



## create

Clone an existing repo

```
$ git clone ssh://user@domain.com/repo.git
```

create a new local repo

```
$ git init
```

## local changes

changed files in your working directory

```
$ git status
```

changes to tracked files

```
$ git diff
```

add all current changes to the next commit

```
$ git add .
```

add some changes in *< file >* to the next commit

```
$ git add -p < file >
```

commit all local changes in tracked files

```
$ git commit -a
```

commit previously staged changes

```
$ git commit
```

change the last commit

*don't amend published commits!*

```
$ git commit --amend
```

## refactor filenames

deletes the file from working directory and stages the deletion

```
$ git rm < file >
```

removes the file from version control but preserves the file locally

```
$ git rm --cached < file >
```

rename *< file >*

```
$ git mv < file -- original > < file -- renamed >
```

## commit history

show all commits, starting with newest

```
$ git log
```

show latest *< n >* commits, with details

```
$ git log -< n > -p
```

show changes over time for a specific file

```
$ git log -p
```

who changed what and when in *< file >*

```
$ git blame
```

## branches & tags

list all existing branches

```
$ git branch -av
```

switch HEAD branch

```
$ git checkout < branch >
```

create a new branch based on your current HEAD

```
$ git branch < new -- branch >
```

*or*

```
$ git checkout -b < new -- branch > create and switch
```

create a new tracking branch based on a remote branch

```
$ git checkout --track < remote/branch >
```

delete a local branch

```
$ git branch -d < branch >
```

mark the current commit with a tag

```
$ git tag < tag -- name >
```

## update & publish

list all currently configured remotes

```
$ git remote -v
```

show information about a remote

```
$ git remote show < remote >
```

add new remote rep, as *< remote >*

```
$ git remote add < remote > < url >
```

download all changes from *< remote >*,

but don't integrate into HEAD

```
$ git fetch < remote >
```

download changes and

directly merge/integrate into HEAD

```
$ git pull < remote > < branch >
```

publish local changes on a remote

```
$ git push < remote > < branch >
```

delete a branch on the remote

```
$ git branch -dr < remote/branch >
```

publish your tags

```
$ git push --tags
```

## merge & rebase

merge *< branch >* into your current HEAD

```
$ git merge < branch >
```

rebase your current HEAD onto *< branch >*

*don't rebase published commits!*

```
$ git rebase < branch >
```

abort a rebase

```
$ git rebase --abort
```

continue a rebase after resolving conflicts

```
$ git rebase --continue
```

use your configured merge tool to solve conflicts

```
$ git mergetool
```

use your editor to manually solve conflicts

and (after resolving) mark file as resolved

```
$ git add < resolved -- file >
```

```
$ git rm < resolved -- file >
```

## undo

discard all local changes in your working directory

```
$ git reset --hard HEAD
```

discard local changes in a specific file

```
$ git checkout HEAD < file >
```

revert a commit (by producing a new commit with contrary changes)

```
$ git revert < commit >
```

reset your HEAD pointer to a previous commit ...

... and discard all changes since then

```
$ git reset --hard < commit >
```

... and preserve all changes as unstaged changes

```
$ git reset < commit >
```

... and preserve uncommitted local changes

```
$ git reset --keep < commit >
```

Compiled by Yi Liu (<http://YiLiu6240.github.io>).

source:

<https://www.git-tower.com/blog/git-cheat-sheet/>

Last Updated June 20, 2016