

```
// PLACEZ-VOUS A LA RACINE DE VOTRE DEPOT GIT !
```

```
echo ".DS_Store" > .gitignore
```

```
// le fichier .gitignore, placé à la racine d'un répertoire git,  
// doit contenir la liste des fichiers qui doivent être ignorés par git :  
// ils ne seront donc pas ajoutés aux commits de ce répertoire, même avec git add .  
// (qui ajoute normalement tous les fichiers du répertoire courant  
// dans le prochain commit)
```

```
// le fichier .DS_Store est un fichier spécifique à macOS :  
// c'est un fichier caché qui stocke des informations sur des paramètres  
// spécifiques au répertoire (disposition des icônes, taille des fenêtres, etc)  
// un tel fichier doit donc être ignoré par git
```

```
// ATTENTION ! EXECUTEZ LA COMMANDE ls -a POUR AFFICHER LE FICHIER .gitignore !  
// (-a = all) : n'ignore pas les fichiers commençant par .
```

```
echo "*~" >> .gitignore // exclus tous les fichiers se terminant par ~  
// (* = wildcard) : remplace n'importe quelle suite de caractères  
// Remarque : on utilise >> pour écrire à la suite du fichier .gitignore,  
// sur une nouvelle ligne  
// cela permet de ne pas écraser ce qu'il contenait déjà !
```

```
// un fichier se terminant par ~ est un fichier de sauvegarde automatique  
// (fichier temporaire), pour prévenir la perte de données en cas de problème  
// il porte le même nom que le fichier original, suivi de ~  
// un tel fichier doit donc aussi être ignorés par git
```

```
cd shell00  
mkdir ex06  
cd ex06
```

```
// DEBUT TEST .gitignore
```

```
touch .DS_Store           // crée le fichier .DS_Store  
touch mywork.c            // crée le fichier mywork.c  
touch mywork.c~           // crée le fichier mywork.c~
```

```
git add .                // on ajoute tous les nouveaux fichiers du repo
                        // dans le commit (sauf les fichiers indiqués dans .gitignore)
                        // ils seront ignorés automatiquement
git commit -m "gitignore" // on commite
                        // résultat :
                        // 2 files changed, 2 insertions(+)
                        // create mode 100644 .gitignore
                        // create mode 100644 mywork.c

git push

                        // on vérifie sur le repo (sur github) que les fichiers
                        // .DS_Store et test.c~ n'ont pas été inclus

                        // FIN TEST .gitignore

git ls-files -io --exclude-standard

                        // liste tous les fichiers ignorés par git (dans le .gitignore)
                        // mais présents dans le repo

                        // ls-files indique à git d'afficher les fichiers suivis par git
                        // (suivis pour leur modification ou leur suppression)

                        // (-i = ignored) : indique de montrer seulement les fichiers ignorés
                        // (dans le .gitignore entre autres)
                        // cette option doit être utilisée avec -o ou -c
                        // (-o = others) : indique d'afficher aussi les fichiers non suivis
                        // par git
                        // lorsque l'option -o est indiquée,
                        // une option --exclude* doit aussi être indiquée
                        // (--exclude-standard) : ajoute les exclusions standard de Git
                        // c'est-à-dire (entre autres) les exclusions de .gitignore

                        // résultat :
                        // .DS_Store
                        // mywork.c~
```

```
vim git_ignore.sh
<i>
#!/bin/sh
git ls-files -io --exclude-standard
<echap>
:wq
<enter>
```

```
chmod +x git_ignore.sh
bash git_ignore.sh | cat -e
```

```
// résultat :
// .DS_Store$
// mywork.c~$
```

```
rm .DS_Store mywork.c mywork.c~
```

```
// ASTUCES :
// man git
// man git ls-files
```