

As for data, when working with code, good management practices are needed. The publication of code is crucial to understand, validate, reuse and repeat the research.

VERSIONING



Versioning systems are powerful code management tools.

The best-known is **Git**, it's free and open. It allows to:

- manage different versions of your code, and **track and undo changes** as needed;
- **automatically back up** code and its changes, being connected to a repository;
- collaboratively **work in a team** on the same code.

SHARING



To **share code and make it visible**, repositories provide various services like versioning systems, wikis, task management, and issues tracking. One of the most used is **Github**.

- EPFL provides c4science.ch for code versioning. Data are stored in Switzerland.
- EPFL provides gitlab.epfl.ch (GitHub open-source alternative). Data are stored at EPFL.

DESCRIBING



The README file ^[1] is fundamental for coding documentation. It allows you to **explain your code**, to yourself and others. On any publication of the code, you should add documentation and rich metadata (e.g., README.txt, license.md, parameter files, comments on code ^[2],...). Tools like [sphinx-doc.org](https://www.sphinx-doc.org) or doxygen.nl help by generating preformatted documentation.

LICENSING



As for data, it is important to explain **how your code can be used or cited** by others (considering related restrictions). Instead of Creative Commons licenses as for data, for code it is recommended to use software-specific licenses, such as:

- Open source licenses (permissive as [MIT](https://opensource.org/licenses/MIT) ^[3] or [GPL](https://opensource.org/licenses/GPL-3.0) ^[4]);
- Academic licenses (restrict commercial usage);
- Commercial licenses (reserve commercial usage);

Check the **Fast Guide #12: DATA & CODE LICENSING** ^[5]

PUBLISHING



Don't forget to **generate a DOI** to uniquely identify a version of your software and to make it easily citable. Most data repositories ^[6] automatically generate a DOI for your code release.

TIP: Github provides a quick and easy integration with Zenodo ^[7]

PRESERVING



Preservation is important for keeping your work secure and also for scientific validation.

- c4science is an EPFL solution to preserve your code **as a backup** solution.
- If you use another code repository, you can **always copy it on c4science**.
- For the **longer term**, a generic data repository (like Zenodo) is appropriate.

Credits and sources

[1] go.epfl.ch/rdm-readme
 [2] peps.python.org/pep-0008/#comments
 [3] opensource.org/licenses/MIT

[4] opensource.org/licenses/gpl-license
 [5] go.epfl.ch/rdm-fastguide12
 [6] go.epfl.ch/datarepo
 [7] docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content