

# Git Cheat Sheet



## Git Basics

<b>git init</b> <b>&lt;directory&gt;</b>	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.	
<b>git clone &lt;repo&gt;</b>	Clone repo located at <repo> onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.	
<b>git config</b> <b>user.name &lt;name&gt;</b>	Define author name to be used for all commits in current repo. Devs commonly use --global flag to set config options for current user.	+
<b>git add</b> <b>&lt;directory&gt;</b>	Stage all changes in <directory> for the next commit. Replace <directory> with a <file> to change a specific file.	
<b>git commit -m</b> <b>"&lt;message&gt;"</b>	Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.	
<b>git status</b>	List which files are staged, unstaged, and untracked.	
<b>git log</b>	Display the entire commit history using the default format. For customization see additional options.	+
<b>git diff</b>	Show unstaged changes between your index and working directory	+

## Undoing Changes

<b>git revert</b> <b>&lt;commit&gt;</b>	Create new commit that undoes all of the changes made in <commit>, then apply it to the current branch.	
<b>git reset &lt;file&gt;</b>	Remove <file> from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.	+
<b>git clean -n</b>	Shows which files would be removed from working directory. Use the -f flag in place of the -n flag to execute the clean.	

## Rewriting Git History

<b>git commit --amend</b>	Replace the last commit with the staged changes and last commit combined. Use with nothing staged to edit the last commit's message.	
<b>git rebase &lt;base&gt;</b>	Rebase the current branch onto <base>. <base> can be a commit ID, a branch name, a tag, or a relative reference to HEAD.	+
<b>git reflog</b>	Show a log of changes to the local repository's HEAD. Add --relative-date flag to show date info or --all to show all refs.	

## Git Branches

<b>git branch</b>	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.	
<b>git checkout -b</b> <b>&lt;branch&gt;</b>	Create and check out a new branch named <branch>. Drop the -b flag to checkout an existing branch.	
<b>git merge &lt;branch&gt;</b>	Merge <branch> into the current branch.	

## Remote Repositories

<b>git remote add</b> <b>&lt;name&gt; &lt;url&gt;</b>	Create a new connection to a remote repo. After adding a remote, you can use <name> as a shortcut for <url> in other commands.	
<b>git fetch</b> <b>&lt;remote&gt; &lt;branch&gt;</b>	Fetches a specific <branch>, from the repo. Leave off <branch> to fetch all remote refs.	
<b>git pull &lt;remote&gt;</b>	Fetch the specified remote's copy of current branch and immediately merge it into the local copy.	+
<b>git push &lt;remote&gt;</b> <b>&lt;branch&gt;</b>	Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.	+