How to modify system in git commands

System

git config --system

User

git config – -global

Project

git config

git config – -global example: git config – -global user.name “myname”

To see the list of my configurations: git config --list, git user.name, git user.email

Locating my config: cd ~

List file/system: ls –la

If a file has . infront it will hide this file by default. To see this file you can use cat .gitconfig (name of the .file)

Command to tell git what text editor to use: git config – -global core.editor “mate” (name of the text editor) –wl1” (this tell the unit to wait before launches)

Color user interface UI: git config – -global color.ui true

Cat command allows me to open any file

Git auto-completion – Downloading from GitHub

Cd ~

curl <https://github.com/git/git/raw/master/contrib/completion/git-completion.bash>

Rename file

mv ~/git-completion.bash ~/.git-completion.bash

Adding to my log config

Edit ~/ .bash\_profile or ~/.bashrc

If the git-completion. Bash exisr then load the file .git-completion.bash. Code below

If [ -f ~/.git-completion.bash ]; then

Source ~/.git-completion.bash

fi

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

**reset** Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

Last login: Sat Sep 15 06:37:00 on ttys000

Sandros-iMac:~ katherinekruk$ git help

usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

reset Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

Sandros-iMac:~ katherinekruk$ git help log

GIT-LOG(1) Git Manual GIT-LOG(1)

NAME

git-log - Show commit logs

SYNOPSIS

git log [<options>] [<revision range>] [[--] <path>...]

DESCRIPTION

Shows the commit logs.

The command takes options applicable to the git rev-list command to control what is shown

and how, and options applicable to the git diff-\* commands to control how the changes each

commit introduces are shown.

OPTIONS

--follow

Continue listing the history of a file beyond renames (works only for a single file).

--no-decorate, --decorate[=short|full|auto|no]

Print out the ref names of any commits that are shown. If short is specified, the ref

name prefixes refs/heads/, refs/tags/ and refs/remotes/ will not be printed. If full

is specified, the full ref name (including prefix) will be printed. If auto is

specified, then if the output is going to a terminal, the ref names are shown as if

short were given, otherwise no ref names are shown. The default option is short.

How to star a new repo:

git init

katherinekruk$ cd Desktop/Git-GitHub/First\_git\_home\_repo/

Sandros-iMac:First\_git\_home\_repo katherinekruk$ **git init**

Initialized empty Git repository in /Users/katherinekruk/Desktop/Git-GitHub/First\_git\_home\_repo**/.git/**

Sandros-iMac:First\_git\_home\_repo katherinekruk$

Command:

* la : will shown hidden/top level of or our project/ file in a folder example: . .. or .git.
* ls -la .git list more elements in the project

Initialing new commit: git add .

Git commit –m “Initial commit”

. : is short for the directory that I am currently.

Viewing commit log: git log

git log –n 2 : to see the numbers of logs

git log --since= 2018-06-11

git log --until=2018-9-17

git log --author="Augusto"

git log --grep="Init" (grep= Global regular expression search)

Git schema:

Three tree

1. Repository (add)
2. Staging index (updating)
3. Working (commit)

Git get refer: (SHA-1)

Git generates a Checksum for each change set

Algorithms

Always equal to sum

It is the long key with 40 characters

HEAD or pointer always point the “tip” of the current branch in the repo, last check out, next-newest point to start recording (committed).

Git status: tells us what is happening in our directory.

(use "git add <file>..." to update what will be committed)

git commit -m "Modified the text of the third file"

Viewing changes with “diff”

git diff

To view what it is in the staging level you will use:

git diff --staged

How to delete file and track deletions:

git rm and the name of the file you want to delete (rm is short for remove)

How to change name to the files:

Git mv name of the file new name of the file

How to move file from one folder to a new folder

git mv third\_file.txt first\_directory/third\_file.txt

How to undo command:

git checkout and the name of the file

git checkout + name of branch to switch between branches

git checkout -- name of the file (this command is more specific to bring back one article/file within the same branch)

how to move all file to stage at once

git add --all

How to use the undo command in one particular committed item:

git log

git checkout and the HEAD ( Key name) -

git checkout –b <new branch name> environment to test your code

how to check git branches

git branch

how to know which branch I am in:

The branch name with \* is where you are currently in

How to get back from the git log window:

Control+z

How merge branches:

First you need to be in the branch that you want the merge to be complete

\*master

git merge and the name of branch you want to merge

How to change branch name:

Git branch –m

Deleting branch command

Git branch –D and the name of branch

How to see all branches at once:

Git branch –a

How to clone github repo:

Git clone and the url

How to mirror a clone from github repo:

Git clone --mirror and the url .git

How to clone a specified repo from github

Git clone –b name of the branch and the url

How to delete a folder from git repo in the terminal:

First go to the folder ~ cd/ folder name

Rm –dfr and the name of the folder ( .git)

How to go back to a folder

Cd .. / name of the folder/

Reviewing changes made:

\*git diff --color-words name of the file (side by side changes)

git diff name of the file

Adding/committing all file at once.

git commit –am “changed 24 hour support number to 4314”

Adding more then one item to your staging area can be done by

Git add name of the item/

How to unstaging commands:

git reset HEAD file Name

How to make changes/amend made to the Repository: in case you make one update and save and yo need to make a new one within the same repo. You can also use to change on the message of your commit.

git commit --amend -m " messages"

|  |  |
| --- | --- |
| SHA | **Secure Hash Algorithm** |

How to retrieve old versions:

git checkout 0a20599a8399add890a -- file name

To checked old stage and changed

git diff –staged

How to revert a commit:

git revet ef3aa841f2048cb8

How to undo multiple commands – dangerous to use:

Option to uses while using git reset (Soft, mixed, hard)

--soft : does not change staging index or index pr working directory

--mixed (default): Changes staging index to match repository, Does not change working directory

--hard: changes staging index and working directory to match repository (be careful using this one)

How to know what the HEAD pointer is pointing at

cat .git/HEAD

cat .git/refs/heads/master

0c2f15b0ea900454daf737b90ec978703f074a55

How to remove untracked files:

git clean –n just inform you about the remove

git clean –f will force to remove the files

: