**Configuring author and email**

**# home directory**$ git config --global --list       
$ git config --global user.name "Tanuj Tripathi"  
$ git config --global user.email "tanujtripathi93@gmail.com"

**# vi is the editor, ~ is the home directory**  
$ vi ~/.gitconfig                                                  
$ cat .gitconfig

**Initializing an empty repository**

**# notice the creation of .git directory**                         
$ git init     
                         
**# "ls -al" command lists all objects, including hidden**                                               
$ ls -al    
        
**# without using any editor**                                                                  
$ echo "this is my first file in empty repository" >> firstFileUsingEcho.txt       
$ cat firstFileUsingEcho.txt

**Command Summary - Accessing Git Help system**

**# general help**  
$ git help

**# lists sub-commands**  
$ git help -a

**# lists concept guides**  
$ git help -g

**# read about a specific sub-command**  
$ git help <command>

**# read about a specific concept**  
$ git help <concept>

**# same as "git status" since "long" option is the default one (compare output)**      
$ git status --long          
$ git status -s                                 # status "??" for untracked file  
$ git add weightLossChart  
$ git status -s                                 # status "A"  
$ git commit -m "1st commit for weightLossChart"  
$ git status -s       
$ vi weightLossChart                 # make some changes to the file  
$ git status -s                              # status "M"  
$ git add weightLossChart    
$ git status -s                                # status "M"  
$ git commit -m "2nd commit for weightLossChart"  
$ git status -s     
$ mv weightLossChart weightLossChart2  
$ git status -s                             # status "D"

**How to check commit history**

**# displays the entire commit history using the default formatting**  
$ git log

**# oneline condensed view of each commit history**                        
$ git log --oneline

**# Only display commits that include the specified file**                               
$ git log <file>

**# Show only commits that occur between <since> and <until>. Both arguments can be either a commit ID ...contd.  
# a branch, name, HEAD, or any other kind of revision reference.**$ git log <since>..<until>

**# Limit the number of commits by <limit>. For example, git log -n 3 will display only 3 commits.**    
$ git log -n <limit>

Rebasing

**SSH**

This works on the concept of public private key.

Suppose **Person A** wants to send any sensitive data to **Person B**, Persosn will encrypt the data with **Person B’s** public key and send to **Person B**. Now **Person B** with the prvate key will decrypt it.

**Check existing ssh keys:**

$ ls -l ~/.ssh

**Generate public-private key**

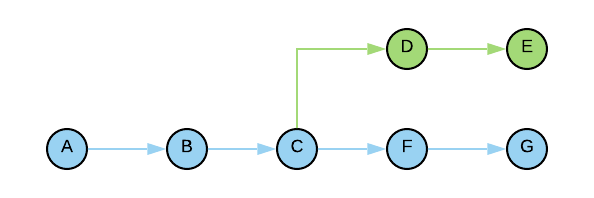
$ ssh-keygen -t rsa -b 4096 -C "key pair generation demo"

**start ssh agent and add key to it**

$ eval "$(ssh-agent -s)"

$ ssh-add ~/.ssh/id\_rsa

**Associate ssh key to the github account**



After Rebasing

