

## Archiver et rendre visible son code source

Améliorer la reproductibilité des expériences ne peut se faire sans l'accès aux données. Mais l'accès aux codes sources et logiciels qui ont servi à traiter ces données est également indispensable. Voici comment on peut garantir cette mise à disposition

### Pourquoi préserver et rendre accessibles les logiciels et codes sources de la recherche ?

Les logiciels et codes sources (scripts) constituent une partie intégrante du patrimoine de la recherche. Ils permettent de reproduire des analyses, de partager des méthodes et de garantir la transparence des résultats rendant possible la reproductibilité de ceux-ci. Il est crucial de les archiver et de les diffuser pour :

- garantir la reproductibilité scientifique
- pérenniser et transmettre les connaissances
- valoriser et reconnaître le travail des chercheurs
- protéger juridiquement et assurer la traçabilité

### Comment rendre visible son code ?

Il y a plusieurs manières de faciliter la réutilisation de son code.

Le premier pas consiste à **rendre ce travail visible depuis la forge où il a été développé**, ce qui peut être fait dès le début du projet.

La forge (Github, Gitlab, Codeberg, etc.) n'est parfois conçue que comme un outil de développement permettant le versionnage et facilitant le travail collaboratif, mais cela peut aussi être utilisé comme une vitrine, une manière de rendre son code visible et trouvable pour faciliter le travail d'autres équipes de recherche.

En plus du code source lui-même et de ses différentes versions, la forge donne accès aux problèmes et demandes de nouvelles fonctionnalités qui ont été postées par des internautes (issues) et aux propositions d'ajouts (*pull requests*), ce qui permet de juger de l'adaptabilité du code source à tel ou tel projet de recherche. Les chercheuses et chercheurs en sciences humaines et sociales peuvent utiliser la [forge d'Humanum](#).

N.B. : Une forge est idéale pour partager du code source, mais ce n'est pas une archive. Des forges sont régulièrement fermées parce que les institutions ou associations qui les soutiennent n'ont plus les moyens de les maintenir.

La seconde étape vise à **lier la thèse ou l'article publié avec le code source** réalisé à la faveur du projet de recherche.

Sur le site de l'éditeur, cette relation peut se faire de différentes manières :

- au moyen d'un paragraphe dédié, un **Software Availability Statement** (à l'instar des *Data Availability Statement*)
- sous la forme d'une **référence en bibliographie**
- lors d'un **passage spécifique dans la méthode** de l'article

Dans tous les cas, ces espaces pointent vers un espace où le code source est décrit et/ou est accessible, par exemple : un plan de gestion de logiciel sous **Zenodo**, une description du code source sur **HAL** (voir plus bas) ou bien le dépôt dans une **forge**.

### Comment conserver son code ? Archiver le code source dans Software Heritage et le signaler dans HAL

[Software Heritage](#) est une archive qui a vocation à archiver de façon pérenne le code source qu'on y dépose. Cette archive moissonne les dépôts publics des forges et il est possible à tout moment d'y envoyer le code source qu'on est en train de développer depuis la forge, soit en entrant l'adresse git (ou équivalent) du dépôt (repository) [depuis le site de Software Heritage](#), soit en utilisant une extension de son navigateur et en cliquant sur un simple bouton.

Un identifiant unique et pérenne (le "Software Heritage ID", ou SWHID) sera automatiquement attribué à la version du dépôt envoyé dans l'archive. Cet identifiant peut être affiné pour pointer vers un seul fichier du dépôt, voire seulement quelques lignes d'un fichier.

Le SWHID va permettre de relier la publication dans HAL au dépôt du code source dans l'archive. En effet, HAL prévoit un lien vers l'archive Software Heritage. Par ailleurs, le SWHID facilite la citation du code source dans une bibliographique.

#### Contact

Pour toute question sur les codes sources et logiciels de la recherche

guichet-ardoise@groupes.renater.fr

Aller plus loin

- [Software Heritage](#)
- [Catalogue des logiciels libres de la recherche académique](#)
- [Gitlab Huma-Num](#)
- [Dessin du mécanisme d'une montre chronographe. Préserver et rendre identifiabl...](#)