**History and Configuration**

**Viewing the Log**

These will show the SHA hash

The Author

The Commit message

**Colorizing the Log**

Usually not easy to read.

You should

git config --global color.ui true

You can format the log

git log --pretty=oneline

This will display one commit per line

**Log Format**

git log --pretty=format: “%h %ad- %s [%an]”

You can format the log anyway you’d like with placeholders as shown above.

The place holders are…

%ad Author date

%an Author name

%h SHA hash

%s subject

%d ref names

**Patch**

git long --oneline -p

This will show line by line what was added and removed for each commit

**Stat**

git log --oneline --stat

This will show how many insertions and deletions included in each file in each commit.

**Graph**

git log --oneline --graph

This will give you a visual representation of the branch merging into the master with commit timeline nodes.

**Date ranges**

If you want to specify a range of dates to see in the log.

git log --until=1.minute.ago

git log --since=1.day.ago

git log --since=1.hour.ago

git log --since=1.month.ago --until=2.weeks.ago

git log --since=2000-01-01 --until=2012-12-21

**Uncommitted Changes**

git diff HEAD

the difference between local and remote

**Earlier Commits**

git diff HEAD^

parent of latest commit

git diff HEAD^^

grandparent of latest commit

git diff HEAD~5

five commits ago

git diff HEAD^..HEAD

Compares against the second most recent and the most recent

You can also compare by SHA hashes

git diff SHAofOne..SHAoftheother

You can also compare branches

git diff master branch

You can also compare based on time

git diff --since=1.week.ago --until=1.minute.ago

**Blame**

git blame filename --date short

**Excluding Files**

git status

if you have a folder you want to exclude just open up the .git folder and look for exclusions and add it in there.

**Exclude Patterns**

tutorial.mp4

Will exclude this file

\*.mp4

Will exclude all mp4’s

directoryName/

Will exclude the directory

logs/\*.log

Will exclude all logs

**Excluding from all copies**

Never include log files in a repo

Why

Because they always changed based on who is using the application

In .gitignore

logs/\*.log

**Removing Files**

git rm README.txt

This will add to the stage

git commit -m “Removed README”

This will delete the readme file

**Untracking Files**

git rm --cached filename

This will stop tracking the file, it won’t be deleted locally only from git.

.gitignore

logs/\*.log

git add .gitignore

git commit -m “Ignore all log files”

**Config**

git config --global core.editor emacs

use emacs for interactive commands

git config --global merge.tool opendiff

use opendiff for merging conflicts

git config --list

Will list all config options you have set

You can have more than one email set, it’s repo specific to see which you are using…

git config user.email

**Aliases**

Aliases for log formats

git config --global alias.mylog \ “log --pretty=format: ‘%h %s [%an]’ --graph”

git config --global alias.lol \ “log --graph --decorate --pretty=oneline --abbrev-commit --all”

Everyday commands

git config --global alias.st status

git config --global alias.co checkout

git config --global alias.br branch

git config --global alias.ci commit