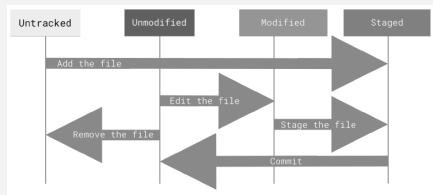


States



Diff

<code>git diff</code>	WORKSPACE vs STAGED
<code>git diff --staged</code> (or <code>--cached</code>)	STAGED vs COMMITTED
<code>git diff HEAD</code>	WORKSPACE + STAGED vs COMMITTED
<code>git diff <SHA-1></code>	WORKSPACE vs any commit (more general case of the above)
<code>git diff master...topic</code>	Changes introduced if topic gets merged into master
<code>-w</code>	IGNORE whitespace

Removal

<code>git rm README</code>	Remove file from git and locally
<code>git rm --cached README</code>	Remove file from git, but keep locally as UNTRACKED
<code>git reset HEAD README</code>	UNSTAGE a file
<code>git checkout -- README</code>	Revert a file back to its form from the last commit

Stashing

<code>git stash</code>	Push all TRACKED modified files (<code>-u</code> to include untracked, <code>-a</code> to include ignored)
<code>git stash list</code>	LIST all stashes
<code>git stash apply</code>	APPLY the last stash
<code>git stash apply stash@{2}</code>	APPLY a chosen stash
<code>git stash pop</code>	APPLY stash and DELETE it
<code>git stash drop</code>	DELETE stash

Remotes

<code>git remote -v</code>	LIST remotes with URLs
<code>git remote show <origin></code>	GET details with tracked branches
<code>git remote add <origin> <url></code>	ADD a remote
<code>git remote rename <origin> <upstream></code>	RENAME remote origin to upstream
<code>git remote rm <origin></code>	DELETE remote
<code>git push <origin> --delete <feature/one></code>	DELETE branch from remote
<code>git fetch origin</code>	FETCH all data from origin
<code>git pull</code>	FETCH and MERGE
<code>git push <origin> <master></code>	PUSH branch to a remote
<code>git branch -vv</code>	Details of local branches and what they track

Tagging

<code>git tag</code>	LIST tags
<code>`-l "119.*"</code>	FILTER tags
<code>git tag -a v2.0 -m "Version 2.0"</code>	CREATE annotated tag
<code>git tag v2.0</code>	CREATE lightweight tag
<code>git show v2.0</code>	GET tag
<code>git tag -d v2.0</code>	DELETE tag
<code>git push <origin> <tag></code>	PUSH single tag
<code>git push <origin> --tags</code>	PUSH all tags

Rebasing

<code>git rebase master</code>	Takes commits from current branch and applies them on top of master. Master stays on previous commit, only current branch is moved
--------------------------------	--

Git Log

<code>-p</code>	See differences in each commit
<code>-2</code>	Only 2 last commits
<code>master feature/one</code>	Only commits between master and feature/one
<code>--all</code>	Commits from all branches
<code>--stat</code>	See file change statistics
<code>--pretty=oneline/short/full- l/fuller/format:{some format}</code>	Formatting of output
<code>--oneline</code>	Shorthand for <code>--pretty= =oneline</code>
<code>--graph</code>	Shows graph
<code>--since=2.weeks/2010-01-15 (or after)</code>	Commits from
<code>--until=2021 (or before)</code>	Commits to
<code>--author=Joe</code>	Only by some author, can be used multiple times
<code>--grep=word</code>	Filter by message content
<code>-S word</code>	Filter by commit that changed occurrence of <code>word</code> in code

Example: ``git log --pretty="%h - %s" --author='Junio C Hamano' --
since="2008-10-01" \
--before="2008-11-01" --no-merges``

--pretty=format options

Option	Description of Output
<code>%H</code>	Commit hash
<code>%h</code>	Abbreviated commit hash
<code>%T</code>	Tree hash
<code>%t</code>	Abbreviated tree hash
<code>%P</code>	Parent hashes
<code>%p</code>	Abbreviated parent hashes
<code>%an</code>	Author name
<code>%ae</code>	Author email
<code>%ad</code>	Author date (format respects the <code>--date=option</code>)
<code>%ar</code>	Author date, relative
<code>%cn</code>	Committer name
<code>%ce</code>	Committer email
<code>%cd</code>	Committer date
<code>%cr</code>	Committer date, relative
<code>%s</code>	Subject

Configuration

Git configuration is stored in 3 places:

- `- /etc/gitconfig (--system)`
- `- ~/.gitconfig (--global)`
- `- .git/config (--local)`

Each level overwrites the previous one.

Setting up:

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
git config --global core.editor code
```

Check some setting:

```
git config user.name
```

See all configuration and sources:

```
git config --list --show-origin
```

