Cheatography

Git Cheat Sheet

by James (realjimbob) via cheatography.com/130441/cs/25956/

Configuration

Set or show Git configuration options:

```
git config [ --global ] [ --get ] <na me> [ <va lue> ]
git config [ -l | --list ]
```

To set user's name and email address:

```
git config --global user.name " -
Fir stname Lastna me"
git config --global user.email "
 fir stn ame.la stn ame @or g.c -
om"
```

By default local settings are stored in the project's repository, global settings are stored in the user's home directory.

Repository Setup

Initialise a local repository:

```
git init
```

Clone a remote repository:

```
git clone <UR L> [ <lo cal _di r
> ]
```

The URL can use HTTP or SSH etc.

Basic Commands

Show the working tree status:

```
git status [ -s | --short ]
```

Add files from the working folder to the index:

```
git add <fi les pec>
```

Save changes from the index to the repository:

```
git commit [ -a ] [ -m <me ssa ge> ]
```

Show changes between references:

```
git diff [ --staged ] [ --no-r enames ]
git diff <co mmi t> [ <co mmi t> ] [
<fi le> ]
git diff <co mmi t>...< com mit>
```

Show commit logs:

```
git log [ -<n> ] [ --oneline ] [ --patch ] [ <br anc h>
]
```

Remove files from working tree and index:

```
git rm [ --cached ] <fi len ame>
```

Move or rename a file:

```
git mv <fi len ame> <de sti nat ion>
```

Reset current HEAD in the repository:

```
git reset [ --soft | --mixed | --hard ] <sh a1>
```

Commit Messages

1. Divide subject and body with blank line
2. Limit the subject length to 50 characters
3. Capitalise the subject line
4. Do not end the subject line with a full stop
5. Use imperative mood in the subject line
6. Wrap the body at 72 characters
7 Use body to explain what & why, not how

https://chris.beams.io/posts/git-commit/

Managing Remotes

Add and track remote repositories:

```
git remote add <re mot e> <UR L>
```

List remote repositories:

```
git remote [ -v | --verbose ]
```

Update local repositories:

```
git pull [ <re mot e> [ <br anc h> ] ]
```

Update remote repositories:

```
git push [ -u ] [ <re mot e> [ <br anc h>
] ]
```

Download changes from remote repository:

```
git fetch [ <re mot e> [ <br anc h> ] ]
```

Managing Branches

List, create, rename and delete branches:

```
git branch [ --all ] [ [ -m [ <ol dbr anc h> ] ] | [ -d ] | [ -D ] <br
> ]
```

Switch branches:

```
git checkout [ -b ] [ --track ] <br anc h>
```

Join branch histories together:

```
git merge <branch>
git merge --abort
```

Rebase your branch:

```
git rebase [ -i ] <br anc h> | <co mmi t>
```

Tag a commit:

```
git tag [ -a ] <ta g> [ -m <me ssa ge> ]
```

Compare branches:

```
git diff <br anc h1> <br anc h2>
```

Stash Management

Stash the changes from the working tree:

```
git stash [ -u ]
git stash [ list | show | pop ] [
```

Plumbing Commands

Show the contents of objects in the repository:

```
git cat-file -t | -p <sh a1>
```

Count the number of objects in the repository:

```
git count- objects
```

```
git show-ref <br anc h>
```

Find common ancestor commit:

```
git merge-base <br anc h1> <br anc
```

Copy specific commits:

```
git cherry -pick <co mmi t>
```

List remote references:

```
git ls-remote
```

Set files to no longer track changes:

```
git update -index --assu me- unc h
i le>
```

By **James** (realjimbob)
cheatography.com/realjimbob/

Published 5th February, 2021.
Last updated 11th February, 2021.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com