## Free software

“Free software” means software that respects users freedom and community. Roughly, the users have the freedom to run, copy, distribute, study, change and improve the software. With these freedoms, the users (both individually and collectively) control the program and what it does for them.

When users don't control the program, the program controls the users. The developer controls the program, and through it controls the users. This non--free or “proprietary” program is therefore an instrument of unjust power.

Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”.

A program is free software if the programs users have the four essential freedoms:
• The freedom to run the program, for any purpose ( freedom 0 ).
• The freedom to study how the program works, and change it so it does your computing as you wish ( freedom 1 ). Access to the source code is a precondition for this.
• The freedom to redistribute copies so you can help your neighbor ( freedom 2 ).
• The freedom to distribute copies of your modified versions to others ( freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

A program is free software if users have all of these freedoms. Thus, you should be free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission to do so.

You should also have the freedom to make modifications and use them privately in your own work or play, without even mentioning that they exist. If you do publish your changes, you should not be required to notify anyone in particular, or in any particular way.
The freedom to run the program means the freedom for any kind of person or organization to use it on any kind of computer system, for any kind of overall job and purpose, without being required to communicate about it with the developer or any other specific entity. In this freedom, it is the users purpose that matters, not the developers purpose; you as a user are free to run the program for your purposes, and if you distribute it to someone else, she is then free to run it for her purposes, but you are not entitled to impose your purposes on her.
The freedom to redistribute copies must include binary or executable forms of the program, as well as source code, for both modified and unmodified versions. (Distributing programs in runnable form is necessary for conveniently installable free operating systems.) It is OK if there is no way to produce a binary or executable form for a certain program (since some

languages dont support that feature), but you must have the freedom to redistribute such forms should you find or develop a way to make them.

In order for freedoms 1 and 3 (the freedom to make changes and the freedom to publish the changed versions) to be meaningful, you must have access to the source code of the program. Therefore, accessibility of source code is a necessary condition for free software. Obfuscated “source code” is not real source code and does not count as source code. Freedom 1 includes the freedom to use your changed version in place of the original. If the program is delivered in a product designed to run someone else modified versions but refuse to run yours — a practice known as “lockdown”, or as “secure boot” — freedom 1 becomes a theoretical fiction rather than a practical freedom. This is not sufficient. In other words, these binaries are not free software even if the source code they are compiled from is free. One important way to modify a program is by merging in available free subroutines and modules. If the programs license says that you cannot merge in a suitably licensed existing module — for instance, if it requires you to be the copyright holder of any code you add — then the license is too restrictive to qualify as free.

Freedom 3 includes the freedom to release your modified versions as free software. A free license may also permit other ways of releasing them; in other words, it does not have to be a copyleft license. However, a license that requires modified versions to be non--free does not qualify as a free license.

“Free software” does not mean “noncommercial”. A free program must be available for commercial use, commercial development, and commercial distribution. Commercial development of free software is no longer unusual; such free commercial software is very important. You may have paid money to get copies of free software, or you may have obtained copies at no charge. But regardless of how you got your copies, you always have the freedom to copy and change the software, even to sell copies.
E.g. The GNU operating system is the Unix--like operating system, which is entirely free software.

## Open source Software

Open--source software (OSS) is computer software that is available in source code form for which the source code and certain other rights. Open source software is software that has been released under an Open Source Initiative (OSI) certified license.
‘The term “open source” software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software: they accept some licenses that we consider too restrictive, and there are free software licenses they have not accepted. However, the differences in extension of the category are small: nearly all--free software is open source, and nearly all--open source software is free.’

— Free Software Foundation, http://www.gnu.org/philosophy/categories.html
Linux is “open source” software meaning, simply, that anyone can get copies of its source code files.

## Proprietary Software

Proprietary software is another name for non--free software. Proprietary software is software that is owned by an individual or a company (usually the one that developed it). There are almost always major restrictions on its use, and its source code is almost always kept secret because this is the company's competitive edge over others in the industry. Here, computer software is licensed under exclusive legal right of the copyright holder. This refers to any computer software that has restrictions on any combination of the usage, modification, copying or distributing modified versions of the software. Proprietary software is usually called closed--source software.

The Free Software Foundation follows the rule that we cannot install any proprietary program on our computers except temporarily for the specific purpose of writing a free replacement for that very program. Aside from that, FSF feel there is no possible excuse for installing a proprietary program.

## Other Software Models

Public domain software
Public domain software is software that is not copyrighted. If the source code is in the public, that is a special case of non--copylefted free software, which means that some copies or modified versions may not be free at all.
Copylefted software
Copylefted software is free software whose distribution terms ensure that all copies of all versions carry more or less the same distribution terms. This means, for instance, that copyleft licenses generally disallow others to add additional requirements to the software (though a limited set of safe added requirements can be allowed) and require making source code available.
Copyleft is a general concept; to copyleft an actual program, you need to use a specific set of distribution terms. There are many possible ways to write copyleft distribution terms, so in principle there can be many copyleft free software licenses. However, in actual practice nearly all copylefted software uses the GNU General Public License. Two different copyleft licenses are usually “incompatible”, which means it is illegal to merge the code using one license with the code using the other license; therefore, it is good for the community if people use a single copyleft license.
Non--copylefted free software
Non--copylefted free software comes from the author with permission to redistribute and modify, and also to add additional restrictions to it. If a program is free but not copylefted,

then some copies or modified versions may not be free at all. A software company can compile the program, with or without modifications, and distribute the executable file as a proprietary software product.

Freeware

The term “freeware” has no clear accepted definition, but it is commonly used for packages, which permit redistribution but not modification (and their source code is not available). These packages are not free software, so we can’t use “freeware” to refer to free software.

Shareware

Shareware is software, which comes with permission for people to redistribute copies, but says that anyone who continues to use a copy is required to pay a license fee. Shareware is not free software, or even semi free. There are two reasons it is not:

• For most shareware, source code is not available; thus, you cannot modify the program at all.

• Shareware does not come with permission to make a copy and install it without paying a license fee, not even for individuals engaging in nonprofit activity.

Private software

Private or custom software is software developed for one user (typically an organization or company). That user keeps it and uses it, and does not release it to the public either as source code or as binaries. A private program is free software if its sole user has the four freedoms. In particular, if the user has full rights to the private program, the program is free. However, if the user distributes copies to others and does not provide the four freedoms with those copies, those copies are not free software. In general we do not believe it is wrong to develop a program and not release it. There are occasions when a program is so important that one might argue that withholding it from the public is doing wrong to humanity. However, such cases are rare. Most programs are not that important, and declining to release them is not particularly wrong. Thus, there is no conflict between the development of private or custom software and the principles of the free software movement.

Commercial software

“Commercial” and “proprietary” are not the same! Commercial software is software developed by a business as part of its business. Most commercial software is proprietary, but there is commercial free software, and there is noncommercial non--free software.

For example, GNU Ada is developed by a company. It is always distributed under the terms of the GNU GPL, and every copy is free software; but its developers sell support contracts. When their salesmen speak to prospective customers, sometimes the customers say, “We would feel safer with a commercial compiler.” The salesmen reply, “GNU Ada is a commercial compiler; it happens to be free software.”

For the GNU Project, the priorities are in the other order: the important thing is that GNU Ada is free software; that it is commercial is just a detail. However, the additional development of GNU Ada that results from its being commercial is definitely beneficial.

## Open Standard

An open standard is a standard that is publicly available and has various rights to use associated with it, and may also have various properties of how it was designed (e.g. open process). There is no single definition and interpretations vary with usage.
The terms open and standard have a wide range of meanings associated with their usage. There are a number of definitions of open standards, which emphasize different aspects of openness, including of the resulting specification, the openness of the drafting process, and the ownership of rights in the standard. The term standard is sometimes restricted to technologies approved by formalized committees that are open to participation by all interested parties and operate on a consensus basis.
*Most definitions of the term standard permit patent holders to impose reasonable and non--discriminatory licensing.*
Principles
1. Availability
Open Standards are available for all to read and implement.
2. Maximize End--User
Choice Open Standards create a fair, competitive market for implementations of the standard. They do not lock the customer in to a particular vendor or group.
3. No Royalty
Open Standards are free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee.
4. No Discrimination
Open Standards and the organizations that administer them do not favor one implementer over another for any reason other than the technical standards compliance of a vendors implementation. Certification organizations must provide a path for low and zero--cost implementations to be validated, but may also provide enhanced certification services.
5. Extension or Subset
Implementations of Open Standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions.
6. Predatory Practices
Open Standards may employ license terms that protect against subversion of the standard by embrace--and--extend tactics. The licenses attached to the standard may require the publication of reference information for extensions, and a license for all others to create, distribute, and sell software that is compatible with the extensions.
Open Source Initiative's definition
The Requirement
An open standard must not prohibit conforming implementations in open source software.
The Criteria
To comply with the Open Standards Requirement, an open standard must satisfy the following criteria. If an open standard does not meet these criteria, it will be discriminating against open source developers.

1. No Intentional Secrets:
The standard MUST NOT withhold any detail necessary for interoperable implementation. The standard MUST define a process for fixing flaws identified during implementation and interoperability testing.
2. Availability: The standard MUST be freely and publicly available (e.g., from a stable web site) under royalty--free terms at reasonable and non--discriminatory cost.
3. Patents: All patents essential to implementation of the standard MUST: • Be licensed under royalty--free terms for unrestricted use, or • Be covered by a promise of non--assertion when practiced by open source software
4. No Agreements: There MUST NOT be any requirement for execution of a license agreement, Non Disclosure Agreements, grant, click--through, or any other form of paperwork to deploy conforming implementations of the standard.
5. No OSR--Incompatible Dependencies: Implementation of the standard MUST NOT require any other technology that fails to meet the criteria of this Requirement.
World Wide Web Consortium's definition
As an important provider of Web technology ICT Standards (HTTP, HTML, CSS etc.), the World Wide Web Consortium (W3C) follows a process that promotes the development of high quality web applications.
• Transparency (due process is public, and all technical discussions, meeting minutes, are archived and reference--able in decision making)
• Relevance (new standardization is started upon due analysis of the market needs, including requirements phase)
• Openness (anybody can participate, and everybody does: industry, individual, public, government bodies, academia, on a worldwide scale)
• Impartiality and consensus (guaranteed fairness by the process and the neutral hosting of the W3C organization, with equal weight for each participant)
• Availability (free access to the standard text, both during development, at final stage, and for translations, and assurance that core Web and Internet technologies can be implemented Royalty-- Free)
• Maintenance (ongoing process for testing, errata, revision, permanent access, validation, etc.)

## Open Content

What does open mean? The word has different meanings in different contexts. Our commonsense, every day experience teaches us that open is a continuous (not 1 or 0) construct. A door can be wide open, mostly open, cracked slightly open, or completely closed, so can your eyes :D .
The open in open content is a similarly continuous construct. In this context, open refers to granting of copyright permissions above and beyond those offered by standard copyright law. Open content, then, is content that is licensed in a manner that provides users with the right to make more kinds of uses than those normally permitted under the law -- at no cost to the user.

The primary permissions or usage rights open content is concerned with are expressed in the 4Rs Framework:
1. Reuse -- the right to reuse the content in its unaltered form (e.g. make a backup copy of the content)
2. Revise -- the right to adapt, adjust, modify, or alter the content itself (e.g. translate the content into another language)
3. Remix -- the right to combine the original or revised content with other content to create something new (e.g. incorporate the content)
4. Redistribute -- the right to share copies of the original content, your revisions, or your remixes with others (e.g. give a copy of the content to a friend)
*Content is open to the extent that its license allows users to engage in the 4R activities.*
Licenses
• Open Content License v1.0 and
• Open Publication License v1.0
Open Content and Education
Over the past decade, open content has been used to develop alternative routes towards higher education. Traditional universities are expensive, and their tuition rates are increasing. Open content allows for a free way of obtaining higher education that is focused on collective knowledge and the sharing and reuse of learning and scholarly content. There are multiple projects and organizations that promote learning through open content, including OpenCourseWare Initiative, The Saylor Foundation and Khan Academy. Some universities from USA, like MIT, Yale, and Tufts are making their courses freely available on the Internet.