GIT Cheat Sheet

Create Repositories

`$ git init <project-name>`
Creates a new local repo named ‹project-name›

`$ git clone <url>`
Downloads a repo and its entire version history

`$ git flow init`
Init git-flow in current repo. Needs install.

Managing changes

`$ git status`
Lists all new or modified files and their status

`$ git add <file>`
Stage the current file content for commiting.

`$ git reset <file>`
Unstage ‹file›, keeping its contents.

`$ git diff <non-staged-file>`
Shows differences from last version to new changes.

`$ git commit -m "<descriptive message>"`
Records file snapshots permanently in version history

Branches

`$ git branch <name>`
Create a new branch.

`$ git checkout <name>`
Switch to the specified branch.

`$ git merge <name>`
Merge the specified branch into the current branch.

`$ git branch -d <name>`
Delete branch.

Stash

`$ git stash`
Stash all modified versioned files. Temporarily!

`$ git stash pop`
Restore the most recently stashed files.

`$ git stash drop`
Discard the most recently stashed files.

History

`$ git log --follow <file>`
Lists version history for a file.

`$ git log --graph --oneline --decorate --all`
Fancy version history.

Oops!

`$ git commit --amend ...`
Don't create a new commit, change the previous one.

`$ git rebase -i <commit>`
Open interactive rebase from current point to specified commit.

`$ git reset <commit>`
Undo changes made after ‹commit›, preserve changes.

`$ git reset --hard`
Undo and discard all changes made after ‹commit›.

Remote sync

`$ git fetch`
Download all new items from repo.

`$ git push -u origin <branch>`
Push branch to remote repo and sync.

`$ git pull`
Update current branch to include remote changes.