

# Git

Dominique DUFOUR domduf.com

10 mars 2018



# table

Présentation

Installer

Premières configurations

Utiliser en local

initialiser un dépôt local

premier commit

les commandes indispensables

les alias

le fichier .gitignore

Le dépôt sur Github

Mise à jour / fetch

Pousser / push

Tirer / pull

Les branches

Politique de travail

Mettre en commun/Merger

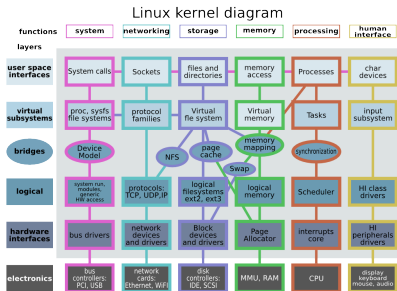
The End



# Qu'est-ce que c'est, à quoi ça sert, qui qu'a fait ça ?

parce que ça n'a pas l'air simple...

Lorsque le noyau de Linux devint un peu compliqué à faire évoluer (un grand nombre de développeurs étaient sur le coup, maintenir un historique des versions commençait à devenir chaud), un homme très important, **Linus Torvalds**, le créateur de Linux se retroussa les manches et créa ce formidable outil.



© 2007-2009 Constantine Shulyupin <http://www.MakeLinux.net/kernel/diagram>

Voilà le lien du **livre de référence Git**, si vous êtes perdus ou curieux.

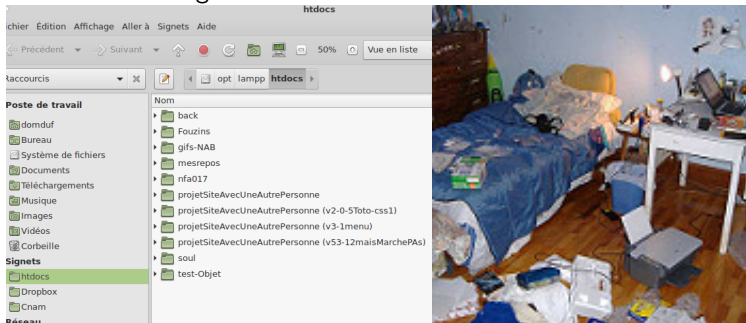
la version 2 du git-book : <https://git-scm.com/book/fr/v2/>



# L'intérêt de Git.

sinon c'est pas la peine...

Si vous avez déjà travaillé sur un projet, tel la construction d'un site internet, **avec une autre personne**, vous vous êtes certainement déjà retrouvé avec ce genre de chose :











C'est à dire un répertoire rempli de dossiers du même site, ayant un nom de version différent, et au bout du compte...**bordélique**.



Git va permettre de faire plusieurs choses **très intéressantes** :

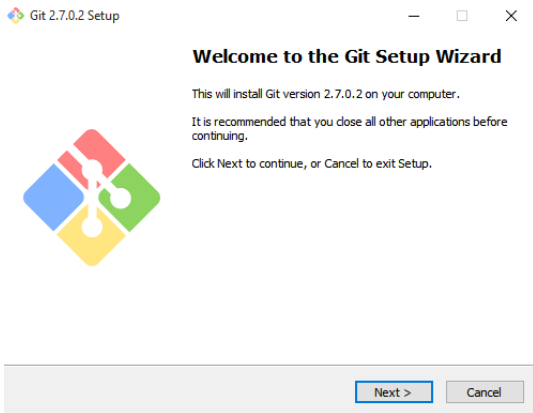
▶ Permettre d'utiliser **un seul dossier**

| Poste de travail  | Nom                                      | Taille      | Type    |
|---|--|-------------|---------|
|  domduf              | back                                     | 7 éléments  | dossier |
|  Bureau              | Fouzins                                  | 0 élément   | dossier |
|  Système de fichiers | gifs-NAB                                 | 9 éléments  | dossier |
|  Documents           | mesrepos                                 | 34 éléments | dossier |
|  Téléchargements     | nfa017                                   | 31 éléments | dossier |
|  Musique             | projetSiteAvecUneAutrePersonne(gitUsing) | 0 élément   | dossier |
|  Images              | soul                                     | 48 éléments | dossier |
|  Vidéos              | test-Objet                               | 3 éléments  | dossier |

- ▶ sauvegarder les anciennes versions du site, c'est à dire tout le boulot précédant, dans ce même dossier, **de façon transparente**.
- ▶ **partager** votre version avec votre binôme, en ligne.
- ▶ gérer les **conflits de version** sans destruction définitive du code.
- ▶ travailler sur une partie du site, pendant que votre binôme s'occupe d'une autre, et ensuite "mélanger" votre travail, de façon **non destructive**.



# Installer



Pour tous les systèmes<sup>1</sup>, lisez d'abord la page ici :  
<https://git-scm.com/book/fr/v2> à **la section 1.5** et laissez vous guider !

1. image source : <https://www.studytonight.com/github/>



Il faut, pour pouvoir par la suite utiliser Git, entrer vos coordonnées.<sup>2</sup>  
*Attention les options sont précédées de deux tirets.*

- ▶ `git config --global user.name "Prenom Nom"`  
Configure vos **Prénom et Nom**, à mettre entre guillemets.
- ▶ `git config --global user.email votre.adresse@email.com`  
Configure votre **adresse email** (utile par la suite avec le dépôt en ligne Github)
- ▶ `git config --list`  
Vous montre les configurations déjà effectuées. Voici ma configuration personnelle<sup>3</sup>

```
domduf@debian-toshiba:/opt/lampp/htdocs/NFA084 (master)$ git config --list
user.name=domduf
user.email=minique.duf@gmail.com
alias.lol=log --oneline --graph --decorate --all
alias.rem=remote show origin
alias.co=checkout
alias.br=branch
alias.ci=commit
alias.st=status
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

2. <https://git-scm.com/book/fr/v2> à la section 1.6

3. pour les alias, voir section 4

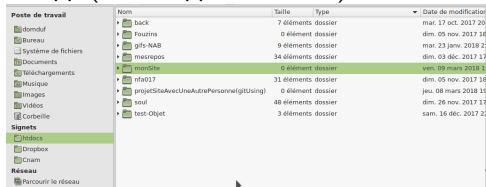


## Utiliser en local

Lorsqu'on parle d'utilisation en local, cela concerne votre travail sur votre propre ordinateur.

Sans pour l'instant parler de partage de votre projet avec votre ou vos collaborateurs.

Vous disposez d'un dossier, par exemple un dossier de site web nommé **monSite**, qui se trouve dans le dossier **htdocs** de votre serveur local xampp ( ou lampp sous Linux) :



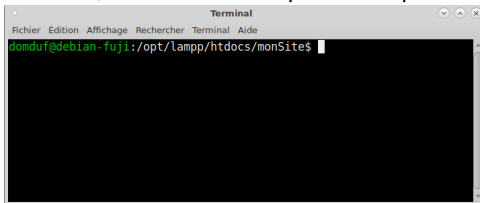
Vous allez ouvrir une console

(sous Windows, un clic droit sur le dossier **monSite** va vous proposer l'option d'ouverture de la console, si git est installé)





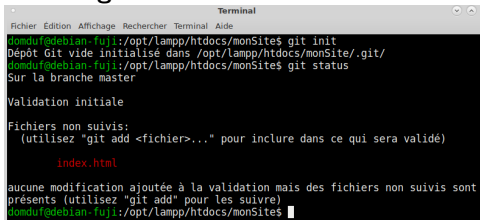
Voici un aperçu d'un terminal (ici sous Linux). Le vôtre doit être différent, mais vous allez pouvoir taper des commandes...



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
domduf@debian-fuji:/opt/lampp/htdocs/monSite$
```

Pour commencer un **"git init"** pour que git puisse être utilisé dans ce dossier.

Puis un **git status** nous montre l'état de notre suivi de version :



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
domduf@debian-fuji:/opt/lampp/htdocs/monSite$ git init
Dépôt Git vide initialisé dans /opt/lampp/htdocs/monSite/.git/
domduf@debian-fuji:/opt/lampp/htdocs/monSite$ git status
Sur la branche master

Validation initiale

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

    index.html

aucune modification ajoutée à la validation mais des fichiers non suivis sont
présents (utilisez "git add" pour les suivre)
domduf@debian-fuji:/opt/lampp/htdocs/monSite$
```



Vous décidez d'inclure dans votre historique le seul fichier `index.html` :  
**git add .** (notez la présence du point, qui signifie ici "tout ce qui n'est pas suivi")

Puis un **git status** nous montre l'état de notre suivi de version :

```
domduf@debian-fuji:/opt/lampp/htdocs/monSite$ git add .
domduf@debian-fuji:/opt/lampp/htdocs/monSite$ git status
Sur la branche master

Validation initiale

Modifications qui seront validées :
  (utilisez "git rm --cached <fichier>..." pour désindexer)

    nouveau fichier : index.html
domduf@debian-fuji:/opt/lampp/htdocs/monSite$
```

Pour l'instant, les modifications ne sont **pas encore "enregistrées"** au niveau des étapes de votre projet, pour cela...

## IL FAUT COMMITER



# Premier commit

"Commit" peut se comprendre comme :

- ▶ Enregistrer une étape de mon travail à un instant "t" pour pouvoir y avoir accès dans le futur.

Par exemple pour pouvoir revenir en arrière si je continue mon travail et que j'estime que ce que j'ai fait n'est pas conforme à mes attentes.



## les commandes indispensables

- ▶ git status  
Permet de **visualiser** les fichiers qui ont été modifiés, ajoutés, supprimés, depuis le dernier commit
- ▶ git add  
Permet d'**ajouter** les fichiers qui devront être **suivis** dans leur version lors du prochain commit.
- ▶ git commit  
Permet de faire **rentrer dans l'historique** tous les fichiers "marqués" par le git add
- ▶ git log  
Va afficher un historique des commits déjà réalisés.

```
domduf@debian-fuji:~/Dropbox/Cnam/PROF/git (master)$ git log
commit b29b111dd9e721e20745c8839ba940831f564a12
Author: domduf <minique.duf@gmail.com>
Date: Thu Mar 8 20:16:20 2018 +0100

    ajout des sections et subSection

commit d4eb3a9f6c721e14dbf6b1de1b5c0331229c5451
Author: domduf <minique.duf@gmail.com>
Date: Thu Mar 8 19:53:20 2018 +0100

    ajout lien vers le Github

commit 444ca9d42ff5cfc4a32777010166fb2c922b6f8b
Author: domduf <minique.duf@gmail.com>
Date: Thu Mar 8 19:47:33 2018 +0100
```



## Les alias

Les alias sont des raccourcis de commande<sup>4</sup>.

Ces raccourcis sont appliqués bien sûr aux commandes les plus utilisées, et vont vous faire gagner du temps.

On les définit soi-même (une fois pour toute) en ligne de commande, en tapant :

**git config --global alias.nomDeLAlias commande**

Par exemple, pour utiliser **git st** au lieu de **git status** , la commande de création de l'alias sera :

**git config --global alias.st status**

```
domduf@debian-fuji:~/Dropbox/Cnam/PROF/git (master)$ git status
Sur la branche master
Votre branche est à jour avec 'origin/master'.
Modifications qui seront validées :
  (utilisez "git reset HEAD <fichier>..." pour désindexer)

    modifié :      git.tex

Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git checkout -- <fichier>..." pour annuler les modifications dans l
a copie de travail)

    modifié :      git.tex

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

    git.aux.bbl
    git.aux.blg
```

```
domduf@debian-fuji:~/Dropbox/Cnam/PROF/git (master)$ git st
Sur la branche master
Votre branche est à jour avec 'origin/master'.
Modifications qui seront validées :
  (utilisez "git reset HEAD <fichier>..." pour désindexer)

    modifié :      git.tex

Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git checkout -- <fichier>..." pour annuler les modifications dans l
a copie de travail)

    modifié :      git.tex

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

    git.aux.bbl
    git.aux.blg
```

4. [https:](https://github.com/domduf/git_trucs_et_astuces/blob/master/alias_utiles.txt)

[//github.com/domduf/git\\_trucs\\_et\\_astuces/blob/master/alias\\_utiles.txt](https://github.com/domduf/git_trucs_et_astuces/blob/master/alias_utiles.txt)



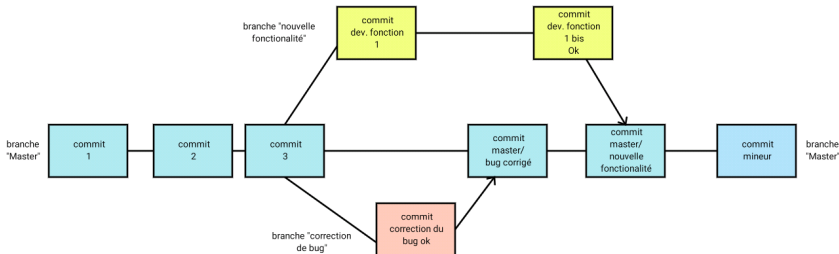
L'intérêt de Git, c'est aussi et surtout de pouvoir travailler sur un même projet à plusieurs...

Un site internet est souvent construit avec au minimum :

- ▶ un responsable de projet
- ▶ un plusieurs codeur de la structure du site
- ▶ un ou plusieurs graphiste



# Politique de travail, diapo 1/2<sup>5</sup>



## Politique de travail, diapo 2/2

- ▶ vous faites une branche pour ajouter une nouvelle fonctionnalité
- ▶ en plein milieu on vous demande de corriger un bug en urgence
- ▶ vous ne le faites pas dans votre nouvelle branche car le code n'est pas encore fonctionnel  
vous ne le faites pas non plus dans la branche principale car celle-ci ne doit pas être utilisée comme branche de développement
- ▶ vous créez une seconde branche depuis la branche principale pour corriger le bug
- ▶ vous fusionnez cette seconde branche avec la branche principale une fois la correction terminée
- ▶ vous terminez votre ajout de fonctionnalité sur la première branche
- ▶ vous fusionnez votre première branche avec la (nouvelle) branche principale en réglant les conflits éventuels





La suite au prochain épisode...

pour les curieux, le code de ce document est là :

<https://github.com/domduf/GitBeamer>

Et la version pdf actualisée à cette adresse Dropbox

[https://www.dropbox.com/sh/6bf9gw8q2mtqf4b/AAAy\\_1ja36MJiDIjezwYSVIga?dl=0](https://www.dropbox.com/sh/6bf9gw8q2mtqf4b/AAAy_1ja36MJiDIjezwYSVIga?dl=0)

