



• Git config

1. Où stocker le fichier de configuration
2. Configurez votre nom d'utilisateur et votre mot de passe

```
$ git config --global user.name "xx"
```

```
$ git config --global user.email xxx@xxx.com
```
3. Configurez votre compilateur

```
$ git config --global core.editor emacs
```
4. Configurez votre outil de comparaison

```
$ git config --global merge.tool vimdiff
```

5. Vérifiez votre configuration

```
$ git config --list
user.name=xx
user.email=xxx@xx.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

6. Obtenir de l'aide

```
$ git config user.name
xx
```

•Git init

1. Il est généralement utilisé pour initialiser le répertoire courant (un répertoire non vide) en tant que référentiel local.
2. `git init -q` exécution silencieuse
3. `git init --bare` généralement utilisé pour initialiser un répertoire vide comme dépôt de stockage à distance
4. `git init --template template_dir - template` équivalent à la copie du référentiel de modèles, `template_dir` répertoire du référentiel de modèles
5. `git init --separate-git-dir git_dir` l'équivalent de la copie du répertoire `.git` depuis le référentiel `git_dir` (soit un référentiel de travail, soit un référentiel de stockage).

•Git status

La commande `git status` est utilisée pour afficher l'état du répertoire de travail et de la zone de transit. Utilisez cette commande pour voir quelles modifications ont été mises à disposition, lesquelles ne l'ont pas été, et quels fichiers n'ont pas été suivis par Git. `git status` ne montre pas d'informations sur ce qui a été livré dans l'historique du projet.

•Git add

1. Suivi.
`git add:`Suivre tous les fichiers du répertoire courant
`git add :<filename>:Fichiers suivis<filename>`
2. le stockage des fichiers modifiés.
`git status` apparaît sous la ligne "Changed but not updated", ce qui

signifie que le contenu du fichier suivi a été modifié mais n'a pas encore été placé dans la zone de transit. `git add` place les mises à jour suivies dans la zone de transit.

`git add :<filename>:Fichiers suivis<filename>`

•Git push

Pousser le référentiel local vers le serveur distant

•Git merge

La commande `git merge` est utilisée pour combiner les branches que vous avez précédemment créées avec la commande `git branch` et le contenu que vous avez développé indépendamment sur cette branche en une seule branche.

•Git diff

`git diff` : permet de voir la différence entre un arbre de travail et un fichier d'index.

`git diff --cached` : permet de voir la différence entre le fichier d'index et le commit.

`git diff HEAD` : permet de voir la différence entre l'arbre de travail et le commit.

•Git blame

`git blame` est utilisé pour retracer l'historique des modifications apportées à un fichier donné