

Université Abderrahmane Mira de Béjaia
Faculté des Sciences Exactes
Département d'informatique

Cours

Logiciels Libres

Introduction

Objectif : Familiarisation AVEC L'outil
Informatique et Internet

Introduction aux Logiciels Libres (Open Source)

- Quelques logos célèbres : Linux, Firefox, LibreOffice, GIMP



- « **La liberté de partager et de collaborer** »

L'informatique a pris beaucoup de place dans nos vies professionnelles et personnelles

Plan du Module (1)

- **Introduction**
- **Chapitre 1: Technologie de l'Information**
 - Définitions
 - Outils
 - **Matériels (Hardware)**
 - **Ordinateurs** : PC, Portable, Serveurs, Station de Travail
 - **Réseaux de Communication**: Routeur, Commutateur, **Modem**, Fibre Optique, **WiFi**
 - **Périphériques**: Imprimantes, Scanners, Claviers, Souris, Caméras, Capteurs
 - **Dispositifs Mobiles**: Smartphones, Tablettes, **PDA**
 - **Puces Intelligentes**:: RFID, NFC, Processeur Embarqués, Capteurs (**IoT**)
 - **Logiciels (Software)**
 - **Systèmes d'Exploitation** : Windows, Linux, macOS, Android, iOS
 - **Applications Métiers**: ERP, CRM, RH, Comptabilité, ...
 - **Logiciels de Bureautique**: Word, Excel, PowerPoint, Google Workspace, Microsoft 365:

Plan du Module (2)

- **Chapitre 1 (suite): Technologie de l'Information**
 - Applications
 - Espaces de Communication : Internet, Intranet, Extranet
 - Multimédia: Audioconférence, Visioconférence
 - Echange de Données Informatisées (EDI)
 - Workflows
- **Chapitre 2: Outils Open Source**
 - Introduction (Historique, Avantages, Inconvénients et Licence)
 - Environnement de Développement:
 - Introduction à **Linux**, aux Editeurs de Code
 - Bureautique (Suite Bureautique : **LibreOffice**)
 - Collaboration (Stockage & Partage)
- Contribuer à un Projet Open Source

Historique

Historique des Logiciels Libres (1)

1. Années 1970 : l'ère du partage

Au début de l'informatique moderne, les chercheurs et universitaires partageaient librement leurs programmes

Le logiciel était souvent fourni avec son code source pour être modifié et adapté

Exemple : le système **UNIX** (développé par AT&T en 1969) a été largement étudié et adapté dans les universités

Historique des Logiciels Libres (2)

2. Années 1980 : naissance du concept moderne

En **1983**, **Richard Stallman**, chercheur au MIT, lance le **projet GNU** (« **GNU's Not Unix!** ») avec l'idée de créer un système d'exploitation entièrement libre

En **1985**, il fonde la **Free Software Foundation (FSF)** et définit les **quatre libertés fondamentales** du logiciel libre :

- Liberté d'utiliser le logiciel,
- Liberté d'étudier son fonctionnement (accès au code source),
- Liberté de le modifier,
- Liberté de le redistribuer.

Historique des Logiciels Libres (3)

3. Années 1990 : diffusion et popularisation

1991 : Linus Torvalds développe le noyau **Linux**, associé aux outils **GNU**, donne naissance au système **GNU/Linux**

1998 : Naissance du terme **Open Source** (source ouverte) pour faciliter l'adoption dans le milieu industriel

Création de la fondation **Apache Software Foundation** et développement de serveurs web libres

Historique des Logiciels Libres (4)

4. Années 2000 : adoption mondiale

Les logiciels libres commencent à rivaliser avec les logiciels propriétaires :

- **Mozilla Firefox** (navigateur web),
- **LibreOffice** (suite bureautique),
- **MySQL** et **PostgreSQL** (bases de données)

Les grandes entreprises (IBM, Google, Red Hat, etc.) commencent à investir massivement dans l'open source

Historique des Logiciels Libres (5)

5. Années 2010 à aujourd’hui : l’open source universel

Les logiciels libres sont au cœur de l’informatique moderne : **Android** (basé sur Linux), **Docker**, **Kubernetes**, **TensorFlow**, etc

Les géants du numérique (Google, Microsoft, Facebook, Amazon) publient de nombreux projets en open source

Historique des Logiciels Libres (6)

6. Extension de la Culture du Libre

La **culture du libre** s'étend à d'autres **domaines**:

- **matériel libre**,
- **ressources éducatives libres**,
- **données ouvertes**

Matériels (Hardware)

Définition du hardware

Le **hardware** désigne l'ensemble des **composants physiques** d'un ordinateur,

par opposition au **software** (logiciels)

→ **Exemple** : clavier, écran, disque dur, carte mère, processeur, carte graphique, carte réseau, etc.

Composants matériels principaux

- **Unité centrale (boîtier / tour)** : contient la carte mère, processeur, mémoire vive (RAM), disque dur/SSD.
- **Processeur** Central Processing Unit (**CPU**) : le « cerveau » de l'ordinateur.
- **Mémoire RAM** : mémoire rapide et temporaire.
- **Disque dur / SSD** : stockage permanent des données.
- **Carte graphique** Graphics Processing Unit (**GPU**) : gestion de l'affichage (et calculs intensifs, IA, jeux vidéo).
- **Carte réseau** : connexion Internet.
- **Périphériques d'entrée** : clavier, souris, scanner.
- **Périphériques de sortie** : écran, imprimante, haut-parleurs.

Types d'ordinateurs et différences (1)

PC de Bureau (Desktop)

- Utilisé à la maison, au bureau, en laboratoire.
- Composants modulables (on peut changer la carte graphique, disque, RAM).
- Relativement bon marché.
- Exemple : un **Dell Optiplex** ou un **HP Pavilion**.
- ➔ **Avantage** : puissance + évolutif.
- ➔ **Limite** : peu transportable.

PC Portable (Laptop)

- Compact, transportable (batterie intégrée).
- Moins puissant qu'un desktop (à prix égal).
- Pièces difficilement remplaçables.
- Exemple : **Lenovo ThinkPad, MacBook Pro**.
- ➔ **Avantage** : mobilité.
- ➔ **Limite** : évolutivité réduite.

Types d'ordinateurs et différences (2)

Serveur

- Conçu pour gérer des données, des applications ou des sites Web 24h/24
- Haute fiabilité, ventilation renforcée, grande mémoire, stockage massif.
Exemple : **Dell PowerEdge**, **HP ProLiant**, serveurs **OVH** ou **AWS**
- ➔ **Avantage** : puissance pour de nombreux utilisateurs
➔ **Limite** : coûteux, nécessite maintenance

Station de Travail (Workstation)

- Ordinateur très puissant pour les professionnels (graphisme, modélisation 3D, IA, calcul scientifique)
- Composants haut de gamme (CPU multi-cœurs, GPU puissant, RAM massive)
- Exemple: **HP Z Workstation**, **Dell Precision**, **Apple Mac Pro**
- ➔ **Avantage** : puissance extrême pour les applications spécialisées
➔ **Limite** : prix élevé

Résumé comparatif

Type	Usage principal	Exemple	Avantage	Limite
PC Bureau	Usage quotidien, bureautique, jeux	HP Pavilion, Dell Optiplex	Puissant et évolutif	Non portable
PC Portable	Mobilité, études, travail nomade	Lenovo ThinkPad, MacBook	Transportable	Peu évolutif
Serveur	Hébergement, bases de données	Dell PowerEdge, HP ProLiant	Haute fiabilité	Coût élevé
Workstation	Calculs, graphisme, IA	HP Z, Dell Precision	Ultra performant	Très cher

Définition du hardware

Le **hardware** désigne l'ensemble des **composants physiques** d'un ordinateur,

par opposition au **software** (logiciels)

→ **Exemple** : clavier, écran, disque dur, carte mère, processeur, carte graphique, etc.

Composants matériels principaux

- **Unité centrale (boîtier / tour)** : contient la carte mère, processeur, mémoire vive (RAM), disque dur/SSD.
- **Processeur (CPU)** : le « cerveau » de l'ordinateur.
- **Mémoire RAM** : mémoire rapide et temporaire.
- **Disque dur / SSD** : stockage permanent des données.
- **Carte graphique (GPU)** : gestion de l'affichage (et calculs intensifs, IA, jeux vidéo).
- **Carte réseau** : connexion Internet.
- **Périphériques d'entrée** : clavier, souris, scanner.
- **Périphériques de sortie** : écran, imprimante, haut-parleurs.

Types d'ordinateurs et différences (1)

PC de Bureau (Desktop)

- Utilisé à la maison, au bureau, en laboratoire.
- Composants modulables (on peut changer la carte graphique, disque, RAM).
- Relativement bon marché.
- Exemple : un **Dell Optiplex** ou un **HP Pavilion**.
- ➔ **Avantage** : puissance + évolutif.
- ➔ **Limite** : peu transportable.

PC Portable (Laptop)

- Compact, transportable (batterie intégrée).
- Moins puissant qu'un desktop (à prix égal).
- Pièces difficilement remplaçables.
- Exemple : **Lenovo ThinkPad**, **MacBook Pro**.
- ➔ **Avantage** : mobilité.

Types d'ordinateurs et différences (2)

Serveur

- Conçu pour gérer des données, des applications ou des sites web 24h/24.
- Haute fiabilité, ventilation renforcée, grande mémoire, stockage massif.
- Exemple : **Dell PowerEdge**, **HP ProLiant**, serveurs **OVH** ou **AWS**.
- ➔ **Avantage** : puissance pour de nombreux utilisateurs.
- ➔ **Limite** : coûteux, nécessite maintenance.

Station de Travail (Workstation)

- Ordinateur très puissant pour les professionnels (graphisme, modélisation 3D, IA, calcul scientifique).
- Composants haut de gamme (CPU multi-cœurs, GPU puissant, RAM massive).
- Exemple : **HP Z Workstation**, **Dell Precision**, **Apple Mac Pro**.
- ➔ **Avantage** : puissance extrême pour les applications spécialisées.
- ➔ **Limite** : prix élevé.

Résumé comparatif

Type	Usage principal	Exemple	Avantage	Limite
PC Bureau	Usage quotidien, bureautique, jeux	HP Pavilion, Dell Optiplex	Puissant et évolutif	Non portable
PC Portable	Mobilité, études, travail nomade	Lenovo ThinkPad, MacBook	Transportable	Peu évolutif
Serveur	Hébergement, bases de données	Dell PowerEdge, HP ProLiant	Haute fiabilité	Coût élevé
Workstation	Calculs, graphisme, IA	HP Z, Dell Precision	Ultra performant	Très cher

1 - Routeur (Router)

Définition

Un **routeur** est un appareil qui connecte plusieurs réseaux entre eux, comme votre réseau domestique à **Internet**. Il décide où envoyer les paquets de données en fonction de leur adresse **IP**

Fonction principale

- Diriger le trafic entre différents réseaux
- Fournir des adresses IP via DHCP
- Filtrer et sécuriser les données (pare-feu intégré)

Exemple concret

- Routeur domestique : **TP-Link Archer AX50**
- Routeur d'entreprise : **Cisco ISR 4000**

Image suggérée

Une photo d'un petit routeur domestique avec antennes

2 - Commutateur (Switch)

Définition

Un **switch** est un appareil qui relie plusieurs appareils dans un même réseau local (LAN) et permet de transférer les données directement entre eux

Fonction principale :

- **Transmettre** les données uniquement à l'appareil concerné (contrôle des adresses MAC)
- **Optimiser** la vitesse et réduire les collisions dans le réseau local

Exemple concret :

- Switch domestique : **Netgear GS105** (5 ports)
- Switch professionnel : **Cisco Catalyst 2960**

3 - Modem

Définition

Un **modem** (modulateur-démodulateur) convertit les signaux numériques de votre ordinateur en signaux analogiques pour le fournisseur d'accès Internet, et vice-versa

Fonction principale :

- Permettre la connexion à Internet via DSL, fibre, câble ou 4G/5G
- Peut être intégré au routeur dans certains modèles

Exemple concret :

- Modem câble : **Arris SURFboard SB6183**
- Modem ADSL : **Netgear DM200**

Différence clé

Le **modem** connecte votre réseau local à Internet.

Le **routeur** distribue Internet aux appareils du réseau local

Image suggérée :

Photo d'un petit modem avec les voyants "Power Online DSL"

4 - Fibre Optique

Définition :

La **fibre optique** est un câble qui transmet des données sous forme de lumière, permettant des vitesses très élevées sur de longues distances

Fonction principale

- **Transfert** ultra-rapide des données (jusqu'à plusieurs Gbps)
- **Réduction** des pertes et des interférences comparé aux câbles cuivre

Exemple concret

- Fibre optique utilisée par Orange, Free, SFR pour Internet haut débit
- Câbles multi-fibres pour les réseaux d'entreprise

Image suggérée

Photo d'un câble de fibre optique avec lumière visible

5 - WiFi

Définition :

Le **Wi-Fi** est une technologie sans fil qui permet de connecter des appareils à un réseau local ou Internet sans câble

Fonction principale :

Connexion des ordinateurs, smartphones et IoT à un réseau.

Communication via ondes radio (2,4 GHz ou 5 GHz)

Exemple concret :

Routeur Wi-Fi domestique : **Asus RT-AX88U**

Points d'accès Wi-Fi pour entreprise : **Ubiquiti UniFi AP**

Différence clé :

Le **Wi-Fi** remplace les câbles Ethernet mais peut être plus lent ou moins stable selon les interférences

Les **câbles Ethernet et fibre** offrent une connexion plus rapide et stable

Image suggérée :

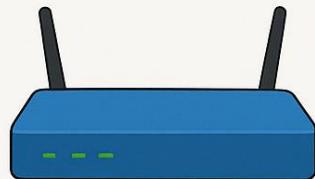
Photo d'un routeur Wi-Fi avec ondes graphiques représentant le signal

Résumé des différences

Matériel	Fonction principale	Réseau concerné	Exemple
Routeur	Connecter plusieurs réseaux	LAN → WAN	TP-Link, Cisco
Switch	Connecter plusieurs appareils dans un LAN	LAN uniquement	Netgear, Cisco
Modem	Convertir signaux pour connexion Internet	LAN → FAI	Arris, Netgear
Fibre optique	Transfert rapide de données	LAN/WAN	Câble optique ISP
Wi-Fi	Connexion sans fil	LAN sans câble	Routeur Wi-Fi Asus

Outils Matériels de Réseaux de Communication

OUTILS MATERIAUX DE RÉSEAUX DE COMMUNICATION



Routeur
Connecter plusieurs réseaux
TP-Link Archer AX50



Commutateur (Switch)
Connecter des appareils dans un réseau local
Netgear GS105



Modem
Convertir les signaux pour connexion Internet
Arris SURFboard SB 6183



Fibre Optique
Transfert rapide de données



Wi-Fi
Connexion sans fil
Routeur Wi-Fi Asus RT-AX88U

Etablir des Classifications de Concepts tout en mettant en Valeur les Différences

- 1. Comparaison des Différents Types de Puces Intelligentes (Déjà Pris)**
- 2. Comparaison des Différents Types d'ordinateurs**
- 3. Comparaison des Différents Outils de Hardware permettant la Connexion à Internet (Wifi, Switch, ...)**

Assimiler les concepts et différences entre libre et propriétaire

- 1.** Définir les 4 libertés du logiciel libre
- 2.** Quelle différence entre « libre » et « gratuit » ? Donnez des exemples
- 3.** Quelle différence entre « libre », “propriétaire” et « gratuit » ? Donnez des exemples

Assimiler les concepts et différences entre libre et propriétaire

- 4.** Comparez : Firefox (libre) vs Google Chrome (propriétaire)
- 5.** Recherchez et classez une dizaine de logiciels que vous utilisez en 2 colonnes : Libres vs Propriétaires

1 - Définition des périphériques matériels

Les **périphériques** sont des composants matériels connectés à un **ordinateur** ou à un **système** pour **entrer, sortir ou stocker des données**. Ils permettent à l'utilisateur d'interagir avec l'ordinateur ou à l'ordinateur d'interagir avec le monde extérieur

On distingue principalement **trois types** :

Périphériques d'**entrée** : permettent à l'utilisateur de saisir des données (ex. **clavier, souris, scanner**)

Périphériques de **sortie** : permettent à l'ordinateur de fournir des informations à l'utilisateur (ex. **imprimante, écran**)

Périphériques d'**entrée/sortie** : peuvent à la fois recevoir et envoyer des données (ex. **caméra, capteurs**)

2 - Péphériques & Différences

Péphérique	Type	Fonction principale	Exemple	Image (idée)
Clavier	Entrée	Permet de saisir du texte et des commandes	Clavier AZERTY ou QWERTY, clavier gaming	Photo d'un clavier filaire ou sans fil
Souris	Entrée	Permet de pointer, cliquer et sélectionner des objets sur l'écran	Souris optique ou laser	Photo d'une souris classique ou ergonomique
Scanner	Entrée	Permet de numériser des documents ou images pour les transformer en fichiers numériques	Scanner à plat, scanner portable	Image d'un scanner plat avec un document à l'intérieur
Imprimante	Sortie	Permet de produire des copies papier de documents numériques	Imprimante laser ou jet d'encre	Photo d'une imprimante en action
Caméra	Entrée/Sortie	Capture des images ou vidéos, parfois pour la visioconférence	Webcam USB, caméra IP	Photo d'une webcam sur un écran
Capteurs	Entrée/Sortie	Mesurent des données physiques ou environnementales et les transmettent à l'ordinateur	Capteur de température, capteur de mouvement	Image d'un capteur connecté à un ordinateur

3 - Explications Supplémentaires

Clavier et souris sont les périphériques les plus courants pour **interagir directement** avec un ordinateur

Scanner et imprimante sont complémentaires : le scanner **numérise**, l'imprimante **matérialise**.

Caméras et capteurs permettent de **relier** le monde réel à l'ordinateur, **utilisés** dans la **visioconférence**, la **surveillance**, ou la **domotique**

4 - Dispositifs mobiles – Définition générale

Les **dispositifs mobiles** sont des appareils électroniques portables permettant d'effectuer diverses tâches informatiques et de communication en mobilité.

Ils se distinguent par leur **taille compacte**, leur **autonomie sur batterie**, et leur capacité à se connecter à internet et à d'autres appareils.

Composants matériels typiques :

- **Écran tactile** : principal moyen d'interaction (smartphone, tablette).
- **Processeur** : “cerveau” de l'appareil, gère les applications et le système.

4 - Dispositifs mobiles – Définition générale (2)

Composants matériels typiques :

- **Écran tactile** : principal moyen d'interaction (**smartphone, tablette**)
- **Processeur** : “cerveau” de l'appareil, gère les applications et le système
- **Mémoire RAM et stockage interne** : pour exécuter des apps et stocker des données
- **Batterie** : alimentation portable
- **Caméra** : frontale et/ou arrière pour **photos, vidéos, visioconférences**
- **Capteurs** : **GPS, accéléromètre, gyroscope**, capteur de **proximité, de Température, Agriculture, Urgences**
- **Connectivité** : **Wi-Fi, Bluetooth, 4G/5G, NFC** (Near Field Communication)

4 – Comparaison des Dispositifs mobiles Principaux

Dispositif	Taille et Portabilité	Usage principal	Exemples	Particularités matérielles
Smartphone	Très compact (5–7 pouces)	Communication (appels, SMS, messagerie), navigation internet, applications diverses	iPhone 15, Samsung Galaxy S23	Écran tactile haute résolution, multiples capteurs (empreinte, reconnaissance faciale), caméras avancées, batterie intégrée
Tablette	Moyen (7–13 pouces)	Consultation de contenus multimédias, jeux, travail léger, lecture	iPad, Samsung Galaxy Tab	Écran tactile plus grand, processeur souvent puissant pour multimédia, peut intégrer stylet, moins de portabilité qu'un smartphone
Assistant Numérique Personnel (PDA)	Compact à moyen (3–7 pouces)	Gestion personnelle : agenda, contacts, notes, petites applications	Palm Pilot, HP iPAQ	Clavier ou écran tactile, processeur et mémoire limités, souvent pas de téléphonie, utilisé historiquement avant smartphones

3 - Explications avec Exemples d'usage

Smartphone

Appel vidéo avec Zoom ou Teams

Navigation GPS en temps réel

Prise de photo ou vidéo haute qualité

Paiement sans contact via NFC (**Near Field Communication**)

Tablette

Lecture de livres numériques et journaux

Visionnage de films ou séries sur grand écran

Dessin ou prise de notes avec un stylet

Cours en ligne ou présentations interactives

PDA (**moins courant aujourd'hui**)

- Organisation de l'agenda et contacts professionnels

- Prise de notes et mémos rapides

- Synchronisation avec ordinateur pour transfert d'informations

4 - Suggestion d'images

Smartphone : photo d'un iPhone ou Samsung Galaxy posé sur une table

Tablette : image d'une iPad avec stylet et applications ouvertes

PDA : photo vintage d'un Palm Pilot ou HP iPAQ

1 - Puces intelligentes – Définition générale

Les **puces intelligentes** (ou **smart chips**) sont des circuits électroniques miniaturisés capables de traiter des données, de communiquer et d'interagir avec leur environnement

Elles sont au cœur des **objets connectés** et des systèmes embarqués

Caractéristiques matérielles typiques :

Microprocesseur intégré pour le traitement de l'information

Mémoire pour stocker des programmes et données

Modules de communication (radio, NFC, Wi-Fi)

Capteurs pour mesurer des paramètres physiques ou environnementaux

2 – Comparaison des Principaux Types de Puces intelligentes

Type de puce	Fonction principale	Exemples d'usage	Particularités matérielles
RFIC (Radio Frequency Integrated Circuit)	Gestion et traitement des signaux radio	Antennes Wi-Fi, modules Bluetooth, RFID	Convertit les signaux radio en signaux numériques; miniaturisé pour intégration dans smartphones, objets connectés
NFC (Near Field Communication)	Communication sans fil à courte portée (≤ 10 cm)	Paiement sans contact (Apple Pay, Google Pay), cartes d'accès, billets électroniques	Intégrée dans smartphones et cartes intelligentes; faible consommation d'énergie; fonctionne à 13,56 MHz
Processeurs embarqués	Exécution de programmes et contrôle des systèmes	Microcontrôleurs Arduino, Raspberry Pi Pico, ESP32	CPU intégré + mémoire RAM/ROM; faible consommation; gère capteurs et périphériques externes
Capteurs IoT (Internet of Things)	Mesure de paramètres physiques ou environnementaux	Capteurs de température, humidité, pression, mouvement (IMU)	Généralement très petits; envoient les données à un microcontrôleur via filaire ou sans fil; essentiels pour smart home, santé, industrie

3 - Explication Avec Exemples – Pratiques

RFIC (Radio-Frequency Integrated Circuit)

- Permet à un smartphone de capter le Wi-Fi et le Bluetooth simultanément

Exemple : puce Wi-Fi Broadcom BCM4375 intégrée aux smartphones.

NFC (Near Field Communication)

Permet de payer simplement en approchant son smartphone d'un terminal

Exemple : puce NFC NXP PN532 dans un smartphone Android

Processeurs embarqués

Contrôle un robot ou un capteur **intelligent**

Exemple : microcontrôleur ESP32 utilisé pour un thermostat intelligent

Capteurs IoT

- Mesurent la température et envoient les données à une application mobile

Exemple : capteur de température DHT22 dans une station météo connectée

4 - Suggestions d'Images

RFIC : image d'un circuit RF miniature intégré sur une carte électronique.

NFC : smartphone approchant une borne de paiement.

Processeurs embarqués : microcontrôleur Arduino ou ESP32 avec câbles et LED.

Capteurs IoT : petit capteur de température ou capteur de mouvement sur un circuit imprimé.

Logiciels (Software)

1 - Systèmes d'Exploitation- Définition Générale

Un **système d'exploitation (OS)** est un logiciel qui permet à un ordinateur, un smartphone ou un autre dispositif informatique de **fonctionner** et de **gérer les ressources matérielles** (processeur, mémoire, stockage, périphériques). Il fournit aussi une **interface utilisateur** pour interagir avec le matériel et les applications.

Fonctions principales d'un OS :

Gestion du **processeur, mémoire et stockage**

Gestion des **périphériques** (clavier, souris, écran, imprimante)

Exécution des **applications**

Sécurité et contrôle des **droits utilisateurs**

Interface graphique (GUI) ou interface en ligne de commande (CLI)

2 – Comparaison des Principaux Systèmes d'Exploitation

OS	Type de dispositifs	Usage principal	Exemples	Particularités
Windows	PC de bureau, ordinateurs portables	Informatique générale, bureautique, jeux vidéo, entreprises	Windows 10, Windows 11	Interface conviviale, compatible avec de nombreux logiciels, grande communauté d'utilisateurs
Linux	PC, serveurs, systèmes embarqués	Informatique avancée, serveurs web, programmation	Ubuntu, Debian, Fedora	Open-source, personnalisable, stable, sécurisé, gratuit
macOS	Ordinateurs Apple	Bureautique, création multimédia, programmation	macOS Ventura, macOS Sonoma	Interface élégante, intégration avec l'écosystème Apple, sécurité renforcée
Android	Smartphones, tablettes	Applications mobiles, jeux, communication	Samsung Galaxy, Google Pixel	Open-source (partiellement), personnalisable par les fabricants, large choix d'applications
iOS	iPhone, iPad	Applications mobiles, jeux, communication	iPhone 15, iPad Pro	Fermé et sécurisé, optimisé pour appareils Apple, interface intuitive

3 - Explication Avec Exemples – Pratiques

Windows

Utilisé pour Microsoft Office, jeux PC et logiciels professionnels.

Exemple : création de documents Word et Excel sur Windows 11.

Linux

Serveurs web, programmation et sécurité informatique.

Exemple : hébergement d'un site web sur Ubuntu Server.

macOS

Montage vidéo, graphisme, programmation iOS.

Exemple : édition vidéo avec Final Cut Pro sur macOS Ventura.

Android

Téléphone mobile pour apps, jeux, messagerie et navigation.

Exemple : utiliser Google Maps et WhatsApp sur Samsung Galaxy.

iOS

Smartphone et tablette Apple, apps et sécurité.

Exemple : utiliser FaceTime et Apple Wallet sur iPhone 15.

4 - Explication Avec Exemples – Pratiques

Windows

Utilisé pour Microsoft Office, jeux PC et logiciels professionnels.

Exemple : création de documents Word et Excel sur Windows 11.

Linux

Serveurs web, programmation et sécurité informatique.

Exemple : hébergement d'un site web sur Ubuntu Server.

macOS

Montage vidéo, graphisme, programmation iOS.

Exemple : édition vidéo avec Final Cut Pro sur macOS Ventura.

Android

Téléphone mobile pour apps, jeux, messagerie et navigation.

Exemple : utiliser Google Maps et WhatsApp sur Samsung Galaxy.

iOS

Smartphone et tablette Apple, apps et sécurité.

Exemple : utiliser FaceTime et Apple Wallet sur iPhone 15.

4 - Suggestions d'Images

Windows : écran d'ordinateur avec bureau Windows 11.

Linux : terminal Linux avec interface graphique Ubuntu.

macOS : écran MacBook affichant le bureau macOS.

Android : smartphone Android affichant le menu d'applications.

iOS : iPhone avec écran d'accueil iOS.

Voici une présentation
claire des **outils**
logiciels (software) –
applications métiers,
avec leurs
différences, exemples
et explications

I. Applications métiers –

Définition générale

Les **applications métiers** sont des logiciels conçus pour répondre aux besoins spécifiques d'une entreprise ou d'une organisation

Elles permettent de:

- gérer efficacement les processus internes,
- automatiser certaines tâches et
- améliorer la productivité

Fonctions principales

- Gestion des ressources humaines
- Gestion des finances et comptabilité
- Gestion de la relation client
- Gestion de la production, stocks et approvisionnement
- Analyse et reporting pour la prise de décision

2. Comparaison des principaux types d'applications métiers

Type d'application	Fonction principale	Exemples	Particularités
ERP (Enterprise Resource Planning)	Planification et gestion intégrée des ressources de l'entreprise (production, stock, finances, RH)	SAP, Oracle NetSuite, Microsoft Dynamics 365	Intègre plusieurs modules pour centraliser les données et automatiser les processus
CRM (Customer Relationship Management)	Gestion des relations et interactions avec les clients	Salesforce, Zoho CRM, HubSpot	Suit les ventes, marketing, service client ; améliore la satisfaction et fidélisation
Logiciels de gestion des ressources humaines (RH)	Gestion du personnel, paie, formation, recrutement	Workday, Sage HR, ADP	Automatisation des tâches RH, suivi des performances et conformité légale
Logiciels de comptabilité	Gestion financière et comptable	QuickBooks, Sage Comptabilité, Xero	Suivi des dépenses, facturation, rapports financiers et audits
Logiciels de gestion de projet	Planification, suivi et coordination de projets	Microsoft Project, Asana, Trello	Gestion des tâches, calendrier, budgets et ressources

3. Explications avec exemples pratiques

- **ERP**

- Exemple : SAP utilisé pour suivre les commandes, gérer les stocks et générer les rapports financiers.
 - Avantage : centralise toutes les informations de l'entreprise dans un seul système.

- **CRM**

- Exemple : Salesforce utilisé pour suivre les prospects et clients, envoyer des emails marketing personnalisés.
 - Avantage : améliore la relation client et optimise les ventes.

- **Logiciels RH**

- Exemple : Workday utilisé pour gérer la paie et le suivi des employés.
 - Avantage : réduit le temps consacré aux tâches administratives RH.

- **Logiciels de comptabilité**

- Exemple : QuickBooks utilisé pour gérer les factures, paiements et rapports financiers.
 - Avantage : simplifie la comptabilité et assure la conformité fiscale.

- **Logiciels de gestion de projet**

- **Exemple :** Trello pour organiser un projet marketing avec suivi des tâches et deadlines.
 - **Avantage :** facilite la collaboration et le suivi des objectifs.

4. Suggestions d'images

- **ERP** : capture d'écran d'un tableau de bord SAP ou Oracle NetSuite
- **CRM** : interface Salesforce montrant le suivi des clients
- **Logiciels RH** : interface Workday avec gestion des employés
- **Logiciels de comptabilité** : capture QuickBooks avec factures et rapports
- **Gestion de projet** : tableau Trello ou diagramme Gantt Microsoft Project

**Voici une
présentation claire et
complète des outils
logiciels de
bureautique, leurs
différences, exemples
et explications**

1. Outils de bureautique – Définition générale

Les **logiciels de bureautique** sont des applications qui permettent de **produire, organiser, analyser et partager des informations** dans un contexte **professionnel ou académique**

Fonctions principales

- Rédaction de documents
- Gestion et analyse de données
- Création de présentations
- Collaboration et partage en temps réel

2. Comparaison des principaux outils de bureautique

Outil	Fonction principale	Exemples	Particularités
Word	Traitement de texte	Microsoft Word, Google Docs	Création de documents, mise en forme avancée, insertion d'images, tableaux et références
Excel	Tableur et analyse de données	Microsoft Excel, Google Sheets	Gestion de tableaux, formules, graphiques, calculs financiers et statistiques
PowerPoint	Présentations multimédias	Microsoft PowerPoint, Google Slides	Création de diaporamas, intégration d'images, vidéos, animations et transitions
Outils de travail collaboratif	Travail collaboratif	Google Workspace (Docs, Sheets, Slides), Microsoft 365 (Word,	Partage en temps réel, commentaires, édition simultanée, intégration

3. Explications avec exemples pratiques

- **Word**
 - Exemple : rédiger un rapport ou un mémoire avec titres, tableaux et images.
 - Avantage : mise en forme professionnelle et export facile en PDF.
- **Excel**
 - Exemple : créer un tableau budgétaire ou analyser des données commerciales avec graphiques et formules.
 - Avantage : automatisation des calculs et visualisation graphique des données.
- **PowerPoint**
 - Exemple : créer une présentation pour un cours ou une réunion avec animations et images.
 - Avantage : support visuel efficace pour exposer des idées.
- **Outils de collaboration**
 - **Exemple :** Google Docs pour rédiger un document simultanément avec plusieurs collègues.
 - **Avantage :** permet le travail à distance, suivi des modifications et sauvegarde dans le cloud.

4. Suggestions d'images

- **Word** : capture d'écran d'un document avec texte et images
- **Excel** : tableau avec formules et graphique
- **PowerPoint** : diapositive avec texte et images
- **Outils de collaboration** : interface Google Docs ou Microsoft Teams avec plusieurs utilisateurs connectés

voici une présentation
claire et complète des

**espaces de
communication, leurs
différences, avec
exemples et
explications**

1. Espaces de communication – Définition générale

Les **espaces de communication** sont des environnements permettant le **partage d'informations et la communication** entre utilisateurs via des réseaux informatiques

1. Espaces de communication – Définition générale

Selon leur **accessibilité et leur portée**,
on distingue :

- Internet,
- Intranet et
- Extranet

2. Comparaison des principaux espaces de communication

Espace	Définition	Accès	Exemples	Particularités
Internet	Réseau mondial public interconnectant des millions d'ordinateurs et serveurs	Ouvert à tous	Google, Wikipedia, YouTube	Accessible partout, utilisation publique, contient sites web, emails, réseaux sociaux, services cloud
Intranet	Réseau privé interne d'une organisation	Réservé aux employés ou membres de l'organisation	Portail interne d'une entreprise, réseau universitaire	Protégé par mot de passe, permet de partager documents internes, annuaires, procédures et applications métier
Extranet	Extension sécurisée de l'intranet donnant accès à des partenaires externes	Accès limité aux partenaires ou clients autorisés	Portail fournisseur, site de suivi client	Facilite la collaboration avec partenaires externes tout en protégeant les données internes de l'entreprise

5. Explications avec exemples pratiques

• Internet

- Exemple : naviguer sur Google pour rechercher des informations ou regarder une vidéo sur YouTube.
- Usage : communication globale, accès à des informations publiques et services variés.

• Intranet

- Exemple : les employés d'une entreprise accèdent à l'annuaire interne, aux documents RH ou aux outils de gestion de projet.
- Usage : communication et partage d'informations internes à l'entreprise.

• Extranet

- **Exemple :** un fournisseur accède au portail d'une entreprise pour suivre les commandes et les stocks.

• **Usage :** collaboration sécurisée avec des partenaires ou clients externes tout en gardant les informations confidentielles.

4. Suggestions d'images

- **Internet** : icône de globe ou photo de personne naviguant sur un site web public
- **Intranet** : capture d'écran d'un portail interne d'entreprise ou université
- **Extranet** : image montrant un accès sécurisé pour partenaires via mot de passe ou VPN

Thèmes déjà Pris

- 1. Comparaison des Différents Types de Puces Intelligentes (Déjà Pris)**
- 2. Comparaison des Différents Types d'ordinateurs**
- 3. Définir les 4 libertés du logiciel libre**
- 4. Comparaison entre Firefox et Google Chrome**
- 5. Comparaison entre Windows et Linux**

Nouveaux Thèmes

1. Quelle différence entre «libre» et «gratuit» ? Donnez des exemples
2. Quelle différence entre «libre», “propriétaire” et «gratuit» ? Donnez des exemples

Nouveaux Thèmes

- 4.** Comparez : Firefox (libre)
vs Google Chrome
(propriétaire)
- 5.** Recherchez et classez une
dizaine de logiciels que vous
utilisez en 2 colonnes : Libres
vs Propriétaires

Nouveaux Thèmes

1. Comparaison des principaux types d'applications métiers
2. Comparaison des principaux outils de bureautique
3. Comparaison des principaux espaces de communication

**Voici une présentation
claire des outils
multimédia pour la
communication, avec
leurs différences,
exemples et
explications**

1. Multimédia – Définition générale

Le **multimédia** combine plusieurs types de médias (texte, audio, image, vidéo, animation) pour faciliter la **communication et l'échange d'informations**

Dans le contexte de la **communication professionnelle**, les outils **multimédias** incluent l'**audioconférence** et la **visioconférence**

2. Comparaison des principaux outils

Outil	Définition	Média utilisé	Exemples	Particularités
Audio-conférence	Réunion ou échange en temps réel uniquement via la voix	Audio uniquement	Zoom (mode audio), Teams (appel vocal), Skype audio	Permet des discussions rapides sans vidéo, nécessite moins de bande passante, idéal pour appels téléphoniques de groupe ou réunions à distance
Visio-conférence	Réunion ou échange en temps réel avec audio et vidéo	Audio + Vidéo	Zoom vidéo, Microsoft Teams vidéo, Google Meet	Permet interaction visuelle, partage d'écran, supports multimédias, rapproche les participants comme en

3. Explications avec exemples pratiques

• **Audioconférence**

- Exemple : un appel téléphonique de groupe entre collègues situés dans différents bureaux pour discuter d'un projet.
- Avantage : facile à organiser, faible consommation de données, pas besoin de caméra.

• **Visioconférence**

- Exemple : réunion Zoom avec caméra pour présentation de projet, partage de documents et discussion en direct.
- Avantage : communication plus interactive, expressions faciales visibles, possibilité de montrer des présentations ou tableaux.

4. Suggestions d'images

- **Audioconférence** : icône de téléphone ou de haut-parleur avec plusieurs participants
- **Visioconférence** : capture d'écran Zoom ou Teams avec plusieurs participants en vidéo et partage d'écran

**Voici une présentation claire de
l'Échange de
Données
Informatisées (EDI)
avec définition, explications et
exemples**

1. Définition

L'Échange de Données Informatisées (EDI – Electronic Data Interchange) est un système automatisé permettant l'échange structuré de documents et d'informations entre organisations ou systèmes informatiques, sans intervention humaine directe.

1. Définition

En d'autres termes, l'EDI permet à des entreprises ou systèmes différents de communiquer et partager des données de manière standardisée et sécurisée, souvent pour des transactions commerciales ou logistiques.

2. Objectifs principaux

- Automatiser les échanges de documents (commandes, factures, bons de livraison...)
- Réduire les erreurs liées à la saisie manuelle
- Accélérer le traitement des transactions
- Garantir la sécurité et la traçabilité des données

3. Exemples d'EDI

Type de document	Exemple d'usage
Commande fournisseur	Une entreprise envoie automatiquement un bon de commande à son fournisseur via EDI pour réapprovisionner un stock
Facturation électronique	Une société envoie ses factures directement au service comptable de son client via un format EDI standard (comme XML ou EDIFACT)
Bons de livraison	Un transporteur reçoit automatiquement les informations de livraison via EDI pour préparer les expéditions
Données douanières	Les entreprises exportatrices transmettent des déclarations douanières directement aux douanes

4. Avantages

- Rapidité et automatisation des échanges
- Réduction des erreurs et duplications
- Standardisation des formats de données (ex. EDIFACT, XML, JSON)
- Sécurité et confidentialité améliorées

5. Suggestion d'images

- Schéma montrant deux entreprises connectées via un **flux EDI** pour envoyer des commandes et factures
- Icône représentant des documents numériques échangés automatiquement entre systèmes

**Voici une présentation
claire des workflows,
avec définition,
explications et
exemples**

1. Définition

Un **workflow** (ou flux de travail) est une **suite organisée et automatisée d'activités ou de tâches** qui permet de réaliser un processus spécifique au sein d'une organisation

Caractéristiques principales

- Séquence de tâches ou activités
- Participants ou rôles définis
- Règles et conditions pour déclencher les étapes suivantes
- Possibilité d'automatisation via un logiciel

2. Objectifs des workflows

- Standardiser les processus métiers
- Améliorer la productivité et la coordination
- Réduire les erreurs et les retards
- Assurer la traçabilité des actions

3. Exemples de workflows

Type de workflow	Exemple concret
Workflow de validation	Un employé soumet une note de frais ; le manager la valide ; le service comptable la traite pour remboursement
Workflow de gestion de projet	Un projet passe par différentes étapes : planification → conception → validation → livraison
Workflow de support client	Une demande d'assistance est reçue; un agent est assigné ; le problème est résolu ; le client reçoit une confirmation
Workflow de publication	Un document est rédigé → révisé par l'éditeur → validé → publié → diffusé dans les médias

4. Avantages

- Automatisation des tâches répétitives
- Meilleure coordination entre services
- Réduction des délais et erreurs
- Suivi et reporting faciles des processus

5. Suggestions d'images

- Schéma en flèche ou diagramme montrant les étapes d'un workflow avec des rôles et transitions
- Exemple visuel d'un **workflow numérique** dans un logiciel (ex. Jira, Trello, Monday.com)

Université Abderrahmane Mira de Béjaia
Faculté des Sciences Exactes
Département d'informatique

Cours Logiciels Libres

Partie II: Outils Open

Source

10 Nov 2025

Introduction au cours : Logiciels Libres (Open Source)

« Savez-vous que des millions de personnes dans le monde utilisent des logiciels qu'elles peuvent modifier, améliorer et partager librement ? C'est l'univers fascinant des logiciels libres et de l'open source ! »

Les **logiciels libres** sont des programmes qui respectent la **liberté des utilisateurs** : ils peuvent **utiliser, étudier, modifier et redistribuer** le logiciel. Contrairement aux logiciels propriétaires (comme Microsoft Office ou Photoshop), les logiciels libres favorisent la **collaboration, l'innovation et l'autonomie**.

Quelques exemples célèbres de Logiciels Libres (Open Source)

- **Quelques exemples célèbres :**
- **Linux** : système d'exploitation libre
- **Firefox** : navigateur web
- **LibreOffice** : suite bureautique
- **GIMP** : retouche photo
- **Python** et beaucoup de bibliothèques open source
- L'open source ne concerne pas seulement la technique ; c'est **une philosophie et une communauté** qui promeut le partage et la transparence.

Objectifs du cours

- Comprendre ce qu'est un logiciel libre et la différence avec le logiciel propriétaire.
- Découvrir les licences libres et leur rôle.
- Explorer des exemples concrets de logiciels libres.
- Savoir installer et utiliser des logiciels libres.
- Découvrir la communauté et contribuer à des projets open source.

Plan du cours

1 : Introduction aux logiciels libres et open source

- Définitions : logiciel libre, open source, freeware, shareware
- Historique : Richard Stallman et le mouvement GNU
- Différence entre libre et gratuit

2 : Licences et aspects juridiques

- Licence GPL, MIT, Apache, Creative Commons
- Droits et obligations des utilisateurs
- Importance des licences pour la collaboration

3 : Écosystème des logiciels libres

- Systèmes d'exploitation (Linux, BSD)
- Logiciels bureautiques (LibreOffice, OpenOffice)
- Logiciels multimédia et développement (GIMP, Blender, VS Code)

Références et ressources

- **Livres :**

- “Free Software, Free Society” – Richard Stallman
- “The Cathedral & the Bazaar” – Eric S. Raymond

- **Sites web :**

- GNU Project
- Open Source Initiative
- Linux.com

- **Tutoriels en ligne :**

- [Ubuntu tutorials](#)
- Git & GitHub

Petits travaux à réaliser chez soi

- **Explorer un logiciel libre**

- Installer et tester **LibreOffice** ou **GIMP**
- Rédiger un court compte rendu : fonctionnalités, avantages et limites

- **Découverte de Linux**

- Installer une machine virtuelle avec **Ubuntu** ou **Linux Mint**
- Naviguer dans le système, créer un fichier texte et un dossier

- **Introduction à GitHub**

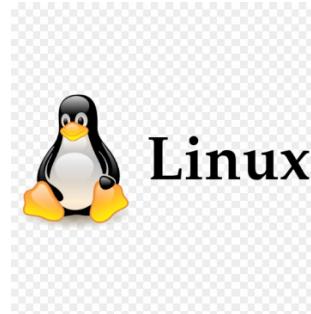
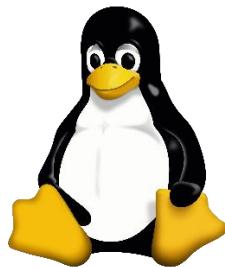
- Créer un compte GitHub
- Forker un petit projet open source
- Essayer de créer un **commit**

- **Réflexion personnelle**

- Faire une liste de logiciels que tu utilises tous les jours et identifier s'ils sont libres ou propriétaires
- Écrire un paragraphe sur “Pourquoi le logiciel libre est important pour l'avenir ?”

Introduction aux Logiciels Libres (Open Source)

Logos de Logiciels Libres



Introduction aux Logiciels Libres (Open Source)

Références

- LibreOffice : <https://www.libreoffice.org/>
- Mozilla Firefox :
<https://www.mozilla.org/firefox/>
- GIMP : <https://www.gimp.org/>
- Inkscape : <https://inkscape.org/>
- Blender : <https://www.blender.org/>
- VLC : <https://www.videolan.org/vlc/>
- Thunderbird : <https://www.thunderbird.net/>

Références

- Ubuntu : <https://ubuntu.com/>
- Debian : <https://www.debian.org/>
- Fedora : <https://getfedora.org/>
- Arch Linux : <https://archlinux.org/>
- Linux Mint : <https://linuxmint.com/>

Objectifs du cours

- Comprendre ce qu'est un logiciel libre et la différence avec le logiciel propriétaire.
- Découvrir l'historique et la philosophie du mouvement du logiciel libre.
- Identifier des exemples concrets de logiciels libres.
- Comprendre pourquoi le libre est important pour la société et la technologie.

Qu'est-ce qu'un logiciel libre ?

Un programme informatique qui respecte **4 libertés fondamentales** :

- **Liberté d'utiliser** le logiciel pour tout usage.
- **Liberté d'étudier** son fonctionnement (accès au code source).
- **Liberté de modifier** le logiciel pour l'adapter à ses besoins.
- **Liberté de redistribuer** des copies, modifiées ou non.
-  Ne pas confondre : **Libre ≠ Gratuit**.

Historique et philosophie

- **1983** : Richard Stallman lance le **projet GNU**
- **1985** : création de la **Free Software Foundation (FSF)**
- **1991** : Linus Torvalds publie le noyau **Linux**
- **Années 2000** : popularisation du terme **Open Source (OSI)**
- **Valeurs clés** : **liberté, transparence, collaboration, innovation**

Logiciel libre vs logiciel propriétaire

- **Libre** : accès au code source, modification et partage autorisés.
- **Propriétaire** : code fermé, restrictions d'usage, dépendance à l'éditeur.
- **Exemple :**
 - Libre → Linux, LibreOffice, Firefox
 - Propriétaire → Windows, Microsoft Office, Photoshop

Exemples

- le système d'exploitation GNU/Linux,
- les suites bureautiques comme LibreOffice, le navigateur web Firefox et le lecteur multimédia VLC

Exemples

- D'autres exemples populaires incluent
- le logiciel de retouche d'image GIMP,
- l'outil d'édition audio Audacity et
- le serveur web Apache

Exemples

Systèmes d'exploitation

- [GNU/Linux](#) : Un système d'exploitation libre qui existe sous de nombreuses formes comme [Ubuntu](#), [Debian](#) ou [Fedora](#).

Bureautique

- [LibreOffice](#) : Une suite bureautique complète avec un traitement de texte, un tableur et un logiciel de présentation, souvent comparée à Microsoft Office.

Internet et réseau

- [Firefox](#) : Un navigateur web populaire connu pour le respect de la vie privée.
- [Thunderbird](#) : Un client de messagerie.
- [Apache HTTP Server](#) : Un serveur web très utilisé.
- [FileZilla](#) : Un client FTP pour le transfert de fichiers.

Exemples

Création et multimédia

- VLC Media Player : Un lecteur multimédia capable de lire presque tous les formats audio et vidéo sans codecs supplémentaires.
- GIMP : Un logiciel puissant de manipulation d'images, souvent vu comme une alternative à Photoshop.
- Audacity : Un éditeur et enregistreur audio.
- Blender : Un outil de création 3D et de modélisation.
- Inkscape : Un éditeur de graphiques vectoriels.

Utilitaires

- 7-Zip : Un logiciel de compression et de décompression de fichiers.
- Pidgin : Un client de messagerie instantanée

Exemples de logiciels libres célèbres

- **Systèmes d'exploitation** : Linux, BSD.
- **Base de systèmes d'exploitation** : Les systèmes d'exploitation basés sur la licence BSD, tels que FreeBSD, OpenBSD et NetBSD, sont des alternatives à Linux.
- **Bureautique** : LibreOffice, OnlyOffice.
- **Navigation Web** : Mozilla Firefox.
- **Graphisme & multimédia** : GIMP, Inkscape, Blender, VLC.
- **Développement** : Python, R, Eclipse, Git.

BSD

- BSD fait référence à la fois à une licence de logiciel libre (la licence BSD) et à une famille de systèmes d'exploitation (les systèmes d'exploitation BSD comme FreeBSD, OpenBSD, etc.).
- La licence BSD est une licence « permissive » qui autorise le libre-usage, la modification et la distribution de logiciels, avec des restrictions minimales, tandis que les systèmes BSD sont des systèmes d'exploitation open source inspirés d'UNIX.

Systèmes d'exploitation BSD

- **Basés sur UNIX** : Une famille de systèmes d'exploitation dérivés d'UNIX, dont les principaux représentants sont FreeBSD, OpenBSD et NetBSD.
- **FreeBSD** : Un système d'exploitation gratuit et open source, souvent utilisé pour les serveurs et les plateformes embarquées.
- **OpenBSD** : Connue pour mettre l'accent sur la sécurité.
- **NetBSD** : Connue pour sa portabilité sur une large gamme de matériels.
- **Utilisations** : Bien qu'il s'agisse d'alternatives à Linux, les systèmes BSD sont utilisés dans de nombreux environnements, y compris au cœur de certains équipements de réseaux et de stockage commerciaux.

Git

- **Git** est un logiciel de gestion de versions décentralisé. C'est un logiciel libre et gratuit, créé en 2005 par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Git est maintenu par Junio C Hamano depuis juillet 2005

R

- R est un langage de programmation et un logiciel libre destiné aux statistiques et à la science des données soutenu par la R Foundation for Statistical Computing. Il fait partie de la liste des paquets GNU^[3] et est écrit en C, Fortran et R.

Pourquoi utiliser du libre ?

- **Liberté** d'utilisation et de partage.
- **Sécurité** (code audité par la communauté).
- **Économie** (réduction des coûts de licences).
- **Innovation** (amélioration continue par les contributeurs).
- **Indépendance technologique** (éviter le verrou propriétaire).

Eclipse

- Eclipse est un **environnement de développement intégré (IDE) open source et libre** écrit en Java. Il est gratuit, personnalisable grâce à des plugins, et prend en charge de nombreux langages de programmation, bien qu'il soit principalement axé sur Java. Le projet est géré par la [Fondation Eclipse](#).

Exercice pratique

- Faites la liste des logiciels que vous utilisez chaque jour.
- Identifiez lesquels sont **libres** et lesquels sont **propriétaires**.
- Réfléchissez : qu'est-ce qui change pour vous si vous remplacez un logiciel propriétaire par son équivalent libre ?

Travail à la maison

- Installer **LibreOffice** ou **GIMP** et tester ses fonctionnalités.
- Rédiger un court compte rendu : points positifs et limites par rapport à un équivalent propriétaire.

Références

- GNU Project : <https://www.gnu.org/>
- Open Source Initiative :
<https://opensource.org/>
- Linux.com : <https://www.linux.com/>
- Tutoriels Ubuntu :
<https://ubuntu.com/tutorials>

Conclusion

Le logiciel libre, ce n'est pas seulement du code : c'est une **philosophie de liberté et de collaboration**.

- ☞ Il est déjà présent dans notre quotidien (ordinateurs, smartphones, Internet).
- ☞ Apprendre le libre, c'est entrer dans une **culture d'innovation partagée**.

Licences et aspects juridiques

Licences et aspects juridiques

- **Objectifs** : comprendre les droits et obligations liés au logiciel libre.
- **Contenu** :
 - Différence entre **GPL, MIT, Apache, BSD**
 - Notion de copyleft vs permissif
 - Exemples d'utilisation en entreprise et recherche
- **Travail pratique** : comparer deux projets open source (par exemple sur GitHub) et identifier leur licence.

Objectifs du cours

- Comprendre pourquoi les licences sont indispensables dans le monde du logiciel libre.
- Distinguer les grandes familles de licences (copyleft vs permissives).
- Connaître les licences les plus utilisées (GPL, MIT, Apache, BSD).
- Être capable d'identifier et de choisir une licence adaptée à un projet.

Pourquoi une licence ?

- Définir les droits et obligations des utilisateurs et développeurs.
- Empêcher l'appropriation privative d'un logiciel libre.
- Garantir la liberté d'utilisation, de modification et de redistribution.
- Favoriser la collaboration et la confiance entre contributeurs.

Les grandes catégories de licences

- **Copyleft (fortes obligations de partage)**
 - Exemples : GNU GPL, AGPL, LGPL
 - Principe : toute modification doit rester libre
 - Avantage : protège la liberté
 - Limite : moins de compatibilité avec certains usages commerciaux
- **Licences permissives (plus flexibles)**
 - Exemples : MIT, Apache, BSD
 - Principe : grande liberté, possibilité d'usage propriétaire
 - Avantage : adoption large
- Limite : risque d'appropriation privative

Licence GPL (General Public License)

- Crée par Richard Stallman dans le cadre du projet GNU.
- Licence copyleft forte.
- Toute version modifiée ou redistribuée doit être sous GPL.
- Exemples de projets : Linux, WordPress, GCC.
- Avantage : préserve la liberté.
- Limite : moins attrayante pour les entreprises voulant garder du code fermé.

Licence MIT

- Très simple et courte.
- Permet l'utilisation, modification, redistribution, y compris commerciale.
- Seule obligation : mentionner le copyright et la licence.
- Exemples : jQuery, Node.js, Ruby on Rails.
- Avantage : adoption massive.
- Limite : possibilité d'appropriation fermée.

Licence Apache 2.0

- Licence permissive avec clauses juridiques solides (brevets).
- Compatible avec usage commercial et propriétaire.
- Exemples : Apache HTTP Server, Hadoop, Android.
- Avantage : adoptée par de grandes entreprises.
- Limite : moins protectrice que la GPL.

Comparaison simplifiée

- **GPL** : protège fortement la liberté → idéale pour les projets communautaires.
- **MIT** : ultra-simple et permissive → idéale pour adoption rapide.
- **Apache 2.0** : permissive + protection juridique → idéale pour entreprises.

Exercice pratique

- Allez sur **GitHub** et choisissez deux projets libres.
- Identifiez leur licence (GPL, MIT, Apache, autre).
- Comparez leurs droits et obligations.
- Réfléchissez : quelle licence choisiriez-vous pour votre propre projet, et pourquoi ?

Références

- Free Software Foundation (FSF) :
<https://www.fsf.org/>
- Open Source Initiative (OSI) :
<https://opensource.org/licenses>
- GNU Licenses : <https://www.gnu.org/licenses/>
- Guide pratique : <https://choosealicense.com/>

Conclusion

Les licences sont le **cœur juridique du logiciel libre.**

Elles garantissent la liberté, définissent les règles du jeu et favorisent la collaboration.

Bien comprendre les licences est essentiel pour tout futur contributeur à l'open source.

Systèmes d'exploitation libres

Systèmes d'exploitation libres

- **Objectifs** : découvrir Linux et ses distributions.
- **Contenu** :
 - Historique de Linux
 - Principales distributions : Ubuntu, Debian, Fedora, Arch
 - Gestionnaire de paquets et mise à jour
- **Travail pratique** : installer Ubuntu/Mint en machine virtuelle et explorer la ligne de commande de base.

Objectifs du cours

- Comprendre ce qu'est un système d'exploitation libre
- Découvrir Linux et ses principales distributions
- Apprendre à installer et manipuler un OS libre
- Explorer les commandes de base en terminal

Qu'est-ce qu'un système d'exploitation libre ?

- Logiciel qui gère le matériel et les applications
- Exemples : Linux, BSD
- Libre : code source ouvert, modifiable, redistribuable
- Avantages : sécurité, transparence, communauté active

Historique de Linux

- **1983** : Projet GNU (Richard Stallman)
- **1991** : Linus Torvalds crée le noyau Linux
- Linux + GNU = système complet
- **Aujourd'hui** : utilisé dans serveurs,
supercalculateurs, smartphones (Android)

Distributions Linux

- **Ubuntu** : accessible aux débutants
- **Debian** : stabilité et fiabilité
- **Fedora** : nouveautés et technologies récentes
- **Arch Linux** : flexibilité, personnalisation avancée
- **Linux Mint** : simplicité et interface proche de Windows

Installation de Linux

Options :

- Machine virtuelle (VirtualBox, VMware)
- Dual boot avec Windows
- Live USB (sans installation)

Étapes clés :

- Télécharger une image ISO
- Créer une clé USB bootable
- Lancer l'installation

Gestion des logiciels

- Gestionnaires de paquets : facilitent installation et mise à jour
 - apt** (Debian, Ubuntu)
 - dnf** (Fedora)
 - pacman** (Arch Linux)
- Exemple :
sudo apt install libreoffice

Commandes de base

- ls : lister fichiers
- cd : changer de dossier
- mkdir : créer un dossier
- cp : copier fichier
- rm : supprimer fichier
- man : afficher l'aide d'une commande

Exercice pratique

- Installer Ubuntu ou Linux Mint en machine virtuelle
 - Créer un dossier « Test » dans votre répertoire personnel
 - Créer un fichier texte (nano ou gedit)
 - Copier et supprimer un fichier via le terminal
- Lister les logiciels installés avec apt

Références

- Ubuntu : <https://ubuntu.com/>
- Debian : <https://www.debian.org/>
- Fedora : <https://getfedora.org/>
- Arch Linux : <https://archlinux.org/>
- Linux Mint : <https://linuxmint.com/>

Conclusion

Linux et les systèmes libres sont au cœur de l'informatique moderne.

Ils offrent **liberté, sécurité et flexibilité**.

Apprendre Linux = ouvrir la porte du monde open source.

Outils libres du quotidien

Outils libres du quotidien

Objectifs : montrer les alternatives libres aux logiciels propriétaires.

Contenu :

- LibreOffice (vs MS Office)
- GIMP (vs Photoshop)
- Inkscape (vs Illustrator)
- VLC (vs Windows Media Player)

Travail pratique : réaliser un mini-document avec LibreOffice et une retouche d'image avec GIMP.

Outils libres du quotidien

Ce **Cours** correspond plutôt aux **outils logiciels libres alternatifs** (bureautique, multimédia, graphisme...), pas aux systèmes d'exploitation.

Objectifs du cours

Découvrir les principales alternatives libres aux logiciels propriétaires

Savoir utiliser quelques logiciels libres pour les tâches courantes

Comparer leurs avantages et limites

Réaliser de petites activités pratiques avec ces outils

Pourquoi utiliser des outils libres au quotidien ?

- Liberté d'usage et gratuité
- Compatibilité multi-plateformes
- Communauté active et mise à jour régulière
- Sécurité (code source audité)
- Soutenir l'innovation collaborative

Bureautique : LibreOffice

- Alternative à **Microsoft Office**
- Applications : Writer (Word), Calc (Excel), Impress (PowerPoint)
- Compatible avec les formats Office (.docx, .xlsx, .pptx)
- Avantages : complet, gratuit, multiplateforme
- Limites : parfois des problèmes de compatibilité avancée avec MS Office

Navigation Web : Firefox

- Alternative à **Google Chrome / Edge**
- Respect de la vie privée (modules de sécurité, bloqueur de traqueurs)
- Personnalisable via extensions
- Utilisé dans de nombreuses distributions Linux
- Limite : moins performant que Chrome sur certains services Google

Graphisme et Multimédia

- **GIMP** → alternative à Photoshop (retouche photo)
- **Inkscape** → alternative à Illustrator (dessin vectoriel)
- **Blender** → alternative à 3ds Max/Maya (3D, animation)
- **Krita** → illustration et peinture numérique
- **VLC** → lecteur multimédia universel

Développement

- **Eclipse** : IDE Java
- **Visual Studio Code (Open Source version)**
- **Geany** : éditeur léger
- **Python + bibliothèques open source**
- Gestion de versions avec **Git**

Autres exemples utiles

- **Thunderbird** : client mail (alternative à Outlook)
- **Nextcloud** : cloud libre (alternative à Google Drive)
- **Jitsi Meet** : visioconférence (alternative à Zoom)
- **OBS Studio** : streaming et capture vidéo

Exercice pratique

- Installer **LibreOffice** et créer un mini-document texte + une présentation simple.
- Télécharger **GIMP** et retoucher une image (redimensionner, ajouter un texte).
- Utiliser **VLC** pour lire une vidéo et tester les options (sous-titres, capture d'écran).
- Comparer l'expérience avec un logiciel propriétaire équivalent.

Références

- LibreOffice : <https://www.libreoffice.org/>
- Mozilla Firefox :
<https://www.mozilla.org/firefox/>
- GIMP : <https://www.gimp.org/>
- Inkscape : <https://inkscape.org/>
- Blender : <https://www.blender.org/>
- VLC : <https://www.videolan.org/vlc/>
- Thunderbird : <https://www.thunderbird.net/>

Conclusion

Les outils libres permettent d'effectuer toutes les tâches quotidiennes (bureautique, multimédia, graphisme, développement) **sans dépendre de solutions propriétaires.**

En les adoptant, on gagne en **liberté, sécurité et autonomie.**

Développement et collaboration

Développement et collaboration

Objectifs : apprendre à contribuer à un projet open source.

Contenu :

- Introduction à Git et GitHub
- Workflow collaboratif : fork, commit, pull request
- Gestion de bugs et documentation

Travail pratique : créer un dépôt GitHub personnel, faire des commits et rédiger un README.

Ce Cours correspond à :

**Développement et collaboration (Git & GitHub,
contribution aux projets open source)**
(et non aux systèmes d'exploitation, qui étaient
déjà traités en **Cours Précédent**).

Objectifs du cours

Comprendre l'importance de la collaboration dans le logiciel libre

Découvrir l'outil Git et la plateforme GitHub

Apprendre les bases du versioning (commits, branches, merges)

Savoir contribuer à un projet open source

La collaboration dans l'Open Source

Les projets libres vivent grâce à la **contribution collective**

Contributions possibles :

- Code
- Documentation
- Traduction
- Signalement de bugs / suggestions

Exemple : Linux a des milliers de contributeurs dans le monde

Qu'est-ce que Git ?

Système de gestion de versions distribué

Créé par Linus Torvalds en 2005

Permet de :

- Suivre l'historique d'un projet
- Travailler à plusieurs sans écraser le code

Revenir en arrière en cas d'erreur

GitHub et autres plateformes

GitHub : la plus grande plateforme de collaboration open source

Alternatives : GitLab, Bitbucket

Fonctionnalités :

- Hébergement de dépôts Git
- Suivi des issues (bugs, tâches)
- Gestion des pull requests

Intégration continue (CI/CD)

Concepts de base de Git

Repository (repo) : dossier d'un projet versionné

Commit : enregistrement d'une modification

Branch : une ligne de développement parallèle

Merge : fusion de branches

Pull Request (PR) : proposition de modification sur un projet collaboratif

Flux de travail collaboratif (Workflow GitHub)

Fork du projet (copie personnelle)

Clone en local

Création d'une **branche**

Écriture du code et **commit**

Push vers GitHub

Création d'une **Pull Request** pour proposer les changements

Exemple pratique

Créer un compte GitHub

Créer un nouveau dépôt avec un fichier README.md

Ajouter un fichier test.txt, le modifier et faire plusieurs commits

Créer une branche dev, y faire un commit puis la fusionner dans main

Exercice pratique à la maison

Créez un compte GitHub (si ce n'est pas déjà fait)

Créez un dépôt nommé mon-premier-projet-libre

Ajoutez un fichier texte avec une description du projet

Faites au moins **3 commits** (ajout, modification, suppression)

Partagez le lien du dépôt avec vos camarades

Références

- Git officiel : <https://git-scm.com/>
- Documentation GitHub :
<https://docs.github.com/>
- GitLab : <https://about.gitlab.com/>
- Tutoriel interactif :
<https://learngitbranching.js.org/>

Conclusion

- Git et GitHub sont les **outils centraux de la collaboration open source.**
- Ils permettent à des milliers de développeurs de travailler ensemble efficacement.
Savoir utiliser Git = clé pour contribuer activement au monde du logiciel libre.

Communauté et contribution

Communauté et contribution

Objectifs : comprendre l'importance de la communauté.

Contenu :

- Forums, mailing-lists, chat (IRC, Discord, Slack)
- Comment contribuer : code, documentation, traduction, test
- Études de cas : Mozilla, Linux Foundation, Apache Foundation

Travail pratique : participer à un forum ou canal de discussion d'un projet libre.

Objectifs du chapitre

- Expliquer le rôle central de la communauté dans l'open source
- Montrer les différentes formes de contribution possibles
- Présenter les canaux de communication utilisés par les communautés libres
- Illustrer avec des exemples concrets de grandes fondations et projets open source

La communauté : pilier de l'open source

Dans l'open source, **le code n'est rien sans la communauté**

La communauté permet :

- Le **développement continu**
- La **maintenance et correction de bugs**
- Le **support et la documentation**
- La **diffusion et la formation**

Exemple : Linux → des milliers de contributeurs bénévoles et professionnels

Les différentes formes de contribution

Contribution technique

- Écriture ou amélioration du code
- Relecture de code (code review)
- Tests et assurance qualité (QA)

Contribution non technique

- Documentation, tutoriels, guides
- Traduction et localisation
- Design graphique et ergonomie
- Communication (forums, blogs, réseaux sociaux)

Tout le monde peut contribuer, pas seulement les développeurs !

Outils et canaux de communication

Forums : ex. Ubuntu Forums, Ask Fedora

Mailing lists : historiques (Debian, GNU)

Chats en ligne : IRC, Matrix, Discord, Slack

Plateformes collaboratives : GitHub, GitLab, SourceForge

Événements communautaires : FOSDEM, LinuxCon, MozFest

Exemples de grandes communautés open source

Mozilla Foundation : Firefox, Thunderbird

Linux Foundation : kernel Linux, Kubernetes,
Hyperledger

Apache Software Foundation : Apache HTTP
Server, Hadoop, Spark

Debian Project : modèle de gouvernance
communautaire exemplaire

Étude de cas

Projet Libre Office

- Communauté mondiale active
- Contributions variées : développement, documentation, traduction
- Communication via mailing-lists, forums, événements (LibreOffice Conference)
- Résultat : logiciel libre utilisé par des millions de personnes

Exercice pratique

- Choisir un projet open source (GIMP, Firefox, LibreOffice, etc.)
- Identifier ses canaux de communication (site officiel, forum, GitHub)
- Lister au moins **3 façons de contribuer** à ce projet
- Partager vos résultats en classe

Travail à la maison

Rejoindre un forum ou une communauté (par ex. Ubuntu, LibreOffice, Debian)

Créer un compte et participer (poser une question, répondre, proposer une idée)

Rédiger un petit rapport (1 page) :

- Quelle communauté avez-vous rejoint ?
- Comment accueille-t-elle les nouveaux ?

Quelle expérience avez-vous vécue ?

Références

- [Mozilla Community](#)
- Linux Foundation
- [Apache Software Foundation](#)
- Debian Community
- Guide pour débuter dans l'open source

Conclusion

La communauté est le moteur du logiciel libre.

Contribuer ne signifie pas seulement coder, mais aussi documenter, traduire, tester, enseigner.

Participer, c'est apprendre, partager et faire grandir l'open source.

Études de cas et projets pratiques

Études de cas et projets pratiques

Objectifs : appliquer les acquis.

Contenu :

- Utilisation du libre dans l'éducation, la recherche, l'administration
- Débat : libre vs propriétaire

Mini-projet étudiant : créer une présentation, un petit programme ou un rapport basé sur des outils libres.

Objectifs du cours

- Découvrir l'utilisation des logiciels libres dans différents secteurs (éducation, administration, recherche, entreprises).
- Analyser des cas réels de réussite du libre.
- Lancer des mini-projets étudiants pour mettre en pratique les connaissances acquises.
- Favoriser la réflexion critique : « Libre vs Propriétaire ».

Logiciels libres dans l'éducation

Utilisation de **Linux** et **LibreOffice** dans les écoles pour réduire les coûts.

Moodle : plateforme libre de e-learning, utilisée dans de nombreuses universités.

Avantages : économie, autonomie pédagogique, adaptation locale.

Exemple : adoption de logiciels libres dans certaines universités africaines et européennes.

Logiciels libres dans l'administration et les entreprises

Adoption gouvernementale :

- Munich (Allemagne) → migration massive vers Linux (LiMux Project).
- France → Gendarmerie nationale utilise Linux et LibreOffice.

Entreprises :

- **Red Hat** : succès économique basé sur les services autour de Linux.

Google : Android (basé sur Linux) + contributions open source.

Logiciels libres dans la recherche et la science

Outils libres pour la science et l'ingénierie :

- **R** : statistiques et data science.
- **Python** : machine learning, IA, analyse de données.
- **Scilab / Octave** : alternatives libres à MATLAB.

Avantage : transparence scientifique et reproductibilité des résultats.

Logiciels libres dans les arts et la créativité

GIMP et **Krita** pour le dessin et la retouche photo

Blender pour la modélisation 3D et l'animation (ex. : courts-métrages « Big Buck Bunny », « Sintel »)

LMMS et **Audacity** pour la création musicale et l'édition sonore

Mini-projets étudiants proposés

Projet bureautique : créer un rapport et une présentation avec **LibreOffice**.

Projet multimédia : retoucher une image avec **GIMP** et créer une affiche.

Projet développement : écrire un script simple en **Python** et le publier sur GitHub.

Projet collaboration : créer un dépôt GitHub de groupe et faire un mini-projet collaboratif.

Débat en classe : Libre vs Propriétaire

Questions à discuter :

- Le libre est-il toujours meilleur que le propriétaire ?
- Quels sont les **avantages** et **limites** du libre pour les entreprises ?

Un monde 100% libre est-il réaliste ?

Références

- Case Studies – Open Source Initiative
- Linux Foundation – Success Stories
- Free Software Foundation – Education
- Blender Open Movies

Conclusion

Les logiciels libres sont présents partout : **écoles, entreprises, administrations, recherche, création artistique.**

Étudier des cas concrets aide à comprendre leur **impact réel.**

Les projets pratiques permettent aux étudiants de devenir **acteurs et contributeurs** du monde open source.

Séries de TD & TP – Logiciels Libres (Open Source)

TD1 : Comprendre le Logiciel Libre

Objectifs : Assimiler les concepts et différences entre libre et propriétaire.

Q1. Définir les 4 libertés du logiciel libre.

Q2. Quelle différence entre « libre » et « gratuit » ? Donnez 2 exemples.

Q3. Comparez : Firefox (libre) vs Google Chrome (propriétaire).

Q4. Recherchez et classez 5 logiciels que vous utilisez en 2 colonnes : *Libres vs Propriétaires*.

TP1 : Découverte de logiciels libres

Objectifs : Installer et tester des logiciels libres.

Installer **LibreOffice** et rédiger un mini-document (rapport d'une page).

Créer une présentation simple avec **Impress**.

Installer **GIMP** et retoucher une image (ajout de texte, recadrage).

Rédiger un court compte rendu des avantages et limites.

TD2 : Licences et aspects juridiques

Objectifs : Identifier et comparer les licences libres.

Q1. Différence entre GPL, MIT et Apache 2.0.

Q2. Sur GitHub, trouvez deux projets avec des licences différentes et comparez-les.

Q3. Pourquoi une entreprise pourrait-elle préférer la licence MIT à la GPL ?

Q4. Donnez un exemple de risque en cas de non-respect d'une licence libre.

TP2 : Installation de Linux

Objectifs : Découvrir un OS libre.

Installer **Ubuntu** (ou Linux Mint) en machine virtuelle.

Créer un dossier “TP_Linux” et un fichier texte à l’intérieur.

Apprendre et tester 5 commandes de base (ls, cd, mkdir, cp, rm).

Installer un logiciel libre via le gestionnaire de paquets (ex : sudo apt install vlc)

TD3 : Outils libres du quotidien

Objectifs : Explorer les alternatives libres.

Q1. Citez un équivalent libre pour : Word, Photoshop, Zoom, Illustrator, Google Drive.

Q2. Pourquoi VLC est-il devenu un standard mondial malgré la concurrence ?

Q3. Débat en classe : *Peut-on vivre uniquement avec des logiciels libres ?*

TP3 : Développement et Git/GitHub

Objectifs : Apprendre à versionner et collaborer.

- Installer **Git**.
- Créer un dépôt local avec un fichier README.md.
- Faire plusieurs commits après modification du fichier.
- Créer un compte **GitHub** et pousser le dépôt en ligne.

Créer une nouvelle branche, ajouter un fichier, et la fusionner dans main.

TD4 : La communauté open source

Objectifs : Comprendre l'importance des communautés.

Q1. Quelles sont les principales façons de contribuer sans coder ?

Q2. Donnez 2 exemples de grandes fondations open source.

Q3. Quelle est la différence entre un forum, une mailing-list et un canal IRC ?

Q4. Pourquoi dit-on que la communauté est le « moteur » du logiciel libre ?

TP4 : Contribution et projet collaboratif

Objectifs : Simuler une contribution réelle.

Sur GitHub, forkez un projet open source simple (ex. un dépôt de cours, une page web).

Modifiez un fichier (exemple : ajouter votre nom dans une liste).

Faites un **pull request** pour proposer vos changements.

En groupe : créer un mini-projet collaboratif (par ex. un site web statique en HTML) et gérer le travail avec GitHub.

TD5 : Études de cas

Objectifs : Analyser des usages réels du libre.

Q1. Donnez un exemple de logiciel libre utilisé dans l'éducation.

Q2. Pourquoi la Gendarmerie française a migré vers Linux ?

Q3. Quels sont les avantages de Blender dans l'industrie créative ?

Q4. Débat : *Faut-il imposer le libre dans l'administration publique ?*

TP5 : Projet final

Objectifs : Synthèse pratique.

Chaque étudiant ou groupe doit réaliser un **mini-projet basé sur un logiciel libre**, par exemple :

- Un rapport et une présentation avec LibreOffice.
- Une affiche réalisée avec GIMP ou Inkscape.
- Un script Python publié sur GitHub.
- Une courte vidéo montée avec un logiciel libre (ex. Kdenlive).

Présentation orale en fin de semestre.