Formation-GIT

Mélodie Angeletti, Éric Mermet

16th December 2024

What is GIT

- Distributed version control system that tracks versions of files.
- Free and open-source software shared under the GPL-2.0-only license

How it works

- Local copy of the entire repository, a.k.a. repo, with history and version-tracking abilities, independent of network access or a central server, viewed as a folder
- Feature to synchronize changes between repos that share history; copied (cloned) from each other, even remotely.

Create/Clone a repository

git init	create a new repository
git clone http	clone locally a remote repository

Add files

git add	followed by a file name, or regular expression (for instance *.py): add file(s) contents to the index
git commit	Record changes to the repository. Can be followed by a message to describe the change
git revert	: Revert some existing commits
git log	: Shows the commit logs.

Save change

git status	Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git
git pull	Incorporates changes from a remote repository into the current branch.
git push	Updates remote refs using local refs, while sending objects necessary to complete the given refs.

Conflict

A pull can be refused if locally you're not up-to-date. You need to do a push before.
A conflict can happen if you modify a file that has been modified in the depot
Git doesn't choose which file to keep
The file where the conflict happens has been modified by Git:

- <<<<<<< and >>>>>>> delimit the conflicts
- First part following **HEAD** is your local modification
- Second part following ======= show the remote modification

The conflict happens only in your local repository: your collaborator doesn't see it

Solve conflict

Three solutions:

Save the local change : use **git checkout –ours** followed by the file

Save the remote change : use **git checkout –theirs** followed by the file

Edit a new version merging the two versions : modify your file

Then use **git add**, **git commit** and **git push**

Branch and forks

Fork	new repository that shares code and visibility settings with the original “upstream” repository.
Branch	Branch allow you to develop features, fix bugs, or safely experiment with new ideas in a contained area of your repository.

Working with Branch

Each repository has one default branch, and can have multiple other branches

git branch newBranch	create a branch named newBranch from the current branch
git branch -d myBranch	delete branch myBranch once its is fully merged
	git switch myBranch swithc to branch myBranch

When you switch branches, Git resets your working directory to look like it did the last time you committed on that branch. It adds, removes, and modifies files automatically to make sure your working copy is what the branch looked like on your last commit to it.

Pull/Merge request

A merge/pull request is simply a request from a user to merge their code from one branch to another, typically to the master branch.

Merge request is used in Gitlab and Pull request is used in Github.

The command is **git merge myBranch**

Git and RStudio

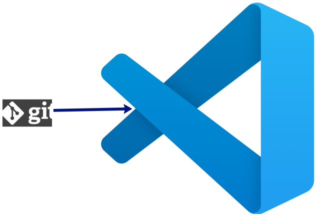
To import Git project in Rstudio: see `https://thinkr.fr/travailler-avec-git-via-rstudio-et-versionner-son-code/` or click on the image to launch the video

To convert a R project into git repository: see `https://rtask.thinkr.fr/fr/transformer-un-dossier-en-projet-git-synchronise-sur-githu` or click on the image to launch the video

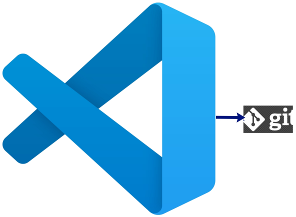
Git and VSCode

To work with Git in VSCode: see `https://code.visualstudio.com/docs/sourcecontrol/intro-to-git`

Video to import Git project in VSCode:

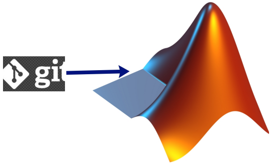


Video to import VScode project into Git:



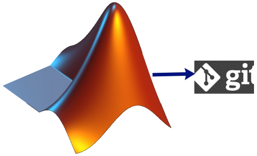
Matlab and Git

To import a git repository in Matlab: see `https://fr.mathworks.com/help/matlab/matlab_prog/use-git-in-matlab.html` or click on the image to launch the



video

To convert Matlab folder into Git repository: see `https://fr.mathworks.com/help/simulink/ug/share-project-on-github.html` or launch the video



You can also use git command by prefixing it with ! in Matlab terminal (for instance !git add *.m)

Stata and Git

You can work with git involving the shell in Stata terminal, that is using ! before the git command (for instance !git add *.do). See `https://medium.com/the-stata-guide/stata-and-github-integration-8c87ddf9784a` and `https://gitlab.univ-lorraine.fr/legall5/git-with-stata`

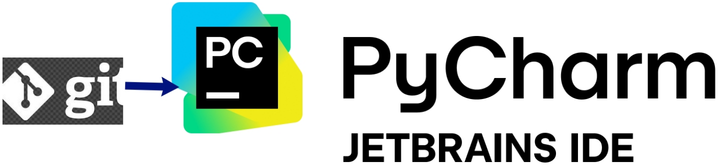
To import git project into Stata

To convert your Stata project into git repository

PyCharm and Git

See `https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#clone-repo`

To import git project into Pycharm



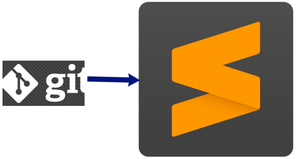
To import Pycharm project into Github

PC
PyCharm
JETBRAINS IDE
git

SublimeText and Git

Since Version 3.2 Git is integrated with SublimeText: see `https://www.sublimetext.com/docs/git_integration.html`
You can also install the package controller in SublimetText: see `https://packagecontrol.io/installation#Manual` then install a git package (for instance Git Sappy)

To import git project into SublimeText



To import files from SublimeText in Git