

Global Configuration

<code>git config --global user.name "John Doe"</code>	Globally configure your committer name
<code>git config --global user.email johndoe@example.com</code>	Globally configure your committer email

Execute these step **before** any other action with your git client

Initialize new repository

<code>git init</code>	Initialize current directory as a new git repository
<code>git remote add origin https://reporurl/repo.git</code>	Add remote repository with name 'origin'
<code>git add --all</code>	Promote all folder content for commit
<code>git commit -am"initial commit"</code>	Commit all files with given comment

Clone existing remote repository

<code>git clone https://giturl/repo.git</code>	Clone remote repository via HTTPS, access credential could be required
<code>git clone ssh://giturl/repo.git</code>	Clone remote repository via SSH, private/public key exchange is required

Inspect your work

<code>git status</code>	Provide you with the difference in commits and push between your local repo and the remote origin
<code>git log</code>	List latest commit on the local repository
<code>git diff \${FILENAME}</code>	Provide a diff view between the local file and the HEAD version on the remote origin

Save your work

<code>git pull origin develop</code>	Retrieve all commit from remote repository named "origin" and branch "develop" and merge them into local repository
<code>git add --all</code>	Promote ALL unstaged files for commit - use with care!
<code>git add \${FILENAME}</code>	Promote given file/folder for commit
<code>git commit -m"my details on commit"</code>	Commit all promoted file to local repository
<code>git push origin develop</code>	Push all commit to remote repository named "origin" and branch "develop". Branch will be created if not present. May fail if remote repository has not been pulled before pushing since local repository is not up-to-date



By **Michele Sacchetti** (mikesac)
cheatography.com/mikesac/
aroundthecode.org

Published 26th January, 2017.
 Last updated 10th April, 2017.
 Page 1 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Branching

git branch -a	List all available branches
git fetch	Fetch all remote repository data into local one, including new branches
git fetch -p	Fetch all remote repository data into local one, including new branches. -p (prune) causes unexisting remote branches to be dropped also locally
git checkout -b newbranch	create a new branch named "newbranch" starting from current commit
git merge otherbranch	Merge local branch named "otherbranch" to be merged into current local branch
git push origin newbranch	Push all commit in local branch newbranch to remote origin, creating remote branch if not already present
git branch -D oldbranch	Delete local branch named "oldbranch". Any tag pushed from deleted branch will be preserved.
git push origin --delete oldbranch	Delete remote branch named "oldbranch". Any tag pushed from deleted branch will be preserved.
git merge --squash featurebranch	Merge branch named "featurebranch" into current one, squashing all commit into a single one. Commit with comment is needed to complete operation.

Tagging

git tag -l	List all available tags
git fetch --tags	Fetch all remote repository data into local one, including tags
git fetch --tags -p	Fetch all remote repository data into local one, including new tags. -p (prune) causes unexisting remote tags to be dropped also locally
git tag mytag	Creates a new tag named "mytag" and attach it to current local commit
git push origin --tags	Push local tags to remote origin
git tag -d oldtag	Delete local tag named "oldtag".
git push origin :refs/tags/oldtag	Delete remote tag named "oldtag".



By **Michele Sacchetti** (mikesac)
cheatography.com/mikesac/
aroundthecode.org

Published 26th January, 2017.
 Last updated 10th April, 2017.
 Page 2 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Cleaning up your repository

<code>git reset --soft origin master</code>	Reset local repository commit to remote origin on branch master. All changes will be preserved as uncommitted files
<code>git reset --hard origin master</code>	Reset local repository commit to remote origin on branch master. All changes will be discarded -use with care!
<code>git clean --dry-run</code>	Remove untracked files - show only candidates no actual removal (alias -n)
<code>git clean -f</code>	Remove untracked files - Actual remove use with care!
<code>git clean -f -d</code>	Remove untracked files and folders - Actual remove use with care!
<code>git clean -f -X</code>	Remove ignored files - Actual remove use with care!
<code>git clean -f -x</code>	Remove ignored and non-ignored files - Actual remove use with care!
<code>git log -- \${FILENAME}</code>	Show history for \${FILENAME} even if deleted, useful to know when a file has been removed



By **Michele Sacchetti** (mikesac)
cheatography.com/mikesac/
aroundthecode.org

Published 26th January, 2017.
Last updated 10th April, 2017.
Page 3 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>