

# 1 Base

```
— export $MES_REPOS='CEQUEVOUSVOULEZ'  
— mkdir $MES_REPOS  
— cd $MES_REPOS  
— git clone https://github.com/git/git.git > /dev/null 2>/dev/null \&  
— mkdir repos_1  
— cd repos_1  
— git init  
— ls -al  
— git -{}-help
```

## 1.1 config

```
— git config -{}-global -{}-list  
— git config -{}-global -{}-add user.name "VOTRE NOM"  
— git config -{}-global -{}-add user.email VOTRE_EMAIL  
— git config -{}-global -{}-add color.ui auto  
— git config -{}-global -{}-add core.editor /usr/local/bin/vim  
— git config -{}-global -{}-list
```

## 1.2 add

```
— cd $MES_REPOS/repos_1  
— git status  
— git add *  
— git commit  
— vi/emacs unfichier  
— git status  
— git add unfichier  
— git commit  
— Faire plusieurs fois les vi, git add, git commit
```

## 1.3 Staged

```
— vi/emacs unfichier  
— git status  
— git add unfichier  
— vi/emacs unfichier  
— git status  
— git commit  
— git status  
— git add unfichier  
— git commit
```

## 1.4 Ajout fichiers partiel

- vi/emacs unfichier
- Faire  $\geq 2$  groupes de modifications
- git status
- git add -p unfichier
- git commit
- git status
- git add -p unfichier
- git commit
- git status

## 1.5 amend

- vi/emacs unfichier
- git add; git commit
- git log
- git commit --amend
- git log
- Pour faire plus git rebase

## 1.6 diff

- vi/emacs unfichier
- git diff
- git add unfichier
- git diff -{}-staged (ou git diff -{}-cached)

## 1.7 logs

- cd \$MES\_REPOS/git
- git log
- git log -{}-oneline
- git log -p
- git log -p XXXX...XXXX
- git log -{}-author="Hamano"
- git log -{}-grep="forget"
- git log -S"function"
- git diff XXXX XXXX
- git diff XXXX XXXX -{}- FICHIER

## 1.8 divers

- cd \$MES\_REPOS/git
- git blame utf8.c
- tig ou gitk
- more .gitignore

- `git checkout HEAD~3`
- `git checkout master`

## 1.9 Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log -{}-oneline`
- `git checkout XXXXX`
- `git checkout master`
- `git checkout XXXXX FICHIER`
- `git reset HEAD -{}- FICHIER`
- `git checkout -{}- FICHIER`

## 1.10 revert

- `cd $MES_REPOS/repos_1`
- `git log -{}-oneline`
- Annuler **un** commit `git revert XXXXX`.
- Revenir à une version `git revert HEAD...XXXX`.
- Option `git revert --no-edit HEAD...XXXX` :
- `git log`

## 1.11 reset

- `cd $MES_REPOS/repos_1`
- `git log -{}-oneline`
- `git reset XXXXX`
- `git log`
- Ne pas faire sur des données pusher.

# 2 À plusieurs

## 2.1 remote

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull`
- `git log`
- Faites quelques modifs dans `repos_2` et commiter
- `git remote -v`

## 2.2 remote add

```
— cd $MES_REPOS/repos_1
— git remote add repos2 $MES_REPOS/repos_2
— git pull
— git pull repos2 master
— Faites quelques modifs dans repos_2 et commiter
— cd $MES_REPOS/repos_1
— git pull repos2 master
```

## 2.3 remote add

```
— cd $MES_REPOS/repos_1
— git remote add origin git@gitlab.obspm.fr:jas/test.git
— git push -u origin --all
— git push -u origin --tags
— cd $MES_REPOS/repos_2
— git remote add repos1 $MES_REPOS/repos_1
— git remote add distant git@gitlab.obspm.fr:jas/test.git
— git remote -v
— Faites quelques modifs dans repos_2 et commiter
— git push distant
— git push repos1
— cd $MES_REPOS/repos_1
— git pull repos2 master
```

## 2.4 conflits

```
— cd $MES_REPOS/repos_1
— git pull repos2 master
— cd $MES_REPOS/repos_2
— git pull repos1 master
— Faites quelques modifs dans repos_1 et commiter
— Faites similaires modifs dans repos_2 et commiter
— cd $MES_REPOS/repos_2
— git pull repos1 master
— git status
— Résoudre les conflits
— git add FICHIER
— git commit
— git status
```

## 3 Branches

```
— cd $MES_REPOS/repos_1
— git branch
```

```

— git checkout -b essai
— git branch
— Faites des modifs dans et commiter plusieurs fois
— git checkout master
— git merge essai
— git log -{}-oneline --graph --decorate
— Utiliser soit tig soit gitk pour visualiser les branches
— cd $MES_REPOS/repos_1
— git branch
— git checkout essai
— Faites des modifs et commiter plusieurs fois
— git checkout master
— Faites des modifs dans un fichier différent et commiter plusieurs fois
— git merge essai
— git log -{}-oneline --graph --decorate
— Utiliser soit tig soit gitk pour visualiser les branches

```

## 4 Les branches et les remotes

```

— cd $MES_REPOS/repos_2
— git fetch
— git branch
— git checkout -b monessai origin/essai
— cd $MES_REPOS/repos_1
— git checkout essai
— Faites des modifs et commiter plusieurs fois
— cd $MES_REPOS/repos_2
— git pull
— Faites des modifs et commiter plusieurs fois
— cd $MES_REPOS/repos_1
— git branch -a
— git fetch repos2
— git branch -a
— git checkout essai
— git branch -{}-set-upstream-to=repos2/monessai essai
— git pull
— git log -{}-oneline -{}-decorate -{}-graph ou tig ou gitk

```

## 5 Tout droit ou pas (Fast forward)

```

— cd $MES_REPOS/repos_1
— git pull repos2 master
— cd $MES_REPOS/repos_2
— git pull repos1 master

```

- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches
- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches
- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull --rebase repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches

## 6 Historique

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter  $\geq 3$  fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`
- `git merge nettoyage`
- `git checkout master`
- `git merge essai`
- `git log`