Cheatography

Git Cheat Sheet

by milad69_1 via cheatography.com/52248/cs/16796/

About Git

Git vs other systems

The major difference between Git and any other VCS (Subversion and friends included) is that they store as a set of files and the changes made to each file over time. Git thinks about its data more like a stream of snapshots.

Git states

Git has three main states that your files can reside in: **1) committed, 2) modified**, and **3) staged**:

* **Committed** means that the data is safely stored in your local database.

* **Modified** means that you have changed the file but have not committed it to your database yet.

* **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot (staging area).

Getting Help

```
git help <verb>
```

```
man git <verb>
```

```
git add -h
```
or
```
git add --help
```

ex: `git help config`

.git folder

About

A folder to save history and version control information about the project

Config

show the git config info

```
git config --list
```

add user name (use --global for global change in system)

```
git config [--global] user.name "[name]"
```

add user email (use --global for global change in system)

```
git config [--global] user.email "[email address]"
```

Set editor

Config (cont)

```
git config --global core.editor nano
```

Checking git setting

```
git config --list
```

```
git config user.name
```

> John Doe

Git config files

Stores in 3 level: 1) system config files 2) user specific config file 3) repository config file

1) System config file: located at /etc/gitconfig. can change if you pass `--system` option to `git config` command

2) user config file: ~/.gitconfig or ~/.config/git/config. Can change if you pass `--global` option to `git config` command.

3) Repository config file: .git/config. Can change if no other option is specified or `--local` option is set.

Staging

Get staging status

git status

"Get information about files that have special status to get. Not previousley added and commited files."

Staging Area

Commit

Commit all files in the staging area into local repository

using editor(command and then save commit message inside editor):

git commit

Commit with inline message

git commit -m "commit message"

Commit with all the changes that it can find

(from tracked files)

git commit -a

Add file

Add file from workspace into staging area

```
git add <file1> <file2> ...
```

Add all files from current dir into staging area

```
git add .
```

Add all tracked file and untracked file from the workspace into staging area

```
git add -A .
```

Add using Regular expression

```
git add *.js
```

Create/Remove a repository

Initialize the current dir to be repository

cd <project_dir>

git init

touch README

Initialize given dir to be repository

git init my_repository

From other repositories

git clone existing_dir new_dir

git clone git://github.com/user/repo.git

git clone https://github.com/user/repo.git

Remove the repository

rm -r repository_folder/.git