

git commands for version tagging

<code>git tag</code>	List all available tags in Git
<code>git tag -l [tag_number]</code>	search for tags with the particular tag_number
<code>git tag -a [tag_number] -m [message]</code>	creates an annotated tag with an optional tagging message
<code>git show [tag_number]</code>	shows the tagger information
<code>git tag [tag_number]-lw</code>	creates a lightweight tag
<code>git tag -a [tag_number] [part_of_checkout_sum]</code>	tags the commit after you've moved passed them
<code>git push origin [tag_number]</code>	push version_number to shared server after you have created them.
<code>git push origin --tags</code>	push tags to shared server
<code>git checkout -b [branch_name] [tag_name]</code>	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

<code>git init</code>	Initializes a new git repository in the current directory
<code>git clone url</code>	Clone an existing Git repository located at in the current directory
<code>git clone url directory_name</code>	Clone an existing Git repository located at in directory_name

git commands to check status

<code>git status</code>	used to determine which files are in which state
<code>git status -s</code>	the short and compact representation of git status

git commands for removing and renaming files

<code>git rm "file_name"</code>	Removes the file from your tracked files and the working directory
<code>git rm --cached "file_name"</code>	Removes the files from your tracked files but keeps the file in the working directory
<code>git mv "file_from" "file_to"</code>	Renames the file from "file_from" to "file_to"

git commands for undoing things

<code>git reset [file_name]</code>	used to unstage the "file_name"
<code>git checkout -- [file_name]</code>	reverts a file to the previous version

git commands for staging files

<code>git add file_name</code>	Stage file_name for commit
<code>git add -A</code>	Stage all modified or new files for commit
<code>git reset file_name</code>	Unstage file_name

git commands for working with remotes

<code>git remote add "name" "url"</code>	Create a new remote named "name" for the repository at "url"
<code>git fetch "remote"</code>	Download all changes from "remote", but don't integrate into current branch
<code>git merge "remote"/"branch"</code>	Merge "remote"/"branch" into current branch
<code>git push "remote" "branch"</code>	Push local changes in "branch" to "remote"/"branch"
<code>git pull "remote" "branch"</code>	Fetch and merge all changes from "remote"/"branch" into current branch

git commands for committing changes

<code>git commit -m "message"</code>	Commit staged changes with "message" describing changes
<code>git commit --amend</code>	Edit the previous commit message or contents
<code>git commit -a -m "message"</code>	Stages all already tracked files and commits them

git commands for branching and merging

<code>git remote add "name" "url"</code>	Create a new remote named "name" for the repository at "url"
<code>git branch "name"</code>	Create a new branch named "name"
<code>git checkout "name"</code>	Switch to branch named "name"
<code>git merge "branch"</code>	Merge "branch" into current branch
<code>git merge --no-ff "branch"</code>	Merge "branch" into current branch, keeping branch history

git commands for inspections and differences

<code>git log</code>	View the commit history of the current branch
<code>git log "branch"</code>	View the commit history of "branch"
<code>git log --stat</code>	Prints below each commit entry a list of modified files, how many files were changed, and how many lines in those files were added or removed

git commands for inspections and differences (cont)

<code>git log --pretty=online</code>	This option changes the log output to formats other than the default
<code>git log --graph</code>	Display an ASCII graph of the branch and merge history beside the log output
<code>git diff</code>	View unstaged changes between the current state of the code and the last commit
<code>git diff --cached</code>	View staged changes between last commit and the current state of the code
<code>git diff "branch1" "branch2"</code>	View differences between "branch1" and "branch2"

