

# Git Cheatsheet

This page presents a list of commonly used commands for git.

Categories: [cheatsheet](#) [git](#)

Access this page by simply typing in "**cs git**" in your browser address bar if you have [bunnylol](#) configured.

## Commands

### New Repo

```
# Create a new local git repo.
git init [project name]

# Clone a git repo.
git clone <url>

# Clone a git repo to a local dir.
git clone <url> <local-dir>
```

[Copy](#)

### Config

```
# Set your user name.
git config --global user.name "name"

# Set your email address.
git config --global user.email "email"

# Edit git global config.
git config --global --edit
```

[Copy](#)

# Branch

```
# List all local branches.
git branch

# List all local and remote branches.
git branch -a

# Check out to another branch.
git checkout <branch-name>

# Check out to the previous branch.
git checkout -

# Create a new local branch and check it out.
git checkout -b <branch-name>

# Delete a local local branch.
git branch -D <branch-name>

# Merge the target branch changes into the current branch.
git merge <branch-name>

# Merge a commit into the current branch.
git cherry-pick <commit-id>

# Rebase the current branch onto the target branch.
git rebase <branch-name>

# Fetch all remote branches from alias.
git fetch [alias]

# Fetch and merge commits from tracking remote branch
git pull

# Push a local branch to remote with same branch name.
git push [alias] [branch]

# Push a local branch to another remote branch.
git push [alias] [local-branch]:[remote-branch]

# Delete a remote branch
git push origin --delete <branch-name>

# Fetch master branch from other branch and rebase on it.
git fetch origin master:master && git rebase master
```

[Copy](#)

# Changes

```
# Show local changes.
git status

# Stage file.
git add <file>

# Unstage file.
git reset <file>

# Stage changes for all tracked files.
git add .

# Commit all staged changes.
git commit -m "commit message"

# Amend last commit
git commit --amend -m "new message"

# Reset everything to last commit.
git reset --hard

# Delete all untracked files and directories.
git clean -df

# Undo local modifications to all files.
git checkout -- .

# Show non-staged changes.
git diff

# Show staged changes.
git diff --staged

# Save the current changes
git stash

# List all the stashed changes.
git stash list

# Pop the last stashed change.
git stash pop

# Drop the last stashed change.
git stash drop

# Show commit logs.
git log

# Show pretty commit logs.
git log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset'
```

Copy

## Tags

```
# List all tags.
git tag

# Get remote tags.
git pull --tags

# Switch to an existing tag.
git checkout tags

# Create a new tag.
git tag -a tag_name -m "tag message"

# Push all tags to the remote repo.
git push --tags
```

[Copy](#)

## Remote

```
# Create a new alias for a git URL.
git remote add <alias> <url>

# Show the remote repo names you have.
git remote

# Show the remote repo names and urls you have.
git remote -v

# Remove a remove repo.
git remote rm <remote-repo-name>

# Change the URL of the origin alias.
git remote set-url origin [git-url]
```

[Copy](#)

## Submodule

```
# Create a submodule.
git submodule add <git-repo-url> <submodule-dir-path>

# Init submodules.
git submodule init

# Update and init submodules recursively.
git submodule update -i -r

# Set a new url for the submodule.
git submodule set-url -- <submodule-dir-path> <new-git-repo-url>

# Sync all submodules.
git submodule sync

# Execute command for each submodule.
git submodule foreach --recursive <command>
```

[Copy](#)

# Aliases

```
git config --global alias.co checkout  
git config --global alias.br branch  
git config --global alias.ci commit  
git config --global alias.st status
```

[Copy](#)