

## 1. Commandes de GIT pour le mode distant

Commandes GIT (mode distant)	Utilités
<b>git init --bare</b>	Initialisation d'un nouveau référentiel vide de type «nu» ou «serveur». (à alimenter par un push depuis un projet originel)
<b>git clone --bare url_repo_existant</b>	Idem mais via un clonage d'un référentiel existant
<b>git clone url_repo_sur_serveur.git</b>	Création d'une copie du projet sur un poste de développement (c'est à ce moment qu'est mémorisée l'url du référentiel « serveur » pour les futurs push et pull)
<b>git pull</b>	Rapatrie les dernières mises à jour du serveur distant (de référence) vers le référentiel local. (NB: <i>git pull</i> revient à déclencher les deux sous commandes <i>git fetch</i> et <i>git merge</i> )
<b>git push</b>	Envoie les dernières mises à jour vers le serveur distant (de référence) <b><i>Attention: le push est irréversible et personne ne doit avoir effectuer un push depuis votre dernier pull !</i></b>
...	

### Exemples:

#script de création d'un nouveau référentiel GIT (coté serveur) dans /var/scm/git ou ailleurs:

```
mkdir p0.git
cd p0.git
git init --bare
git update-server-info
mv hooks/post-update.sample hooks/post-update

#nb www-data est le groupe de apache2
cd ..
sudo chgrp -R www-data p0.git

# ce repository initial et vide pourra être alimenté par un push depuis un projet "original"
# depuis ce projet original , on pourra lancer git config remote.p0.url http://localhost/git/p0.git
#                               puis git push p0 master
echo "fin ?"; read fin
```

ou bien

```
# construira p1.git
git clone --bare file:///home/formation/Bureau/tp/tmp-test-git/original/p1
cd p1.git
git update-server-info
```

## 1. Commandes de GIT pour le mode distant

#récupération d'une copie du projet sur un poste de développement

```
git clone http://localhost/git/p1.git
```

#pull from serv:

```
cd p1  
git pull
```

#push to serv:

```
cd p1  
git push
```

## 2. Gestion des branches avec GIT

Tout projet commence avec une seule branche «**master**» .

Commandes GIT (branches)	Utilités
<b>git branch</b>	Affiche la liste des branches et précise la branche courante (*) .
<b>git branch</b> <i>nomNouvelleBranche</i>	Créer une nouvelle branche (qui n'est pas automatiquement la courante)
<b>git checkout</b> <i>nomBrancheExistante</i>	Changement de branche (avec mise à jour « checkout » des fichiers pour refléter le changement de branche) .
<b>git checkout master</b> <b>git merge</b> <i>autreBranche_a_fusionner</i>	Modifie la branche courante (ici «master») en fusionnant le contenu d'une autre branche
<b>git branch -d</b> <i>ancienneBrancheAsupprimer</i>	Supprime une ancienne branche (avec -d : vérification préalable fusion, avec -D : pas de verif , pour forcer la perte d'une branche morte)
...	