***Configure Git***

$ which git – Verified git installed successfully.

git config --system – System

git config --global – User

git config - Project

$ git config --global user.name “Alex Berghell” – Config your name.

$ git config--global user.email “ someone@nowhere.ocm” – Config your email.

$ git config--list – To view your configurations.

$ git config --global core.editor “mate -wl1” – Set up default text editor.

$ git config --global color.ui true – Set up different colors for better user experience.

$ git help log – To get full git commands for explanations.

$ man git-log - To get full git commands for explanations.

***Using Git***

$ git init – Initialize a git in the particular folder. Set up a folder as a home repository.

$ git add . – Add all files to staging index.

$ git add file.txt – Add only file.txt to staging index.

$ git commit -m “Desired message” – Commit the file into remote repository and add a message to specify the change. Message should not be longer than 72 characters.

$ git log – Shows the log of all our commits one after another. Each commit has own unique ID.

$ git log -n 2 – Shows last 2 commits. You can specify the number of commits you want to see.

$ git log --since=2018-03-14 – shows all commits since the particular date.

$ git log --until=2018-03-14 – shows all commits before the particular date.

$ git log --author=”Alex Berghell” – Look for commits made by person/author.

$ git log --grep=”Init” – search for something has “Init” pattern.

***Changes in our three-tree architecture***

Unique ID of a commit calls - **Checksum** / **Hash algorithm** / **SHA-1**. It is *40-character hexadecimal string (0-9, a-f)*.

**HEAD** – pointer to “tip” of current branch in repository. Points to our last commit – parent. To find out what file HEAD points to, you can go to $ cd .git>refs>heads>$ cat master – to view inside of a file. You should see the checksum.

$ git status – Shows the difference between our trees: ***working – staging – repository***.

***Changes between files***

$ git diff – Shows all changes of the old(remote repository) and new(working directory) files. (diff compares two files between each other). Shows line by line all changes. -- stands for remove lines, ++ stands for adding new lines.

$ git diff file.txt – show changes only in “file.txt” file between working and repository.

$ git diff --staged – compare files between staging and repository. Old versions “*$ git diff --cached*”.

***Deleting files***

$ git rm file.txt – completely delete file

$ git status – just to verify file was deleted

$ git commit -m “File.txt removed” – commit our changes/deletion to our repository

***Move and Rename files***

$ git mv second\_file.txt secondary\_file.txt – Renamed the second\_file.txt to the secondary\_file.txt.

$ git status – Verify rename.

$ git mv third\_file.txt first\_directory/third\_file.txt – Moved third\_file.txt to the other directory /first\_directory.

$ git commit -m “Reorganized file structure by moving files” – Don’t forget to commit our changes.

***Track files***

$ git commit -am “Initial commit” – commit to the remote repository “-a” avoiding $git add .

$ git checkout -- file.txt – undo our changes. Return everything back in file.txt