Common Git Commands Cheat Sheet

getting stuff from GitHub

Command	Purpose
git clone https://github.com/...	make a local copy of a repo
git fetch	fetch any changes on GitHub
git status	describe any local vs. GitHub diffences
git pull	pull changes to local repo copy

recording you changes

Command	Purpose
git add *filename*	add file *filename* to staging area
git add -A	add all changed files to staging area
git commit -m “message”	commit changes with message “message”
git commit -a -m “message”	stage and commit *all* changes with message “message”
git push	push your changes up to GitHub

danger zone

Command	Purpose
git rm *filename*	remove (delete) file *filename*
git rm -r *directory name*	remove (delete) directory and all contents

working with branches

Command	Purpose
git branch	list local branches (* denotes active branch)
git branch -a	last all local *and* remote branches
git branch “branchname”	make a new branch called “branchname”
git checkout “branchname”	switch to branch “branchname”
git checkout -b “branchname”	create new branch and switch to it in one go

Command	Purpose
git merge “branchname”	merge “branchname” into current branch
git merge “sourcebranch” “targetbranch”	merge “sourcebranch” into “targetbranch”

Typical workflow

Open a terminal in the local repo folder

Use git to git - er, “get” - current GitHub version

`git fetch` - fetch any changes on GitHub
`git status` - look at any changes
`git pull` - pull (only if there are changes)

Edit files (usually code)

Now get GitHub current with your local changes

`git add -A` - stage all changes
`git commit -m "commit message"` - commit the changes to your local repo
`git push` - send them up to GitHub

or

`git commit -a -m "commit message"` - stage & commit the changes to your local repo
`git push` - send them up to GitHub