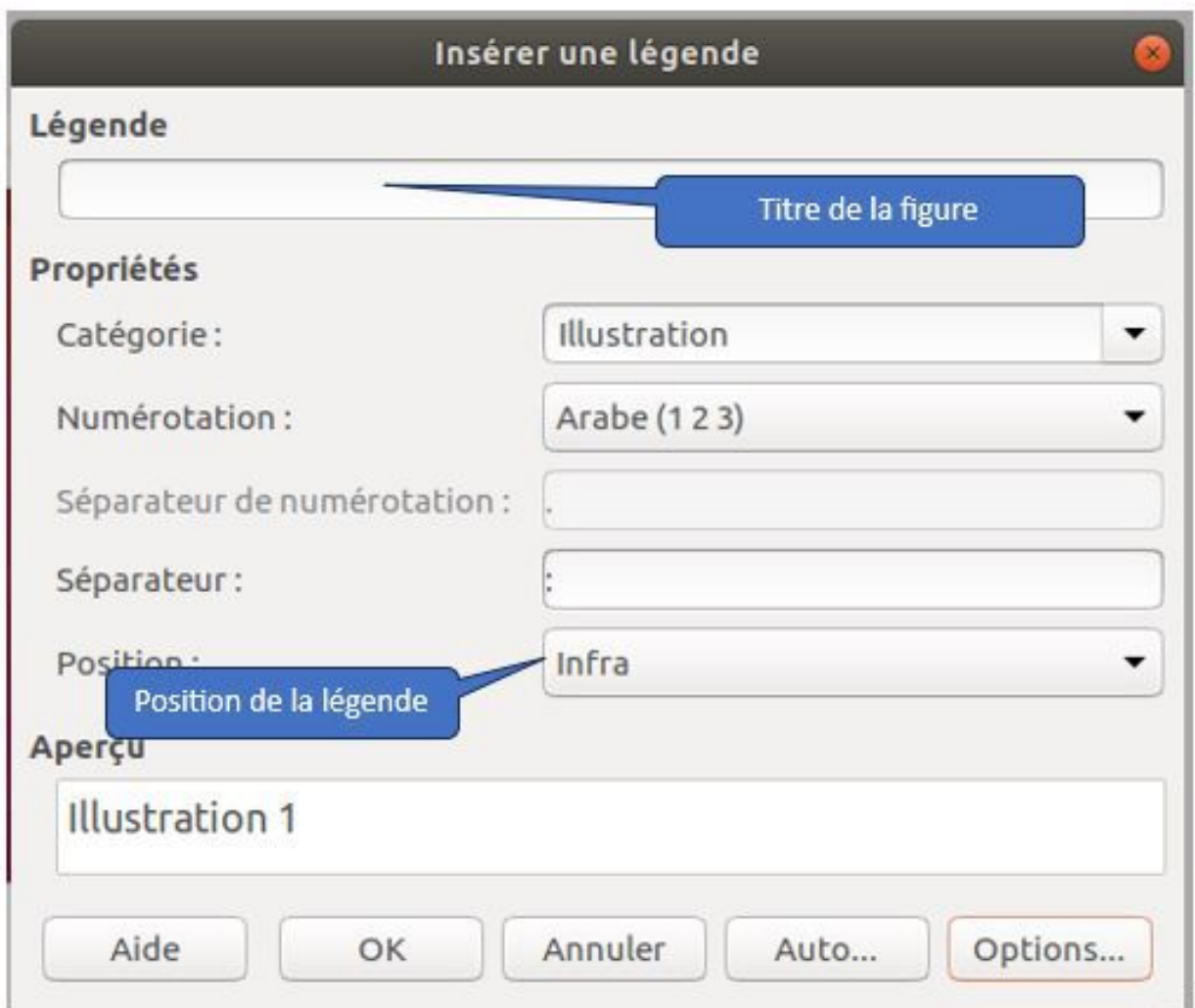
- **Insertion > Image...** puis choisir le fichier.
- Redimensionner avec les poignées (coins) pour garder les proportions.



Légende (Figure 1, Figure 2... automatique)

Ne pas taper "Figure 1" à la main.

- Clic droit sur l'image > **Insérer une légende...**
- Catégorie : **Illustration** (ou Figure selon version)
- Writer numérote automatiquement : **Figure 1, Figure 2...**

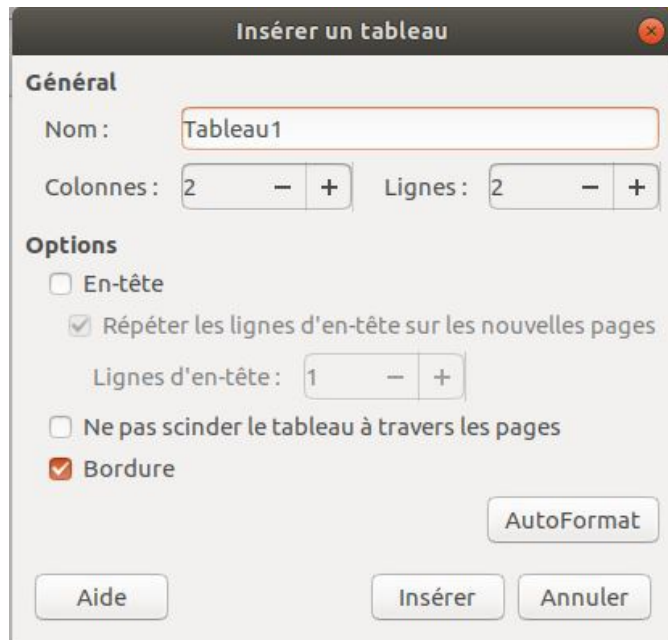


2.6. Tableaux

Un tableau dans un rapport doit être **clair, aligné**, et surtout accompagné d'un **titre (légende) automatique** pour éviter le bricolage et pouvoir générer la **liste des tableaux**.

Insérer un tableau

- **Tableau > Insérer un tableau...**
- Choisir le nombre de **colonnes** et de **lignes**
- Valider, puis remplir les cellules



Titre du tableau (légende automatique)

Ne pas écrire "Tableau 1" à la main.

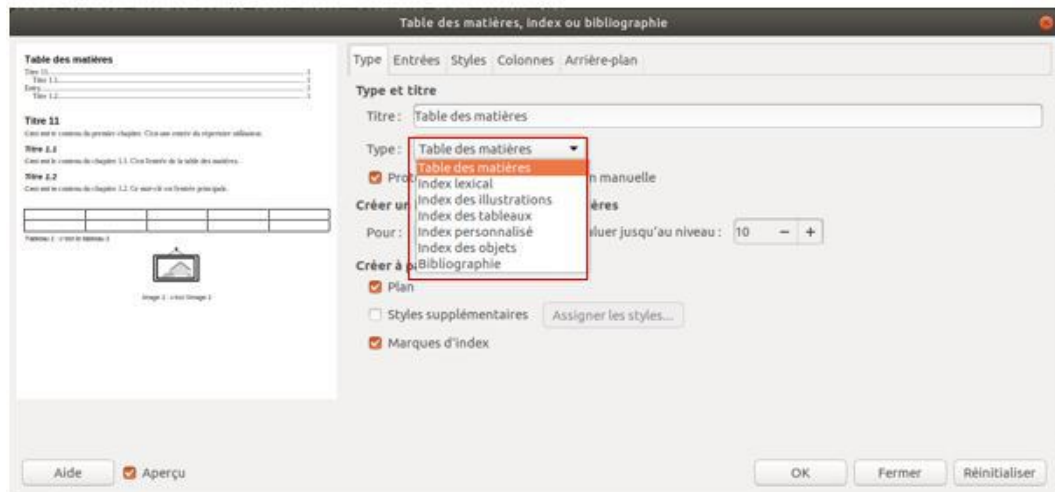
- Clic sur le tableau → **Insérertion** → **légende...**
- Catégorie : **Tableau**
- Writer crée automatiquement : **Tableau 1, Tableau 2...**
- Ajouter le texte : "Tableau 1 : ..."



2.7. Table des matières, Liste des figures et liste des tableaux(automatique)

Ces trois éléments servent à rendre le document **professionnel** et surtout **automatique** : dès qu'on ajoute, supprime ou déplace une section, une figure ou un tableau, tout se met à jour sans refaire le travail à la main.

Insertion > Table des matières et index > Table des matières > Puis choisir le type (Table des matières, index figures , index des tableaux)

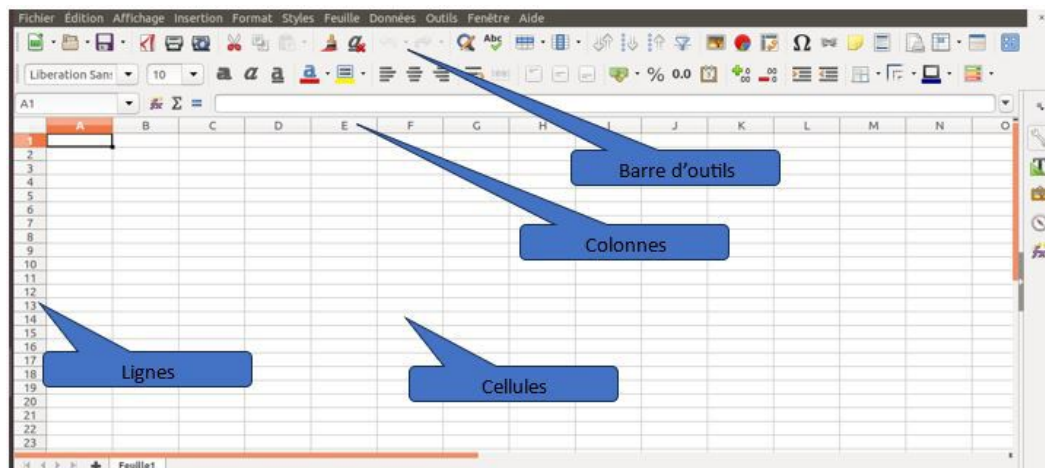


3. Calc (tableur)

Calc est le tableur de LibreOffice, l'équivalent d'Excel, conçu pour organiser des données et faire des calculs. Il permet de créer des tableaux structurés, d'utiliser des formules et des fonctions (sommations, moyennes, conditions, recherches), de trier et filtrer des listes, et de représenter les résultats avec des graphiques. Il est particulièrement utile pour des contextes pédagogiques où il faut apprendre à gérer des notes, analyser des résultats de TP, faire des statistiques simples ou automatiser des calculs répétitifs. Calc propose aussi des outils pratiques comme la mise en forme conditionnelle, les tableaux croisés dynamiques (pour résumer rapidement des données), la validation de données (pour limiter les erreurs de saisie) et l'export en PDF, ce qui aide à produire des documents propres et facilement partageables.

3.1. Interface Calc

LibreOffice Calc est un **logiciel tableur**. Son interface est organisée pour permettre la saisie, le calcul et l'analyse des données.



3.2. Notions fondamentales dans LibreOffice Calc

Pour bien utiliser un tableur, il faut comprendre 4 notions essentielles :

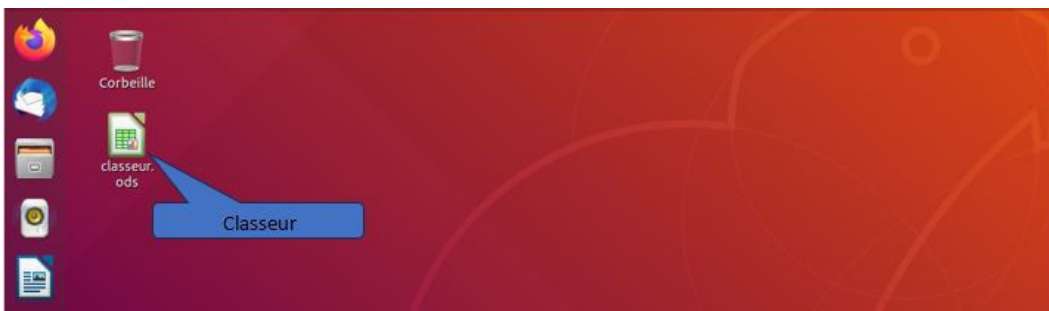
- Classeur
- Feuille
- Cellule
- Adresse

a) Classeur

Un **classeur** est le fichier complet que l'on enregistre sur l'ordinateur.

C'est l'équivalent d'un document dans Writer.

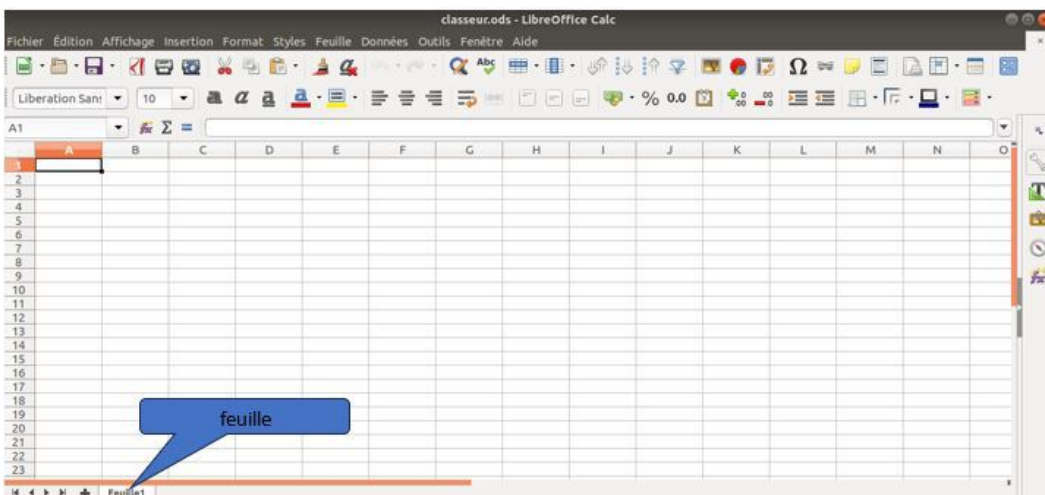
- Il peut contenir plusieurs feuilles.
- Il porte une extension : .ods
- Exemple : Notes_Etudiants.ods



b) Feuille

Une **feuille** est une page de travail à l'intérieur du classeur.

- Chaque classeur peut contenir plusieurs feuilles.
- Les feuilles apparaissent sous forme d'onglets en bas (Feuille1, Feuille2...).
- On peut :
 - Ajouter une feuille
 - Supprimer une feuille
 - Renommer une feuille



c) Cellule et Adresse

Une **cellule** est l'unité de base du tableau.

C'est l'intersection :

- d'une colonne (A, B, C...)
- et d'une ligne (1, 2, 3...)

Chaque cellule possède une **adresse unique**.

Elle est composée :

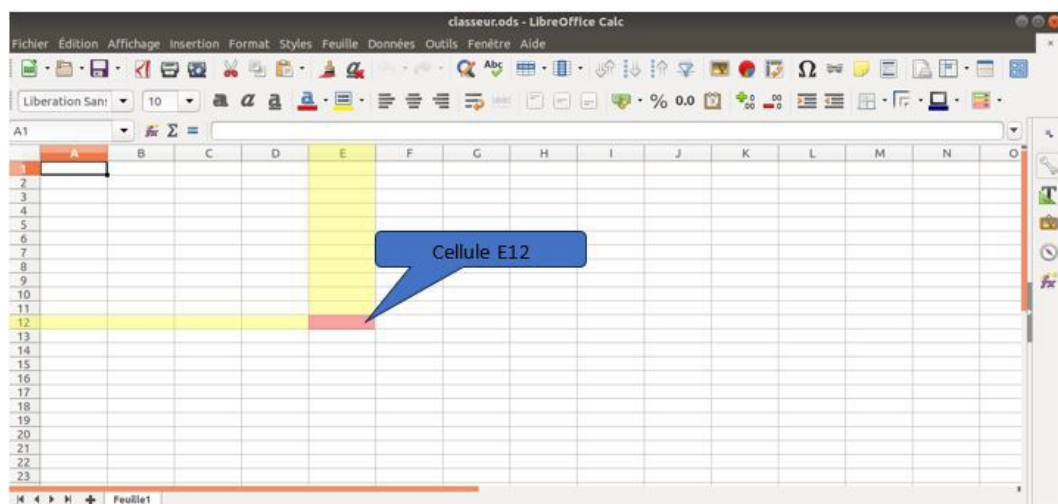
- De la lettre de la colonne
- Du numéro de la ligne

👁 Exemple

- A1 → colonne A, ligne 1
- B5 → colonne B, ligne 5

Chaque cellule peut contenir :

- Du texte
- Un nombre
- Une date
- Une formule



3.3. Mise en forme dans LibreOffice Calc

La **mise en forme** consiste à modifier l'apparence des données sans changer leur valeur.

Elle permet de rendre un tableau :

- Plus lisible
- Plus professionnel
- Plus organisé

Elle permet de régler :

- Format des nombres (monétaire, pourcentage...)
- Alignement

- Bordures
- Couleurs
- Fusionner des cellules
- Largeur / hauteur des colonnes

3.4. Formules et calculs

Les formules permettent d'effectuer des calculs automatiques dans LibreOffice Calc.

a) Qu'est-ce qu'une formule ?

- Une formule est une expression de calcul.
- Elle commence toujours par le signe =.
- Elle peut contenir :
 - Des nombres
 - Des opérateurs (+, -, *, /)
 - Des adresses de cellules

=A1+B1

👁 Exemple

b) Opérations de base

- Addition : =A1+B1
- Soustraction : =A1-B1
- Multiplication : =A1*B1
- Division : =A1/B1

i) Fonctions principales

Calc propose des fonctions prêtes à utiliser :

- =SOMME (A1 : A5) → additionne plusieurs cellules
- =MOYENNE (A1 : A5) → calcule la moyenne
- =MIN (A1 : A5) → valeur minimale
- =MAX (A1 : A5) → valeur maximale
- =NB (A1 : A5) → compte les cellules numériques

c) Références de cellules

Il existe deux types :

- Référence relative : A1
 - Change lorsqu'on copie la formule
- Référence absolue : \$A\$1
 - Ne change pas lors de la copie

3.5. Fonctions conditionnelles dans LibreOffice Calc

Les **fonctions conditionnelles** permettent d'effectuer un calcul ou d'afficher un résultat **en fonction d'une condition**.


Elles introduisent une logique proche de l'algorithmique :

Si une condition est vraie → faire quelque chose

Sinon → faire autre chose

- **Fonction SI()**

La fonction **SI()** permet de tester une condition.

 Syntaxe

=SI(condition ; valeur_si_vrai ; valeur_si_faux)

 Exemple

=SI(A1>=10 ; "Admis" ; "Ajourné")

3.6. Les graphiques dans LibreOffice Calc

Les **graphiques**

permettent de représenter des données sous forme visuelle afin de mieux les comprendre et les analyser.

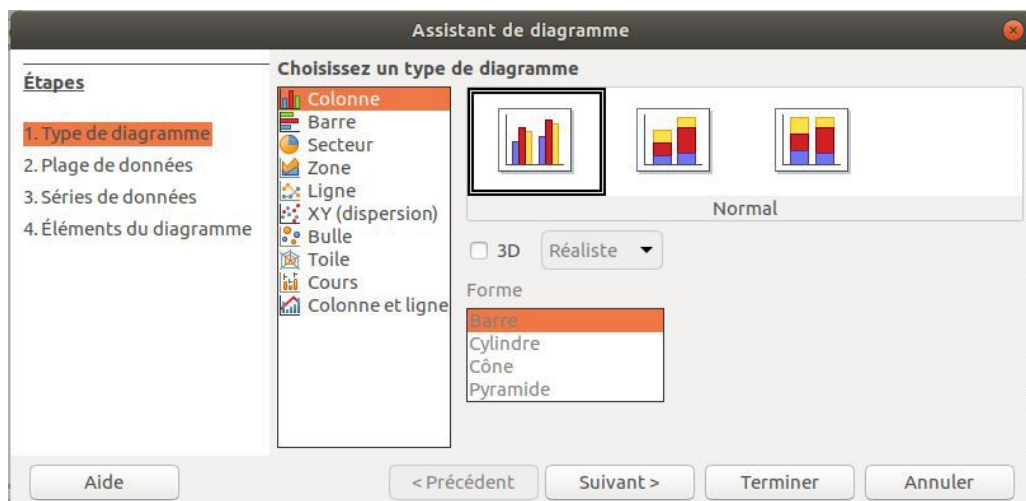
Ils facilitent l'interprétation des résultats et rendent les tableaux plus clairs et plus professionnels.

a) Création d'un graphique

Étapes principales :

1. Sélectionner les données (tableau)
2. Cliquer sur **Insertion** → **Diagramme**
3. Choisir le type de graphique
4. Vérifier les paramètres
5. Valider

Le graphique apparaît dans la feuille et peut être déplacé ou redimensionné.



i) Types de graphiques les plus utilisés

Histogramme (diagramme en colonnes)

- Compare des valeurs
- Exemple : notes des étudiants

Courbe (diagramme en lignes)

- Montre une évolution
- Exemple : évolution des ventes par mois

Camembert (diagramme circulaire)

- Montre une répartition en pourcentage
- Exemple : répartition des dépenses

1 Éléments d'un graphique

Un graphique contient généralement :

- Un titre
- Des axes (horizontal et vertical)
- Une légende
- Des séries de données

Annexes

Liste de logiciels propriétaires et équivalents libres

Domaine	Logiciel propriétaire	Équivalent libre / gratuit	Utilité
Traitement de texte	Microsoft Word	LibreOffice Writer	Rédiger rapports, mémoires, lettres
Tableur	Microsoft Excel	LibreOffice Calc	Calculs, tableaux, graphiques
Présentation	Microsoft PowerPoint	LibreOffice Impress	Exposés, soutenances, diaporamas
Base de données	Microsoft Access	LibreOffice Base	Gestion simple de bases de données
Suite bureautique complète	Microsoft Office	LibreOffice	Travail bureautique complet
Mathématiques numériques	MATLAB	GNU Octave, Scilab	Calcul scientifique, matrices, scripts
Calcul symbolique	Mathematica	SageMath, Maxima	Calcul formel, algèbre, équations
Statistiques	SPSS	R + RStudio	Analyse statistique, graphiques
Analyse de données	Minitab	R, PSPP, JASP	Statistiques pour étudiants
DAO / CAO 2D-3D	AutoCAD	FreeCAD, LibreCAD	Dessin technique, modélisation
Modélisation 3D	SolidWorks	FreeCAD, Blender	Conception 3D
Retouche photo	Adobe Photoshop	GIMP, Krita	Retouche d'images
Dessin vectoriel	Adobe Illustrator	Inkscape	Logos, schémas, affiches
Mise en page	Adobe InDesign	Scribus	Affiches, brochures, magazines
Montage vidéo	Adobe Premiere Pro	Kdenlive, Shotcut	Montage vidéo
Audio	Adobe Audition	Audacity	Enregistrement, montage audio
Lecture PDF	Adobe Acrobat Pro	Okular, PDF24, LibreOffice Draw	Lire, annoter, parfois modifier PDF

Édition PDF	Acrobat Pro	PDFsam Basic, LibreOffice Draw	Fusion, découpage, modification simple
Développement / code	Visual Studio	VS Code, VSCodium, Code::Blocks	Programmation
Simulation électronique	Proteus	KiCad, QUCS	Schémas électroniques
PCB électronique	Altium Designer	KiCad	Conception de circuits imprimés
Machine virtuelle	VMware Workstation Pro	VirtualBox	Virtualisation
Compression	WinRAR	7-Zip, PeaZip	Compresser et décompresser