

Open source software licences explained

Licences are about giving people extra rights. If you don't use any licence people can't legally make use of your code. So you need to use a licence to let them know what they can do with it. But which one should you use? There are two main classes of licences widely used for open source software:

- “BSD-style” licences (such as the MIT licence): very permissive and allow almost any reuse of the code;
- The GNU General Public License (GPL): allows reuse only in open source software also under the GPL.

First, review your code! Have you incorporated any other code that you might not have permission to distribute under any or all of these licences? Are you reusing code that you don't have permission to distribute at all? Make sure you know who owns the copyright in your work; whether your institution or funding body has any specific policies about publication and licensing; and whether you have permission to apply particular licences to your software.

Quick checklist...

Is this an implementation of an algorithm that is in a published paper?

Consider a BSD-style licence. If you want your code to be referred to by other researchers; it's not a complete application; and you don't intend to explicitly maintain or support it yourself, the BSD option makes for the simplest possible legal landscape. The idea is to eliminate obstacles to other people doing research in the future. Also: do not forget to state alongside the licence how people should cite your software if they use it for their own publications.

Does my code contain other software, e.g. software under GPL?

If your application contains code released under the GPL, you have two options: either apply the GPL to your own code or re-implement the original yourself. If the 3rd party code is released under a BSD-style licence, you are free to choose the license you prefer. Remember to always keep the original 3rd party code license and authorship information together with your own when releasing the final version of the software.

Will I be around and interested for 20 years to answer questions and maintain the software?

Chances are... not really! In this case you should consider a BSD-style licence. You'll probably not always be around to ask for permission to use your code, or the technical environment may change and make less permissive licence requirements unexpectedly hard to comply with. BSD terms are also the simplest for people who wish to cite or quote your work.

What if I wish to publish a self-contained library, and I intend to maintain it?

Consider using the LGPL (GNU Lesser Public License). The LGPL aims to strike a balance between the free-for-all of BSD and the tighter line associated with the GPL, and it can work well for libraries that will be actively maintained by their authors—learn more about the LGPL by reading the full article online.

Does my research funder or university have a policy about publishing code and data that supports published research?

If so, you should follow the existing policy (or at least discuss it with your supervisor). If you're uncertain whether such a policy exists, you might find the GPL provides more reassurance than a more permissive licence (for instance, the GPL offers some protection against certain sorts of commercial exploitation).

Do I want to allow other people to release applications derived from my software?

If you wish to allow others to release applications based on your code, you can go for either licence. Just remember that, for instance, the Apple App store doesn't permit using the GPL, so please take this into account if you would like to see others releasing your software via this distribution channel.

Do I want to submit it to SciPy or Matlab Central?

If you're writing code for an existing framework, or you wish to submit it to an external site, please check whether it has a licence policy already. For example, Python code contributed to SciPy should be under a BSD licence.

Remember, too, that the licence is something that the copyright holder provides and people who receive the code need to comply with. It doesn't bind the copyright holder. You can always provide your own code under more than one licence, or change the licence later on, although you can't retract any licensed code that has already been published.

Read the full article at: <http://soundsoftware.ac.uk/open-source-software-licences-explained>