TechWayFit: Git Quick Reference

Essential Git Commands

git init

Initialize a new Git repository in your project folder.

git clone <url>

Download a project and its entire version history.

git status

Check the status of changes as untracked, modified, or staged.

git add <file>

Add a file to the staging area.

git commit -m 'message'

Commit changes with a message.

git log

View commit history.

git branch

List all branches in your repository.

git checkout <branch>

Switch to a different branch.

git checkout -b <branch>

Create and switch to a new branch.

git merge <branch>

Merge the specified branchs history into the current branch.

git pull

Fetch from the remote repository and merge.

git push

Push your commits to the remote repository.

git reset --hard <commit>

Reset your repository to a specific commit (dangerous).

git revert <commit>

Create a new commit that undoes changes from a previous commit.

git stash

Temporarily save changes that you dont want to commit immediately.

Best Practices for Git Usage

- Commit frequently with clear, concise messages.
- Use branches to isolate features, fixes, or experiments.
- Avoid committing sensitive data like passwords or API keys.
- Always pull the latest changes before pushing to avoid conflicts.
- Review code via pull requests to maintain quality.
- Use `.gitignore` to exclude unnecessary files from version control.
- Regularly prune merged or stale branches.
- Keep commit history clean using rebase (if appropriate).

Recommended Git Tools for Developers

Windows

- Git Bash
- GitHub Desktop
- TortoiseGit
- Sourcetree

Mac

- Git (via Homebrew)
- GitHub Desktop
- Tower
- Sourcetree

Linux

- Git CLI (preinstalled or apt/yum)
- GitKraken
- SmartGit