

# DOCUMENTATION GIT



git

# **Sommaire:**

## **1- Introduction**

- Histoire-----
- Utilité-----
- Installation-----

## **2- Commandes de bases**

- init-----
- add-----
- push-----
- clone-----
- commit-----
- pull-----
- diff-----
- config-----
- status-----
- branch-----
- checkout-----
- remote-----
- merge-----
- fetch-----
- stash-----
- log-----

# INTRODUCTION

## Histoire:

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.

En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.

## Utilité:

Git est un système de gestion de version gratuit et à open source, conçu pour gérer tout projet, du plus petit au plus grand, avec rapidité et efficacité.

Le logiciel git est utilisé notamment par les développeurs du noyau Linux. Il s'agit d'un logiciel de contrôle de version, comme Subversion, CVS, Arch ou encore Darcs, qui utilise des dépôts décentralisés. Il permet ainsi de travailler tout en gardant une trace des modifications apportées successivement, et de retrouver ainsi l'état antérieur de ses données.

## Installation:

Installer Git sur Windows est très facile. Le projet msysGit fournit une des procédures d'installation les plus simples. Téléchargez simplement le fichier exe d'installateur depuis la page GitHub, et lancez-le :

<http://msysgit.github.io>

Après son installation, vous avez à la fois la version en ligne de commande (avec un client SSH utile pour la suite) et l'interface graphique standard.

# Commandes de bases

## git init:

Cette commande est utilisée pour créer un nouveau dépôt GIT.

Par exemple si vous commencez à suivre un projet existant dans Git, vous n'avez qu'à vous positionner dans le répertoire du projet et saisir :

```
g.unique@SFDA3 MINGW64 /c/Users/g.unique/Documents/si7
$ git init
Initialized empty Git repository in C:/Users/g.unique/Documents/si7/.git/
```

## git add:

La commande git add peut être utilisée pour ajouter des fichiers à l'index. Par exemple, la commande suivante ajoute un fichier nommé temp.txt dans le répertoire local de l'index:

```
g.unique@SFDA3 MINGW64 /c/Users/g.unique/Documents/si7 (master)
$ git add doc-git.odt
```

## git push:

Met à jour les références distantes à l'aide de références locales, tout en envoyant les objets nécessaires pour compléter les références données.

```
g.unique@SFDA3 MINGW64 /c/Users/g.unique/Documents/si7 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 75.74 KiB | 15.15 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/gregoire-unq/doc-git.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

### git pull:

Pour fusionner toutes les modifications présentes sur le dépôt distant dans le répertoire de travail local

```
g.unique@SFDA3 MINGW64 ~ (master)
$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=<remote>/<branch> master
```

### git diff:

Affiche les modifications entre l'arborescence de travail et l'index ou une arborescence, les modifications entre l'index et une arborescence, les modifications entre deux arborescences, les modifications entre deux objets blob ou les modifications entre deux fichiers sur le disque.

```
g.unique@SFDA3 MINGW64 ~ (master)
$ git diff
```

### git config:

La première chose à faire après l'installation de Git est de renseigner votre nom et votre adresse e-mail. C'est une information importante car toutes les validations dans Git utilisent cette information et elle est indélébile dans toutes les validations que vous pourrez réaliser :

```
g.unique@SFDA3 MINGW64 ~ (master)
$ git config --global user.name "gregoire.unique"

g.unique@SFDA3 MINGW64 ~ (master)
$ git config --global user.email gregoire.unique37@gmail.com
```

### git status:

L'outil principal pour déterminer quels fichiers sont dans quel état est la commande git status. Si vous lancez cette commande juste après un clonage, vous devriez voir ce qui suit :

```
$ git status
On branch master
nothing to commit, working directory clean
```

### git branch:

Créer une nouvelle branche.

```
g.unique@SFDA3 MINGW64 ~ (master)
$ git branch
```

### git checkout:

Met à jour les fichiers de l'arborescence de travail afin qu'ils correspondent à la version de l'index ou de l'arborescence spécifiée. Si aucun chemin n'est donné, git checkout se mettra également à jour.

```
git checkout [<branche>]
```

### git remote:

Gérez l'ensemble des référentiels ("télécommandes") dont vous suivez les branches.

```
g.unique@SFDA3 MINGW64 /c/Users/g.unique/Documents/si7 (master)
$ git remote add origin https://github.com/gregoire-unq/doc-git.git
```

### git merge:

Intègre les modifications des commits nommés (à partir du moment où leurs historiques ont divergé de la branche actuelle) dans la branche actuelle.

### git fetch:

Récupérez les branches et / ou les balises (collectivement, les "références") d'un ou de plusieurs autres référentiels, ainsi que les objets nécessaires pour compléter leur historique.

```
g.unique@SFDA3 MINGW64 ~ (master)
$ git fetch
```

### git clone:

Si vous souhaitez obtenir une copie d'un dépôt Git existant — par exemple, un projet auquel vous aimeriez contribuer — la commande dont vous avez besoin s'appelle git clone.

ex: `$ git clone git://github.com/schacon/grit.git`

## git commit:

Enregistrer les modifications dans le référentiel

```
g.unique@SFDA3 MINGW64 /c/Users/g.unique/Documents/si7 (master)
$ git commit -m "Documentation GIT"
[master (root-commit) f5a2c12] Documentation GIT
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 doc-git.odt
```