Git Cheat Sheet

Getting Started

`$ git clone [repository]`	Clone an existing repository
`$ git init`	Create a new local respository

Local

`$ git status`	Show changed files in your working directory
`$ git diff`	Show changes to tracked files
`$ git add -A`	Stage all changes
`$ git add -u`	Stage modified and deleted files, but not untracked files
`$ git add .`	Stage new and modified files, but not deleted files
`$ git add [file]`	Stage *[file]*
`$ git commit`	Commit previously staged changes
`$ git commit -m "[message]"`	Commit previously staged changes with a message
`$ git commit -a`	Commit all local changes in tracked files
`$ git commit --amend`	Change the last commit
`$ git log`	Show all commits, starting with most recent
`$ git log -p [file]`	Show changes over time to *[file]*
`$ git blame [file]`	Show changes for *[file]* with time and author

Branches & Tags

`$ git branch`	List local branches
`$ git branch -a`	List both local and remote branches
`$ git branch -r`	List remote branches
`$ git checkout [branch]`	Switch HEAD branch

`$ git branch [new branch]`	Create new branch based on current HEAD
`$ git branch --track [new branch] [remote branch]`	Create new tracking branch based on a remote branch
`$ git branch -d [branch]`	Delete local branch
`$ git tag [tag name]`	Mark the current commit with a tag

Remote

`$ git remote update`	Refresh list of remotes
`$ git remote -v`	List all currently configured remotes
`$ git remote show [remote]`	Display information about *[remote]*
`$ git remote add [remote] [url]`	Add a new remote, named *[remote]*
`$ git fetch [remote]`	Download all changes from *[remote]*, don't integrate into HEAD
`$ git pull [remote] [branch]`	Download changes and merge to HEAD
`$ git push [remote] [branch]`	Publish local changes
`$ git push [remote] : [branch]`	Delete a branch on the remote
`$ git push --tags`	Publish tags

Merge & Rebase

`$ git merge [branch]`	Merge *[branch]* into your current HEAD
`$ git rebase [branch]`	Rebase your HEAD onto *[branch]*
`$ git rebase --abort`	Abort a rebase
`$ git rebase --continue`	Continue a rebase after resolving conflicts
`$ git mergetool`	Use configured merge tool to resolve conflicts

Stash

`$ git stash`	Temporarily store all modified tracked files
`$ git stash pop`	Restore most recently stashed files
`$ git stash list`	List all stashed changes
`$ git stash drop`	Discard most recently stashed changeset

Reset & Revert

`$ git reset --hard HEAD`	Discard all local changes in your working directory
`$ git checkout HEAD [file]`	Discard local changes in *[file]*
`$ git revert [commit]`	Revert a commit
`$ git reset --hard [commit]`	Reset HEAD to a previous commit and discard all changes since
`$ git reset [commit]`	Reset HEAD to a previous commit and preserve all changes as unstaged changes
`$ git reset --keep [commit]`	Reset HEAD to a previous commit and preserve uncommitted local changes