**DOCUMENTATION GIT**



SOMMAIRE

1- Introduction

* Histoire
* Utilité
* Installation

2- Commandes

* git init
* git add
* git push
* git pull
* git diff
* git config
* git status
* git branch
* git checkout
* git remote
* git merge
* git fetch
* git stach
* git log
* Explication
* Exemple
* Options principales

**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

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.

Il permet également, de travailler à plusieurs sur un même projet. Mais contrairement à CVS ou Subversion, par exemple, Git ne fait pas de différence entre un dépôt principal et les copies locales chez les différents contributeurs.

Ce système permet également une organisation hiérarchisée de gros projets, comme c'est le cas du noyau Linux, en rendant complètement naturelle l'existence de dépôts alternatifs pour chaque sous-projet.

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 et l'interface graphique standard.

**Commandes**

git init : Cette commande permet de créer un dépôt.

git add : La commande 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.

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.

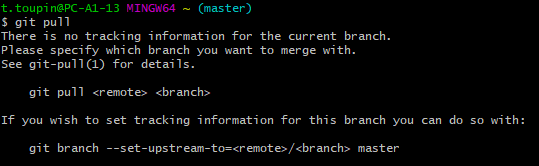
git pull : Récupère les dernières modifications distantes du projet et les fusionner dans la branche courante.

git diff : Affiche les modifications entre l’arbre de travail et l’index ou un arbre, les modifications entre l’index et un arbre, les modifications entre deux arbres, les modifications entre deux objets blobs ou les modifications entre deux fichiers sur disque.





Exemple : $ git push origin master





git config : Vous pouvez spécifier les paramètres de configuration de Git avec la commande git config. Une des premières choses que vous avez faites a été de paramétrer votre nom et votre adresse e-mail :

git status : La commande git status affiche la liste des fichiers modifiés ainsi que les fichiers qui doivent encore être ajoutés ou validés.

git branch : La commande git branch fait plus que créer et effacer des branches. Si vous la lancez sans argument, vous obtenez la liste des branches courantes

git checkout : Change de branche ou restaure les fichiers de l'arbre de travail.

git remote : Gérer un ensemble de référentiels suivis

git merge : Joindre deux ou plusieurs historiques de développement

