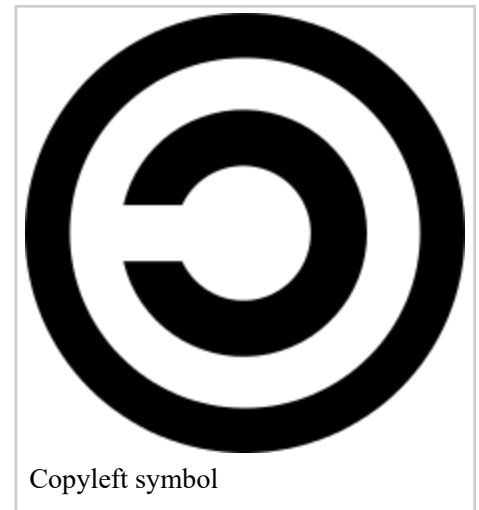


Copyleft

Copyleft (a play on the word *copyright*) is the practice of offering people the right to freely distribute copies and modified versions of a work with the stipulation that the same rights be preserved in derivative works down the line.^[1] Copyleft software licenses are considered *protective*, as contrasted with permissive free software licenses.^[2]

Copyleft is a form of licensing, and can be used to maintain copyright conditions for works ranging from computer software, to documents, to art. In general, copyright law is used by an author to prohibit recipients from reproducing, adapting, or distributing copies of their work. In contrast, under copyleft, an author may give every person who receives a copy of the work permission to reproduce, adapt, or distribute it, with the accompanying requirement that any resulting copies or adaptations are also bound by the same licensing agreement.



Copyleft licenses for software require that information necessary for reproducing and modifying the work must be made available to recipients of the binaries. The source code files will usually contain a copy of the license terms and acknowledge the authors.

Copyleft type licenses are a novel use of existing copyright law to ensure a work remains freely available. The GNU General Public License (GPL), originally written by Richard Stallman, was the first software copyleft license to see extensive use, and continues to dominate in that area.^[3] Creative Commons, a non-profit organization founded by Lawrence Lessig, provides a similar license provision condition called share-alike.^[4]

Reciprocity

Copyleft is a copyright licensing scheme in which an author surrenders some, but not all, rights under copyright law. Instead of allowing a work to fall completely into the public domain (where no ownership of copyright is claimed), copyleft allows an author to impose some restrictions on those who want to engage in activities that would more usually be reserved by the copyright holder. Under copyleft, derived works may be produced provided they are released under the compatible copyleft scheme.

The underlying principle is that one benefits freely from the work of others, but any modifications one makes public must be released under similar terms. For this reason some copyleft licenses are also known as reciprocal licenses, they have also been described as "viral" due to their self-perpetuating terms.^[5] Under fair use, however, the copyleft license may be superseded, just like regular copyrights. Therefore, any person utilizing a copyleft-licensed source for their own work is free to choose any other license provided they meet the fair use standard.^[6]

According to Free Software Foundation compliance engineer David Turner, the term *viral license* creates a misunderstanding and a fear of using copylefted free software.^[7] David McGowan has written that there is no reason to believe the GPL could force proprietary software to become free software, but could "try to enjoin the firm from distributing commercially a program that combined with the GPL'd code to form a derivative work, and to recover damages for infringement." If the firm "actually copied code from a GPL'd program, such a suit would be a perfectly ordinary assertion of copyright, which most private firms would defend if the shoe were on

the other foot."^[8] Richard Stallman has described this view with an analogy, saying, "The GPL's domain does not spread by proximity or contact, only by deliberate inclusion of GPL-covered code in your program. It spreads like a spider plant, not like a virus."^[9]

While copyright law gives software authors control over copying, distribution and modification of their works, the goal of copyleft is to give all users of the software the freedom to carry out these activities. In this way, copyleft licenses are distinct from other types of free software licenses, which do not guarantee that all subsequent recipients of the program receive these rights, or the source code needed to make them effective. In particular, permissive free software licenses such as BSD allow re-distributors to remove some or all these rights, and do not require the distribution of source code.

Reuses of open source software which subvert the freedom of the software by adding extra restrictions are called open source hijacking. Other forms of open source licensing are susceptible to hijacking, so creators who desire that their work remain free may choose to use copyleft. The Open Directory Project (ODP, also known as DMOZ), which is created and maintained entirely by volunteer editors, can be considered an example of open-source hijacking. Since all of the top search engines use DMOZ, it is an extremely important public commodity which is entirely built by volunteer contributions. Despite this, the DMOZ has remained a commercial product for its entire existence.^[10]

The economic incentives to work on copyleft content can vary. Traditional copyright law is designed to promote progress by providing economic benefits to creators. When choosing to copyleft their work, content creators may seek complementary benefits like recognition from their peers. The open-source culture had been described as a gift culture, where social status is determined by an individual's contributions.^[11] Working on free software may also be an outlet for programmers to fill a need they have noticed. For some creators, keeping their work open is an incentive in and of itself. For these programmers, preventing commercial enterprises from absorbing and selling their product is another incentive.

Copyleft software has economic effects beyond the individual creators. The presence of quality copyleft software can force commercial producers to increase the quality of their products, which must compete with free software.^[12] This may also have the effect of preventing large commercial entities from applying monopoly prices. However, competition with proprietary software can also be a reason to forego copyleft. The Free Software Foundation recommends that when "widespread use of the code is vital for advancing the cause of free software," allowing the code to be copied and used freely is more important than a copyleft.

History

An early use of the word "copyleft" was in Li-Chen Wang's Palo Alto Tiny BASIC's distribution notice "@COPYLEFT ALL WRONGS RESERVED" in June 1976, but Tiny BASIC was not distributed under any form of copyleft distribution terms, so the wordplay is the only similarity.^{[13][14]}

The concept of copyleft was described in Richard Stallman's GNU Manifesto in 1985, where he wrote:

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

Stallman worked a few years earlier on a Lisp interpreter. Symbolics asked to use the Lisp interpreter, and

Stallman agreed to supply them with a public domain version of his work. Symbolics extended and improved the Lisp interpreter, but when Stallman wanted access to the improvements that Symbolics had made to his interpreter, Symbolics refused. Stallman then, in 1984, proceeded to work towards eradicating this emerging behavior and culture of proprietary software, which he named *software hoarding*. This was not the first time Stallman had dealt with proprietary software, but he deemed this interaction as a "turning point". He justified software sharing, protesting that when sharing, the software online can be copied without the loss of the original piece of work. Everyone is a winner. The software can be used multiple times without ever being damaged or wearing out.^{[15][16]}

As Stallman deemed it impractical in the short term to eliminate current copyright law and the wrongs he perceived it perpetuated, he decided to work within the framework of existing law; in 1985,^[17] he created his own copyright license, the Emacs General Public License,^[18] the first copyleft license. This later evolved into the GNU General Public License, which is now one of the most popular Free Software licenses. For the first time a copyright holder had taken steps to ensure that the maximal number of rights be perpetually transferred to a program's users, no matter what subsequent revisions anyone made to the original program. This original GPL did not grant rights to the public at large, only those who had already received the program; but it was the best that could be done under existing law.

The new license was not at this time given the copyleft label.^[19] Richard Stallman stated that the use of "Copyleft" comes from Don Hopkins, who mailed him a letter in 1984 or 1985, on which was written: "Copyleft – all rights reversed."^[19] The term "copyleft" with the notation "All Rights Reversed" was also in use in the early 1970s, within the *Principia Discordia*, which may have inspired Hopkins or influenced other usage. And in the arts Ray Johnson had earlier coined the term independently as it pertained to his making of and distribution of his mixed media imagery in his mail art and ephemeral gifts, for which he encouraged the making of derivative works. (While the phrase appears briefly as (or on) one of his pieces in the 2002 documentary *How to Draw a Bunny*, Johnson himself is not referenced in the 2001 documentary *Revolution OS*.)

Some have suggested that copyleft became a divisive issue in the ideological strife between the Open Source Initiative and the free software movement.^[20] However, there is evidence that copyleft is both accepted and proposed by both parties:

- Both the OSI and the FSF have copyleft and non-copyleft licenses in their respective lists of accepted licenses.^{[21][22]}
- The OSI's original Legal Counsel Lawrence Rosen has written a copyleft license, the Open Software License.
- The OSI's licensing how-to recognises the GPL as a "best practice" license.^[23]
- Some of the software programs of the GNU Project are published under non-copyleft licenses.^[24]
- Stallman himself has endorsed the use of non-copyleft licenses in certain circumstances, most recently in the case of the Ogg Vorbis license change.^[25]

Applying copyleft

Common practice for using copyleft is to codify the copying terms for a work with a license. Any such license typically gives each person possessing a copy of the work the same freedoms as the author, including (from the Free Software Definition):

- Freedom 0 – the freedom to use the work,
- Freedom 1 – the freedom to study the work,
- Freedom 2 – the freedom to copy and share the work with others,
- Freedom 3 – the freedom to modify the work, and the freedom to distribute modified and therefore derivative works.

(Note that the list begins from 0 as a reference to computer programming, where zero-based numbering is prevalent.)

These freedoms do not ensure that a derivative work will be distributed under the same liberal terms. In order for the work to be truly copyleft, the license has to ensure that the author of a derived work can only distribute such works under the same or equivalent license.

In addition to restrictions on copying, copyleft licenses address other possible impediments. These include ensuring the rights cannot be later revoked and requiring the work and its derivatives to be provided in a form that facilitates modification. In software, this requires that the source code of the derived work be made available together with the software itself.

Copyleft licenses necessarily make creative use of relevant rules and laws. For example, when using copyright law, those who contribute to a work under copyleft usually must gain, defer or assign copyright holder status. By submitting the copyright of their contributions under a copyleft license, they deliberately give up some of the rights that normally follow from copyright, including the right to be the unique distributor of copies of the work.

Some laws used for copyleft licenses vary from one country to another, and may also be granted in terms that vary from country to country. For example, in some countries it is acceptable to sell a software product without warranty, in standard GNU GPL style (see articles 11 and 12 of the GNU GPL version 2), while in most European countries it is not permitted for a software distributor to waive all warranties regarding a sold product. For this reason the extent of such warranties are specified in most European copyleft licenses. Regarding that, see the European Union Public Licence EUPL,^[26] or the CeCILL license,^[27] a license that allows one to use GNU GPL (see article 5 of the EUPL and article 5.3.4 of CeCILL) in combination with a limited warranty (see article 7 and 8 of the EUPL and 9 of CeCILL). For projects which will be run over a network, a variation of the GPL is provided in the Affero General Public License, which ensures that the source code is available to users of network software.

Types and relation to other licenses

See also: Free software licences § Restrictions

Copyleft is a distinguishing feature of some free software licenses. Many free software licenses are not copyleft licenses because they do not require the licensee to distribute derivative works under the same license. There is an ongoing debate as to which class of license provides the greater degree of freedom. This debate hinges on complex issues such as the definition of freedom and whose freedoms are more important, or whether to maximize the freedom of all potential future recipients of a work (*freedom from* the creation of proprietary software). Non-copyleft free software licenses maximize the freedom of the initial recipient (*freedom to* create proprietary software).

In common with the Creative Commons share-alike licensing system, GNU's Free Documentation License allows authors to apply limitations to certain sections of their work, exempting some parts of their creation from the full copyleft mechanism. In the case of the GFDL, these limitations include the use of invariant sections,

which may not be altered by future editors. The initial intention of the GFDL was as a device for supporting the documentation of copylefted software. However, the result is that it can be used for any kind of document.

Strong and weak copyleft

The strength of the copyleft governing a work is an expression of the extent that the copyleft provisions can be efficiently imposed on all kinds of derived works. "Weak copyleft" refers to licenses where not all derived works inherit the copyleft license; whether a derived work inherits or not often depends on the manner in which it was derived.

"Weak copyleft" licenses are generally used for the creation of software libraries, to allow other software to link to the library, and then be redistributed without the legal requirement for the work to be distributed under the library's copyleft license. Only changes to the weak-copylefted software itself become subject to the copyleft provisions of such a license, not changes to the software that links to it. This allows programs of any license to be compiled and linked against copylefted libraries such as glibc (the GNU project's implementation of the C standard library), and then redistributed without any re-licensing required.

The most well known free software license that uses strong copyleft is the GNU General Public License. Free software licenses that use "weak" copyleft include the GNU Lesser General Public License and the Mozilla Public License. Examples of non-copyleft free software licenses include the X11 license, Apache license and the BSD licenses.

The Design Science License is a strong copyleft license that can apply to any work that is not software or documentation, such as art, music, sports photography, and video. It is hosted on the Free Software Foundation website's license list,^[28] but it is not considered compatible with the GPL by the Free Software Foundation.

The concrete effect of strong vs. weak copyleft has yet to be tested in court.^[29]

Full and partial copyleft

"Full" and "partial" copyleft relate to another issue: Full copyleft exists when all parts of a work (except the license itself) may only be modified and distributed under the terms of the work's copyleft license. Partial copyleft exempts some parts of the work from the copyleft provisions, thus permitting distribution of some modifications under terms other than the copyleft license, or in some other way does not impose all the principles of copylefting on the work. For example, the GPL linking exception made for some software packages (see below).

Share-alike

Share-alike imposes the requirement that any freedom that is granted regarding the original work must be granted on exactly the same or compatible terms in any derived work: this implies that any copyleft license is automatically a share-alike license, but not the other way around, as some share-alike licenses include further restrictions, for instance prohibiting commercial use. Another restriction is that not everyone wants to share their work and some share-alike agreements require that the whole body of work be shared, even if the author



only wants to share a certain part. The plus side for the author of the source code is that any modification to the code will not only benefit the company, but the author will be recognized and hold equal claim over the changed code.^{[30][31]} Some permutations of the Creative Commons licenses are examples of share-alike.

Viral licensing

Main article: Viral license


Viral license is a pejorative name for copyleft licenses.^{[32][33][34][35][36]} It originates from the terms 'General Public Virus' or 'GNU Public Virus' (GPV), which dates back to 1990, a year after the GPLv1 was released.^{[37][38][39]} The name "viral licenses" refers to the fact that any works derived from a copyleft work must preserve the copyleft permissions when distributed. Some BSD License advocates used the term derisively in regards to the GPL's tendency to absorb BSD licensed code without allowing the original BSD work to benefit from it, while at the same time promoting itself as "freer" than other licenses.^{[40][41][42]} Microsoft vice-president Craig Mundie remarked, "This viral aspect of the GPL poses a threat to the intellectual property of any organization making use of it."^[43] In another context, Steve Ballmer declared that code released under GPL is useless to the commercial sector (since it can only be used if the resulting surrounding code is licensed under a GPL compatible license), describing it thus as "a cancer that attaches itself in an intellectual property sense to everything it touches".^[44] In response to Microsoft's attacks on the GPL, several prominent Free Software developers and advocates released a joint statement supporting the license.^[45] According to FSF compliance engineer David Turner, it creates a misunderstanding and a fear of using copylefted free software.^[46] David McGowan has written that there is no reason to believe the GPL could force proprietary software to become free software, but could "try to enjoin the firm from distributing commercially a program that combined with the GPL'd code to form a derivative work, and to recover damages for infringement." If the firm "actually copied code from a GPL'd program, such a suit would be a perfectly ordinary assertion of copyright, which most private firms would defend if the shoe were on the other foot."^[47]

Popular copyleft licenses, such as the GPL, have a clause allowing components to interact with non-copyleft components as long as the communication is abstract, such as executing a command-line tool with a set of switches or interacting with a Web server.^[48] As a consequence, even if one module of an otherwise non-copyleft product is placed under the GPL, it may still be legal for other components to communicate with it normally. This allowed communication may or may not include reusing libraries or routines via dynamic linking – some commentators say it does,^[49] the FSF asserts it does not and explicitly adds an exception allowing it in the license for the GNU Classpath re-implementation of the Java library. This ambiguity is an important difference between the GPL and the LGPL, in that the LGPL specifically allows linking or compiling with the covered work.^[50]

Symbol

The copyleft symbol is a backwards C in a circle (copyright symbol © mirrored). It has no legal significance.^[51]

Because it is currently unavailable on Unicode, it can be approximated with character U+2184 ◌ LATIN SMALL LETTER REVERSED C or the more widely available character U+0254 ◌ LATIN SMALL LETTER OPEN O between parenthesis '◌)' or, if supported by the application, by combining it with the character U+20DD ◌COMBINING

ENCLOSING CIRCLE: 'C'^[52] In 2016, a proposal for encoding the coryleft symbol^[53] was accepted by the Unicode Consortium to be added to a future version of Unicode at code point U+1F12F  CORYLEFT SYMBOL.^[54]


See also

- All rights reversed
- Opposition to copyright
- Comparison of open source and closed source
- Commercial use of coryleft works
- Copyright
- Creative Commons licences
- Free content
- Free Culture movement
- Free Software Movement
- GNU General Public License
- HESSLA – a license which prohibits uses that violate human rights or add spyware
- History of free and open-source software
- Open content
- Patentleft
- Permissive free software licence
- Public copyright license
- Public domain
- Share-alike

Notes and references

1. ↑ "What is Coryleft?". Retrieved 29 August 2010.
2. ↑ Open Source from a Proprietary Perspective at the Wayback Machine (archive index)
3. ↑ Stallman, Richard (June 29, 2007). "GNU General Public License". *GNU*. Free Software Foundation. Retrieved August 14, 2015.
4. ↑ "Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)". *Creative Commons*. Creative Commons. Retrieved August 14, 2015.
5. ↑ Mundie, Craig (3 May 2001). "Prepared Text of Remarks by Craig Mundie, Microsoft Senior Vice President – The Commercial Software Model". New York University Stern School of Business. Archived from the original on 21 June 2005. Retrieved 1 October 2009.
6. ↑ Kirk St.Amant & Brian Still (2008). "Examining Open Source Software Licenses through the Creative Commons Licensing Model". *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. Information Science Reference. pp. 382 of 728. ISBN 1-59140-999-3.
7. ↑ Byfield, Bruce (29 August 2006). "IT Manager's Journal: 10 Common Misunderstandings About the GPL". Retrieved 23 August 2008.
8. ↑ David McGowan (2005), "Legal Aspects of Free and Open Source Software", in Joseph Feller; Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani, *Perspectives on Free and Open Source Software*, MIT Press, p. 382, ISBN 0-262-06246-1
9. ↑ Poynder, Richard (21 March 2006). "The Basement Interviews: Freeing the Code". Retrieved 5 February 2010.
10. ↑ Ciffolilli, Andrea (6 September 2004). "The Economics of Open Source Hijacking and the Declining Quality of Digital Information Resources". *First Monday*. Retrieved 1 May 2014.
11. ↑ Maher, Marcus (2000). "Open Source Software: The Success of an Alternative Intellectual Property Incentive Paradigm" (PDF). Fordham Intellectual Property, Media & Entertainment Law Journal. Retrieved 1 May 2014.
12. ↑ Mustonen, Mikko. "Coryleft - The Economics of Linux and Other Open Source Software" (PDF). Information Economics and Policy. Retrieved 1 May 2014.
13. ↑ Wang, Li-Chen (May 1976). "Palo Alto Tiny BASIC". *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Running Light Without Overbyte*. **1** (5): 12–25. (NB. Source code begins with the following six lines. "TINY BASIC FOR INTEL 8080; VERSION 1.0; BY LI-CHEN WANG; 10 JUNE, 1976; @CORYLEFT; ALL WRONGS RESERVED". The June date in the May issue is correct. The magazine was behind schedule, the June and

July issues were combined to catch up.)

14. ↑ Rauskolb, Roger (December 1976). "Dr. Wang's Palo Alto Tiny BASIC". *Interface Age*. **2** (1): 92–108. (NB. The source code begins with the following nine lines: "TINY BASIC FOR INTEL 8080; VERSION 2.0; BY LI-CHEN WANG; MODIFIED AND TRANSLATED TO INTEL MNEMONICS; BY ROGER RAUSKOLB; 10 OCTOBER, 1976 ; @COPYLEFT; ALL WRONGS RESERVED")
15. ↑ Carver, Brian W. "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses". Berkeley Technology Law Journal. Retrieved 6 February 2012.
16. ↑ Williams, Sam (March 2002). "7". *Free as in Freedom – Richard Stallman's Crusade for Free Software*. O'Reilly Media. ISBN 0-596-00287-4.
17. ↑ Moody, Glyn (2002). *Rebel Code*. p. 26.
18. ↑ "Emacs General Public License". 5 July 2001. Retrieved 23 August 2008.
19. 1 2 Stallman, Richard (21 January 2008). "About the GNU Project". Free Software Foundation. Retrieved 23 August 2008.
20. ↑ Biancuzzi, Federico (30 June 2005). "ESR: "We Don't Need the GPL Anymore" ". ONLamp.com. Retrieved 23 August 2008.
21. ↑ Tiemann, Michael (18 September 2006). "Licenses by Name". Retrieved 23 August 2008.
22. ↑ "Licenses". 5 June 2008. Retrieved 23 August 2008.
23. ↑ Raymond, Eric Steven (9 November 2002). "Licensing HOWTO". Retrieved 23 August 2008.
24. ↑ "What the GPLv3 Means for MS-Novell Agreement". Yro.slashdot.org. 8 March 2007. Retrieved 14 May 2014.
25. ↑ Stallman, Richard (26 February 2001). "LWN.net: RMS on the Ogg Vorbis license". Retrieved 23 August 2008.
"[M]y agreement with the idea of a lax [Ogg/Vorbis] license in this special case is just as pragmatic as my preference for the GPL in most cases. In both cases it is a matter of how we can attain freedom."
26. ↑ "The EUPL – European Union Public Licence". European Commission. Retrieved 9 January 2007.
27. ↑ "Free Software Licensing Agreement CeCILL" (PDF). INRIA. Retrieved 24 August 2010.
28. ↑ Various Licenses and Comments about Them
29. ↑ Patrice-Emmanuel Schmitz (April 3, 2015). "The meaning of "Copyleft" in EUPL". *Joinup*.
30. ↑ Engelfriet, Arnoud. "Choosing an Open Source License". Athens University of Economics and Business. Retrieved 6 February 2012.
31. ↑ Rob Myers, NonCommercial Sharealike is not Copyleft
32. ↑ "Microsoft license spurns open source". *CNET*. CBS Interactive.
33. ↑ "Some rights reserved: the alternatives to copyright". *Wired UK*.
34. ↑ "Glossary". *a2knetwork.org*.
35. ↑ Jason Greenberg; Daniel Glazer (March 2013). "Inoculating Your Purchase – Contractual Protection from Viral Licenses in M&A Transactions" (PDF). *Fried Frank*. Association of Corporate Counsel. Retrieved June 10, 2016.
36. ↑ Philip Wood; Amy Ryburn; Allan Yeoman; Andrew Matangi; Steve Nightingale (July 1, 2013). "Legal update on Information and Communication Technology – July 2013". Archived from the original on 21 September 2013. Retrieved June 10, 2016.
37. ↑ Vixie, Paul (6 March 2006). "Re: Section 5.2 (IPR encumbrance) in TAK rollover requirement draft". IETF Namedroppers mailing list. Archived from the original on 27 September 2007. Retrieved 29 April 2007.
38. ↑ "General Public Virus". *Jargon File 2.2.1*. 15 December 1990. Retrieved 29 April 2007.
39. ↑ Hackv  n, Stig (September 1999). "Reverse-engineering the GNU Public Virus – Is copyleft too much of a good thing?". *Linux Journal*. Archived from the original on 18 July 2011. Retrieved 29 April 2007.
40. ↑ Stewart, Bill (8 October 1998). "Re: propose: 'cypherpunks license' (Re: Wanted: Twofish source code)". Cypherpunks mailing list. Retrieved 29 April 2007.
41. ↑ Buck, Joe (10 October 2000). "Re: Using of parse tree externally". GCC mailing list. Retrieved 29 April 2007.
42. ↑ Griffis, L. Adrian (15 July 2000). "The GNU Public Virus". Retrieved 29 April 2007.
43. ↑ Mundie, Craig (3 May 2001). "Speech Transcript – Craig Mundie". New York University Stern School of Business. Archived from the original on 21 June 2005. Retrieved 23 August 2008.
44. ↑ Newbart, Dave (1 June 2001). "Microsoft CEO takes launch break with the Sun-Times". Chicago Sun-Times. Archived from the original on 15 June 2001.(Internet archive link)
45. ↑  *Free Software Leaders Stand Together*. Wikisource.

46. ↑ Byfield, Bruce (29 August 2006). "IT Manager's Journal: 10 Common Misunderstandings About the GPL". Retrieved 23 August 2008.

47. ↑ David McGowan (2005). "Legal Aspects of Free and Open Source Software". In Joseph Feller; Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani. *Perspectives on Free and Open Source Software*. MIT Press. p. 382. ISBN 0-262-06246-1.

48. ↑ "Frequently Asked Questions about the GNU Licenses". Free Software Foundation. 24 June 2008. Retrieved 23 August 2008.

49. ↑ Raymond, Eric Steven (9 November 2002). "Licensing HOWTO". Retrieved 21 March 2010.

50. ↑ Stallman, Richard. Why you shouldn't use the Lesser GPL for your next library. Free Software Foundation official website.


51. ↑ Hall, G. Brent (2008). *Open Source Approaches in Spatial Data Handling*. Springer. p. 29. ISBN 3-540-74830-X. Additional ISBN 978-3-540-74830-4. See *Open Source Approaches in Spatial Data Handling*. p. 39.


52. ↑ "Unicode copyleft inquiry". 6 May 2000.

53. ↑ "Proposal to add the Copyleft Symbol to Unicode" (PDF).

54. ↑ "Time to rejoice! COPYLEFT SYMBOL was accepted at #UTC147 at code point U+1F12F for a future version of #unicode". 10 May 2016.

External links

- What is copyleft? – by Richard Stallman.
 - GNU's Bulletin, vol. 1 no. 5 – first appearance of article on *What is copyleft?*
 - Freedom or Power by Richard Stallman and Bradley Kuhn.
 - Copyleft: Pragmatic Idealism – by Richard Stallman.
 - "Copyleft and Copyright article". *Eye Magazine*.
 - Two explanations of copyleft and its history by Richard Stallman: one in April 2006, and one in June 2006.
 - Article: Working Without Copyleft by Bjørn Reese and Daniel Stenberg, 19 December 2001.
 - David M. Berry (2008). *Copy, Rip, Burn: The Politics of Copyleft and Open Source*. London: Pluto Press.
 - Steven Weber (2005). *The Success of Open Source*. Harvard: Harvard University Press.
-  Wikimedia Commons has media related to ***Symbols of Copyright-Copyleft***.

 Look up ***copyleft*** in Wiktionary, the free dictionary.

Free and open-source software

- Alternative terms for free software ·
- Comparison of open-source and closed-source software ·
- General

Comparison of source code hosting facilities · Free software ·

Free software project directories · Gratis versus libre · Long-term support ·

Open-source software · Open-source software development · Outline
- Audio · Bioinformatics · Codecs · Collaboration · Configuration management ·
- Device drivers (Graphics · Wireless) · Geophysics · Health · Mathematics ·
- Software packages

Operating systems · Programming languages · Routing · Statistics · Television ·

Video games · Web applications (Content management systems · E-commerce) ·

Word processors · Android apps · iOS apps · Commercial · Trademarked ·

Formerly proprietary

Community	Free software movement · History · Open-source software movement · Organizations · Events
Licenses	Apache · APSL · Artistic · Beerware · Boost · BSD · CC0 · CDDL · EPL · GNU GPL · GNU LGPL · ISC · MIT · MPL · Ms-PL/RL · WTFPL · zlib
License types and standards	Comparison of free and open-source software licenses · Contributor License Agreement · Copyfree · Copyleft · Debian Free Software Guidelines · Definition of Free Cultural Works · Free license · The Free Software Definition · The Open Source Definition · Open-source license · Permissive free software licence · Public domain · Viral license
Challenges	Binary blob · Digital rights management · Hardware restrictions · License proliferation · Mozilla software rebranding · Proprietary software · SCO/Linux controversies · Secure boot · Software patents · Software security · Trusted Computing
Related topics	<i>The Cathedral and the Bazaar</i> · Forking · Microsoft Open Specification Promise · <i>Revolution OS</i>
<div>  Book ·  Category ·  Commons ·  Portal </div>	
Intellectual property activism	
Issues	Copyright infringement · Digital rights management · Gripe site · Intellectual property · Legal aspects of file sharing · Mashup (digital · videos · music) · Music piracy · Orphan works · Patents (biological · software · software patent debate · trolling) · Public domain
Concepts	All rights reversed · Alternative compensation system · Anti-copyright notice · Business models for open source software · Copyleft · Commercial use of copyleft works · Commons-based peer production · Free content · Free software license · Infoanarchism · Libertarian positions · Open content · Open design · Open Music Model · Open patent · Open-source hardware · Open source software · Prize system (contests) · Share-alike · Video on demand
Movements	Access to Knowledge movement · Anti-copyright · Cultural environmentalism · Free culture movement · Free software movement
Organizations	Copyright Alliance · Creative Commons · · Electronic Frontier Foundation · Free Software Foundation · Open Rights Group · Organization for Transformative Works · The Pirate Bay · Piratbyrån · Pirate Party · Sci-Hub · Students for Free Culture
People	Alexandra Elbakyan · Rickard Falkvinge · Lawrence Lessig · Richard Stallman · Peter Sunde · Peter Suber · Aaron Swartz
Documentaries	<i>Steal This Film</i> (2006, 2007) · <i>Good Copy Bad Copy</i> (2007) · <i>RiP!: A Remix Manifesto</i> (2008) ·

TPB AFK: The Pirate Bay Away From Keyboard (2013) • *The Internet's Own Boy* (2014)

This article is issued from Wikipedia (<https://en.wikipedia.org/wiki/Copyleft?oldid=752328231>) - version of the 11/30/2016. The text is available under the Creative Commons Attribution/Share Alike (<http://creativecommons.org/licenses/by-sa/3.0/>) but additional terms may apply for the media files.