SCIENCE OUVERTE

CODES ET LOGICIELS

PASSEPORT POUR LA
SCIENCE
OUVERTE

Légende
Le texte souligné renvoie au glossaire.
Le triangle ▼ signale des outils donnés à titre d' exemple.
Le symbole ↗ indique un lien externe.

La version numérique de ce guide est disponible sur www.ouvrirlascience.fr ↗

Inscrit dans la collection *Passeport pour la science ouverte*, ce livret aborde les enjeux particuliers de l'ouverture des codes et logiciels produits et utilisés dans le cadre de la recherche scientifique.

Quel est le statut juridique des codes et logiciels ? Pourquoi et comment les ouvrir ? Comment organiser leur développement et garantir leur archivage pérenne ? Comment envisager leur valorisation ? Vous trouverez dans ce guide des réponses et des outils directement activables.

Voici des clés pour comprendre le rôle du logiciel dans l'écosystème de la science ouverte. Nous espérons qu'il vous donnera l'envie de contribuer de manière durable aux travaux de la communauté.

le logiciel c'est quoi ?

Le logiciel dans tous ses états

Un **logiciel** (ou programme d'ordinateur) est la description, dans un ou plusieurs langages informatiques, d'un processus de traitement de données que l'on souhaite faire réaliser par un ordinateur.

Le **code source** est l'élément de base de tout logiciel. Il s'agit du texte écrit dans des langages informatiques par un ou plusieurs auteurs humains. Il existe de nombreux langages qui correspondent à des besoins précis : C, Java, Python, R, Ocaml, Scilab, etc.

Les ordinateurs sont uniquement capables d'exécuter des instructions élémentaires, écrites dans un langage machine de bas niveau. Pour exécuter un code source sur un ordinateur, il faut donc :

- soit le **compiler**, c'est-à-dire le traduire au préalable en un code objet fonctionnellement équivalent, écrit en langage machine exécutable sur l'ordinateur ;
- soit l'**interpréter**, c'est-à-dire le traduire à la volée, au sein d'un environnement dédié (interpréteur Python, etc.). Les codes sources destinés à être interprétés sont souvent appelés scripts. Les scripts de petite taille sont parfois appelés macro-instructions, ou macros.

BON À SAVOIR Tout le monde crée du code, parfois sans le savoir ! Même les formules de calcul d'un tableur sont du code source : ce sont de petits scripts, écrits dans le langage de macros de ce logiciel, et destinés à être interprétés à chaque fois que l'utilisateur modifie la valeur d'une cellule. Les représentations graphiques de calculs au sein de logiciels de laboratoires virtuels sont aussi du code, que l'on peut exprimer sous forme de texte.

On parle de logiciel lorsqu'un code a suffisamment d'utilité par lui-même pour qu'il soit considéré comme ayant une existence propre en termes de préservation, d'évolution et de diffusion. Dans ce cas, le logiciel peut exister en plusieurs versions, se succédant dans le temps ou coexistant en parallèle, au gré des adaptations faites par ses auteurs initiaux ou par d'autres contributeurs.

Certains logiciels sont utilisables de façon autonome ; d'autres, conçus pour rendre un service spécifique, sont intégrés en tant que briques logicielles au sein d'autres logiciels.

Enfin, un processus de calcul peut être décrit dans un notebook (ou cahier computationnel) qui combine des fragments de logiciels avec des informations descriptives (documentation ou explication).

Statut juridique

Le logiciel est considéré comme une **œuvre de l'esprit, protégée par le droit d'auteur**. Ce droit a deux facettes :

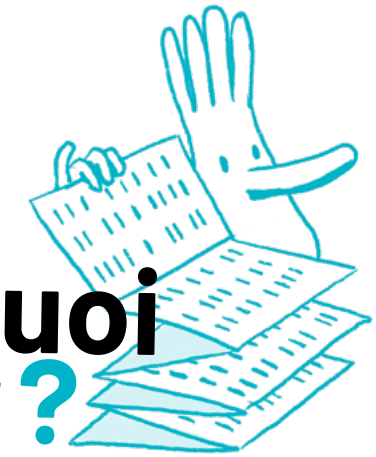
- le droit patrimonial, qui concerne l'exploitation de l'œuvre. Ce droit est exercé par les ayants droit, personnes physiques ou morales qui peuvent être différentes de l'auteur. Par exemple, il peut s'agir de l'employeur de l'auteur lorsque celui-ci est salarié ou agent public ;
- le droit extra-patrimonial (ou droit moral), qui concerne le respect de la paternité d'une œuvre. Même si l'œuvre est exploitée par des ayants droit, la paternité revient toujours aux auteurs, c'est-à-dire aux personnes physiques qui ont écrit le code source.

Lorsqu'un logiciel est mis en œuvre ou diffusé, il peut lui être adjoint une **licence**. Ce contrat d'adhésion définit les droits et devoirs des personnes qui reçoivent ou utilisent le logiciel. **Cette licence va définir le degré d'ouverture du code.**

Tout ce qui n'est pas explicitement autorisé par la licence est interdit. Si le code source d'un logiciel est diffusé sans licence, seule la consultation du code est autorisée.

Un logiciel est souvent accompagné d'une documentation, également couverte par le droit d'auteur. Cette documentation peut être :

- interne au code, sous forme de commentaires ;
- externe au code, sous forme de manuels d'utilisation ou de maintenance.



pourquoi ouvrir ?

Beaucoup de résultats scientifiques s'appuient sur des traitements informatisés de données. La reproductibilité de ces résultats nécessite de donner à la communauté scientifique les moyens de réaliser à nouveau les expérimentations s'appuyant sur des logiciels. Or il est presque impossible de garantir qu'un logiciel s'exécute à l'identique, pour tous les utilisateurs, sur le long terme. Ouvrir les logiciels est un moyen de remédier à cette difficulté. Cette démarche doit être combinée avec l'ouverture des données de recherche. Consultez le ▼**MOOC Recherche reproductible** pour aller plus loin.

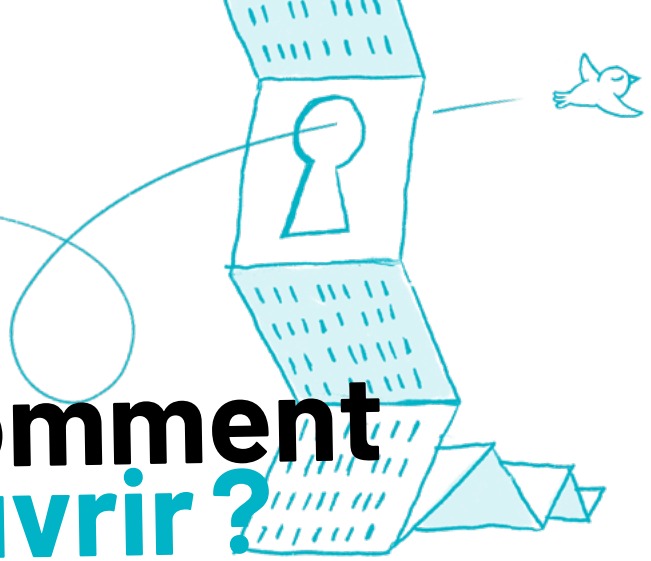
Différentes initiatives facilitent aujourd'hui l'intégration des logiciels aux publications, comme ▼**IPOL**, en traitement d'images, ▼**eLife**, en sciences de la vie, ou encore la ▼**Graphics Replicability Stamp Initiative**.

Parce qu'elle facilite la comparaison des résultats, **l'ouverture d'un logiciel favorise sa citation par les pairs**. Elle encourage les collaborations autour du patrimoine logiciel créé, en élargissant la communauté d'utilisateurs ou de contributeurs. Ces communautés peuvent être fédérées grâce à des plateformes collaboratives, telles que l'initiative ▼**ropensci**, pour le langage R.

L'ouverture des codes et logiciels contribue à leur **amélioration** et à l'**émergence de nouvelles idées**. Elle stimule la recherche scientifique dans une logique de coopétition (coopération et compétition).

L'accès au logiciel permet de **valoriser la compétence** des auteurs et de leurs équipes et de leur apporter une notoriété supplémentaire.

Enfin l'ouverture des logiciels fait partie intégrante des politiques nationales et européennes de science ouverte, depuis la **loi pour une République numérique** (2016) et le **deuxième Plan national pour la science ouverte** (2021).



comment ouvrir ?

Si vous souhaitez partager votre logiciel et le rendre utilisable par d'autres, vous devez en organiser l'accès sur les plans juridique et technique. Il vous faut également réfléchir au devenir du code source du logiciel sur le long terme : maintien d'une activité de recherche à son sujet ou mise en place d'une démarche active de valorisation.

Choisir une ou plusieurs licences

Le choix de la licence est une étape importante de l'ouverture, qui peut avoir des conséquences importantes sur les évolutions du logiciel.

La première étape consiste à identifier qui détient les droits patrimoniaux sur le logiciel : lorsque les auteurs ne sont pas les ayants droit, le choix de la licence ne leur revient pas.

La personne, groupe ou entité à qui revient la responsabilité de choisir la licence peut s'engager dans une démarche d'ouverture complète, en utilisant une licence de logiciel libre. Le département ▼**Etalab** de la ▼**DINUM** fournit une ▼**liste de licences utilisables pour les productions de l'administration publique française**. Dans les autres cas, on peut se référer aux listes maintenues par l' ▼**Open Source Initiative** ou la ▼**Free Software Foundation**.
Dans certaines situations, il est possible d'utiliser des licences plus restrictives, qui n'autorisent que la consultation du code ou restreignent l'utilisation par des tiers. Par exemple, on peut limiter l'usage à un cadre académique pour la seule reproductibilité scientifique.

Si le logiciel utilise des briques préexistantes, il faut s'assurer :

- de la compatibilité juridique de la licence choisie avec celles des différentes briques préexistantes ;
- de la compatibilité des licences des différentes briques entre elles.

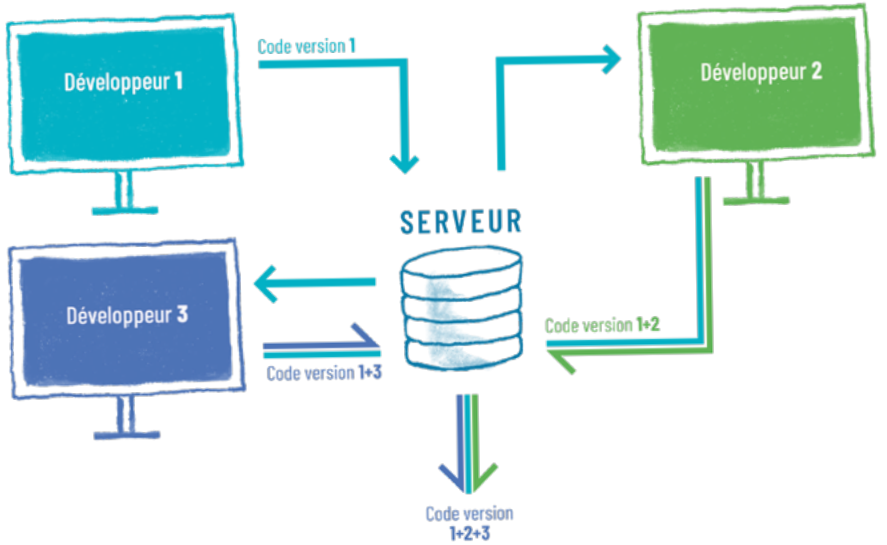
EXEMPLES PRATIQUES Il existe différents types de licences, chacune adaptée à des pratiques de réutilisation différentes. **Scikit-learn** est une bibliothèque d'apprentissage statistique qui intègre l'état de l'art du domaine et le rend accessible au plus grand nombre. Conçue pour être intégrée dans d'autres logiciels, cette bibliothèque est distribuée sous une licence permissive, la BSD. **WebObs** est un outil pluridisciplinaire d'observation en temps réel utilisé dans le cadre de l'étude de phénomènes naturels. Adopté par plusieurs pays pour surveiller leurs volcans (Indonésie, Singapour, Pérou), ce logiciel est distribué sous une licence diffusive, la GPL 3.0. **Scikit-learn** et **WebObs** sont deux exemples des logiciels lauréats des prix Science ouverte du logiciel de la recherche (2022).

Organiser le développement du logiciel

Maintenir, organiser et développer un logiciel d'une certaine taille nécessite la mise en œuvre de bonnes pratiques. Les évolutions du code source doivent être tracées avec un système de contrôle de version : ▼**Git**, ▼**Mercurial** ou ▼**Subversion**.

De nombreuses plateformes web, appelées forges logicielles (▼**bitbucket.org**, ▼**gitlab**, ▼**github**, etc.), simplifient ces tâches et facilitent l'agrégation de communautés de contributeurs. Des instances de ces plateformes peuvent être localement mises en place par l'établissement d'appartenance ou par une infrastructure de recherche, comme ▼**HumaNum** pour les sciences humaines et sociales.

Les communautés de chaque plateforme peuvent apporter des bénéfices appréciables : recherche de bogues, ajout de nouvelles fonctionnalités, pérennisation du logiciel, gestion des contributeurs et de leurs permissions, mise en place de tests de qualité, etc. Pour bénéficier de l'apport d'une communauté, il faut investir le temps et l'énergie nécessaires pour organiser son fonctionnement : évaluer et gérer rapidement les suggestions et contributions apportées, planifier la feuille de route du développement logiciel…



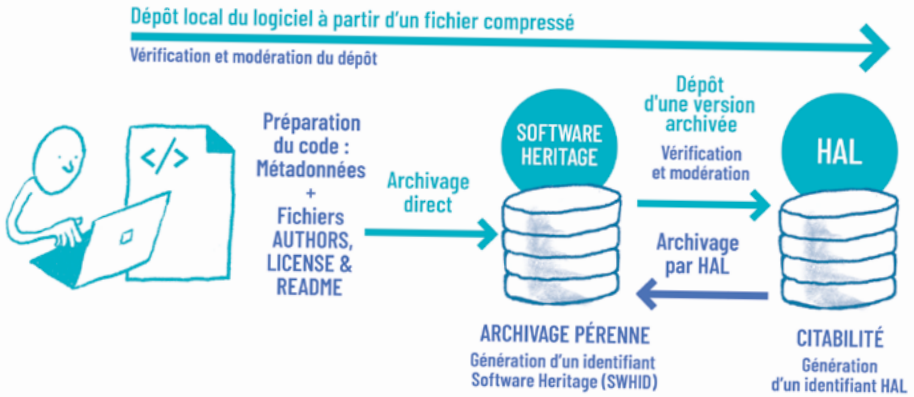
BON À SAVOIR **Les forges logicielles ne sont pas des archives pérennes : les projets qu'elles contiennent peuvent être modifiés ou effacés.** Des forges entières peuvent aussi disparaître, comme ce fut le cas de Google Code et Gitorious en 2015 et d'une partie de BitBucket en 2020. Il est donc préférable de choisir une forge gérée par une organisation publique et recommandée par votre institution et de vous assurer de l'archivage de vos codes sur le long terme dans des archives dédiées aux logiciels, comme ▼**Software Heritage**.

et après ? préparer l'avenir

Archiver, référencer et décrire les logiciels

Comme pour les publications et les données, il convient d'assurer la préservation à long terme d'un logiciel de recherche en l'archivant dans une plateforme adaptée. Les versions utilisées doivent être référencées avec précision, faute de quoi il sera difficile de reproduire les résultats obtenus. Il faut aussi prendre le temps de décrire le logiciel, en utilisant des fichiers README et AUTHORS et en ajoutant des métadonnées qui facilitent l'indexation.

La plateforme Software Heritage permet de s'assurer très simplement de l'archivage et du référencement des logiciels disponibles publiquement et sans embargo. Le dépôt peut être fait aussi directement sur ▼**HAL**. L'archive ouverte HAL permet, en collaboration avec ▼**Software Heritage**, de gérer la description et la citation des logiciels de recherche.



Valoriser le logiciel

Ouvrir ne suffit pas. Pour que votre code ou que votre logiciel soit utilisé, quelques actions de communication s'imposent : des annonces sur des listes de diffusion, des réseaux sociaux, des relais sur des sites institutionnels. Vous devez expliquer synthétiquement pourquoi vous avez créé cet outil. Pour les spécifications plus techniques, renvoyez vers la documentation que vous aurez conçue. Après la période de lancement, la communication s'organise sur la durée : **vous organisez une formation à votre logiciel, une démo ? Vous avez une publication en lien avec votre logiciel ? Une montée de version ? Faites-le savoir.**

Lorsqu'un logiciel devient stratégique pour des membres de sa communauté, celle-ci peut décider de s'organiser de façon plus formelle afin de garantir la pérennité du logiciel et d'assurer son développement. Un travail de valorisation peut conduire à la création d'un club de partenaires (ou consortium), financé par ses membres, ou bien d'une entreprise dédiée.

Selon la politique définie collectivement par les ayants droit, le même logiciel peut être mis à disposition sous des licences différentes, auprès de publics différents. On peut ainsi mettre une version communautaire à la libre disposition de la communauté académique, tandis qu'une version commerciale sera proposée à des partenaires industriels, éventuellement sous une autre licence plus adaptée à leurs besoins, dans le cadre de contrats rémunérés. Afin de faciliter la gestion des droits d'un logiciel, un mandat de valorisation peut être donné à un ayant droit par les autres, afin de négocier en leur nom. Il est aussi possible d'organiser au fil de l'eau la cession de la titularité des droits des contributeurs à un ayant droit unique, comme une fondation dédiée, plus à même de représenter leurs intérêts.

SUR LE TERRAIN

Arnadi Murtiyoso, jeune docteur de l'Université de Strasbourg en photogrammétrie et géomatique, actuellement post-doctorant à l'ETH de Zürich

Dans le cadre de ma thèse, nous avons développé ARcH, un benchmark de documentation patrimoniale 3D. Nous avons utilisé une forge logicielle et placé notre code sous une licence GPL 3.0, ce qui permet à d'autres chercheurs de réutiliser notre code pour développer un benchmark dans leur domaine d'étude. L'ouverture des logiciels de recherche est essentielle pour la compréhension et l'analyse des résultats, ainsi que pour la maîtrise des processus.

Citer les logiciels

Lorsqu'on rédige un article scientifique, il est souhaitable de créditer les auteurs des logiciels utilisés en ajoutant une citation appropriée. Des styles bibliographiques dédiés aux logiciels existent désormais et sont en cours d'adoption. Vous pouvez aussi signaler les codes et les logiciels que vous produisez dans votre compte ▼**ORCID**.

Le paquetage ▼**biblatex-software** disponible sur ▼**CTAN** sait gérer les entrées bibliographiques pour le logiciel. Celles-ci peuvent aussi être importées dans ▼ **Zotero** grâce à l'extension ▼**BetterBibLatex**.

Pour aller plus loin

ARCHIVAGE, RÉFÉRENCEMENT, DESCRIPTION ET CITATION

Guide pour archiver et référencer des codes sources dans ▼Software Heritage.
https://www.softwareheritage.org/howto-archive-and-reference-your-code/
Guide pour le dépôt dans ▼HAL des logiciels avec des fiches de métadonnées.
https://doc.archives-ouvertes.fr/deposer/deposer-le-code-source/
Présentation synthétique de la citation des logiciels.
https://www.softwareheritage.org/2020/05/26/citing-software-with-style/?lang=fr

ASPECTS JURIDIQUES

▼Liste des licences utilisables pour les productions de l'administration publique française.
https://www.data.gouv.fr/fr/pages/legal/licences
Liste de licences commentées du Système exploitation GNU (soutenu par la ▼*Free Software Fondation*). https://www.gnu.org/licenses/license-list.html
Texte de l'Organisation mondiale de la proprieté intellectuelle sur la protection des logiciels informatiques par le droit d'auteur.
https://www.wipo.int/copyright/fr/activities/software.html
P. Moreau, C. Moulin, J. Pappalardo, F. Pellegrini (2017). Fondamentaux juridiques. Collaboration et innovation ouverte, 2ème édition, *Les livrets bleus du logiciel libre*. https://cnll.fr/media/LivretBleu_Juridique-2eEdition_GT-LogicielLibre_Systematic_Nov2016_web.pdf

RÉSEAUX AUTOUR DU LOGICIEL DE RECHERCHE

Réseau français sur la recherche reproductible. https://www.recherche-reproductible.fr/
Réseau des acteurs du Développement LOGiciel au sein de l'Enseignement Supérieur et de la Recherche. https://www.devlog.cnrs.fr/

INFORMATIQUE (GÉNÉRIQUE)

V. Doutaut (2021). Informatique et culture scientifique du numérique, une transcription des MOOC réalisés par le Learning Lab Inria. https://hal.inria.fr/hal-03346079

MÉTADONNÉES

Des plateformes pour vérifier le code et générer des métadonnées :
▼CodeMeta Projet (https://codemeta.github.io/) et
▼CodeMeta generator (https://codemeta.github.io/codemeta-generator/)

CONTRIBUTIONS

▼CRediT, une description des différents rôles de contributeurs dans une production scientifique. https://credit.niso.org/

SYSTÈMES DE CONTRÔLE DE VERSION

Git, un portail avec de ressources autour de Git. https://git-scm.com/
▼Learn Git Branching, une application pour vous aider à saisir les concepts derrière les branches en travaillant avec Git. https://learngitbranching.js.org/?locale=fr_FR

VÉRIFICATION ET PUBLICATION DU CODE

▼CODECHECK, une plateforme informatique pour la vérification des logiciels sous-jacents aux articles scientifiques. https://codecheck.org.uk
Conseils pour la publication du logiciel. Dépôt GitHub, *Tips for Publishing Research Code*. https://github.com/paperswithcode/releasing-research-code/

Sources

P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M. S. Hacid, A. Legrand, N. Rougier (2019). Attributing and Referencing (Research) Software: Best practices and outlook from Inria. *Computing in Science and Engineering, Institute of Electrical and Electronics Engineers*, pp.1-14. https://hal.archives-ouvertes.fr/hal-02135891v2

Article L. 113-9 du code de la propriété intellectuelle. https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000039279818/

L. A. Barba. Terminologies for reproducible research. arXiv:1802.03311 [cs.DL]. https://doi.org/10.48550/arXiv.1802.03311

Décret n° 2017-638 du 27 avril 2017 relatif aux licences de réutilisation à titre gratuit des informations publiques et aux modalités de leur homologation. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000034502557

G. Colavizza, I. Hrynaszkiewicz, I. Staden, K. Whitaker, B. McGillivray (2020). The citation advantage of linking publications to research data. PLOS ONE 15(4): e0230416. https://doi.org/10.1371/journal.pone.0230416

L. Courtès (2021). Reproduire les environnements logiciels: un maillon incontournable de la recherche reproductible. Bulletin de la Société informatique de France, n° 18, pp. 15-22. https://dx.doi.org/10.48556/SIF.1024.18.15

R. Di Cosmo. biblatex-software – BibLATEX stylefiles for software products. https://ctan.org/pkg/biblatex-software

L. Desquilbet, S. Granger, B.Hejblum, *et al.* Vers une recherche reproductible : faire évoluer ses pratiques. https://hal.archives-ouvertes.fr/hal-02144142. Disponible en version collaborative sur https://rr-france.github.io/bookrr/

M. Gruenpeter, J. Sadowska, E. Nivault, A. Monteil (2022). Create software deposit in HAL: User guide and best practices. [Technical Report] Inria; CCSD; Software Heritage. https://hal.inria.fr/hal-01872189

J. Kitzes, D. Turek, et F. Deniz (2018). The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences. Oakland: University of California Press. www.practicereproducibleresearch.org/

J.-F. Pimentel, L. Murta, V. Braganholo, et J. Freire (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks, *Empirical Software Engeneering*, 26, 65. https://doi.org/10.1007/s10664-021-09961-9

H. E. Plesser (2018). Reproducibility vs. Replicability: A Brief History of a Confused Terminology, *Frontiers in Neuroinformatics*,11:76. https://doi.org/10.3389/fninf.2017.00076

K. Powell (2020). Tech tools to make research more open and inclusive. *Nature*, 578, 181–182. https://doi.org/10.1038/d41586-020-00216-z

Prix Science ouverte du logiciel libre de la recherche. https://www.enseignementsup-recherche.gouv.fr/fr/remise-des-prix-science-ouverte-du-logiciel-libre-de-la-recherche-83576

G.K.Sandve, A. Nekrutenko, J. Taylor, E. Hovig (2013). Ten Simple Rules for Reproducible Computational Research. *PLOS Computational Biology* 9(10): e1003285. https://doi.org/10.1371/journal.pcbi.1003285

Utrecht University (n.d.). Best Practices for Writing Reproducible Code. Workshop-Computational-Reproducibility. https://utrechtuniversity.github.io/workshop-computational-reproducibility/

Glossaire

Archiver : organiser de façon pérenne la sauvegarde, l'accès et l'indexation d'une version donnée d'un logiciel. La mise à disposition à travers une page web professionnelle ou un site de laboratoire n'étant pas suffisante pour atteindre ces objectifs, le recours à des plateformes dédiées est à privilégier.

Bibliothèque (ou Brique logicielle ou Module) : logiciel fournissant un service particulier et destiné à être incorporé au sein d'un autre logiciel.

Code exécutable : code écrit en langage machine, directement exécutable par un ordinateur d'un type particulier. Il s'agit généralement d'un code objet.

Code objet : code qui n'a pas été directement écrit par un auteur humain. Il est le résultat de la traduction d'un code source au moyen d'un outil tel qu'un compilateur.

Code source : code écrit par un auteur humain.

Compiler : traduire un code source en code objet, en général pour obtenir un code exécutable sur un ordinateur d'un type particulier. Le compilateur est le logiciel permettant d'effectuer cette traduction de façon automatique.

Club de partenaires / Consortium : coalition d'acteurs ayant un intérêt stratégique commun à promouvoir le développement et/ou la maintenance d'un logiciel ou d'un standard donné.

Communauté : ensemble de personnes utilisatrices ou susceptibles de contribuer à un projet logiciel.

Compatibilité juridique : capacité de deux licences couvrant deux modules distincts à coexister au sein du même logiciel. Par exemple, on ne peut pas incorporer dans le même logiciel un module sous licence libre GNU GPL et un module sous licence non libre sauf exception.

Contributeur : personne participant au développement d'un projet logiciel, que ce soit par la production de code source ou par d'autres activités : test, traduction de l'interface ou de la documentation, animation de la communauté, etc.

Droit d'auteur : régime juridique régissant l'exploitation des œuvres de l'esprit, auxquelles appartiennent notamment les logiciels, leur documentation, les icônes et sons de l'interface graphique.

Forge : environnement de développement logiciel facilitant le travail collaboratif autour d'un projet logiciel. Une forge contient des outils tel que le dépôt versionné de code source, des forums de discussion, un environnement de tests automatisés, etc.

Interface : ensemble de conventions permettant d'utiliser un module logiciel.

Langage informatique : langage spécifiquement conçu pour décrire des processus calculatoires susceptibles d'être effectués par un ordinateur.

Langage machine : langage rudimentaire de bas niveau permettant d'écrire du code directement exécutable par un ordinateur d'un type donné. Les programmeurs préfèrent écrire leurs codes sources dans des langages de plus haut niveau, plus expressifs. Les codes sources doivent alors être traduits en code objet exécutable sur cet ordinateur, notamment par compilation.

Licence : document contractuel par lequel l'ayant droit d'une œuvre accorde un certain nombre de permissions à l'utilisateur. Une licence est dite « libre » si elle offre simultanément les quatre libertés suivantes : d'usage, de copie, de modification et de redistribution du logiciel modifié.

Logiciel : texte, écrit dans un ou plusieurs langages informatiques, décrivant des calculs destinés à être exécutés par un ordinateur.

Macro : script de petite taille.

Mandat (de valorisation) : accord entre les co-ayants droit afin de désigner celui d'entre eux qui entreprendra les démarches administratives au nom de tous.

Métadonnées : données associées à une ou plusieurs données afin de les contextualiser (horodatage et géolocalisation d'une photo, par exemple).

Notebook, Calepin ou Cahier computationnel : document pouvant contenir du code exécutable, du texte, des formules mathématiques, des graphiques et des médias interactifs (par exemple, un ▼**Jupyter Notebook**).

Œuvre de l'esprit : création de forme protégeable au titre du droit d'auteur du fait qu'elle exprime la personnalité de son auteur (critère dit « d'originalité » au sens du droit d'auteur).

Référencer : identifier sans équivoque des objets. Ceux-ci peuvent avoir différents niveaux de granularité : un logiciel complet dans une version donnée, un fichier, un bloc de lignes de code. Le référencement nécessite l'attribution d'identifiants uniques et pérennes comme le SoftWare Heritage persistent IDentifier (SWHID).

Reproductibilité : capacité à reproduire les résultats d'une étude en répliquant le processus original.

Script : fragment de code qui sera interprété au sein d'un environnement de travail tel qu'un logiciel de bureautique, un interpréteur de commandes, etc.

Version : exemplaire d'un logiciel disponible à une date donnée, mis à disposition d'un public donné. À une version donnée d'un logiciel peuvent être associés une licence ou des services spécifiques (assistance à l'utilisation, correction de bogues, etc.).

Crédits

Direction de la publication
Ministère de l'Enseignement supérieur et de la Recherche

Coordination éditoriale
Université de Lille

Conseil scientifique
Collège Compétences et formation et collège Codes sources et logiciels du Comité pour la science ouverte

Cheffe de projet
Mónica Michel Rodríguez

Rédacteurs
François Pellegrini, Roberto Di Cosmo, Laurent Romary, Sabrina Granger, Sacha Hodencq, Joanna Janik, Romane Coutanson, Madeleine Géroudet

Design graphique
Studio 4 minutes 34
Studio Lendroit.com

Impression
L'Artésienne, Liévin

1ère édition: août 2022
Achevé d'imprimer :
septembre 2022 à 3000 exemplaires

Remerciements

Arnadi Murtiyoso, qui a partagé son expérience en matière de science ouverte

Les jeunes chercheurs qui ont participé aux échanges sur la première version
Julien Cicero, Erwin Gerard, Aldo Gonzalez Lorenzo, Loïck Kléparski, Cédric Marinel, Jean-Paul Martischang, Martin Mion-Mouton, Antoine Olczak

Experts consultés
Isabelle Blanc, Danielle Bourcier, Franck Macrez, Sofia Papastamkou

Ce guide appartient à la collection *Passeport pour la science ouverte*.

La version numérique de ce guide est disponible sur www.ouvrirlascience.fr

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
Liberté
Égalité
Fraternité

Ouvrir la science !

Université de Lille