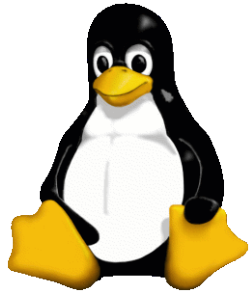
Comprendre le logiciel libre

Utiliser, comprendre, modifier et partager

Première NSI

Des libertés, des exemples et des choix à discuter



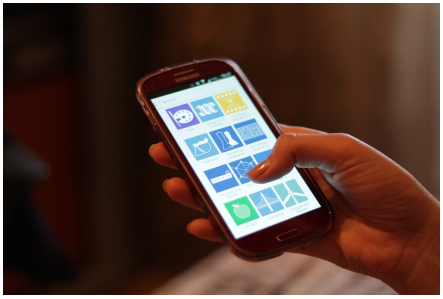
Tux : Larry Ewing, créé avec The GIMP

Une question pour commencer

Une application est gratuite.

Est-elle forcément **libre** ?

- Peut-on voir comment elle fonctionne ?
- Peut-on l'adapter à ses besoins ?
- Peut-on partager sa version modifiée ?

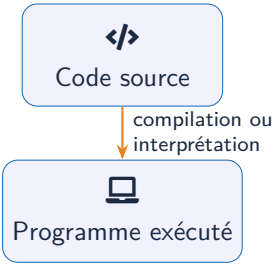


oceanosydados, CC0 1.0

Projet GNU : définition du logiciel libre. Références complètes sur la dernière diapositive.

Avant tout : le code source

- Le **code source** est le texte du programme, lisible et modifiable par une personne qui programme.
- Une version **exécutable** permet à la machine de faire fonctionner le programme.
- Voir le code ne suffit pas : il faut aussi avoir le **droit** de le modifier et de le partager.



Projet GNU ; Open Source Initiative. Références complètes sur la dernière diapositive.

Les quatre libertés fondamentales



0. Exécuter
pour tout usage



1. Étudier et modifier
accès au code source



2. Redistribuer
des copies

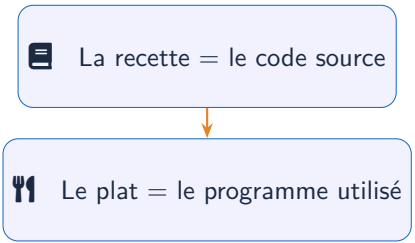


3. Partager
ses versions modifiées

Ces libertés s'exercent dans le respect de la licence. Les libertés 1 et 3 nécessitent l'accès au code source.

Projet GNU : les quatre libertés. Références complètes sur la dernière diapositive.

Une histoire de recette de cuisine



- Cuisiner le plat pour qui l'on veut.
- Lire la recette et changer un ingrédient.
- Donner la recette ou sa variante.

Attention : l'analogie éclaire les libertés ; les droits réels dépendent de la licence du logiciel.

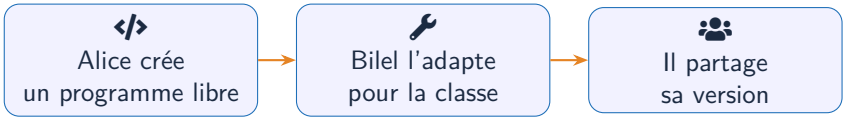
Libre, gratuit, ouvert : ne pas confondre

Mot	Question à se poser
Gratuit	Combien coûte l'accès ? La gratuité ne donne pas les quatre libertés.
Libre	Quels droits ai-je sur le programme et son code ?
Open source	La licence respecte-t-elle la définition de l'OSI ? Les préoccupations mises en avant diffèrent.
Code visible	Peut-on seulement le lire, ou aussi le modifier et le redistribuer ?

Un logiciel libre peut être vendu ; un logiciel gratuit peut être propriétaire.

Projet GNU ; Open Source Initiative. Références complètes sur la dernière diapositive.

Exemple : partager un outil de classe



- Le code est accessible et la licence autorise modification et redistribution.
- Bilel respecte aussi les conditions applicables : mention des auteurs, licence, accès au code selon les cas.

À quoi sert une licence ?

- Les auteurs conservent généralement leurs **droits d'auteur**.
- La licence précise les **permissions** et les **conditions** d'utilisation et de redistribution.
- Publier du code sans licence ne signifie pas automatiquement qu'il est libre.

Le libre n'est pas l'absence de règles : il garantit des libertés à travers une licence.



LICENCE

droits + conditions

Open Source Initiative ; projet GNU. Références complètes sur la dernière diapositive.

Deux familles de licences libres

Copyleft : exemple GPL

Si l'on distribue une version modifiée couverte par la GPL, on doit notamment la distribuer sous GPL et respecter les conditions d'accès au code source.

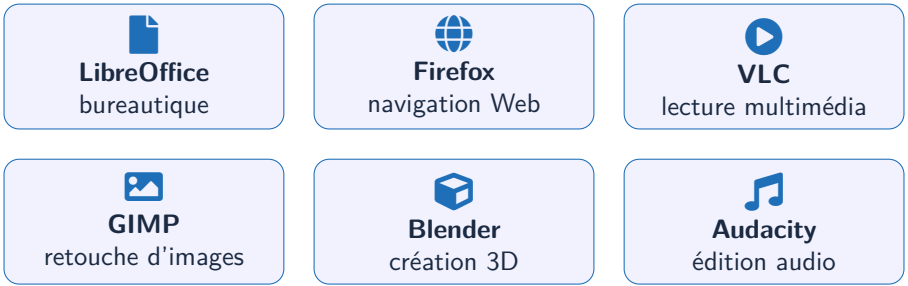
Permissive : exemple MIT

Elle permet la réutilisation, y compris dans un logiciel propriétaire, sous réserve de conserver notamment les mentions de copyright et de licence.

Dans les deux cas : utilisation, étude, modification et partage sont permis.

GNU GPL version 3 ; licence MIT, Open Source Initiative. Références complètes sur la dernière diapositive.

Des logiciels libres dans la vie quotidienne



Documentation officielle des projets ; voir les liens en dernière diapositive. Références complètes sur la dernière diapositive.

Du code à l'infrastructure

- **Python** : un langage et une implémentation de référence utilisés en NSI.
- **Git** : un outil de suivi des versions et de collaboration.
- **Linux** : un noyau libre, présent dans de nombreux systèmes.
- **Debian** : une distribution réunissant noyau et nombreux logiciels.

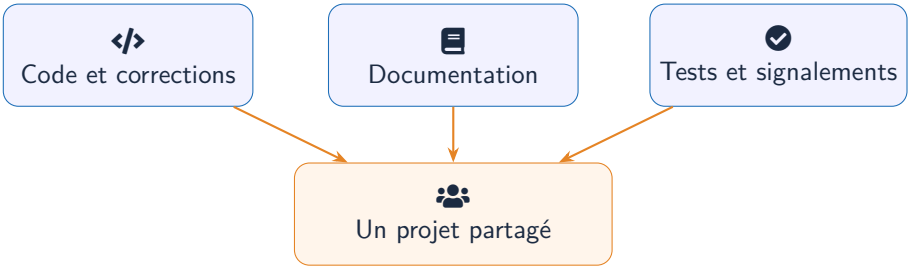
Libre ne signifie pas forcément visible : ces outils peuvent fonctionner en arrière-plan.



Raspberry Pi 4 : Suyash.dwivedi, CC BY-SA 4.0

Python Software Foundation ; Git ; Debian ; documentation du noyau Linux. Références complètes sur la dernière diapositive.

Le libre peut se construire à plusieurs



Particuliers, associations, entreprises et institutions peuvent contribuer ; une communauté n'est pas forcément composée uniquement de bénévoles.

Quels avantages possibles ?

Comprendre
Le code peut être étudié et audité.

Adapter
On peut modifier le logiciel selon ses besoins ou faire appel à quelqu'un pour le faire.

Partager
Les copies et améliorations peuvent circuler.

Choisir
La possibilité de changer de prestataire peut réduire certaines dépendances.

Projet GNU ; Open Source Initiative. Références complètes sur la dernière diapositive.

Des limites et des coûts à examiner

- **Compétences** : modifier ou auditer le code demande du temps et des connaissances.
- **Compatibilité** : formats, matériel ou logiciels spécialisés ne sont pas toujours bien pris en charge.
- **Maintenance** : mises à jour, formation, hébergement et support peuvent coûter de l'argent.
- **Pérennité** : certains projets manquent de contributeurs ou de financement.



Évaluer
besoins et moyens

Libre ne veut pas dire automatiquement sûr

Une possibilité

Le code accessible permet à des tiers d'examiner le programme et de proposer des corrections.

Pas une garantie

Personne n'a nécessairement vérifié tout le code. Des failles peuvent rester présentes longtemps.



Vérifier la provenance, maintenir à jour, évaluer le projet et les besoins.

Comment un projet libre peut-il vivre ?

- Dons et associations
- Support et formation
- Hébergement et services
- Financement public
- Contributions d'entreprises

> Vendre une copie ou un service est compatible avec le libre : ce sont les libertés accordées aux utilisateurs qui comptent.

Projet GNU : logiciel libre et activité commerciale. Références complètes sur la dernière diapositive.

À vous de jouer : libre ou non libre ?

1. Une application est téléchargeable sans payer ; son code est secret et sa copie interdite.
2. Un programme montre son code, mais interdit de distribuer les versions modifiées.
3. Un outil peut être utilisé, étudié, modifié et redistribué, y compris modifié, sous une licence libre.
4. Une entreprise vend de l'assistance pour un logiciel sous licence GPL.

Pour chaque cas : quelles libertés sont accordées ou refusées ? Le prix suffit-il à conclure ?

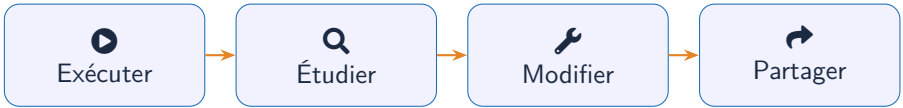
Correction : raisonner sur les libertés

1. **Non libre** : gratuit ne donne pas le droit de copier ni d'étudier le code.
2. **Non libre** : la liberté de redistribuer des versions modifiées manque.
3. **Libre** : les quatre libertés sont accordées sous une licence libre.
4. **Compatible avec le libre** : vendre de l'assistance ne retire pas les libertés de la GPL.

Réflexe : chercher la licence et examiner les droits accordés, pas seulement le prix ou l'accès au code.

Projet GNU : définition du logiciel libre. Références complètes sur la dernière diapositive.

Bilan : quatre mots à retenir



- Le logiciel libre concerne les **libertés**, pas nécessairement la gratuité.
- Sa **licence** fixe les droits et les conditions.
- Ses avantages sont réels, mais le choix dépend aussi des besoins, des moyens et du contexte.

Sources et crédits des illustrations

Définitions et licences — pages consultées le 22 novembre 2026.

- Projet GNU : les quatre libertés ; catégories de logiciels ; GNU GPL v3.
- Open Source Initiative : définition open source ; licence MIT.

Exemples — sites et documentations des projets :

- LibreOffice, Firefox / Mozilla, VLC, GIMP, Blender, Audacity.
- Python, Git, Linux, Debian.

Illustrations — images réduites ; schémas originaux réalisés avec TikZ.

- Tux : Larry Ewing (`lewing@isc.tamu.edu`), créé avec The GIMP ; réutilisation autorisée avec attribution.
- Smartphone Use : oceanosydados, CC0 1.0.
- Raspberry Pi 4 : Suyash.dwivedi, CC BY-SA 4.0.
- Icônes : Font Awesome Free, CC BY 4.0.