Git Version control

CONCEPTS

Run git status often: it shows what changed and what to do next.

Working tree your files
↓ `git add`
Staging area next commit
↓ `git commit`
Local repo history
↓ `git push`
Remote origin

A branch is just a movable pointer to a commit, so branching is cheap.

SETUP

`git init`
Create a repository here

`git clone <url>`
Copy a remote repository

`git config --global user.name "<name>"`
Set your name

`git config --global user.email "<email>"`
Set your email

SNAPSHOTS

`git status`
Show changed and staged files

`git add <file>`
Stage a file

`git add .`
Stage everything here

`git commit -m "<msg>"`
Commit staged changes

`git diff`
Show unstaged changes

`git diff --staged`
Show staged changes

`git add -p`
Stage changes hunk by hunk

`git restore <file>`
Discard changes in a file

`git restore --staged <file>`
Unstage a file

SNAPSHOTS (cont.)

`git commit --amend`
Rewrite the last commit

HISTORY

`git log --oneline --graph`
Compact branch history

`git show <commit>`
Show one commit

`git blame <file>`
Who changed each line

BRANCHES

`git branch`
List local branches

`git switch <branch>`
Switch branch

`git switch -c <branch>`
Create and switch

`git merge <branch>`
Merge branch into current

REMOTES

`git pull`
Fetch and integrate changes

`git push`
Upload commits

`git fetch`
Download without merging

`git push -u origin <branch>`
Push and track a new branch

UNDO

`git revert <commit>`
New commit that undoes one

What reset moves

reset	HEAD	Index	Files
`--soft`	moves	kept	kept
`--mixed`	moves	reset	kept
`--hard`	moves	reset	replaced

STASH & TAGS

`git stash`
Shelve local changes

`git stash pop`
Restore the latest stash