

QUESTIONS?

GIT AND COMMAND LINE CHEAT SHEETS

Git Documentation



GIT CHEAT SHEET

presented by Tower - the best Git client for Mac and Windows

CREATE	BRANCHES & TAGS	MERGE & REBASE
Clone an existing repository <code>\$ git clone <url></code>	List all existing branches <code>\$ git branch -a</code>	Merge <branch> into your current HEAD <code>\$ git merge <branch></code>
Create a new local repository <code>\$ git init</code>	Switch HEAD branch <code>\$ git checkout <branch></code>	Rebase your current HEAD onto <branch> <i>Don't rebase published commits!</i> <code>\$ git rebase <branch></code>
	Create a new branch based on your current HEAD <code>\$ git branch <new-branch></code>	Abort a rebase <code>\$ git rebase --abort</code>
LOCAL CHANGES	Create a new tracking branch based on a remote branch <code>\$ git checkout -b <new-branch> <remote/></code>	Continue a rebase after resolving conflicts <code>\$ git rebase --continue</code>
Changed files in your working directory <code>\$ git status</code>	Delete a local branch <code>\$ git branch -d <branch></code>	Use your configured merge tool to solve conflicts <code>\$ git mergetool</code>
Changes to tracked files <code>\$ git diff</code>	Mark the current commit with a tag <code>\$ git tag <tag-name></code>	Use your editor to manually solve conflicts and (after resolving) mark file as resolved <code>\$ git add <resolved-file></code>
Add all current changes to the next commit <code>\$ git add .</code>		Use your editor to manually solve conflicts and (after resolving) mark file as resolved <code>\$ git add <resolved-file></code>
Add some changes in <file> to the next commit <code>\$ git add -p <file></code>	UPDATE & PUBLISH	UNDO
Commit all local changes in tracked files <code>\$ git commit -a</code>	List all currently configured remotes <code>\$ git remote -v</code>	Discard all local changes in your working directory <code>\$ git reset --hard HEAD</code>
Commit previously staged changes <code>\$ git commit</code>	Show information about a remote <code>\$ git remote show <remote></code>	Discard local changes in a specific file <code>\$ git checkout HEAD --<file></code>
Change the last commit <i>Don't amend published commits!</i> <code>\$ git commit --amend</code>	Add new remote repository, named <remote> <code>\$ git remote add <remote> <url></code>	Revert a commit (by producing a new commit with contrary changes) <code>\$ git revert <commit></code>
COMMIT HISTORY	Download all changes from <remote>, but don't integrate into HEAD <code>\$ git fetch <remote></code>	Reset your HEAD pointer to a previous commit ...and discard all changes since then <code>\$ git reset --hard <commit></code>
Show all commits, starting with newest <code>\$ git log</code>	Download changes and directly merge/integrate into HEAD <code>\$ git pull <remote> <branch></code>	...and preserve all changes as unstaged changes <code>\$ git reset <commit></code>
Show changes over time for a specific file <code>\$ git log -p <file></code>	Publish local changes on a remote <code>\$ git push <remote> <branch></code>	...and preserve uncommitted local changes <code>\$ git reset --keep <commit></code>
Who changed what and when in <file> <code>\$ git blame <file></code>	Delete a branch on the remote <code>\$ git branch -d <remote/></code>	
	Publish your tags <code>\$ git push --tags</code>	

30-day free trial available at www.git-tower.com

TOWER
The best Git Client for Mac & Windows



COMMAND LINE CHEAT SHEET

presented by Tower - the best Git client for Mac and Windows

DIRECTORIES	FILES	SEARCH
<code>\$ pwd</code> Display path of current working directory	<code>\$ rm <file></code> Delete <file>	<code>\$ find <dir> -name "<file>"</code> Find all files named <file> inside <dir> (use wildcards [?] to search for parts of filenames, e.g. "file.*")
<code>\$ cd <directory></code> Change directory to <directory>	<code>\$ rm -r <directory></code> Delete <directory>	<code>\$ grep "<text>" <file></code> Output all occurrences of <text> inside <files> (add -r for case-insensitivity)
<code>\$ cd ..</code> Navigate to parent directory	<code>\$ rm -f <file></code> Force-delete <file> (add -r to force-delete a directory)	<code>\$ grep -i "<text>" <dir></code> Search for all files containing <text> inside <dir>
<code>\$ ls</code> List directory contents	<code>\$ mv <file-old> <file-new></code> Rename <file-old> to <file-new>	
<code>\$ ls -la</code> List detailed directory contents, including hidden files	<code>\$ mv <file> <directory></code> Move <file> to <directory> (possibly overwriting an existing file)	NETWORK
<code>\$ mkdir <directory></code> Create new directory named <directory>	<code>\$ cp <file> <directory></code> Copy <file> to <directory> (possibly overwriting an existing file)	<code>\$ ping <host></code> Ping <host> and display status
	<code>\$ cp -r <directory1> <directory2></code> Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)	<code>\$ whois <domain></code> Output whois information for <domain>
OUTPUT	<code>\$ touch <files></code> Update file access & modification time (and create <file> if it doesn't exist)	<code>\$ curl -O <url/>files</code> Download <file> (via HTTP[S] or FTP)
<code>\$ cat <file></code> Output the contents of <file>		<code>\$ ssh <username>@<host></code> Establish an SSH connection to <host> with user <username>
<code>\$ less <file></code> Output the contents of <file> using the less command (which supports pagination etc.)		<code>\$ scp <file> <user>@<host>:<remote/path></code> Copy <file> to a remote <host>
<code>\$ head <file></code> Output the first 10 lines of <file>	PERMISSIONS	PROCESSES
<code>\$ <cmd> > <file></code> Direct the output of <cmd> into <file>	<code>\$ chmod <rb> <file></code> Change permissions of <file> to <rb>	<code>\$ ps ax</code> Output currently running processes
<code>\$ <cmd> >> <file></code> Append the output of <cmd> to <file>	<code>\$ chmod -R <rb> <directory></code> Change permissions of <directory> (and its contents) to <rb>	<code>\$ top</code> Display live information about currently running processes
<code>\$ <cmd1> <cmd2></code> Direct the output of <cmd1> to <cmd2>	<code>\$ chown <user>:<group> <file></code> Change ownership of <file> to <user> and <group> (add -R to include a directory's contents)	<code>\$ kill <pid></code> Quit process with ID <pid>
<code>\$ clear</code> Clear the command line window		

30-day free trial available at www.git-tower.com

TOWER
The best Git Client for Mac & Windows