git_commands_cheat_sheet Cheat Sheet

by __i-Sage__ (__i-Sage__) via cheatography.com/189705/cs/39545/

git commands for version tagging

`git tag`	List all available tags in Git
`git tag -l [tag_n umber]`	search for tags with the particular tag_number
`git tag -a [tag_n umber] -m [message]`	creates an annotated tag with an optional tagging message
`git show [tag_n umber]`	shows the tagger information
`git tag [tag_n umb er]-lw`	creates a lightweight tag
`git tag -a [tag_n umber] [part_ of_ che ck_ sum]`	tags the commit after you've moved passed them
`git push origin [tag_n umber]`	push version_number to shared server after you have created them.
`git push origin --tags`	push tags to shared server
`git checkout -b [branc h_name] [tag_name]`	put a version of the repository in your working directory that looks like a specific tag

By default, the git push command doesn't transfer tags to remote servers. You will have to explicitly push tags to a shared server after you have created them. This process is just like sharing remote branches, you can run git push origin "tag_name"

git commands to clone and create repositories

`git init`	Initializes a new git repository in the current directory
`git clone url`	Clone an existing Git repository located at in the current directory
`git clone url direct ory _na me`	Clone an existing Git repository located at in directory_name

git commands to check status

`git status`	used to determine which files are in which state
`git status -s`	the short and compact representation of git status

git commands for removing and renaming files

`git rm " fil e_n ame "`	Removes the file from your tracked files and the working directory
`git rm --cached " fil e_n am e "`	Removes the files from your tracked files but keeps the file in the working directory
`git mv " fil e_f rom " " fil e_t o"`	Renames the file form "file_from" to "file_to"

git commands for undoing things

`git reset [file_ name]`	used to unstage the "file_name
`git checkout -- [file_ name ]`	reverts a file to the previous version

git commands for staging files

`git add file_name`	Stage file_name for commit
`git add -A`	Stage all modified or new files for commit
`git reset file_name`	Unstage file_name

git commands for working with remotes

`git remote add " nam e" " u rl "`	Create a new remote named "name" for the repository at "url"
`git fetch " rem ote "`	Download all changes from "remote", but don't integrate into current branch
`git merge " rem ote " /"br - anc h"`	Merge "remote]/[branch" into current branch
`git push " rem ote " " bra - nch "`	Push local changes in "branch" to "remote"/"branch"
`git pull " rem ote " " bra - nch "`	Fetch and merge all changes from "remote"/"branch" into current branch

git_commands_cheat_sheet Cheat Sheet

by __i-Sage__ (__i-Sage__) via cheatography.com/189705/cs/39545/

git commands for commiting changes

`git commit -m " mes sag e"`	Commit staged changes with "message" describing changes
`git commit --amend`	Edit the previous commit message or contents
`git commit -a -m " mes sag e"`	Stages all already tracked files and commits them

git commands for branching and merging

`git remote add " nam e" " u rl "`	Create a new remote named "name" for the repository at "url"
`git branch " nam e"`	Create a new branch named "name"
`git checkout " nam e"`	Switch to branch named "-name"
`git merge " bra nch "`	Merge "branch" into current branch
`git merge --no-ff " bra nch "`	Merge "branch" into current branch, keeping branch history

git commands for inspections and differences

`git log`	View the commit history of the current branch
`git log " bra nc h "`	View the commit history of "branch"
`git log --stat`	Prints below each commit entry a list of modified files, how many files where changed, and how many lines in those files where added or removed

git commands for inspections and differences (cont)

`git log --pret ty= online`	This option changes the log output to formats other than the default
`git log --graph`	Display an ASCII graph of the branch and merge history beside the log output
`git diff`	View unstaged changes between the current state of the code and the last commit
`git diff --cached`	View staged changes between last commit and the current state of the code
`git diff " bra nch 1" " bra - nch 2"`	View differences between "branch1" and "branch2"

By __i-Sage__ (__i-Sage__)
cheatography.com/i-sage/

Published 12th July, 2023.
Last updated 13th July, 2023.