Git Commands

Git Commands (common)

Command	Description
`git help`	Show help information about git commands
`git init`	Init new git project in current folder (folder needs to be empty)
`git clone git@github.com:myusername/myrepositoryname.git`	Clone a git repository from github.com
`git add --all`	Stage all files (or `git add file.txt` for adding a single file)
`git commit -m "my message"`	Create a commit of your stages files (do not forget to stage your files/changes before that, e.g.: `git add --all`)
`git pull`	Pull updates from remote to current branch
`git push`	Push your commits to remote repository
`git push -u origin master`	Push your commits to (new) remote repository and set local branch to track remote branch upstream (-u is equal to --set-upstream). Useful if you don't have any upstream set up yet.
`git push --follow-tags`	Push your commits + tags
`git status`	Show status of unstaged files, project info, branch info, ...

`git branch`	Show all git branches (option -a for showing remotes too)
`git branch mynewbranch`	Create a new branch
`git checkout myotherbranch`	Switch to another git branch
`git log`	Show latest git commits and messages (or "git show")
`git fetch --all`	Fetch from all remote repositories
`git tag -a v0.1.0`	Tag current as "v0.1.0" (or `git tag -am "My Release Message" v0.1.0`). You can use `git push --follow-tags` to push all your tags to the remote.
`git tag --delete v0.1.0`	Remove Tag "v0.1.0"
`git revert commitHash`	Revert changes of a specific commit

Git Commands (advanced)

Command	**Description**
git stash --include-untracked	Save all changes to the stash
git stash --include-untracked -m "stashname"	Save all changes to the stash with a name
git stash list	Show all saved stashes
git stashpop 0	Apply the latest stash + remove it from the stash list
git stash apply 0	Apply the latest stash
git stash drop 0	Remove latest stash from stash list
git stash clear	Remove all stash items
git blame pathtomyfile.txt	Show information who to blame for a file change

git diff	Show changes between two commits
git grep "text-to-search"	Search for a specific string in commits
git format-patch -1 <commitID>	Create a patch of a specific commit
git apply <patchfile>	Apply a patch
git bisect start	Start Bugfinding / Begin using the git bisect command
git bisect good	Identify if a commit is "good", without the bug
git bisect bad	Identify if a commit is "bad", with the bug
git bisect reset	Return to original head
git bisect log	Displays commits you've identified as "good" or "bad"
git bisect visualize	View commits that have not yet been checked
# Set a new remote upstream and push changes/commits git remote add origin git@github.com:username/repository-name.git git push -u origin master	1. Command - Set a remote (if you don't have any) to your local branch 2. Command - Push changes/commits to this new remote branch (-u is equal to --set-upstream)

Revision #27
Created 2023-07-31 19:12:56 UTC by Admin
Updated 2023-08-29 11:01:50 UTC