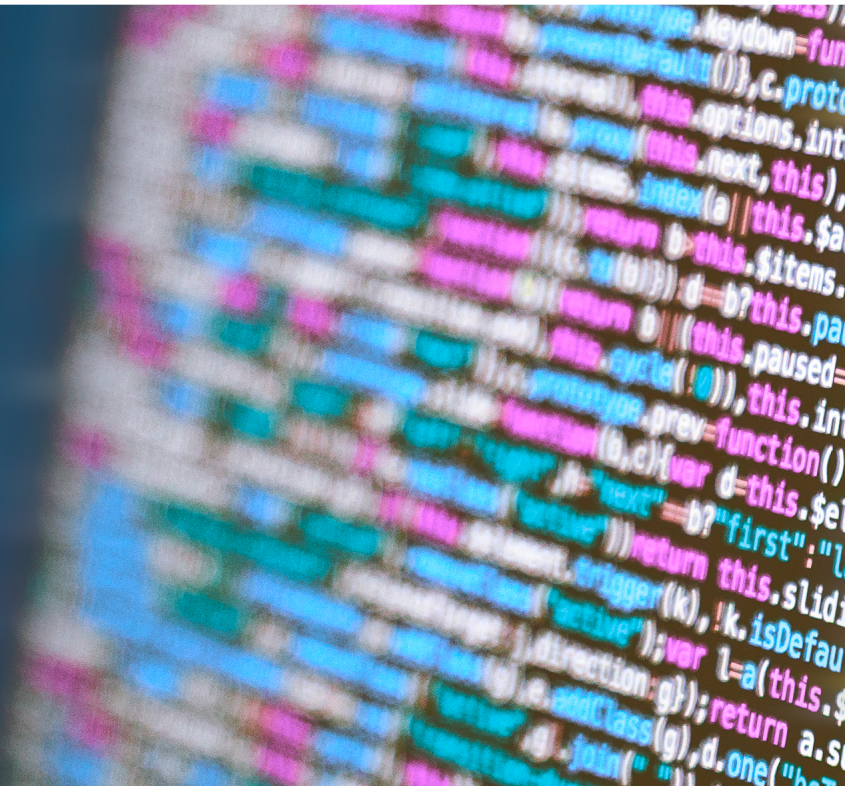


How to teach free software in your IT classroom



What Is Free Software?

Free software is free as in freedom, not as in price. It can be used, shared and improved by everyone.

Free software is everywhere. You'll find free programs in millions of schools, businesses and homes all over the world. It runs most servers on the internet and is even onboard the Perseverance rover that touched down on the Martian surface in 2021!

The movement for free software began in the 1980s with the inception of the GNU Project. GNU — short for "GNU's Not UNIX" — is an operating system written especially to give its users freedom. Today GNU is part of GNU/Linux, usually known simply as "Linux".

Why You Should Teach Free Software

Free software gives students the chance to learn from real, professional-quality programs. And there's a lot of them: billions of lines of free code are out there for them to explore.

But rights come with responsibilities. The majority of free programs are under software licences that use copyright law to guarantee users' freedom. It's important students understand how to obey free software licences both to protect others' rights and for when they are in work or publishing their own projects.

What Your Students Need to Know

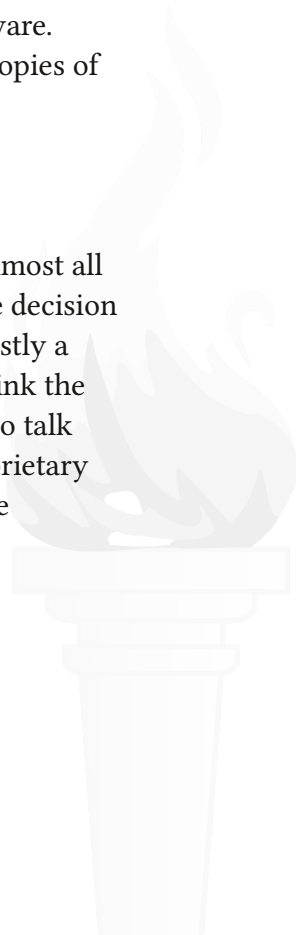
Free software vs. Open Source vs. Proprietary Software

Free software gives users *all* of these freedoms:

0. **Use** — Anyone can use free software for anything.
1. **Study** — Anyone can study and make changes to free software. Because of this, free software always has source code available.
2. **Share** — Anyone can share and copy free software.
3. **Improve** — People are free to share modified copies of free software.

Proprietary software is all software that isn't free.

Almost all free software counts as open source; almost all open source software counts as free software. The decision to talk about 'free software' or 'open source' is mostly a matter of politics: many free software activists think the four freedoms are absolutely essential; people who talk about open source are more likely to believe proprietary software is ethically acceptable and emphasise the practical benefits of software freedom.



Different Attitudes Towards Software Ownership

It's not necessary to go into what proprietary software developers believe about software copyrights. However, the free software movement's position might be new to you.

Briefly, the free software movement believes that decent societies allow people to help each other and to share. Software ought to be no exception.

The four freedoms are designed to make sure people have control over their computing. Free software activists point out that software has an immense power over people's lives. In this light, people's agency over their computing is more important than the interest software developers have in profiting from or controlling the use of their programs.

Links to more information about the arguments for the four freedoms are in the back of this booklet.

Copyleft

The vast majority of free programs are copyrighted.

Authors release their software under licences that explicitly grant users the four freedoms. This way of using copyright against itself is called copyleft, because it's almost like the opposite of copyright.

What the Licences Say

Whenever people share or build on free software they must:

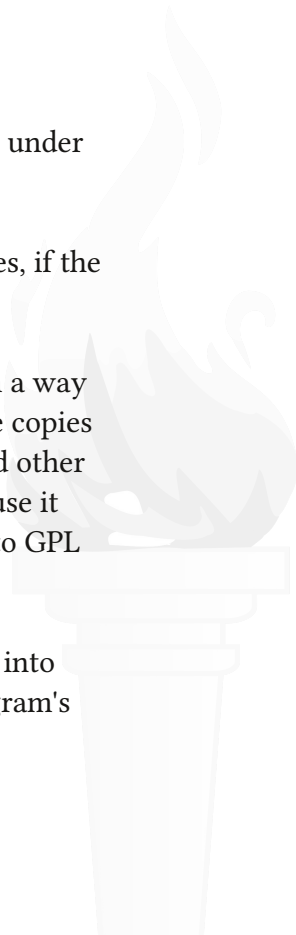
- Give credit to the authors, and
- Include a copy of the licence.

There are only a few cases where this isn't necessary.

When the program is under a "strong copyleft" licence, like most of the GNU GPL family, they must also:

- Make the source code available;
- Share the software and any changes made under the same licence;
- Don't remove copyright and licence notices, if the program displays them;
- Make sure the software isn't distributed in a way that prevents people from making changes to the copies they have. Therefore, you aren't allowed to upload other people's GPL'd code to the Apple App Store because it enforces "anti-tamper" restrictions. (This applies to GPL version 3 only.)

The details of licences vary. It's important to look into what you're allowed to do according to your program's particular licence.



Suggestions for Teaching

- Break the class into small groups and give each one one of the discussion questions on the next page. Ask them to make a short presentation to the class about their conclusions.
- Pick out a free software or open source program and look at their licences. Get your students to find out exactly what people can do with that piece of software and what they have to do in return.
- Ask your students to take up the roles of free software activists and proprietary software developers and hold a debate. More information about the reasoning for free software is on the More Information page later in this booklet.
- Free software isn't always non-commercial. Ask your students to come up with their own ideas to build a business around free software.
- Many people use free software without realising it. Ask your students to investigate what free software they use.

Discussion Questions

- Why might free software authors find it important people give them credit for their work even though they've freed it? What might happen if no one gives credit?
- Lawsuits against companies that violate free software licences are rare, but there can still be consequences. What might those consequences be?
- You're a developer for a company that's building a proprietary program. Your colleagues say that they found some GPLv3 code they could use, and no one would do anything if they ignored the licence. What is the right thing to do?
- Free software activists say proprietary software licences give software developers unjust power over users. Do you agree? Can you think of an example from your own life?



More Information

Free software

"Why Software Should Be Free"

<https://www.gnu.org/philosophy/shouldbefree.en.html>

"Let's Talk Freedom"

<https://letstalkfreedom.missionlibre.org/>

"What Is Free Software"

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

Open Source

"Open Source Definition"

<https://opensource.org/osd>

Licences

MIT License (unofficial site)

<https://mit-license.org/>

BSD 3-Clause License (unofficial site)

<https://opensource.org/license/BSD-3-clause>

GNU GPL v3.0

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

"FAQ About the GNU Licenses"

<https://www.gnu.org/licenses/gpl-faq.html>

This pamphlet is ©2026 Mission:Libre Limited.

You are encouraged to share this pamphlet under the terms of Creative Commons Attribution 4.0. <https://creativecommons.org/licenses/by/4.0/>

Mission:Libre is not responsible for the contents of external sites.

Comments on this pamphlet should be sent to carmen@missionlibre.org

The torch image is by "B1mbo" on Wikimedia Commons, and is CC-BY-SA 3.0; modified by Mission:Libre. <https://creativecommons.org/licenses/by-sa/3.0/deed.en>



```
function(b,d){this
loadingText:"loading
data();b="T
function(){
this.isLoad
manager.buttons");if
(c.prop
c.prop("che
button;a.fn.bu
click.is.button.data-a
preventDefault(
,c.DEFAULTS,
b?e.t
this.keydown,
this.sliding=this.int
this.pause,this)
this.prototype.keydown=fun
preventDefault()),c.prot
this.options.int
this.next,this),
index(a||this.$a
return b>this.$items
return b))d=b?this.pa
this.paused=
this.cycle(!0),this.in
this.prototype.prev=function()
function(b,c){var d=this.$e
"next"=b?"first":"l
return this.slidi
trigger(k,!k.isDefau
direction:g);var l=a(this.$
e.addClass(g);return a.s
d.join(" ")
```