

SIMPLE GIT CHEAT SHEET

Create a repository

```
git init
```

```
git add .
```

Copy an existing repo

```
git clone user@host:dir/<repo>.git
```

Status of current repo

```
git status
```

Changes to tracked files

```
git diff
```

What you are about to commit

```
git diff --cached
```

History

```
git log
```

Fetch (from remote to local, not merging)

```
git fetch
```

Pull (from remote to local, and merging)

```
git pull
```

Add a file to the staging area

```
git add <file>
```

Add all files and sub directories to the staging area

```
git add .
```

Remove file from the staging area

```
git rm <file>
```

Commit to local repo

```
git commit -m "<your_message>"
```

Push to remote repo

```
git push <remote_alias> <branch>  
(typically git push origin master)
```

Revert the current changes

```
git checkout -f
```

Reset current changes (wipes out staging area)

```
git reset --hard
```

Revert the last commit

```
git revert HEAD
```

Revert specific commit

```
git revert <commit_id>
```

Stash changes

```
git stash <optional_id>
```

Unstash last stashed changes

```
git stash pop
```

Unstash changes

```
git stash apply stash@{0}
```

List stashed changes

```
git stash list
```

List all branches (even remote ones)

```
git branch -a
```

Switch to a branch

```
git checkout <branch>
```

Create a new branch from current branch

```
git checkout -b <branch>
```

Merge branch b1 into branch b2

```
git checkout b2
```

```
git merge b1
```

Delete a branch

```
git branch -d <branch>
```

Rebase current branch

```
git rebase <remote_alias>
```

If conflicts during rebase, fix the conflicts and then:

```
git rebase --continue
```

Abort a rebase because of conflicts

```
git rebase --abort
```