

Suivre et partager ses sources avec Gitlab: un exemple avec \LaTeX

Matthieu Boileau, Alexis Palaticky

CNRS - Université de Strasbourg

19 janvier 2016



&



GitLab

Outline

- 1 Intérêt et applications de git
- 2 GitLab comme serveur git
- 3 Démonstration
- 4 Conclusion

Pourquoi utiliser un suivi de version ?

- **Enregistrer** les modifications d'un jeu de fichiers au cours du temps
- Rester **réversible** :
 - pouvoir retourner à une version antérieure,
 - comparer avec une version antérieure
- **Documenter** les modifications (date, auteur et message d'accompagnement)
- Un logiciel de suivi de version (VCS pour *Version Control System* en anglais) comme Git gère très bien tout projet qui se présente sous la forme de **fichiers sources**
- C'est le meilleur moyen de **collaborer** sur des sources !

Pourquoi utiliser un suivi de version ?



Ce que Git gère très bien :

- ✓ les scripts ou code de calcul
- ✓ les documents \LaTeX
- ✓ les fichiers texte de configuration
- ✓ les sources html
- ✓ etc.

Ce que Git gère mal :

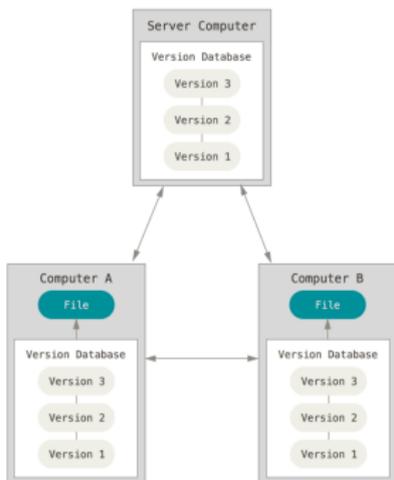
- ✗ les gros fichiers binaires
- ✗ les documents Microsoft Office ou OpenOffice
- ✗ le texte formaté en général
- ✗ les bases de données (type mysql)
- ✗ etc.

Comparaison avec Owncloud

	Owncloud	Git
Type de fichiers	<p>✓ tous types</p>	<p>✓ suivi pour les fichiers sources</p> <p>✗ pas de suivi pour les binaires</p> <p>✗ pas adapté aux gros fichiers (sauf avec git-lfs)</p>
Suivi de version	<p>✗ très limité</p>	<p>✓ outil avancé</p>
Partage	<p>✗ modèle centralisé uniquement</p> <p>✗ synchronisations automatiques uniquement</p>	<p>✓ modèle distribué</p> <p>✓ on contrôle les synchronisations</p>
Prise en main	<p>✓ très simple</p>	<p>✗ demande un apprentissage</p>

Un suivi de version distribué

- les clients possèdent un miroir complet de la base de données du serveur
- on peut travailler en mode déconnecté et synchroniser quand on le souhaite
- indirectement, on crée des sauvegardes multiples



(Source: Pro Git book <http://git-scm.com/book>)

Git en pratique

Git en ligne de commande dans le Terminal :

```
git add
git commit -m "My commit message"
git status
git log
git push
git pull
git checkout
git diff
etc.
```

```
boileau@boileau: ~$ git help --all
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path<gitpath>] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<gitpath>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

available git commands in '/Applications/Xcode.app/Contents/Developer/usr/libexec/git-core'

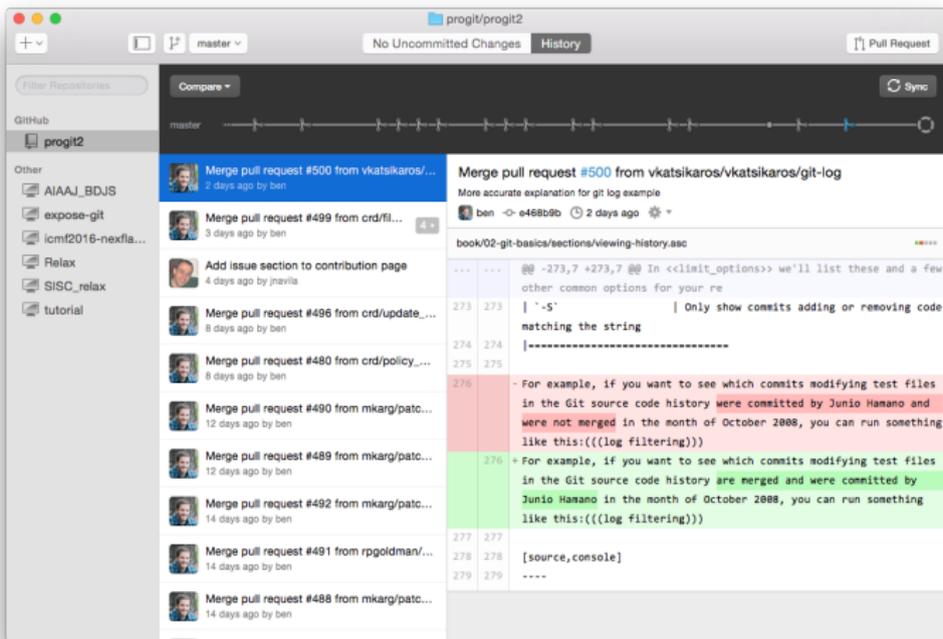
add             diff-files           merge-octopus       reset
add-interactive diff-index           merge-one-file      rev-list
am              diff-tree            merge-ours          rev-parse
annotate        difftool              merge-recursiv      revert
apply           difftool--helper     merge-resolve       rm
archimport      fast-export           merge-subtree       send-email
archive         fast-import          merge-tree           send-pack
bisect          fetch                 mergetool           sh-110m--envsubst
bisect--helper fetch-pack            aktag               shell
blame           filter-branch        aktree              shortlog
branch          fast-merge-sg        av                  show
bundle          for-each-ref          name-rev            show-branch
cat-file        format-patch         notes               show-index
check-attr      fsck                 pack-objects        show-ref
check-ignore    gc                   pack-redundant      stage
check-mailmap   get-tar-commit-id    pack-refs           stash
check-ref-format grep                  patch-id            status
checkout         gui--askpass         prune                stripspace
checkout-index  hash-object          prune-packed         submodule
cherry-pick     help                 pull                subtree
citool          http-backend         push                svn
clean           http-fetch           quiltimport         symbolic-ref
clone           http-push            read-tree           tag
column          imap-send            rebase              unpack-file
commit          index-pack           receive-pack        unpack-objects
commit-tree     init                 reflow             update-index
config          init-db              relink              update-ref
count-objects   instaweb             remote              update-server-info
credential      interpret-trailers   remote-ext          upload-archive
credential-cache log                   remote-fd          upload-pack
credential-cache-daemon ls-remote            remote-ftp         var
credential-osxkeychain ls-remote            remote-ftps        verify-commit
credential-store ls-tree              remote-http        verify-tag
cvsimportcommit mailinfo             remote-https       web--browse
cvsimport       mailsplit            remote-testsvn     watchchanged
cvsserver       merge                repack             worktree
daemon          merge-base           replace            write-tree
describe        merge-file           request-pull
diff            merge-index          rerere

git commands available from elsewhere on your $PATH
diff-cmd.sh latexdiff lfs

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
boileau@boileau: ~$
```

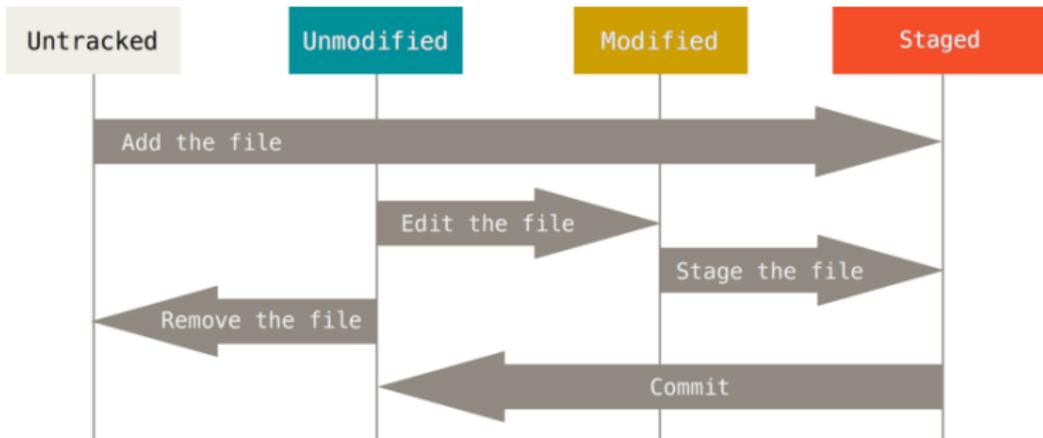
Git en pratique

Avec une interface graphique très simple comme GitHub Desktop, on couvre $\approx 90\%$ de l'utilisation courante de git :



Les quatre statuts des fichiers suivis

- Le cycle de vie d'un fichier suivi avec Git

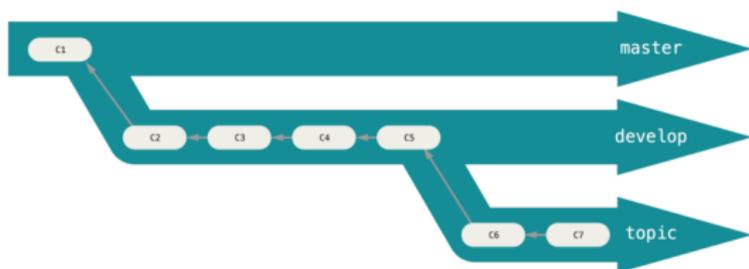


(Source: Pro Git book <http://git-scm.com/book>)

- Les fichiers qui ne sont pas des sources (fichiers objets, fichiers de compilations, exécutables, etc.) peuvent être ignorés.

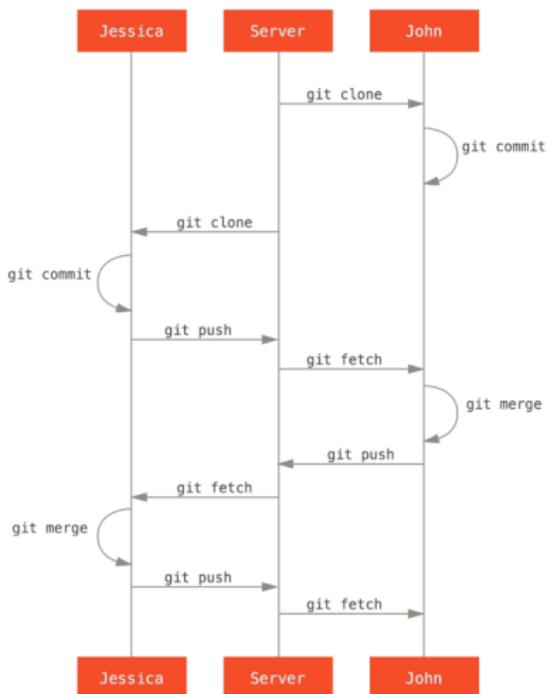
Git et le système des branches

- Git permet de créer et fusionner très facilement des branches
- un système de branches permet de préserver une version stable (branche *master*) sans limiter les développements (branche *develop*)
- les branches sont particulièrement utiles pour le travail collaboratif et par sujet (*topic*).



(Source: Pro Git book <http://git-scm.com/book>)

Travail collaboratif avec Git : le workflow typique d'une petite équipe



En pratique :

- Côté serveur : **GitLab**
- Côté clients (John et Jessica) : ligne de commande ou client graphique (GitHub Desktop, par exemple)

(Source: Pro Git book <http://git-scm.com/book>)

Gitlab comme serveur git



GitLab

Quelques précisions sur GitLab :

- GitLab ? Outil opensource de gestion de projets git
- Pourquoi GitLab à l'IRMA ?
 - Besoin de travail collaboratif sur du code ou des publications
 - Pour avoir la maîtrise sur le paramétrage et la création des comptes
 - Gérer des fonctionnalités avancées comme l'intégration continue

Gitlab comme serveur git

GitLab à l'IRMA :

- `https://gitlab.math.unistra.fr`
- Serveur sauvegardé en interne (VDP et Netbackup)
- Mise à jour de GitLab et mises à jour systèmes régulières
- Serveur accessible de l'extérieur, possibilité de comptes externes

GitLab comme serveur git

Les fonctionnalités de GitLab :

- Création rapide de projets avec gestion des droits (Public, privé,...)
- Historique des commits
- Outils annexes : statistiques, wiki
- Notifications par mail
- Intégration continue

GitLab comme serveur git

Les fonctionnalités de GitLab :

- Création rapide de projets avec gestion des droits (Public, privé,...)

Projects

Signed in successfully.

Your Projects | Starred Projects | Explore Projects

Filter by name

- E** formation-info / expose-git ★ 0
- F** formation-info / formation-git
Formation git ★ 0
- E** irma-admin / expose_irma
Exposé de la journée des nouveaux entrants ★ 0
- T** Palaticky / test ★ 0

GitLab

Go to dashboard

- Project
- Activity
- Builds 0
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Settings

palatick

Palaticky / test2

Search in this project

Project 'test2' was successfully created.



test2

★ 0 SSH HTTPS git@gitlab.math.unistra.fr:pal

The repository for this project is empty

If you already have files you can push them using command line instructions below.
Otherwise you can start with [adding README](#) file to this project.

Command line instructions

Git global setup

```
git config --global user.name "Palaticky"  
git config --global user.email "alexis.palaticky@math.unistra.fr"
```

Create a new repository

GitLab comme serveur git

Les fonctionnalités de GitLab :

- Historique des commits

📅 11 Jan, 2016

5 commits

encore des nouvelles diapos alexis palaticky authored 7 days ago [0a1d8129](#)[Browse Code](#)**Nouvelles diapo** alexis palaticky authored 7 days ago [2210bb53](#)[Browse Code](#)**Nouvelle diapo** alexis palaticky authored 7 days ago [7d8c0371](#)[Browse Code](#)**Add figures** alexis palaticky authored 7 days ago [f204aeed](#)[Browse Code](#)**ignore compilation files** alexis palaticky authored 7 days ago [d3f528de](#)[Browse Code](#)

📅 08 Jan, 2016

1 commit

Expose Palaticky authored 10 days ago [406b0b2a](#)[Browse Code](#)

📅 07 Jan, 2016

3 commits

Update Palaticky authored 11 days ago [1b06e8aa](#)[Browse Code](#)**Se** Palaticky authored 11 days ago [35eea4ca](#)[Browse Code](#)**"Modification Alexis"** Palaticky authored 11 days ago [b98d863d](#)[Browse Code](#)

📅 18 Dec, 2015

3 commits

Some changes Matthieu Boileau authored about a month ago [d7ccec49](#)[Browse Code](#)**Plan expose** Matthieu Boileau authored about a month ago [21081158](#)[Browse Code](#)**A few notes** Matthieu Boileau authored about a month ago [dec8f930](#)[Browse Code](#)

palatick

GitLab comme serveur git

Les fonctionnalités de GitLab :

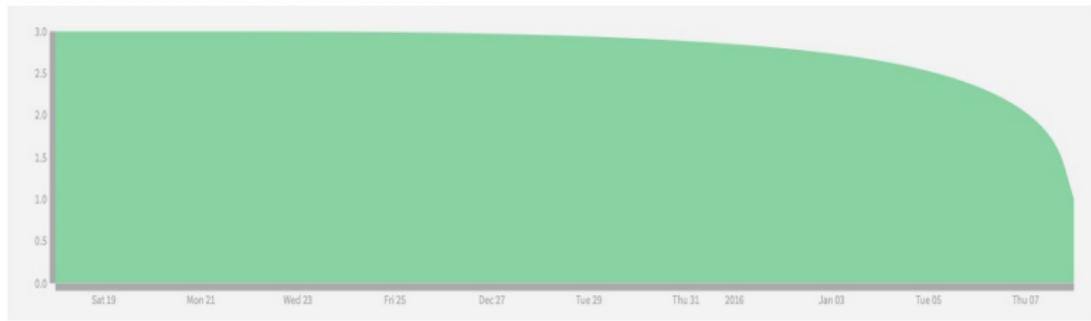
- Outils annexes : statistiques, wiki

- Go to group
- Project
- Activity
- Files
- Commits
- Network
- Graphs**
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Settings
- palatick

formation-info / expose-git · Graphs

Search in this project

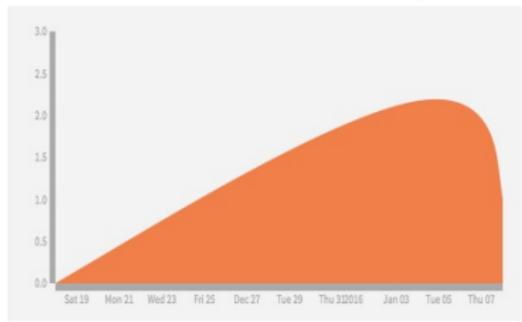
Commits to master, excluding merge commits. Limited by 6,000 commits



Palaticky

4 commits

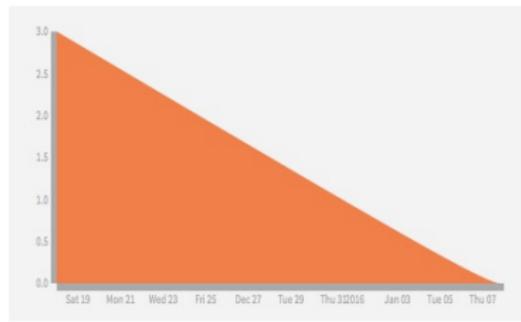
alexis.palaticky@math.unistra.fr



Matthieu Boileau

3 commits

matthieu.boileau@math.unistra.fr



GitLab

Go to group

- Project
- Activity
- Files
- Commits
- Network
- Graphs
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki**
- Settings

Avatar: palatick

Editing - home

[+ NEW PAGE](#) [PAGE HISTORY](#) [EDIT](#)

Format

To link to a (new) page you can just type `[Link Title](page-slug)`.

Content [Write](#) [Preview](#) [Edit in fullscreen](#)

Wiki en Markdown

Markdown tip: Emoji can be added by :emoji_name;, for example :thumbsup: [Attach a file](#)

Commit message

GitLab comme serveur git

Les fonctionnalités de GitLab :

- Notifications par mail

Matthieu Boileau pushed to branch master at [formation-info / expose-git](#)

Commits:

- [2266e14b](#)
by Matthieu Boileau at 2016-01-18T09:53:47Z
Change in README

1 changed file:

- [README.md](#)

Changes:

- [README.md](#)

```
--- a/README.md  
+++ b/README.md  
@@ -78,6 +78,8 @@ Outils annexes : wiki, statistiques
```

```
-### Annoncer une formation initiation git  
+### Annoncer :  
  
+- un tuto d'initiation git  
+- un sondage pour une formation
```

[View it on GitLab](#)

You're receiving this email because of your account on gitlab.math.unistra.fr. If you'd like to receive fewer emails, you can adjust your notification settings.

Démonstration

- **Objectif** : montrer l'intérêt d'utiliser Git pour collaborer sur un document \LaTeX
- **Programme de la démo** :
 - création et partage de projet dans GitLab
 - importation d'un article en \LaTeX
 - modifications, synchronisation (push, pull)
 - notifications par mail
 - édition collaborative (avec et sans conflits)
 - compilation des différences (`git latexdiff`)

Démonstration

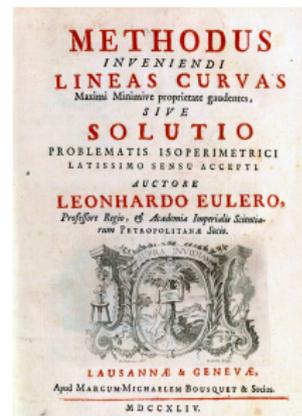
Imaginons une collaboration fictive entre **Johann Bernoulli** et **Leonhard Euler** sur l'appendice II d'un ouvrage réel : le *Methodus Inveniendi*.



Johann Bernoulli
(1667 - 1748)



Leonhard Euler
(1707 - 1783)

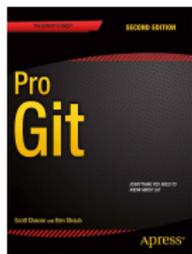


Le *Methodus Inveniendi*
d'Euler (1744)

C'est parti !

Documentation et bonnes pratiques

Documentation



ProGit book : une référence (libre)

- Pro Git book : <http://git-scm.com/book>
- la documentation officielle de Git : <http://git-scm.com/documentation>
- un manuel concis : <http://gitref.org/index.html>
- la doc GitLab et son aide contextuelle : <https://gitlab.math.unistra.fr>

Quelques recommandations

- En \LaTeX , ne pas écrire plus d'**une phrase par ligne**
- Faire de **nombreux commits** (Git est fait pour ça) contenant des modifications **petites et cohérentes** plutôt que l'inverse
- Utiliser les branches !

À venir

- **Un tutorial** pour démarrer avec GitLab à l'IRMA
- Un sondage pour **une formation d'initiation** à Git

Merci de votre attention !

info@math.unistra.fr