**Les commandes de git**

Haoling ZHANG

29/09/2019

**Sommaire**

Abstract………………………………………………………………………………..1

Introduction………………………………………………………………………..…..2

Processus………………………………………………………………………………31. Les commandes de git………………………………………………………........3

1.1 Git config…….......……………………………………………………..…....3

1.2 Git init……………………………………………………………….............4

1.3 Git status……………………………………………………………..………5

1.4 Git merge …………………………………………………………………….5

1.5 Git diff…………………………………………………………………….....5

1.6 Git blame……………………………………………………………………..5

2. Avantage de git...........……………………………………………………………7

Conclusion………………………………………………………………….………….8

Documents de références………………………………………………..……………..8

**Abstract**

Git a été spécialement développé et optimisé pour le noyau Linux. Linus Torvalds a commencé son écriture en avril 2005, pour remplacer le programme propriétaire BitKeeper. La première version a été publiée le 7 avril 2005. Depuis, le développement de cet outil s'est poursuivi. Il a été progressivement amélioré, se voyant doté d'interfaces graphiques, d'interfaces web ou de scripts évolués. Le développement du Git est très actif. La parution de la première version stable (1.0) date du 21 décembre 2005, soit moins d'un an après le démarrage du projet. [1]

Dans cet article, il y a des détailles sur les commandes de git ( git config, git init, git status, git merge, git diff, git blame ) et aussi des avantages de git, incluant pourquoi il y a plus de 70 pour cent des développeurs utilisent Git.

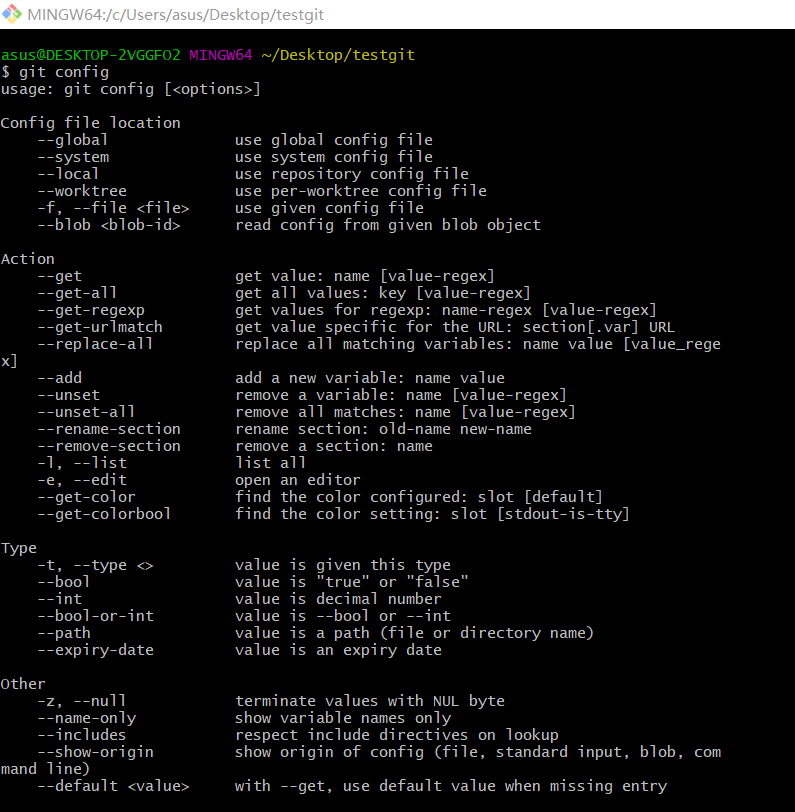
**Introduction**

Git est un [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions" \o "Logiciel de gestion de versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions" \l "Gestion_de_versions_d.C3.A9centralis.C3.A9e" \o "Gestion de versions). C'est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre" \o "Logiciel libre) créé par [Linus Torvalds](https://fr.wikipedia.org/wiki/Linus_Torvalds" \o "Linus Torvalds), auteur du [noyau Linux](https://fr.wikipedia.org/wiki/Noyau_Linux" \o "Noyau Linux), et distribué selon les termes de la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU" \o "Licence publique générale GNU) version 2. En 2016, il s’agit du [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions" \o "Logiciel de gestion de versions) le plus populaire qui est utilisé par plus de douze millions de personnes.[2]

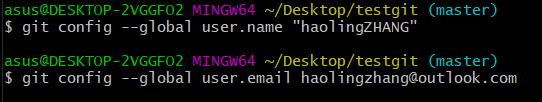
Git est un exemple d'un système de contrôle de version distribuée (DVCS) couramment utilisé pour le développement de logiciels open source et commerciaux. Git n'a pas besoin d'une connexion constante à un dépôt central. Les développeurs peuvent travailler n'importe où et collaborer de façon asynchrone à partir de n'importe quel fuseau horaire. Pour éliminer les travaux inutiles, Git et d'autres VCS donnent à chaque contributeur une vue unifiée et cohérente d'un projet, faisant surface des travaux qui sont déjà en cours. Voir une histoire transparente des changements, qui les a faits, et comment ils contribuent à l'élaboration d'un projet aide les membres de l'équipe à rester alignés tout en travaillant de façon indépendante. [3]

**Processus**

1. **Les commandes de git**
   1. git config : pour configurer l’environnement de travail du Git.

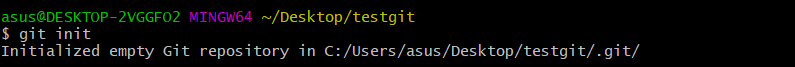
Dans un nouveaux système, on a besoin de configurer l’environnement de travail du Git d'abord. La configuration n'a pas besoin d'être effectuée qu'une seule fois, et la configuration actuelle suivra lorsque vous mettrez a jour plus tard. Et vous pouvez modifier votre configuration existante avec la même commande à tout moment si vous le souhaitez.[5]

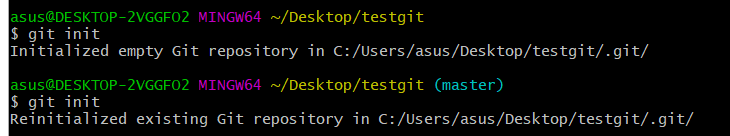
Pour le premier fois d’utiliser Git, c’est necessaire d’utiliser “git config -- global” pour configurer les information de l’utilisateur car il a besoin de savoir qui a soumis la mise à jour, et l’enregistre dans l’histoire.



* 1. git init

Il permet d’initialiser un repository vide et c’est aussi un sous-répertoire ( .git ) dans le répertoire actuel ou réinitialiser un répertoire qui est déjà existe. C’est sûr d’exécuter ce commande dans un répertoire qui est déjà existe parce qu’il ne remplacera pas de choses qui sont déjà là. Le raison d’exécuter ce commande dans un répertoire qui est déjà existe est d’ajouter des templates qui sont nouveaux ajoutées.



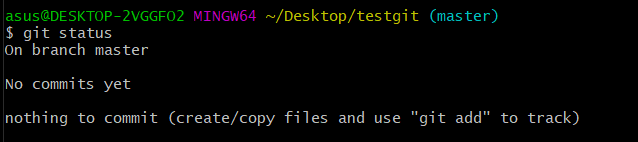


Il y a aussi un option -q

Par exemple : --quiet permet de n’afficher que les erreurs et les warnings. [4]

* 1. git status

La commande est utilisée pour afficher l'état du répertoire de travail et de “staged”. Cependant, il n’affichage pas des messages qui sont déjà “commit”.



1.4 git merge

Faire les deux ou plusieurs histoires de développement ensemble.

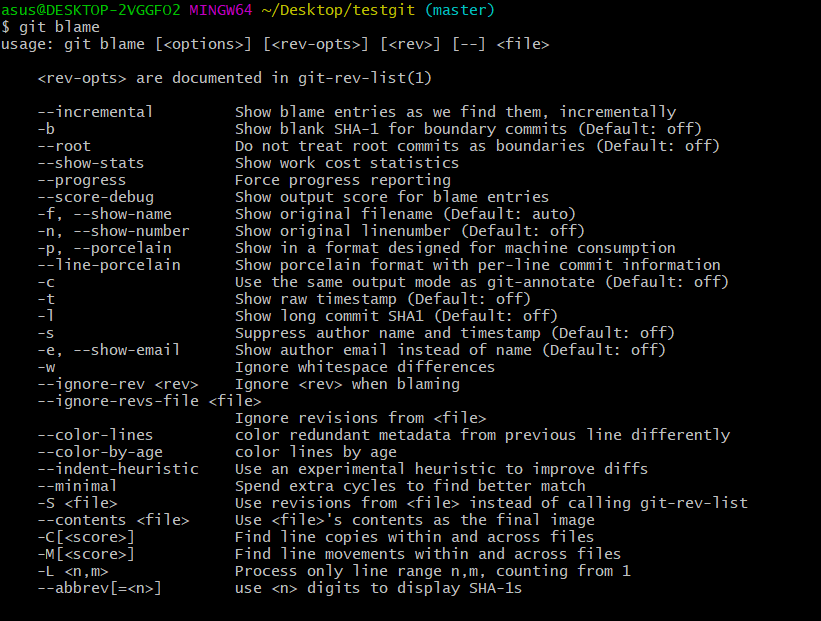
1.5 git diff

Afficher les changements entre les commits, les commits et les arbres de travail, etc.

1.6 git blame

On peut voir tous les modifie dans le fichier incluant qui a modifie et le date et autre information.

Ce sont des options precises pour ce commande.



1. **Avantage**

Selon la dernière enquête des développeurs Stack Overflow, plus de 70 pour cent des développeurs utilisent Git, ce qui en fait le VCS le plus utilisé dans le monde. Git est couramment utilisé pour le développement de logiciels open source et commerciaux, avec des avantages significatifs pour les particuliers, les équipes et les entreprises.  
 Git permet aux développeurs de voir toute la chronologie de leurs modifications, décisions et progression de tout projet en un seul endroit. Dès le moment où ils accèdent à l'historique d'un projet, le développeur a tout le contexte dont il a besoin pour le comprendre et commencer à contribuer.  
 Les développeurs travaillent dans chaque fuseau horaire. Avec un DVCS comme Git, la collaboration peut se produire à tout moment tout en maintenant l'intégrité du code source. À l'aide de succursales, les développeurs peuvent proposer en toute sécurité des modifications au code de production.  
 Les entreprises utilisant Git peuvent éliminer les barrières de communication entre les équipes et les garder concentrées sur leur meilleur travail. De plus, Git permet d'aligner des experts sur une entreprise pour collaborer à des projets d'envergure.[3]

**Conclusion**

Git --- The stupid content tracker[6] . Il est vraiment un outil excellent pour nous aide a gérer les fichiers dans le github. Et il va développer de plus en plus fort a l’avenir.

**Documents de références**

1. <https://fr.wikipedia.org/wiki/Git#Développement>
2. [« Git 2.8.2 Popular Source Code Management System Released with Over 18 Bug Fixes »](http://news.softpedia.com/news/git-2-8-2-popular-source-code-management-system-released-with-over-18-bug-fixes-503591.shtml) [[archive](http://archive.wikiwix.com/cache/?url=http://news.softpedia.com/news/git-2-8-2-popular-source-code-management-system-released-with-over-18-bug-fixes-503591.shtml" \o "archive sur Wikiwix)], sur [Softpedia](https://fr.wikipedia.org/wiki/Softpedia" \o "Softpedia), 2 mai 2016 (consulté le 2 mai 2016)
3. <https://guides.github.com/introduction/git-handbook/>
4. <http://web.mit.edu/~mkgray/project/silk/root/afs/sipb/project/git/git-doc/git-init.html>
5. <https://git-scm.com/book/zh/v1/%E8%B5%B7%E6%AD%A5-%E5%88%9D%E6%AC%A1%E8%BF%90%E8%A1%8C-Git-%E5%89%8D%E7%9A%84%E9%85%8D%E7%BD%AE>
6. <https://baike.baidu.com/item/GIT/12647237#2>