

```
cd shell00
mkdir ex05
cd ex05
```

```
// A EFFECTUER SEULEMENT SI VOUS N'ETES PAS DEJA DANS UN REPO GIT
```

```
git init          // transforme le répertoire courant en répertoire git (git repository)
```

```
// A EFFECTUER SEULEMENT SI VOUS N'AVEZ PAS DEJA EFFECTUE 5 COMMITS
```

```
echo "1" > test.txt    // écrit 1 dans le fichier test.txt
git add .              // ajoute tous les fichiers modifiés (ou créés)
                        // dans le repo au commit
                        // pour préciser un fichier en particulier
                        // (le fichier test.txt par exemple), utiliser git add test.txt
git commit -m "1"      // crée un commit avec comme message descriptif "1"
                        // (pour expliquer les modifications apportées dans ce commit)
```

```
echo "2" > test.txt
git add .
git commit -m "2"
```

```
echo "3" > test.txt
git add .
git commit -m "3"
```

```
echo "4" > test.txt
git add .
git commit -m "4"
```

```
echo "5" > test.txt
git add .
git commit -m "5"
```

```
// FIN DES 5 COMMITS
```

```
git log --pretty=format:%H -n 5
```

```
// git log affiche l'historique des commits avec le hash du commit
// (l'identifiant du commit, représentant son empreinte)
// mais aussi
// l'auteur et la date à laquelle le commit a été effectué
// ainsi que le message du commit
// (expliquant les modifications apportées dans ce commit)
// pour indiquer seulement le hash des commits,
// on utilise l'option --pretty=format%H
// (pour agir sur le format de sortie du log)
// -n 5 indique de limiter le résultat aux 5 derniers commits
```

```
vim git_commit.sh // crée le fichier git_commit.sh et l'ouvre avec vim
// l'extension .sh indique qu'il s'agit d'un script shell
// (fichier texte contenant une séquence de commandes
// interprétables par un interpréteur de shell)
```

```
<i>
```

```
// indique le shell à utiliser pour exécuter le script
// (voir page 2 du sujet :
// les exercices en Shell doivent être exécutables par /bin/sh)
```

```
#!/bin/sh
```

```
git log --pretty=format:%H -n 5;
```

```
echo "";
```

```
// pour ajouter un saut de ligne après le dernier hash affiché
```

```
<echap>
```

```
:wq
```

```
<enter>
```

```
chmod +x git_commit.sh // rends le script exécutable (donne le droit d'exécution (x)
// au propriétaire, au groupe et aux autres)
```

```
bash git_commit.sh | cat -e // exécute le script shell git_commit.sh
// avec le shell et redirige la sortie
// de la commande bash git_commit.sh, vers l'entrée
// de la commande cat -e (son argument)
```

```
// cat -e : l'option -e de cat permet d'afficher le contenu du résultat  
// en montrant les caractères de fin de ligne (EOL, End Of Line) :  
// chaque fin de ligne est alors représentée par le symbole $
```

```
// N'OUBLIEZ PAS DE SUPPRIMER LE FICHIER test.txt SI VOUS L'AVEZ CREE !
```

```
rm test.txt
```