# Git-HOL-Solutions2

## Objective

Use .gitignore to ignore unwanted files and folders (e.g., \*.log files and log/ folders).

## Prerequisites (quick check)

1) Verify Git is installed:  
 git --version  
2) Configure your username and email (if not already):  
 git config --global user.name "Your Name"  
 git config --global user.email "your.email@example.com"  
3) (Optional) Notepad++ as editor:  
 git config --global core.editor "notepad++ -multiInst -notabbar -nosession -noPlugin"  
4) Ensure you have a local repo cloned or initialized, and a remote set (GitLab/GitHub).

## Step 1: Prepare a sample repo (if needed)

mkdir GitIgnoreDemo  
cd GitIgnoreDemo  
git init  
echo "# Demo for .gitignore" > README.md  
git add README.md  
git commit -m "Initial commit"

## Step 2: Create files/folders to be ignored

# Create log files and folders  
echo "some logs" > app.log  
mkdir -p log logs/app  
echo "more logs" > log/run.log  
echo "more logs" > logs/app/service.log  
echo "keep me" > keep.log  
echo "a normal file" > notes.txt  
  
# See what Git sees right now  
git status

## Step 3: Add rules to .gitignore

.gitignore content (create this file in the repo root):  
  
# Ignore all .log files  
\*.log  
  
# Ignore common log directories  
log/  
logs/  
  
# If you want to keep a specific .log file tracked, unignore it (place this AFTER the ignore rules)  
!keep.log

# Add and commit the .gitignore  
git add .gitignore  
git commit -m "Add .gitignore for \*.log and log folders"  
  
# Verify status now (log files/folders should NOT appear as untracked)  
git status

## Step 4: Fix already-tracked files (if needed)

If you accidentally committed log files before adding .gitignore, remove them from the index (keep them on disk):  
  
git rm -r --cached log logs \*.log  
git commit -m "Stop tracking log files now ignored by .gitignore"  
  
# Re-check  
git status

## Step 5: Push to remote

# Add your remote if not present  
git remote add origin <REMOTE\_URL>  
  
# Pull latest (resolve conflicts if any)  
git pull origin master  
  
# Push your commit  
git push origin master

## Verification Checklist

✓ git status shows README.md, .gitignore, and other tracked files, but NOT \*.log nor log/ or logs/ contents  
✓ Committing does not include \*.log files or log folders  
✓ Pushing reflects .gitignore rules on the remote

## Extras (optional but useful)

Ignore only within this repo:  
 - Use .gitignore (project-scoped)  
  
Ignore for all repos on your machine (global):  
 git config --global core.excludesFile ~/.gitignore\_global  
 echo "\*.log" >> ~/.gitignore\_global  
  
Ignore locally without sharing rules (per-repo, not committed):  
 - Use .git/info/exclude