

Browsing the Git History Summary

Commands	Description
<code>git log</code>	all commits in order with all details
<code>git log --oneline</code>	all commits in order but with commit key and only message
<code>git log --oneline --stat</code>	see the changes in the files with each commit in oneline
<code>git log --oneline --patch</code>	see the actual changes in the files in more details
<code>git log --oneline --author="CodeBlessYou"</code>	gets only commits whose author is "CodeBlessYou"
<code>git log --oneline --before="2024-01-22"</code>	gets all commits which committed before "2024-01-22"
<code>git log --oneline --after="yesterday"</code>	gets all commits which committed after yesterday
<code>git log --oneline --grep="update"</code>	gets all commits which commit message has "update" word
<code>git log --oneline -S"tag active"</code>	gets all commits which add or remove "tag active" word
<code>git log --oneline -5</code>	gets recent last 5 commits
<code>git log --oneline file.txt</code>	only shows those commits which added, removed or changed file.txt filea
<code>git log --oneline -- file.txt</code>	does the same as above but with --(double dash) we can separate file names from git command
<code>git log --oneline --patch -- file.txt</code>	shows the actual which part change in all file.txt commits
<code>git config --global alias.lg "log --oneline"</code>	for setting the alias(shortcut) lg which runs this "log —oneline" command
<code>git show 345ff9e</code>	get the full details about '345ff9e' commit
<code>git show HEAD</code>	get the full details about last(HEAD) commit

Commands	Description
<code>git show HEAD~2</code>	get the full details about 3rd last(2 steps above HEAD) commit
<code>git show HEAD~2 --name-only</code>	get the names of only files which changed in this specific commit
<code>git show HEAD~2:card/card.html</code>	get the full content of card.html file in this specific commit
<code>git diff</code>	get the difference between our working stage code and last commit
<code>git diff HEAD~2 HEAD</code>	get the difference by comparing two commits(HEAD~2 and HEAD)
<code>git diff HEAD~2 HEAD index.html</code>	get the difference in file index.html by comparing two commits(HEAD~2 and HEAD)
<code>git checkout ee3460b</code>	restore our working directory code to this commit code
<code>git checkout master</code>	return to our master or last commit code
<code>git bisect start</code>	start the bisect(finding the bug commit) process
<code>git bisect bad</code>	mark commit as the bad commit in bisect process
<code>git bisect good ee3460b</code>	mark known good commit in the bisect process
<code>git bisect reset</code>	remove our bisecting mode to our git bash
<code>git shortlog</code>	get the contributor list with number of commits and commit messages
<code>git shortlog -n</code>	short the list by the number of commits happen by contributor
<code>git shortlog -n -s</code>	do the same as above, just remove the commit messages from the output
<code>git blame src/App.js</code>	get all lines of App.js file with the Commit reference, Author name and date & time
<code>git blame -L 3,5 src/App.js</code>	get only 3, 4 and 5th line with the Commit reference, author name and date & time
<code>git tag</code>	to see tags we added to our project
<code>git tag v1.0 1c8b0a4</code>	to add tag v1.0 to this 1c8b0a4 commit

Commands	Description
<code>git tag -a v1.01 -m "Release version 1.01"</code>	to add annotated tag with message and more details to current commit
<code>git tag -d v1.01</code>	to remove the tag v1.01 to the commit