



git

CHEET SHEET BY @CHARLSTOWN

CREATOR: LINUS TORVALDS

YEAR: 2005

STABLE RELEASE (2021): 2.34

POPULAR GUIs: GERRIT, PHABRICATOR, GITHUB,

SOURCEFORGE, BITBUCKET, GITLAB, ETC.

SETUP & INIT	cmd	>Description
git config --global *		> Show system date
*user.name ""		> Save a name as config
*user.email ""		> Save an email as config
*color.ui auto		> Set line coloring
git init		> Initialize workdir as repo
git clone [url]		> Retrieve a repo via url

DAY-TO-DAY WORK	cmd	[opt1 opt2]
git status		> Show system date
git add [*] [file]		> Add file/s to staging area
git commit		> Record changes to repo
-m "msg"		> Add commit message
git diff [*] [file]		> Changes dir vs. stage area
--staged		> Changes stage vs repository
git checkout -- [f]		> Discard changes in workdir
git reset [file]		> Unstage but keep changes
git rm [file]		> Remove file from dir
git stash *		> Current changes for later
*pop		> Apply stored stash
*drop		> Delete specific stash

REVERTING CHANGES	cmd	[opt1 opt2]
git reset		>Switches the current branch
--hard		> all changes are discarded
[target]		> Target reference to switch
git revert [sha]		> New commit revert changes

GIT BRANCHING	cmd	[opt1 opt2]
git branch		> List local branches
-a		> List all branches
-d [name]		> Remove selected branch
git checkout [b]		> Switch to selected branch
-b [branch]		> Creates & switch to branch
git merge [name]		> Join branch to current

REVIEW YOUR WORK	cmd	>Description
git log		> List commit history
-n count		> Last n commits
--oneline		> One commit per line
--graph		> Graph indented
--decorate		> Add colors
ref..		>Only branch commits
..ref		> Commit presents on ref
git reflog		> Operations in local repo

GIT TAG	cmd	[opt1 opt2]
git tag		> list all tags
[name]		> Create a tag reference
[hash]		> Applies tag to specific hash
-a [name]		> Create a tag object
-d [name]		> Remove tag from local repo

SYNCHRONIZING REPOS	cmd	[opt1 opt2]
git remote add [alias] [url]		> Add git url
git fetch [remote]		> Pull, no merge
--prune [remote]		> Delete remote Refs
git pull [remote]		> Fetch & merge
git push [remote]		> Push local changes
-u [remote] [br]		> Push local branch

IGNORING PATHS	cmd	[opt1 opt2]
\$ cat .gitignore		> Creates file to ignore files
/dir/*		> All contents form dir

COMMON CMDS WORKFLOWS
Create a new repository with init files
git init
> Create local repository & .git dir
git add [. file1 file2]
> Add all/some files to the staging area
git commit -m "first commit"
> Create first commit with init files
git branch -M main
> Create and rename branch to "main"
git remote add origin [url]
> Add local repo to remote origin
git push -u origin main
> Push local repo to remote. -u to make it principal

Discard unstaged local changes
git status
> Confirm there are't unstaged files
git reset
> [Optional] Unstage all files if there where
git checkout [-- file]
> Overwrite local changes. Optional [--file]
git stash
> [Optional] Save local changes for later
git reset --hard
> Discard local changes to all files

Unify commits into one (2 ways)	
git reset --soft [hash HEAD~3]	
> [A] Squash until hash or last three (HEAD~3)	
git commit -m "New message"	
> [A] Overwrite local changes. Optional [--file]	
git push origin +[Branch name]	
> [A] Forces remote branch to accept history	

git rebase -i [hash HEAD~3]	
> [B] Squash commits by editor (s to squash)	