Check the name 🡺 $ git config user.name

Create the name 🡺$ git config --global user.name "your name"

Create the emailed🡺 $ git config --global user.email "your email"

Check the emailed🡺 $git config user.email

Edit emailed and user name 🡺 git config –global –edit

Git status

Make a simple folder git repository 🡺Git init

Show hidden file 🡺 ls -a

.git folder is most important folder don’t delete this folder if delete its make normal folder

Commits 🡺 it is a check point

Stage 🡺 before commits file must be stage using ( git add home.html )

For all file stage 🡺 git add .

Commit 🡺 git commit -m“message”

Commits to unstage 🡺 git rm -- cached home.html

Add and commit New file 🡺 git add . && git commit -am “your message”

Add and commit already track file 🡺 git commit -am “your message”

Check the checkpoint/log 🡺 git log

Exit from log 🡺 press q

Restore the previous version 🡺 git restore hello.html

Check current branch 🡺 git branch

Create new branch 🡺 git branch name

Method 1

Switch one branch to another branch 🡺 git switch branch name

Switch to old branch 🡺 git switch –

Branch create and switch 🡺 git swich -c new branch name

Delete branch force delete 🡺 git branch -D study

Delete branch if the branch is merge 🡺 git branch -d study

Method 2

Switch one branch to another branch 🡺 git checkout branch name

Switch to old branch 🡺 git switch –

Branch create and switch 🡺 git checkout -b new branch name

Open and write 🡺 vim filename and press { i } for insert

Discart change also e revarseble 🡺git check out file name

Get one line log 🡺 git log –oneline

Go to specifig hade🡺 git checkout “sdfsfad”

Now branch is detach stage means head don’t point any hade it point another commit

Create a backup 🡺 git checkout -b name

Remove Heade with position 🡺 git checkout Head~2 ( 2 hoto koto number ba commite pechone jeta chay )

**CHAPTER 7**

If you already commit but now you wany to some change and include in previous commit

Or edit in previous commit

In vim 🡺 git commit –amend

Commit with massage 🡺 git commit –amend -m“message ”

**CHAPTER 8**

if you ignore the file not commit

ignore 🡺 touch .gitignore (make this in project root ) and write at the last of file .gitignore

**CHAPTER 9**

What is branches ?

Ans= a branch is a lightweight movable pointer to a commit . and branch is a copy of codebase

When the commit in specific branch this file only show in that particular branch . if we switch in another branch we can not show that file

Merge 2 different branch 🡺step1-> switch branch where we want to merge

Step2->git merge branch name  **which one we want to merge**

* SCENARIO FILE ARE AVAILABLE IN BOTH BRANCH IF EDIT that file IN SAME LINE IN BOTH BRANCH THAT TIME IF WE TRY TO MERGE IT BECOME “AUTOMATIC MERGE FAILED”