



GIT CHEAT-SHEET

SETUP / CONFIGURE

Git config --global user.name "[firstname lastname]"
(set a name that is identifiable for credit when review version history)

Git config --global user.email "[valid-email]"
(set an email address that will be associated with each history maker)

Git config --global color.ui auto
(set automatic command line coloring for git for easy reviewing)

REWRITE HISTORY

Git rebase [branch]
(apply any commits of current branch ahead of specified one)

Git reset --hard [branch]
(clear staging area,rewrite working tree from specified commit)

REMOTE UPDATE

List Remotes

Git remote -v

Show Information

Git remote show remote

Add Remote

Git remote add path/url

Fetch Changes

Git fetch remote

Fetch + Merge

Git pull remote branch

Publish Local To Remote

Git push remote branch

Delete Remote Branch

Git push origin -d branch-name

Publish Tags

Git push origin/upstream --tags

TEMPORARY COMMITS

git stash
(save modified and staged changes)

git stash list
(list stack-order of stashed file changes)

git stash pop
(write working from top of stash stack)

git stash drop
(discard the changes from top of stash stack)

CREATE GIT

From Existing Directory

cd proj_dir
(change to project dir)

git init
(create new git repo)

git add .
(add everything in repo to staging area)

From Other Repository

git clone existing_dir new_dir

git clone [git repo url]
(git clone <https://github.com/user/repo.git>)

MERGE/REBASE

Merge Branch Into Current

git merge branch

Rebase Into Branch

git rebase branch
git rebase master branch

Abort rebase

git rebase --abort

Merge tool to resolve Conflicts

git mergetool

To View The Merge Conflicts

git diff
(complete conflict diff)

git diff --base \$file
(against base file)

git diff --ours \$file
(against your changes)

git diff --theirs \$file
(against others changes)

To Discard Conflicting Patch

git reset --hard
git rebase --skip

After Resolving Conflicts

git add \$conflicting_file
(do for all resolved file)
git rebase --continue

BRANCHING

List Branches

git branch

Switch To Branch

git checkout branch

Create New Branch

git branch new

Create Branch From Existing

git branch new existing

Delete Branch

git branch -D branch

Tag Current Commit

git tag tag name

Create branch and Switch

git checkout -b branch

LOCAL CHANGES

Changed in Working Directory

Git status

Tracked File Changes

Git diff

Add Changed Files

Git add file1 file2 file3

Remove File

Git rm file
Git rm dir/ -r
(recursive under directory)

Move File

Git mv [existing-path] [new-path]

See files ready for commit

Git status
Git diff --cached

Commit Changes

Git commit
Git commit -m "My Message"
Git commit -a -m "Message"
(tracked files only,auto add)

Change Last Commit

Git commit --amend

Revert Changes To File

Git revert HEAD

Return To Last Committed Sta

Git reset --hard HEAD