

### CONFIGURE TOOLING

<code>git config --global user.name "[name]"</code>	Set the name you want attached to your commit transactions
<code>git config --global user.email "[email address]"</code>	Set the email you want attached to your commit transactions

### CREATE REPOSITORIES

<code>git init</code>	Creates a new local repository with the specified name
<code>git clone [repo url]</code>	Downloads a project and its entire version history

### MAKE CHANGES

<code>git status</code>	List changed files in your working directory
<code>git diff</code>	List changes to tracked files
<code>git diff --staged</code>	List file differences between staging and the last file version
<code>git add .</code>	Add <i>all</i> current changes to the next commit
<code>git add -p [file-name]</code>	Add <i>some</i> changes in [file-name] to the next commit
<code>git commit -m "[commit msg]"</code>	Commit local changes and add msg
<code>git commit -a</code>	Commit all local changes in tracked files
<code>git commit --amend</code>	Change the last commit*
<code>git reset &lt;file&gt;</code>	Unstage the file, but preserve its contents

\* Don't amend published commits!

### GROUP CHANGES

<code>git branch</code>	List all local branches in the current repository
<code>git branch [branch-name]</code>	Create new [branch] based on your current HEAD
<code>git branch -d [branch-name]</code>	Delete specified branch
<code>git checkout [branch-name]</code>	Switch HEAD branch
<code>git merge [branch-name]</code>	Merge [branch-name] into your current HEAD
<code>git tag [tag-name]</code>	Mark the current commit with a tag
<code>cat .git/HEAD</code>	See what HEAD points too

You can think of the HEAD as the "current branch". When you switch branches with `git checkout`, the HEAD revision changes to point to the tip of the new branch.



### SYNCHRONIZE

<code>git remote show [remote-name]</code>	Show information about a remote
<code>git remote -v</code>	List all currently configured remotes
<code>git branch -dr [remote]/[branch]</code>	Delete a branch on the remote
<code>git fetch [remote]</code>	Download all changes from [remote] but do'nt integrate into HEAD
<code>git pull [remote] [branch]</code>	Download changes and directly merge/integrates into HEAD
<code>git push [remote] [branch]</code>	Publish local changes on a remote
<code>git push --tags</code>	Publish your tags
<code>git merge [remote]/[branch]</code>	Combine the remote [branch] into your current HEAD
<code>git pull</code>	Download bookmark history and incorporates changes
<code>git add &lt;resolved-file&gt;</code>	Use your editor to manually solve conflicts and (after resolving) mark file as resolved
<code>git rm &lt;resolved-file&gt;</code>	Use your editor to manually solve conflicts and (after resolving) mark file as resolved

### COMMIT HISTORY

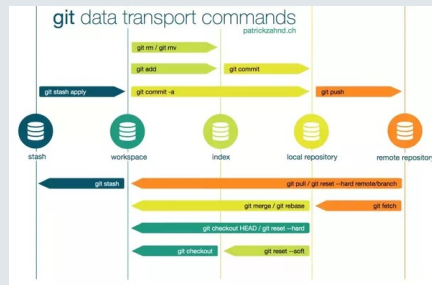
<code>git log</code>	Show all commits, starting with newest
<code>git log -p [file-name]</code>	Show changes over time for a specific file
<code>git blame [file-name]</code>	Who changed what and when in [file-name]
<code>git show [commit-ID]</code>	Output metadata and content changes a specific commit

### UNDO

<code>git reset --hard HEAD</code>	Discard all local changes in your working directory
<code>git checkout HEAD [file]</code>	Discard local changes in a specific file
<code>git revert [commit]</code>	Revert a commit (by producing a new commit with contrary changes)
<code>git reset --hard [commit]</code>	Discard all history and changes back to the specified commit
<code>git reset [commit]</code>	Revert a commit (by producing a new commit with contrary changes) <i>and preserve all changes as unstaged changes</i>
<code>git reset --keep [commit]</code>	Revert a commit (by producing a new commit with contrary changes) <i>and preserve uncommitted local changes</i>



### WORKFLOW



workspace ↔ working directory

index ↔ staging area

local repository ↔ HEAD

### SOURCES

GitHub Git Cheat Sheet

Git Tower

Making Sense of Git – A Visual Perspective

C

By **gsalem**  
[cheatography.com/gsaalem/](https://cheatography.com/gsaalem/)

Not published yet.  
Last updated 15th January, 2018.  
Page 3 of 3.

Sponsored by **Readability-Score.com**  
Measure your website readability!  
<https://readability-score.com>