

3TC37: Contribution à un logiciel libre

Introduction, théorie et principes du logiciel libre

Marc Jeanmougin
Théo Zimmermann

Télécom Paris, Institut Polytechnique de Paris

CC BY-SA 4.0



Introduction

Objectifs du cours

- Développement de vos compétences en programmation (3TC3[5-7])

Notamment en apprenant à :

- naviguer dans du code existant,
 - planifier une modification,
 - prendre en compte les retours automatiques et les revues de code,
 - justifier de choix techniques.
- Connaissance des principes et des enjeux du logiciel libre :
 - définition et notions historiques,
 - compréhension et respect des licences,
 - capacité à évaluer un projet avant de l'utiliser ou d'y contribuer.

Modalités pratiques

Cours très orienté vers la pratique. Objectif : contribuer à un logiciel libre.

Séances :

- Théorie & principes, histoire, licences
- Modèles de gouvernance, communautés, modèles économiques
- TP préparatoires au projet (recherche de projets, analyse de projets)
- TP d'aide au projet (mars)
- Contrôle de connaissance (20/03)
- Office hours d'aide au projet (avril)
- Soutenances de projet (22/04)

Évaluation :

- Point d'étapes notés = merge requests sur le dépôt du cours (30%)
- Contrôle de connaissance (30%)
- Présentation finale du projet (40%)

Qui sommes-nous ?

- Marc Jeanmougin, co-fondateur du MOOC Open Source Masterclass, co-mainteneur du logiciel libre Inkscape.
- Théo Zimmermann, enseignant-chercheur sur le(s) modèle(s) de développement collaboratif du logiciel libre, co-mainteneur du logiciel libre Rocq Prover.

C'est quoi un logiciel libre ? Un logiciel propriétaire ?

C'est quoi un logiciel libre ? Un logiciel propriétaire ?

Un *logiciel libre*, c'est un logiciel distribué sous une *licence* qui autorise ceux qui le reçoivent à l'**utiliser**, en **étudier le code source**, le **modifier** et le **redistribuer** (inchangé ou modifié).

Le contraire d'un logiciel libre est un *logiciel propriétaire*.

Il s'agit donc de garantir des *libertés* à l'utilisateur (là où les licences de logiciels propriétaires restreignent l'utilisation, l'accès au code source et la redistribution).

Mais au-delà de la question des libertés, cela permet aussi un autre modèle de développement, collaboratif et ouvert, qu'on désigne plus souvent sous le terme d'*open source*.

Vous utilisez déjà des logiciels libres

Avec vos voisins, établissez une liste de :

- logiciels libres que vous utilisez,
- logiciels propriétaires que vous utilisez.

Vous utilisez déjà des logiciels libres

Directement :

- Votre système d'exploitation si vous êtes sous GNU/Linux,
- Votre navigateur web si c'est Firefox ou Chromium,
- Une partie de votre téléphone si vous êtes sous Android,
- Votre éditeur de texte si c'est Emacs, Vim, VS Code ou Atom,
- Le compilateur de votre langage préféré,
- Le gestionnaire de version git,
- etc.

Indirectement (dépendances à des bibliothèques libres) :

- La plupart des sites web sur lesquels vous naviguez,
- La plupart des applications mobiles que vous utilisez.

Pourquoi ce cours est utile pour vous

- En tant que futur·es ingénieur·es.
- En tant que futur·es chercheur·ses.
- En tant que citoyen·nes.
- À titre personnel.

En tant que futur·es ingénieur·es

En tant que futur·es ingénieur·es

- Pour vous faire recruter : votre profil GitHub est votre deuxième CV. (Attention aux dérives !)
- **L'utilisation de logiciel libre** est essentielle dans presque toutes les entreprises (dans et hors secteur informatique) :
 - Réduction des coûts,
 - Adaptabilité,
 - Durabilité / Indépendance / Souveraineté.
- Le logiciel libre peut aussi être moteur de **collaborations** au-delà des frontières de l'entreprise.

En tant que futur·es chercheur·ses

En tant que futur·es chercheur·ses

Trois piliers de la **science ouverte** : open access, open data, open source

- Partage de code source en open source ⇒
 - reproduire les résultats de la recherche,
 - construire à partir des résultats existants,
 - comparer différentes techniques.
- Opportunité pour créer des communautés internationales de chercheur·ses autour de projets (exemple : Rocq, Scikit-learn, YAGO, etc.).

En tant que citoyen·nes

En tant que citoyen·nes

- Loi pour une République numérique (2016) : les administrations doivent publier leurs codes et leurs données.
 - <https://code.gouv.fr/> (2021)
- On peut inspecter le code source du calcul des impôts, de ParcourSup, TousAntiCovid, etc. (mais ces logiciels ne sont pas nécessairement développés de manière ouverte).
- Des collectifs développent du logiciel libre au service de la démocratie participative : voir Code for France, Decidim, etc.

À titre personnel

À titre personnel

Lorsque vous **utilisez** des logiciels libres, vous pouvez :

- en rapporter des bugs,
- regarder et modifier le code source pour corriger des bugs, l'adapter à vos besoins,
- faire partie d'une communauté.

Si vous **développez** un logiciel ou une bibliothèque et que vous souhaitez le partager et recevoir des contributions.

Définition et histoire

Free software, logiciel libre et open source

Ces trois termes veulent fondamentalement dire la même chose, mais en mettant l'accent sur différents aspects.

Free software, logiciel libre et open source

- Le terme “free software” a été popularisé par Richard Stallman et la Free Software Foundation (1985).
- Il met l’accent sur les **libertés** (“Free Software Definition”) :
 0. Liberté d’**utiliser** le programme, pour n’importe quel usage.
 1. Liberté d’**étudier** comment le programme fonctionne et de le **modifier** (requiert l’accès aux sources).
 2. Liberté de **redistribuer des copies** du programme.
 3. Liberté de **redistribuer des versions modifiées** du programme.
- Son problème est que “free” peut vouloir dire “libre” ou “gratuit”. L’expression “*Free as in free speech, not as in free beer*” rappelle que ce n’est pas une question de prix.

Free software, logiciel libre et open source

- Le terme “logiciel libre” (“software libre” en espagnol) est la traduction de “free software”.
- Il n’a pas ce problème de double sens.
- Certaines personnes choisissent de l’utiliser en anglais pour éviter la confusion (→ LibreOffice).

Free software, logiciel libre et open source

- Le terme “open source” a été popularisé par Eric Raymond et l’Open Source Initiative (1998).
- Il met l’accent sur le **partage** du code permettant la **collaboration ouverte** (“Open Source Definition”).
- Son problème est que beaucoup de gens confondent “publication du code source” et “open source”, alors que le code source peut être publié sous une licence restrictive. De nos jours, de nombreuses entreprises et organisations publient du code sans qu’il soit libre / open source pour autant.
- À l’heure des grands modèles d’IA, certains sont tentés d’abuser du terme open source pour désigner certains modèles en accès ouvert, mais cela soulève de nombreuses questions.

Exercice : des modèles d'IA open source ?

Chercher des informations sur internet sur les questions que soulève l'usage du terme “open source” pour désigner des modèles d'IA.

Free software, logiciel libre et open source

Pour éviter toute confusion et éviter de prendre parti dans un débat philosophique, certains parlent de FOSS ou FLOSS :
pour **F**ree, (**L**ibre,) and **O**pen **S**ource **S**oftware.

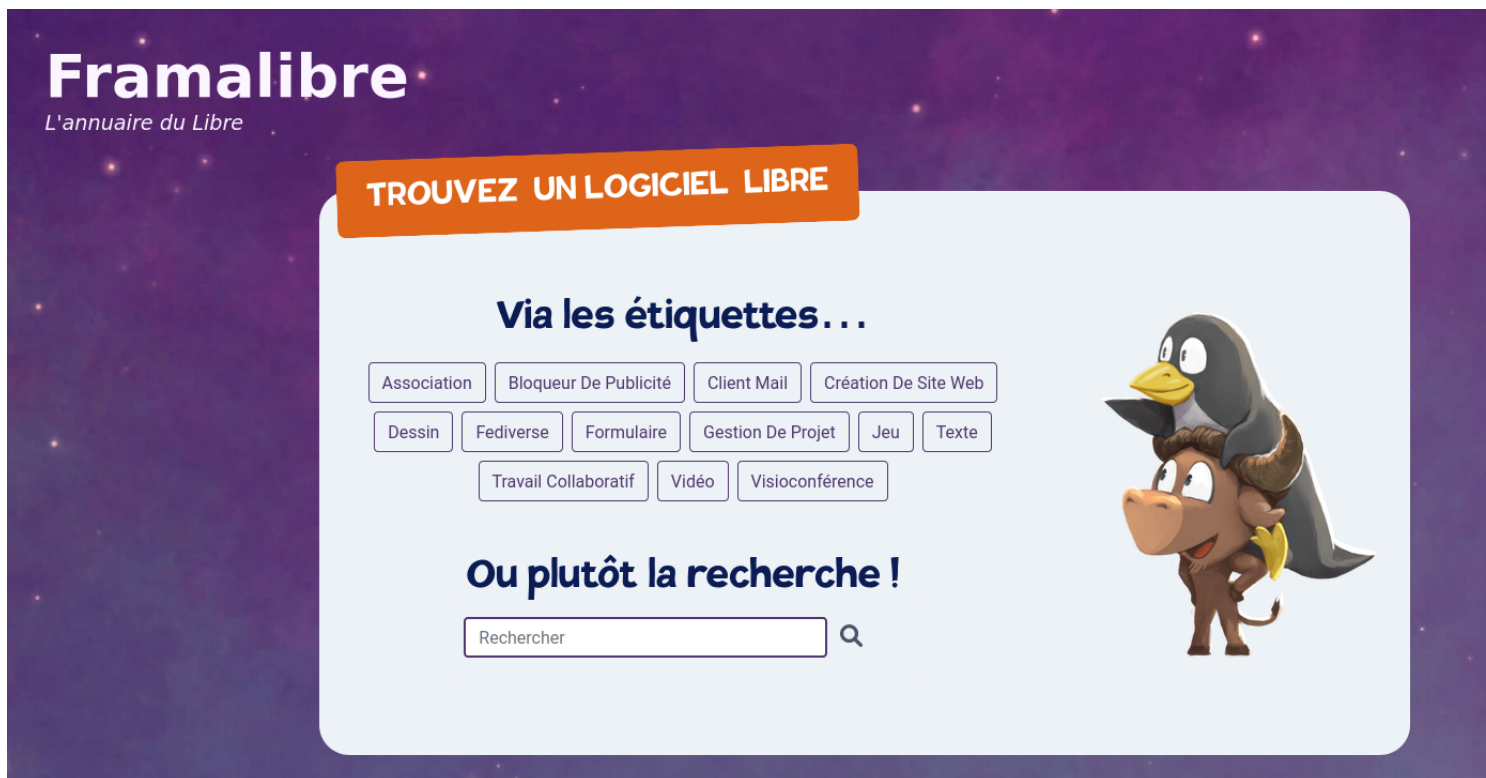
En pratique, la quasi-totalité des logiciels libres sont open source et inversement.

Travail à faire pour la semaine prochaine

- Trouvez :
 - 2 logiciels libres utilisant **un langage de programmation / une technologie que vous maîtrisez**,
 - 2 logiciels libres qui vous **intéressent personnellement** mais dans d'**autres langages de programmation** que les précédents (que vous maîtrisez ou non).
- Pour chacun, ouvrir une **merge request** dans le dépôt 3tc37-2026 en remplissant le template (vérifier qu'il n'y a pas déjà une autre merge request proposant le même projet avant).
- Reviewez 4 pull requests de vos camarades, approuvez si le fichier est complet et respecte les consignes ou commentez si des changements sont nécessaires.

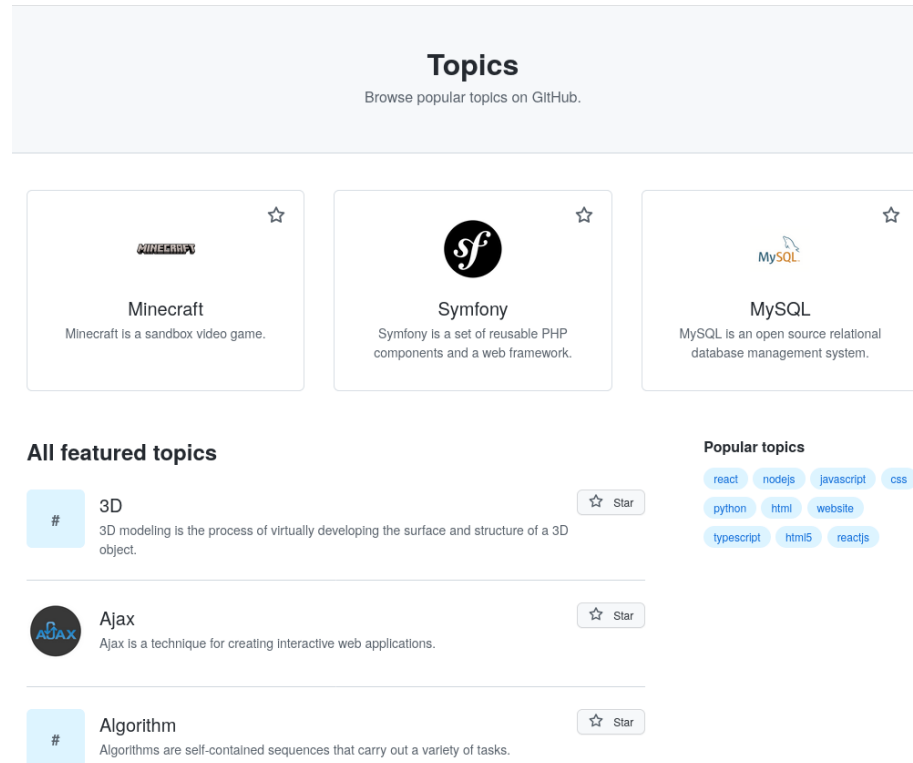
Pour vous aider à trouver des projets

Annuaire du Libre de l'association Framasoft : <https://framalibre.org>



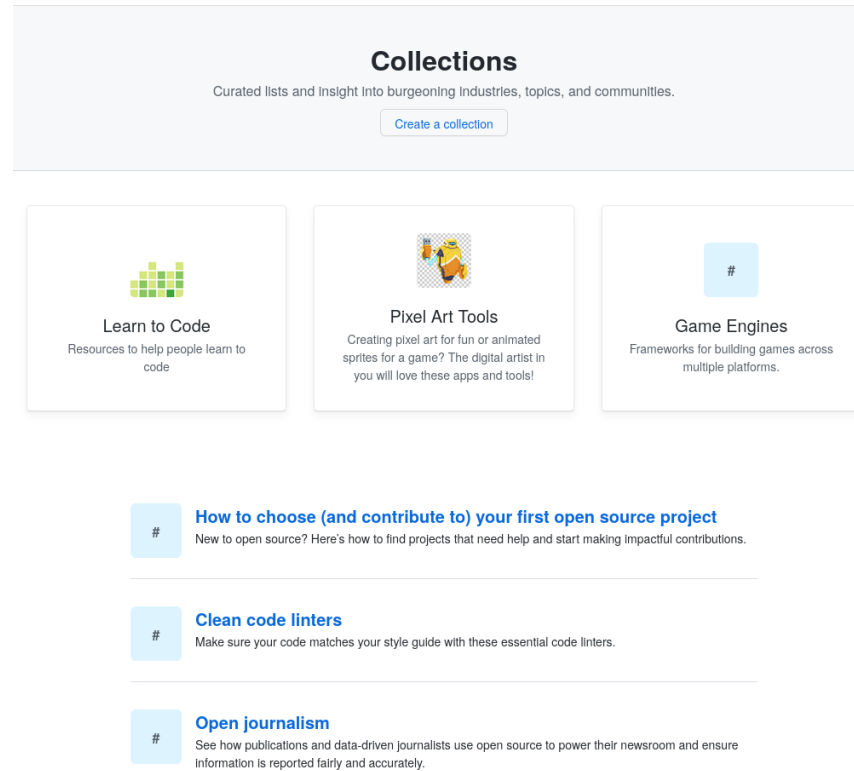
Pour vous aider à trouver des projets

GitHub Topics : <https://github.com/topics>



Pour vous aider à trouver des projets

GitHub Collections : <https://github.com/collections>



Pour vous aider à trouver des projets

Awesome Lists : <https://github.com/sindresorhus/awesome>

An awesome list is a list of awesome things curated by the community. There are awesome lists about everything from [CLI applications](#) to [fantasy books](#). The [main repository](#) serves as a curated list of awesome lists.



Created by [Sindre Sorhus](#) and the community

Released July 11, 2014

[sindresorhus/awesome](#)

[awesome.re](#)

Here are 6,048 public repositories matching this topic...

Language: All ▼

Sort: Most stars ▼



[sindresorhus](#) / [awesome](#)

♥ Sponsor

☆ Star 236k ▼

<> Code ⓘ Issues 🔄 Pull requests

👤 Awesome lists about all kinds of interesting topics

[lists](#) [awesome](#) [unicorns](#) [resources](#) [awesome-list](#)

Updated 1 hour ago

Histoire du logiciel libre

- De quand date le logiciel libre ?
- Mais en fait, de quand date le logiciel ?
- Quel âge ont certains logiciels libres célèbres ?

Avant le logiciel libre

- Dans les débuts de l'informatique, **les ordinateurs coûtent très cher** et sont distribués avec des logiciels spécialement conçus pour eux (dont la valeur est secondaire).
- Des programmes sont modifiés, conçus et partagés par leurs utilisateurs, sans règles particulières.
- On finit par considérer que le logiciel a une valeur en soit, peut être vendu et est protégé par le **copyright** (droit d'auteur).
- **Microsoft prend son envol** à travers un accord avec IBM :
IBM vend des PC (personal computers) avec le système d'exploitation de Microsoft (MS-DOS), sur lequel ce dernier conserve les droits. L'industrie du logiciel est née.

Walter Isaacson

L'auteur de la biographie de référence de Steve Jobs

Les Innovateurs



Comment un groupe de génies,
hackers et geeks a fait la révolution numérique

9.

Le logiciel

Lorsque Paul Allen s'approcha du kiosque à journaux encombré au milieu de Harvard Square et aperçut l'Altair en couverture du *Popular Electronics* daté de janvier 1975, il fut à la fois enthousiasmé et consterné. Bien qu'il soit emballé par la révélation que l'ère de l'ordinateur individuel avait commencé, il craignait d'arriver trop tard pour participer à la fête. Il posa rageusement ses soixante-quinze *cents* sur le comptoir, s'empara du magazine et trotta dans la neige fondue jusqu'à la chambre en cité universitaire de Bill Gates, son pote de lycée de Seattle, mordu d'informatique lui aussi, qui l'avait convaincu de décrocher de la fac et de s'installer à Cambridge. « Hé ! ce truc est en train de nous passer sous le nez ! », déclara Allen. Gates se mit à osciller d'avant en arrière, comme cela lui arrivait souvent dans des moments particulièrement intenses. Quand il eut fini de lire l'article, il se rendit compte qu'Allen avait raison. Pendant les huit semaines suivantes, ils se lancèrent dans une frénésie programmatrice qui allait changer la nature commerciale de l'informatique¹.

Les premiers logiciels libres

Unix (pas un logiciel libre)

- Lancé à partir de 1969 chez AT&T (Bell Labs).
- Distribué à de nombreux industriels et universités qui en créent des **variants**.
- Aujourd'hui les descendants (spirituels) d'Unix sont :
 - La famille BSD (Berkeley Software Distribution) :
 - FreeBSD, OpenBSD, NetBSD,
 - Darwin, le noyau (libre) de macOS.
 - À l'origine de la famille de **licences BSD**.
 - GNU / Linux (GNU = GNU's Not Unix)

Les premiers logiciels libres

Le projet GNU



By Aurelio A. Heckert, CC BY-SA 2.0

- Lancé à partir de 1983 par Richard Stallman.
- Pour créer un Unix complètement libre.
- Une **collection de projets** divers : GNU Emacs, GCC, glibc et GDB, GNU Bash, GNU Core Utilities (cat, ls, rm, etc.), GRUB, GParted, GIMP, GPG, etc.
- Et la famille de **licences GNU** (GPL, LGPL, AGPL).
- Création de la Free Software Foundation en 1985.

X Window System (X11)



By Sven, CC BY-SA 3.0

- Protocole de système de fenêtrage (encore) très utilisé sous Linux.
- Lancé à partir de 1984 au MIT.
- Distribué sous licence X11, à l'origine de la **licence MIT**.
- Implémentation actuelle X.Org, depuis 2004 par la X.Org Foundation.

Le noyau Linux



By Larry Ewing, Simon Budig, Garrett LeSage, CC0

- Lancé à partir de 1991 par Linus Torvalds.
- Premier logiciel libre **développé de manière collaborative** sur internet (envoi de patchs par newsgroups, puis par e-mail).
- Sous **licence GNU GPL 2.0** depuis 1992.
- A inspiré l'essai "The Cathedral and the Bazaar" à Eric Raymond.
- Création de la **Linux Foundation** en 2000, qui collecte aujourd'hui plus de 15 millions de dollars annuels de la part de ses plus de 1000 entreprises membres.

Les premières distributions GNU/Linux



By Debian, CC BY-SA 3.0



By Red Hat

- Debian :
 - Lancée à partir de 1993 par Ian Murdock.
 - L'une des toutes premières distributions Linux
 - Entièrement **communautaire** et presque entièrement libre.
- Red Hat (distribution et entreprise) :
 - Distribution lancée à partir de 1994 par Marc Ewing.
 - L'une des toutes premières entreprises à produire une distribution Linux et en **vendre la maintenance**.
 - Version communautaire : Fedora.
 - L'entreprise Red Hat est **rachetée par IBM** en 2019 pour 34 milliards de dollars.