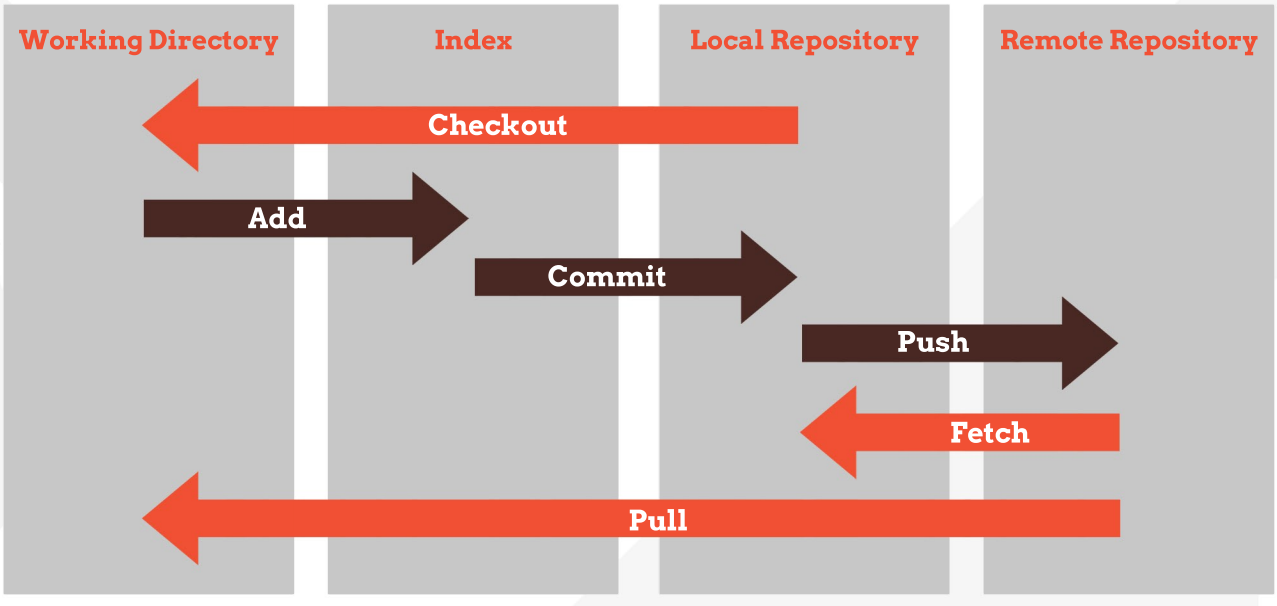
git Cheat Sheet

presented by clearvision

Infrastructure



References

HEAD
Current working position, as shown by the contents of your working directory.

Master
Default branch.

Origin
Default upstream repository.

Origin/Master
Remote branch tracking the location of the master branch in the origin repository.

HEAD^ or HEAD~n
An ancestor of the current head where ^ = 1 or n = any number above 1.

Create

git init <path>
Turns the directory at <path> into a git repository, even if not empty. If <path> does not exist, this command will create it.

git clone <../existing/repo/path> <../new/repo/path>
Copies the repository at <../existing/repo/path> to <../new/repo/path> and records <../existing/repo/path> as the origin

Change

git add <file>
Stages a change to be included in the next commit.

git add * or git add .
Stages all changes.

git add -u .
Stages all local modifications, (recursively from the current directory, does not work on new (untracked) files.

git add -u :/
Stages all local modifications from the repository root, does not work on new (untracked) files.

git add -A
Stages all local modificaions.

git commit
Converts staged changes into binary objects and puts a new commit at the top of current branch.

git commit -m <message>
Converts staged changes into binary objects and puts a new commit at the top of current branch with comment <message>.

git commit -a
Stages all local modifications and commits (does not work on new files).

git commit <file(s)>
Creates a new commit ignoring all changes other than <file(s)>.

git rm <file>
Deletes <file> and stages the change.

git mv <file> <newfile>
Move or rename <file> to <newfile>.

git stash save/apply
Save or re-apply local modifications to / from a stash.

Fixing Errors

git commit --amend
Creates a new commit in place of the current HEAD (correct the staging area first).

git reset <file>
Unstage <file>.

git reset --hard
Resets all modifications back to the HEAD, this cannot be undone unless you Stash first.

git revert <SHA-1>
Revert the delta of <SHA-1> creating a new commit at the top of current branch.

git checkout <SHA-1> <file>
recover <file> from specific commit <SHA-1>.

Log

git log (-n)
Shows the history of the current branch to the (nth) ancestor.

git log -p <file>
Shows the history of <file>

git show <SHA-1>
Show details of the object with the id <SHA-1>.

git show <SHA-1>:<file>
Show details of <file> referenced by commit <SHA-1>

git blame <file>
Show details of who changed <file> and when on a line by line basis.

Diff

git status
Show any changes in the working directory and/or index.

git diff
Show changes to unstaged files.

git diff --cached
Show changes to staged files.

git diff <SHA-1>
Show changes since <SHA-1>, this can also be HEAD or <branch>.

git diff <SHA-1><SHA-1>
Compare two commits.

Branching & Merging

git checkout <SHA-1>
Switch to the <SHA-1> or replace with a <branch>.

git merge <branch>
Merge <branch> into current branch.

git branch <branch>
Create branch named <branch> based on the HEAD.

git checkout -b <branch>
Create branch <branch> based on <SHA-1> and switch to it.

git branch -d
Delete branch <branch>.

Updates

git fetch <remote>
Recover remote changes to remote branches <remote>.

git pull <remote>
Recover remote changes to remote branches from <remote> and merge to local versions.

git push <remote> <branch>
send local changes up to remote server <remote> <branch>.