

# Logiciel Libre

## Cours 6 — Aspects Philosophiques et Légaux

Stefano Zacchioli  
zack@pps.univ-paris-diderot.fr

Laboratoire PPS, Université Paris Diderot

2013-2014

URL <http://upsilon.cc/zack/teaching/1314/freesoftware/>  
Copyright © 2014 Stefano Zacchioli  
© 2007-2013 Ralf Treinen  
© 2004-2006 Roberto Di Cosmo  
License Creative Commons Attribution-ShareAlike 4.0 International License  
[http://creativecommons.org/licenses/by-sa/4.0/deed.en\\_US](http://creativecommons.org/licenses/by-sa/4.0/deed.en_US)



## Rappel du cours précédent

- Logiciel libre avant la lettre
- Les origines de UNIX chez AT&T
- Les origines d'internet
- BSD = UNIX + Internet
- La propriété des logiciels
- Richard Stallman et GNU
- Linux, et Open Source

# Outline

- 1 Le cadre légal
- 2 Licences et domaine public
- 3 Les libertés fondamentales
- 4 Logiciel libre selon Debian
- 5 Quelques exemples de licences libres

- Le cadre légal :  
secret industriel, marques, droit d'auteur, brevets.
- Le rôle des licences.
- Le domaine public.
- Les libertés fondamentales.
- Le logiciel libre selon Debian.
- Quelques exemples de licences libres et non libres.

- 1 Le cadre légal
- 2 Licences et domaine public
- 3 Les libertés fondamentales
- 4 Logiciel libre selon Debian
- 5 Quelques exemples de licences libres

Il y a quatre instruments majeurs de protection de la « propriété intellectuelle » :

**le secret industriel** : on ne dévoile pas les secrets de fabrication

Il y a quatre instruments majeurs de protection de la « propriété intellectuelle » :

**le secret industriel** : on ne dévoile pas les secrets de fabrication

**la marque** : créer une image publique reconnue du produit

Il y a quatre instruments majeurs de protection de la « propriété intellectuelle » :

**le secret industriel** : on ne dévoile pas les secrets de fabrication

**la marque** : créer une image publique reconnue du produit

**le droit d'auteur** : des « licences d'utilisation » du logiciel



Il y a quatre instruments majeurs de protection de la « propriété intellectuelle » :

**le secret industriel** : on ne dévoile pas les secrets de fabrication

**la marque** : créer une image publique reconnue du produit

**le droit d'auteur** : des « licences d'utilisation » du logiciel

**le brevet** : obtenir un monopole garanti par l'État sur des « inventions » ou « techniques » qui ne s'identifient pas à un produit

secret industriel : usage massif dans l'électronique et les jeux

**secret industriel** : usage massif dans l'électronique et les jeux  
**droit des marques** : très rentable, compatible avec l'Open Source

- secret industriel** : usage massif dans l'électronique et les jeux
- droit des marques** : très rentable, compatible avec l'Open Source
- droit d'auteur** : usage massif pour le logiciel

# L'expérience de 30 ans d'Informatique

**secret industriel** : usage massif dans l'électronique et les jeux

**droit des marques** : très rentable, compatible avec l'Open Source

**droit d'auteur** : usage massif pour le logiciel

**brevet** : très utilisé pour les médicaments, les voitures, les avions, usage presque uniquement détourné dans le monde du logiciel

# Outline

- 1 Le cadre légal
- 2 Licences et domaine public**
- 3 Les libertés fondamentales
- 4 Logiciel libre selon Debian
- 5 Quelques exemples de licences libres

- Contrat entre le détenteur du *copyright* (ou le droit d'auteur) et un utilisateur.
- Peut concerner par exemple :
  - ▶ l'utilisation
  - ▶ les modifications (quand code source disponible)
  - ▶ la rétro-ingénierie (quand code source pas disponible)
  - ▶ les copies, la redistribution

## Le cas des logiciels propriétaires

- *shrink-wrap licence* : il y a une notice avec le logiciel qui dit que le fait d'ouvrir la boîte constitue une acceptation de la licence (que souvent le client n'a pas lue).
- *click-through licence* : lors de l'installation, ou de la première utilisation, l'utilisateur doit cliquer sur un bouton pour accepter la licence (sinon le logiciel n'est pas utilisable).
- Souvent restriction forte de l'utilisation (nombre d'utilisateurs, nombre de processeurs, parfois du champs d'applications, ...)
- Licence de *source* : parfois possible (par exemple AT&T UNIX) mais normalement très cher.
- Copie, redistribution normalement pas permise.



## Le cas des logiciels libres

- La licence est normalement délivrée avec le logiciel (pas de *click-through*).
- L'utilisateur devient *propriétaire de la copie* qu'il a acquis (mais pas propriétaire du copyright!).
- L'utilisateur peut ne pas accepter la licence. Quand il l'accepte il obtient les droits que la licence lui donne (par exemple redistribution).
- Utilisation normalement pas restreinte.
- Modifications : le code source est disponible, les modifications sont permises mais parfois sujet à certaines conditions (par exemple de distribuer les modifications séparément du logiciel original).
- Redistribution : autorisée mais sujet à des conditions, en particulier sur la licence de produit redistribué.

# Pourquoi est-ce que les licences sont nécessaires dans le libre ?

- D'après la *convention de Berne* (1886), tous les droits restent avec le créateur (mais voir le cas du *domaine public*). Pas besoin de faire enregistrer les droits.
- Transfert des droits par un acte explicite (licence).
- Nécessité de connaître exactement le statut légal d'un logiciel pour les
  - ▶ Éditeurs de distributions (Debian etc.)
  - ▶ Opérateur de sites web qui veulent mettre des copies en ligne (miroirs)
  - ▶ Éditeurs et revendeurs de CD, DVD, ...
  - ▶ Programmeurs qui veulent produire des produits dérivés
  - ▶ Utilisateurs qui veulent baser leur infrastructure informatique sur ces logiciels

## Le cas du *domaine public*

- Logiciel libre n'est *pas* synonyme avec **domaine public**.
- Domaine public : absence de détenteur de droits.
- Conséquence : tout le monde peut utiliser le produit sans besoin d'une autorisation.
- En particulier pour les oeuvres d'art.
- Directive européenne : les oeuvres entrent dans le domaine public 70 ans après le décès de l'auteur (sauf cas particuliers).
- Logiciels : un auteur peut déclarer que son programme appartient au domaine public, et ainsi renoncer à ses droits.
- Les oeuvres produites par certaines agences du gouvernement américain sont dans le domaine public.

# Outline

- 1 Le cadre légal
- 2 Licences et domaine public
- 3 Les libertés fondamentales**
- 4 Logiciel libre selon Debian
- 5 Quelques exemples de licences libres

## Pas si simple ...

Même les meilleurs peuvent se tromper :

Rasmus Lerdorf Jun 8 1995, 9:00 am  
Announcing the Personal Home Page Tools (PHP Tools)  
version 1.0 [...]

## Pas si simple ...

Même les meilleurs peuvent se tromper :

Rasmus Lerdorf Jun 8 1995, 9:00 am  
Announcing the Personal Home Page Tools (PHP Tools)  
version 1.0 [...]  
The tools are in the [public domain](#)

## Pas si simple ...

Même les meilleurs peuvent se tromper :

Rasmus Lerdorf Jun 8 1995, 9:00 am  
Announcing the Personal Home Page Tools (PHP Tools)  
version 1.0 [...]  
The tools are in the [public domain](#)  
distributed under the [GNU Public License](#).

## Pas si simple ...

Même les meilleurs peuvent se tromper :

Rasmus Lerdorf Jun 8 1995, 9:00 am  
Announcing the Personal Home Page Tools (PHP Tools)  
version 1.0 [...]

The tools are in the [public domain](#)  
distributed under the [GNU Public License](#).  
Yes, that means they are [free](#)!



# Libre n'est pas gratuit

non libre, gratuit :

Internet Explorer, MacTCP, Acrobat Reader, freeware,  
etc.

# Libre n'est pas gratuit

non libre, gratuit :

Internet Explorer, MacTCP, Acrobat Reader, freeware,  
etc.

non libre, non gratuit :

*no comment ...*

# Libre n'est pas gratuit

non libre, gratuit :

Internet Explorer, MacTCP, Acrobat Reader, freeware, etc.

non libre, non gratuit :

*no comment . . .*

libre, gratuit :

Firefox, Linux, VLC, LibreOffice, FreeBSD, OpenBSD, Python, etc.

# Libre n'est pas gratuit

## non libre, gratuit :

Internet Explorer, MacTCP, Acrobat Reader, freeware, etc.

## non libre, non gratuit :

*no comment ...*

## libre, gratuit :

Firefox, Linux, VLC, LibreOffice, FreeBSD, OpenBSD, Python, etc.

## libre, non gratuit :

- on achète une distribution (un DVD)
- on *paye* des développeur pour produire du logiciel libre

# Les quatre libertés essentielles

Un programme est un **logiciel libre** si vous, en tant qu'utilisateur de ce programme, avez les **quatre libertés essentielles** :

- 0 la liberté d'**exécuter** le programme, pour tous les usages ;
- 1 la liberté d'**étudier** le fonctionnement du programme, et de le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez ;<sup>1</sup>
- 2 la liberté de **redistribuer des copies**, donc d'aider votre voisin ;
- 3 la liberté de **distribuer** aux autres des copies de vos **versions modifiées**.<sup>2</sup>

<https://www.gnu.org/philosophy/free-sw.fr.html>

---

1. l'accès au code source est une condition nécessaire

2. en faisant cela, vous donnez à toute la communauté une possibilité de profiter de vos changements ; l'accès au code source est une condition nécessaire.

# Outline

- 1 Le cadre légal
- 2 Licences et domaine public
- 3 Les libertés fondamentales
- 4 Logiciel libre selon Debian**
- 5 Quelques exemples de licences libres

# Principes du logiciel libre selon Debian

- **Debian Free Software Guidelines** (DFSG) : partie du *contrat social* du projet Debian (distribution de logiciel libre).
- 9 critères pour décider si une licence est libre ou pas (10-ème clause : exemples de licences).
- Il s'agit des critères du projet Debian qui sont largement acceptés par la communauté, mais pas par tout le monde !
- Dans un cas concret la question de la liberté ou pas d'une licence peut être épineuse (question d'interprétation des critères).



### 1 Redistribution libre et gratuite.

*La licence d'un composant de Debian ne doit pas empêcher quiconque de vendre ou de donner le logiciel sous forme de composant d'un ensemble (distribution) constitué de programmes provenant de différentes sources. La licence ne doit en ce cas requérir ni redevance ni rétribution.*

Par exemple quand la licence dit seulement « redistribution gratuite autorisée » ⇒ non libre.

Permet une plus grande distribution du LL, et (au moins avant Internet à haut débit) une forme de commerce autour du LL.



### ② Code source.

*Le programme doit inclure le code source et sa diffusion sous forme de code source comme de programme compilé doit être autorisée.*

Problème : il y a des cas dans lesquels il n'est pas évident en quoi consiste le code source (par exemple dans le cas des documentations, des programmes engendrés, des *firmware*).  
Source = la forme préférée par le développeur pour faire des modifications.

Pas d'*obfuscation* de code.

- Obfuscation : rendre le code source impénétrable pour un lecteur humain, en utilisant des codages autorisés par le langage de programmation.
- Techniquement, un code obfusqué reste un code source, un *compilateur* peut toujours le traduire vers un exécutable.
- Certains langages de programmation sont très propices à l'obfuscation (parfois malgré le programmeur), par exemple C.

# Exemple obfuscation

```
#include <stdio.h>
main(t,_,a)char *a;{return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2,_,+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+,#'/*{w+/w#cdnr/+,}{r/*de}+,/*{**+,/w{%,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\
;#q#n+,/+k#;*,/, 'r : 'd* '3 ,}{w+K w'K: '+}e#';dq# 'l \
q#'+d'K#!/+k#;q# 'r }eKK#}w' r }eKK{nl]}'/#;#q#n ')}{#}w ')}{nl]}'/+#n';d}rw' i;# \
){nl]!'/n{n#'; r{#w' r nc{nl]}'/#{l,+ 'K {rw' iK {;[{nl]}'/w#q#n 'wk nw' \
iwk{KK{nl]}'/w{%'l##w# ' i; :{nl]}'/*{q# 'ld;r '}{nlwb!/*de}'c \
; ;{nl}'-}{rw]}'/+,}{## '*}#nc, '#nw]}'/ +kd'+e}+;# 'rdq#w! nr '/' ')}+}{rl# '{n' ')}#
}'+'}##(!/'")
:t<-50?_==*a?putchar(31[a]):main(-65,_,a+1):main((*a=='/')+t,_,a+1)
:0<t?main(2,2,"%s"): *a=='/'||main(0,main(-61,*a,
"!ek;dc i@bK '(q)-[w]*%n+r3#l ,{ }:\ nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);}
```

Qu'est-ce que ce programme fait ?

### ③ Applications dérivées.

*La licence doit autoriser les modifications et les applications dérivées ainsi que leur distribution sous les mêmes termes que ceux de la licence du logiciel original.*

Exige seulement que la licence *autorise* de redistribuer sous les mêmes conditions.

N'exige pas que la licence demande que les modifications soient distribuées sous la même licence.

- 4 Intégrité du code source de l'auteur.  
*La licence peut défendre de distribuer le code source modifié seulement si elle autorise la distribution avec le code source de fichiers correctifs destinés à modifier le programme au moment de sa construction. La licence doit autoriser explicitement la distribution de logiciels créés à partir de code source modifié. Elle peut exiger que les applications dérivées portent un nom ou un numéro de version différent de ceux du logiciel original [ ... ]*

On autorise les auteurs de protéger leur réputation, et l'identité unique d'un certain logiciel (par exemple T<sub>E</sub>X).

Pont avec le droit des marques.

- 5 Aucune discrimination de personne ou de groupe.  
*La licence ne doit discriminer aucune personne ou groupe de personnes.*

Par exemple : « L'utilisation de ce programme est interdite à toute personne travaillant pour l'entreprise ... » ⇒ non libre.

Certains pays ont des restrictions d'exportations vers d'autres pays (États-Unis), ces restrictions ne doivent pas être reprises par la licence.

C'est une des clauses les plus délicates à interpréter.

## Le scénario « île désertique »

*Est-ce qu'un naufragé sur une île désertique (qui a sauvé son ordinateur portable, ainsi qu'une antenne satellite qui lui permet de recevoir des communications, mais pas d'en envoyer) peut satisfaire les conditions de la licence ?*

Exclus des licences qui exigent que toutes les modifications soient envoyées à quelqu'un d'autre.

*Est-ce qu'un utilisateur dans un état totalitaire, et qui ne peut pas risquer de révéler son identité ou le fait qu'il utilise un certain programme, peut satisfaire les conditions de la licence ?*

Exclus les licences qui exigent que les modifications soient envoyées à quelqu'un d'autre que le récepteur des binaires modifiés.



- ⑥ Aucune discrimination de champ d'application.  
*La licence ne doit pas défendre d'utiliser le logiciel dans un champ d'application particulier. Par exemple, elle ne doit pas défendre l'utilisation du logiciel dans une entreprise ou pour la recherche génétique.*

La licence ne doit non plus interdire l'utilisation d'un logiciel pour la création des logiciels propriétaires ...

### 7 Distribution de licence.

*Les droits attachés au programme doivent s'appliquer à tous ceux à qui il est distribué sans obligation pour aucune de ces parties de se conformer à une autre licence.*

Interdiction de faire le logiciel non libre par autre contraintes supplémentaires.

- ⑧ La licence ne doit pas être spécifique à Debian.  
*Les droits attachés au programme ne doivent pas dépendre du fait de son intégration au système Debian. Si le programme est extrait de Debian et utilisé et distribué sans Debian mais sous les termes de sa propre licence, tous les destinataires doivent jouir des mêmes droits que ceux accordés lorsqu'il se trouve au sein du système Debian.*

- 9 La licence ne doit pas contaminer d'autres logiciels.  
*La licence ne doit pas placer de restriction sur d'autres logiciels distribués avec le logiciel. Elle ne doit par exemple pas exiger que tous les autres programmes distribués sur le même support soient des logiciels libres.*

Attention cette clause ne parle que des autres logiciels distribués avec, **pas** du cas d'une *combinaison* de logiciel comme par exemple l'utilisation d'une bibliothèque.

# Outline

- 1 Le cadre légal
- 2 Licences et domaine public
- 3 Les libertés fondamentales
- 4 Logiciel libre selon Debian
- 5 Quelques exemples de licences libres**

- *quelques dizaines de licences libres*, généralement très permissives, *pour des centaines de milliers de logiciels...* ;
- on vous laisse modifier, distribuer, vendre et utiliser le logiciel, sous des conditions pas toujours équivalentes,
- d'où le *problème de compatibilité des licences*, et l'importance d'*éviter la prolifération des licences*.

Ayant une « oeuvre » (logiciel, bout de code) libre A,

- on peut *toujours* réaliser une « oeuvre dérivée » B (on change le code de A, ou on incorpore dans B du code de A), si la licence de B le permet ;
- mais le statut de l'« oeuvre dérivée » change selon la licence de A.

- A sous licence MIT ou BSD (version récente) : presque aucune contrainte sur B. Exemple : Utilisation de BSD en MacOS.
- A sous licence GPL : B doit être redistribué sous licence GPL (**copyleft**).  
Ses détracteurs parlent pour cette raison d'une licence *virale*.
- A sous licence LGPL : selon les cas, B peut être redistribué sous une licence non LGPL, mais avec des contraintes techniques.



En aucun des cas susmentionnés on a l'obligation de diffuser les modifications privées, mais

- la QPL originaire (Trolltech) comporte une obligation de divulgation *tout court*;
- l'Affero GPL comporte une obligation de divulgation envers les usagers du logiciels.

Bien de confusion sur les licences libres vient de la technicité du logiciel : lequel des actes suivants produit une « oeuvre dérivée » de A?

**incorporation de sources** : le source de B **contient** du code source de A,

**liaison statique** : B est **lié statiquement** avec le code de la librairie A,

**liaison dynamique** : B est **lié dynamique** avec le code de la librairie A.

Bien de confusion sur les licences libres vient de la technicité du logiciel : lequel des actes suivants produit une « oeuvre dérivée » de A?

**incorporation de sources** : le source de B **contient** du code source de A,

**liaison statique** : B est **lié statiquement** avec le code de la librairie A,

**liaison dynamique** : B est **lié dynamiquement** avec le code de la librairie A.

Les avis sont partagés, même parmi les experts.

Bien de confusion sur les licences libres vient de la technicité du logiciel : lequel des actes suivants produit une « oeuvre dérivée » de A?

**incorporation de sources** : le source de B **contient** du code source de A,

**liaison statique** : B est **lié statiquement** avec le code de la librairie A,

**liaison dynamique** : B est **lié dynamiquement** avec le code de la librairie A.

Les avis sont partagés, même parmi les experts.  
Tout cela n'est pas spécifique au Logiciel Libre.

## Licence BSD

<http://opensource.org/licenses/BSD-3-Clause>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

[...]



freeBSD<sup>®</sup>



<http://www.gnu.org/licenses/gpl-3.0.html>

Extrait de la traduction française, légèrement abrégé :

**Article 2 :** *Vous pouvez modifier votre copie ou vos copies du Programme ou partie de celui-ci, ou d'un travail basé sur ce Programme, et copier et distribuer ces modifications selon les termes de l'article 1, à condition de Vous conformer également aux conditions suivantes :*

- *Distribuer sous les termes de la GPL l'ensemble de toute réalisation contenant tout ou partie du Programme, avec ou sans modifications.*



# Le cas de la licence AGPL

<http://www.gnu.org/licenses/agpl-3.0.html>

- *Affero General Public License* de la maison GNU
- S'adresse à des logiciels qui ne sont pas installés chez l'utilisateur mais utilisable via le réseaux (la GPL ne s'applique pas dans ce cas)
- Comme GPL, mais contient en plus une clause qui exige à un *fournisseur de service via le réseaux* de publier les modifications qu'il a apporté au logiciel.



*<phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return.  
[Poul-Henning Kamp].*

Est-ce une licence libre ?



*... wrote this file. As long as you retain this notice you can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you **must** buy me a beer in return..*

Toujours libre?

# Free Document Dissemination Licence

<http://bat8.inria.fr/~lang/licence/v1/fddl.html>

Extraits :

*This document may be freely read, stored, reproduced, disseminated, translated or quoted by any means and on any medium provided the following conditions are met :*

- *no modification is made other than cosmetic, change of representation format, translation, correction of obvious syntactic errors, or as permitted by the clauses below ;*
- *comments and other additions may be **inserted**, provided they clearly appear as such ; translations or fragments must clearly refer to an original complete version, preferably one that is easily accessed whenever possible ;*

Licence libre ?

# Matrice de compatibilité (exemple, YMMV)

## Matrice de compatibilité

Licences compatibles ?

	CeCill	GPL	LGPL	ZPL	ApacheSL	MPL	Artistic	MIT	BSD	QPL
QPL	P	NC	L(LGPL):OPL L(QPL):LGPL, sinon NC	P	P	L(QPL):P/ NC	QPL/P	QPL	3:OPL/ 4:QPL	Piegel
BSD	(3:CeCill/4: NC), L(I): Original	3:GPL /4: NC	3:I/4:L(LGPL): BSD, sinon NC	3:ZPL ou P / 4: BSD	3:ASL / 4: Hybr	3:I ou P / 4:BSD ou P	(3:Artistic / 4: P ou Hybr), L(I) = Original	3:I/ 4: BSD	BSD4B ou P	
MIT	CeCill/P	GPL	L(LGPL):I, sinon LGPL	ZPL ou P	ASL	MPL ou P	Artistic	X		
Artistic	NC/CeCill	GPL	L(LGPL):I, sinon LGPL	Artistic ou P	ASL	MPL ou P	X			
MPL	MPL/CeCill	NC	L(LGPL):MPL, sinon LGPL(MPL v1.1) ou NC (MPL v1)	P	ASL 1.0/1.1: NC, ASL 2.0: P	X				
ApacheSL	NC/P	NC	L(LGPL):ASL, sinon NC	ZPL + ASL 2.0 = ASL 2.0, ZPL + ASL 1.0/1.1 = P	X					
ZPL	CeCILL/P	GPL	L(LGPL):ZPL, sinon LGPL	X						
LGPL	L(LGPL):CeCill, sinon GPL	GPL	X							
GPL	GPL/P	X								

gage que moi... [1] il étai...

## légende

- n/a : case non traitée
- le nom d'une licence : licence sous laquelle le programme résultant pourra être mis
- NC : les licences ne sont pas compatibles
- I : le résultat peut être mis indifféremment sous l'une ou l'autre des licences
- L(X) : on considère une librairie sous la licence X
- ? : deux groupes ne sont pas d'accord sur le résultat et en débattent
- Hybr : Licence Hybride reprenant les clauses de chaque licence
- P : Le programme est composé de morceaux de codes avec des licences différentes .

## Qui décide si une licence est libre ou pas ?

- Les quatre libertés fondamentales sont incontestées parmi les acteurs du logiciel libre.
- La **Free Software Foundation** maintient une liste de licences libres <http://www.gnu.org/licenses/license-list.html>
- La **Open Source Initiative** fait la même chose <http://opensource.org/licenses>, sur la base d'une définition d'Open Source inspiré aux DFSG
- Dans le projet Debian, il y a les **Debian Free Software Guidelines** pour déterminer si un licence est libre
- Mais on n'est pas toujours d'accord dans les détails de certaines licences.

# Champs d'application des licences

Est-ce que les libertés fondamentales sont requises pour :

- Code programme : bien sûr !
- Firmware ?
- La source des graphismes ?
- La source des polices de caractères ?
- Les documentations ?

# Champs d'application des licences

Est-ce que les libertés fondamentales sont requises pour :

- Code programme : bien sûr !
- Firmware ?
- La source des graphismes ?
- La source des polices de caractères ?
- Les documentations ?

Les avis sont partagés, sauf sur la liberté du code programme.

Voir :

- la dispute entre GNU et Debian à propos de la *GNU Free Documentation Licence (GFDL)*
- le mouvement *Free Culture*—inspiré du livre homonyme <http://www.free-culture.cc/>—partisan pour l'application de même règles des libertés à tout forme de contenu numérique