Checklist

Contents

Make Use of Git

- Make your project version controlled by initialising a Git repository in its directory using `git init`.
- Add and commit all your files to the repository using `git add .` then `git commit`.
- Continue to add and commit changes as your project progresses. Stage the changes in specific files to be committed with `git add filename`, and add messages to your commits.
    - Each commit should make one simple change.
    - No generated files are committed.
    - Commit messages are meaningful, with a ~50 character summary at the top.
    - Commit messages are in the present tense and imperative.
- Develop new features on their own branches, which you can create via `git checkout -b branch_name` and switch between via `git checkout branch_name`.
    - Make sure branches have informative names.
    - Make sure the main branch is kept clean.
    - Make sure each branch has a single purpose and only changes related to that purpose are made on it.
- Once features are complete, merge their branches into the main branch by switching to the feature branch and running `git merge main`.
    - Merge other's changes into your work frequently.
    - When dealing with merge conflicts, make sure you fully understand both versions before trying to resolve them.

Print to PDF

Contribute to Someone Else's Project

- Clone their project's repository from GitHub `git clone repository_url`.
- Make and commit changes.
- Push your changes to you GitHub version of the project.
- Make use of issues to discuss possible changes to a project.
- Make pull requests on GitHub to share your work.
  - Clearly explain the changes you have made (and why) in your pull request.

Make Sure That Your Data Is Version-Controlled

- If your projects involve data, check whether [Git LFS](#), [git-annex](#), or [DataLad](#) fits your needs for version-controlling it.
- Share the data along with your project to help others reproduce your results.