Cheatography

Git Essentials Cheat Sheet

by hayoola via cheatography.com/40063/cs/12363/

Git Stash

Stash unfinished work

git stash or git stash save "message"

List of stash

git stash list

Checkout or restore stash & then delete

git stash pop or git stash pop <stash>

Delete stash

git stash drop or git stash drop <stash>

To delete all stash

git stash clear

To apply a stash

git stash apply <stash@>

git stash pop and git stash drop would take action on the last stash, where a specific stash is not indicated.

Git Branch

Create a branch

git branch <branch name>

Create and checkout to branch

git checkout -b <branch new>

List branches from local repo

git branch

List branches from remote repo

git branch -a

Rename a branch

Git Branch (cont)

git branch -m <oldname> <newname>

Switch or checkout to a branch

git checkout <branch name>

Delete branch locally

git branch -d <branch name>

Delete branch remotely

git push origin --delete <branch_name>

Merge branches into a current branch

git merge <branch to merge>

By analogy, a branch is more like a module. A commit is more like a sub module. You would typically want to use a branch to create a feature set, a part of the program with multiple sub part. Etc

How To

Discard uncommit changes

git checkout .

Solve Merge conflict by discarding uncommited changes

git reset && git checkout . [reset will unstage, and checkout will discard)

Git commit

List all commits under the current branch

git log

List all commits ever made

git reflog

Make a commit

git commit -m "commit message"

Amend commit

git commit --amend "amend message"

Remove / forget a tracked file

git rm --cached <file>

Git stage

Stage a file i.e add file to staging

git add <file name>

List stage and unstage files

git status

Stage all files

git add .

Unstage a file

git reset <file name | path to file> or git rm —cached <file name | path to file>

Remove files from staging

git checkout -- <file name | path to file>

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

Cheatography

Git Essentials Cheat Sheet

by hayoola via cheatography.com/40063/cs/12363/

Git merge

List branches that are merged into the current branch

git branch —merged

Abort or quit a merge

git merge —abort

List branches that hasn't been merged yet LOCALLY

git branch --no-merged

List branch that hasn't been merged yet REMOTELY

git branch -r --merged

At different point, remember to merge often into master after testing out

Git Remote

Push all of your local branches to the specified remote.

git push <remote> --all

Fetch a particualr branch on the remote

git fetch <branch name>

Fetch and prune at the same time

git fetch -p

Prune fetch configuration

git config --global fetch.prune true

Git log

Show all logged commit

git log

Show all commit of a particular branch

git log <branch name>

Show all commit in remote branch

git log <remote/branch name>

General knowledge

List of git alias

git alias

Config an alias

git config —global alias.aliasname “the command e.g commit”

List all git config

git config —list

THE DIFFERENCE BETWEEN A SOFT, MIXED AND HARD RESET

Soft reset will basically reset your head pointer to a hash in time in such a way that your working index are preserved and your working directory are overwritten. (v)

The Mixed reset will basically cause your staging index unstage, and your working directory overwritten. (V)

The hard reset will revert to a point in time, and overwrite both your working directory and staging index. (V)

Undo Changes

Undo current change and revert back to a more recent commit

git checkout -- <file name | path to file>

This will basic discard all changes and revert back to a point of most recent commit

Git diff | comparism

Compare branches

git diff <branchA> <branchB>

Compare commit

git diff <commit1> <commit2>

Compare working directory with staged files/index

git diff —staged

Compare a commit with a file / narrow the diff in a commit to what changed in file

git diff commit1 filename.ext

Compare a commit with HEAD

git diff commit <HEAD>

Compare commits side by side in oneline

git diff <commit a> <commit b> --word-diff=color

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