Git Tutorials:

1. Git Tutorials #1 | What is Git/GitHub & Why do we need it?
2. Git Tutorials #2| Installing Git + Initial Setup?
3. Git Tutorials #3| Git: Three Stage Architecture
4. Git Tutorials #4 | Tracking Our first Git Project
5. Git Tutorials #5 | Cloning a Remote Git Repository from GitHub
6. Git Tutorials #6 | Git: File Status Lifecycle
7. Git Tutorials #7 | gitignore: Ignoring Files in Git
8. Git Tutorials #8 | Git Diff: Showing Changes Between Commits/Staging Area & Working Directory
9. Git Tutorials #9 | Git: Skipping The Staging Area
10. Git Tutorials #10 |Moving and Renaming Files In Git
11. Git Tutorials #11 | Git Log: Viewing & Changing Commits In Git
12. Git Tutorials #12 | Unstaging & Unmodifying Files In Git
13. Git Tutorials #13 | GitHub: Working with Remote Repositories
14. Git Tutorials #14 | Setting Alias In Git
15. Git Tutorials #15 | Git: Creating & Switching Branches In Git
16. Git Tutorials #16 | Branching & Merging a Production Grade Project
17. Git Tutorials #17 | Resolving Merge Conflicts (With Example)
18. Git Tutorials #18 | Git Branching Workflow in Production
19. Git Tutorials #19 | Pushing Git Branches To Remote Repositories

(1) git status
(2) git init
(3) (i) git add --a
(ii) ↳ git add .
(i) git commit -m "Initial commit"
(ii) git log
(ii) git add first.txt
5) rm . -rf .git
(6) git clone "https:// ..... website name"
↔ Folder
7) pwd
8) ls
(9) cd
(10) (i) touch "new file.txt"
(ii) touch .gitignore
11) (i) git diff
(ii) git diff --staged
(iii) git commit -a -m
(i) git rm "file name"
3) git mv "first.txt"
(ii) git rm -- file name"

5(ii) git log -p

5(iii) git log -p -3

5(iv) git log --stat

5(v) git log --pretty = oneline

5(vi) git log --pretty = short

5(vii) git log --pretty = full

5(viii) git log --since = 2 days

5(ix) git log --since = 2 weeks

5(x) git log --since = 2 months

5(xi) git log --pretty = format: "%h --- an
%an = author name ; %ae = author ema

(iii) git commit --ammend

14) git restore --staged first_renamed

5) (i) git checkout --first_renamed.txt

5(ii) git checkout --f

6(i) git remote

(ii) git remote -V

16(ii) git remote add origin

git@github.com:codewithharry/GitTutorials.g

17) (i) git push -u origin master
(ii) git push origin bugfix

18) (i) git config --global alias.st status

(ii) git config --global alias.unstage "re
-- staged --

(iii) git config --global alias.last "log-

15(iii) git checkout -b develop

15) (iv) git checkout master

19) (i) git branch

20) (i) git merge develop

(19) (ii) git branch --merged

19) (iii) git branch --no-merged

(19) (iv) git branch -d develop

(~~20~~) (17) (11) git push origin bugfix : m

(20) (1) git push -d origin bugfix
(remote_branch

Tracking our first project | Git Tutorials ;

(i) git status

(ii) git init (on branch mas
(Untracked files)

(iii) git add --a

(iv) git commit -m " Initial commit

v) git log

(vi) git add first.txt

(vii) git commit -m " change first.txt and added better design "

(viii)

Git Tutorials #5

Cloning a Remote Git Repository from:

⇒ open github → tensorflow website
→ click clone or download
→ copy the sites "https://....."

⇒ `rm -rf .git` ** ** **

it will delete ".git" folder, thus it is very sensitive command. will not able to track in future.

⇒ `git clone "https://....tensorflow"` we

`pwd`

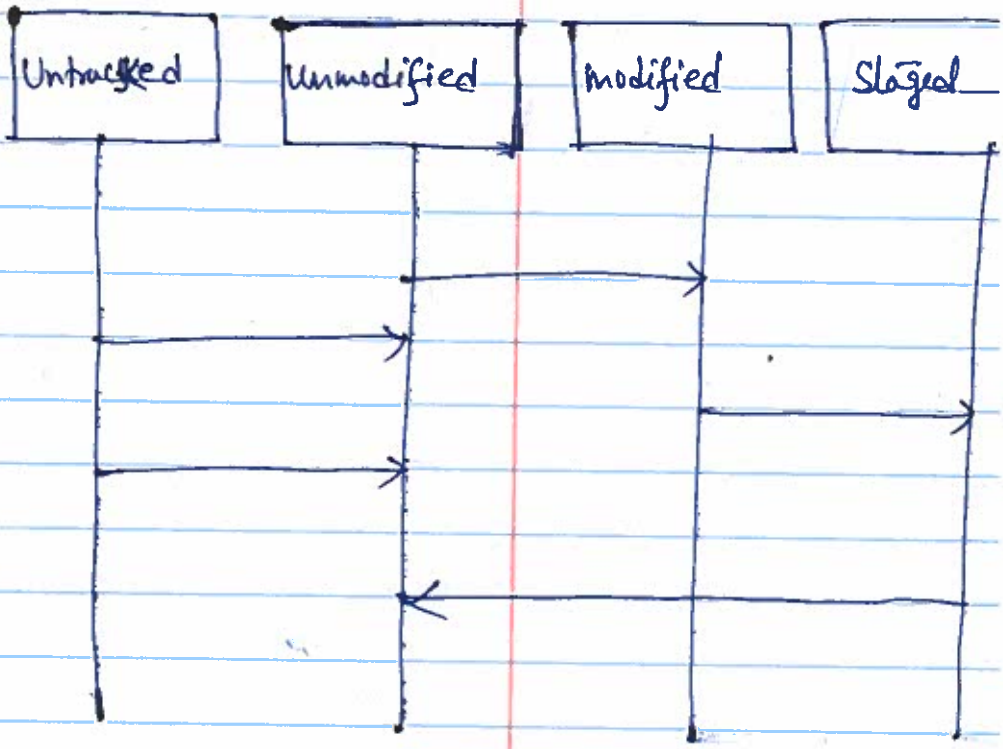
⇒ `ls`

⇒ `git add --a`

⇒ `git commit -m "Changed to tes`

`git log`

Git: File Status lifecycle || Git tutorials #6



i) git init

ii) git add --a

iii) git add "file name"

iv)

it Tutorials #7 || .gitignore! Ignoring files in

(i) .git status

(ii) git commit -m "git tutorial 7"

(iii) touch "newfile.txt"

(iv) touch .gitignore

(v) open ".gitignore" in the repository. then type "newfile.txt"

(vi) so now if you run git. it will not show "newfile.txt"

(vii) git add --a / git add .

(viii) git commit -m "Added gitignore"

(ix) you can use "*" in ".gitignore"

(x) if you want to ignore folder/dir open the .gitignore file and type directory name / e.g NewFolder/

20) *

it Tutorials 8 :- Git diff : Showing Changes betw Commits / Staging area & working Directo

(i) **git diff** [find the difference between working directory and staging area]

(ii) **git diff --staged**

[find the difference between production and current staging area]

Git Tutorials # 9

Git: Skipping The Staging Area

i) `git commit -a -m "Direct commit"` to skip the staging Area.

ii) But you have any untrack file then it will not go to commit area directly. you have to specifically add them.

(ii) `git log`

(iii) q → Exit

Git Tutorials # 10

Moving and Renaming files in Git

Note:- if you rename a file then git thinks the old file you have delet and added a new file.

Note:- if you want to remove a file u can use [git rm "file name"]

he advantage is that it will delet the file, plus it will move the curren status in Staging Area.

① if you want to re-name a file with help of git ("mv" command)

git mv first.txt first_renamed.txt

is command also re-name the file plu ove the file into the staging area.

if you add a file in origin cently then run 'git status' command. and t the same you have done some changes

then it will show the file Staging area. because it was tracking already.

-Now if want the file will not be track please use the command below:-

"git rm --cached filename"

Git Log : Viewing & Changing Commits in Git
Tutorial #11

i) rm -rf .git

ii) git clone "https://github.com/pandas-
pandas.git my panda

first we need to copy the url fro
github then press shift + Insert to
copy then give any folder nam
like "my panda".

iii) git status

If we run this command now we would
this "my panda"
because "my panda" is
a new git repository now.
so to view this we have to go insid
"my panda" repository.

iv) git log → if you see logs fr
initial point

v) git log -p
this command helps us to view the diffe

(vi) q

(vii) git log -p -3

if we write three (3) with this com it will show us 3 (three) commit with all changes.

(viii) git log --stat

it shows us commits summar in short

(ix) git log --pretty = oneline

it shows us all commits in one line.

(x) q

xi) git log --pretty = short

it shows us all commits in short

xii) git log --pretty = full

t will shows us little bit more info

xiii) git log --since = 2 days

it will show us what happen in last two days.

xiii) like this we can write below

git log --since=2.weeks

~~xiv~~ git log --since=2.monts

~~git log => since~~

xv) git log --pretty=format:"%h --an

~~%h~~ %h : abbreviated commit ha

%an : ~~abbreviated~~ author name

%ae: author email

xvi) git commit --amend

it will open "vim" Editor to insert this fi

please press "i" button and then to

save it press "wq" to quit and sav

git commit --ammend

This command helps to change the

commit @ comment.

git scm is the website for git.

git scm useful options for git logs format

staging & Unmodifying Files in Git

Git tutorials #12

```
git init
git status
git add .
git commit -m "Initial commit"
git add first_renamed.txt
git restore --staged first_renamed.txt
```

↓

command is used to unstage any file

```
git checkout -- first_renamed.txt
```

this command helps us to get the previous ~~commit~~ stage.

but I belive it does not change on git ignore file.

```
git checkout --f
```

king with Remote Repositories

Git Tutorials #13

first go to github website and create
↓ gitlab/bitbucket
n account

pull

Push

[to paste in git bash
Please press
shift + Insert]

git remote add origin → we can use different o

git@github.com:codewithHarry/GitTutorialDem

git remote
origin

git remote -V

in git@github.com:CodewithHarry/GitTutorialDe
• git (fetch)

in git@github.com:CodewithHarry/GitTutorialDemo
• git (push)

) git push -u origin master

you don't have permission. Please make su
you have the correct access rights and
the repository exists.

How to grant permission:-

o to github settings : SSH and GPG k

ew SSH keys

Title: - - - - - - - -

ey: "Add SSH Key" showing

~~figure SSH key based secure authentica~~

to google and search "SSH keys github"

enerating a new SSH key and adding it
T the SSH-agent"

-keygen -t rsa -b 4096 -C "your email@
xample.com"

Please check every lines and continue press
Enter

n type the following command if
SH is not working

val $(ssh-agent -s)
Agent pid 1688

1→ ssh -add ~/.ssh/id_rsa
→ add your SSH keys to github account
→ copy the SSH keys to your clipboard.
i/ ~/.ssh/id_rsa.pub
n git push -u origin master

we can see our git repository in "github"
ebsite
u can see all the commits now etc

) suppose we create a new file.
ien run following commands
git status (ii) git add .
) git commit -m "Added this.txt" in github

et push -u origin master
w it go to github then you can see
n this.txt" file

(k

tting Alias in Git (GIT CONFIG etc.)

ppose I want to use "git st" instead o
't status" ie status will be "st" we
n use Alias. How to configure it
Please check

't config --global alias.st status

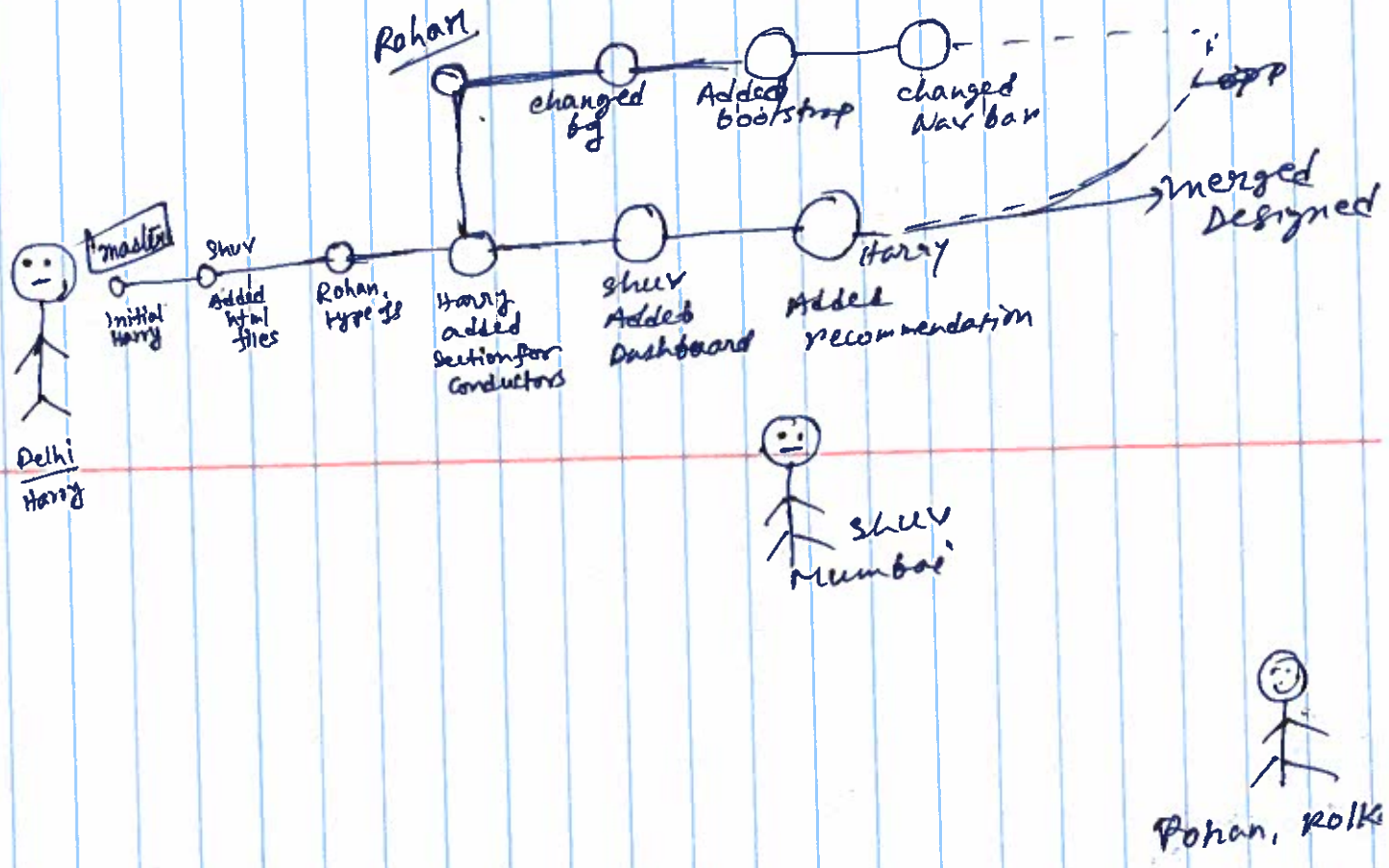
uppose I want to alias for unstage com
't config --global alias.unstage "restore .
--staged --"

ow you can use "git unstage" as a
shortcut

git config --global alias.last 'log -p -1

17

ting & Switching Branches in Git
Git Tutorials #15



git checkout -b develop
It will create new branch and it at the same time it will take you to the new branch.

git add .
git commit -m "beautified structure".
git checkout master
"to switch into the master branch"
rm -rf .git
to remove git directory
git init
initialized git repository
git commit -a -m "Changed ----"
it will directly commit skipping staging
git checkout -b develop
to switch into "develop" branch.
git branch --- it will show all the branches.

Branching & Merging a Production Grad Project | Git Tutorial #16

First you download VS code. VS code is a code editor from Microsoft.

While installing the software, make sure following options are checked

(i) Add "open with code" action to Windows Ex file context menu.

(ii) Add "open with code" action to windows Explorer directory context menu.

(1) Create new project/Folder named "My we right click and select "open wit

(2) Go to "Terminal" and select "new Terminal"

(3) you can use git commands in this ter

But prefers

4) Harry "git bash" terminal, because he can types all the linux commands

5) Goto VS code add a file "index.ht add "!" mark, press Enter, it will

(6) Now you goto "boot strap". com.
Bootstrap.com is an CS and html library
Using the help of bootstrap.com you
create beautiful website within a few min

i) Please copy starter template (you can
ii) it under documentation .... not sur

(7) Now go back to VS code studio,
delete the ~~html~~ html auto fill basic
and paste the starter template from boot

) Now we need to install the 'one ex
- live server".
if it is showing ~~reload~~ disabled the
click on that .... it will show ... reload
.... click on that .... Now click on "Enable

once it will be enabled you will see
"go live" at the bottom right cor.

If you click on the "go live" it will ope
it browser as "Hello world".

(v) Now go to "bootstrap.com" again. Click on "Components" and select "Navba

Now you can copy the Navbar code
Now you paste the code in place of "Hello World"

(vi) (i) git status (ii) git add .
(iii) git commit -m "Initial Commit".
(iv) git checkout -b try cleanup
(v) git checkout master.
(vi) git checkout try cleanup
(vii) git branch to show the branche
(viii) git checkout master
(ix) git merge try cleanup

you can get "Merge conflict" e in Visual Code Studio. --- most of times you get three options. ---
(a) You want these changes, please acc
(b) you want that etc.

(x) git commit [if you not use "-m option]

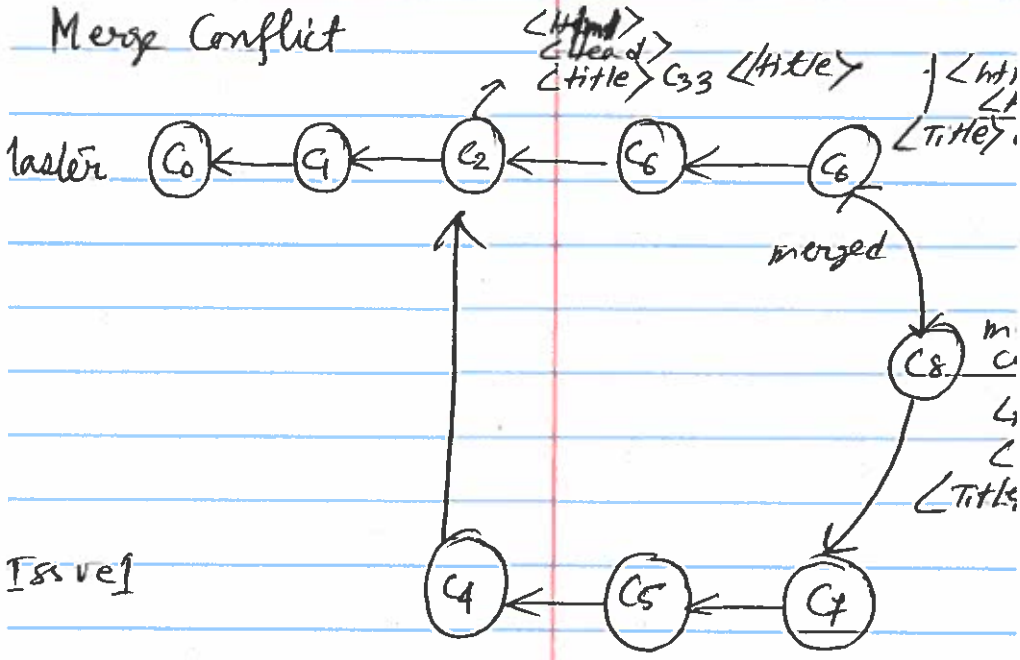
It will open Vim Editor

So, press "Esc" the ":wq" it as save and quit.

"Vim Editor" only available in "git

Resolving Merge Conflicts (with Example) || Git Tutorials #17

Merge Conflict



Conflict resolution markers

<<<<<< Head: index.html

=======

>>>>>> Issue 1: index

Branch Management :-

git Branch

~~master~~
develop
system

git branch -V

* master	Commit hash	
develop	97b13C	"Thi

git branch --merged → already m branc

git branch --no-merged → Not alre branch

Deleting branches

git branch -d develop ⇒ gives error if de is not merged.

git branch -D develop ⇒ change 'd' caps 'D' to dele non-merged bran

---

(i) git status

(ii) git init

(iii) (a) git status

(iv) git add .

(v) first create an index.html

(vi) a) git status (b) git add .

(c) git commit -m " added index.html

(d) git status

. . . please watch the video . . . . .

. . . . I am not writing some lines -

(vi) git checkout -b issue 1

(vii) git commit -a -m " this is c4 ".

. . . . . Again I am not writing some.

. . . . please watch the video . . . .

(viii) git checkout master

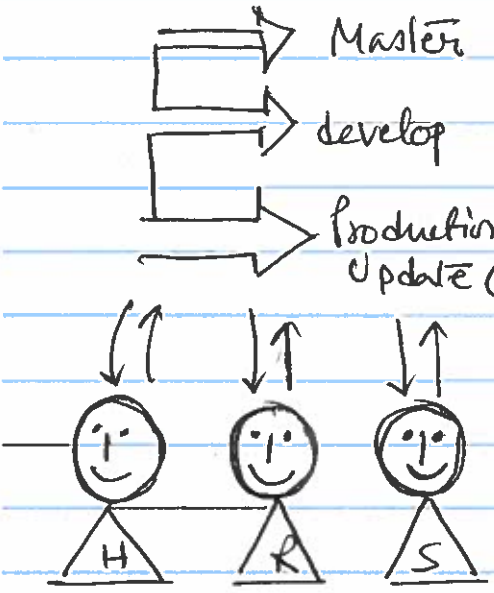
(ix) git merge issue 1

throw "merge conflict" error.

Git Branching Workflow in Product
Git Tutorials #18

Branching Workflow

Long Running Branches



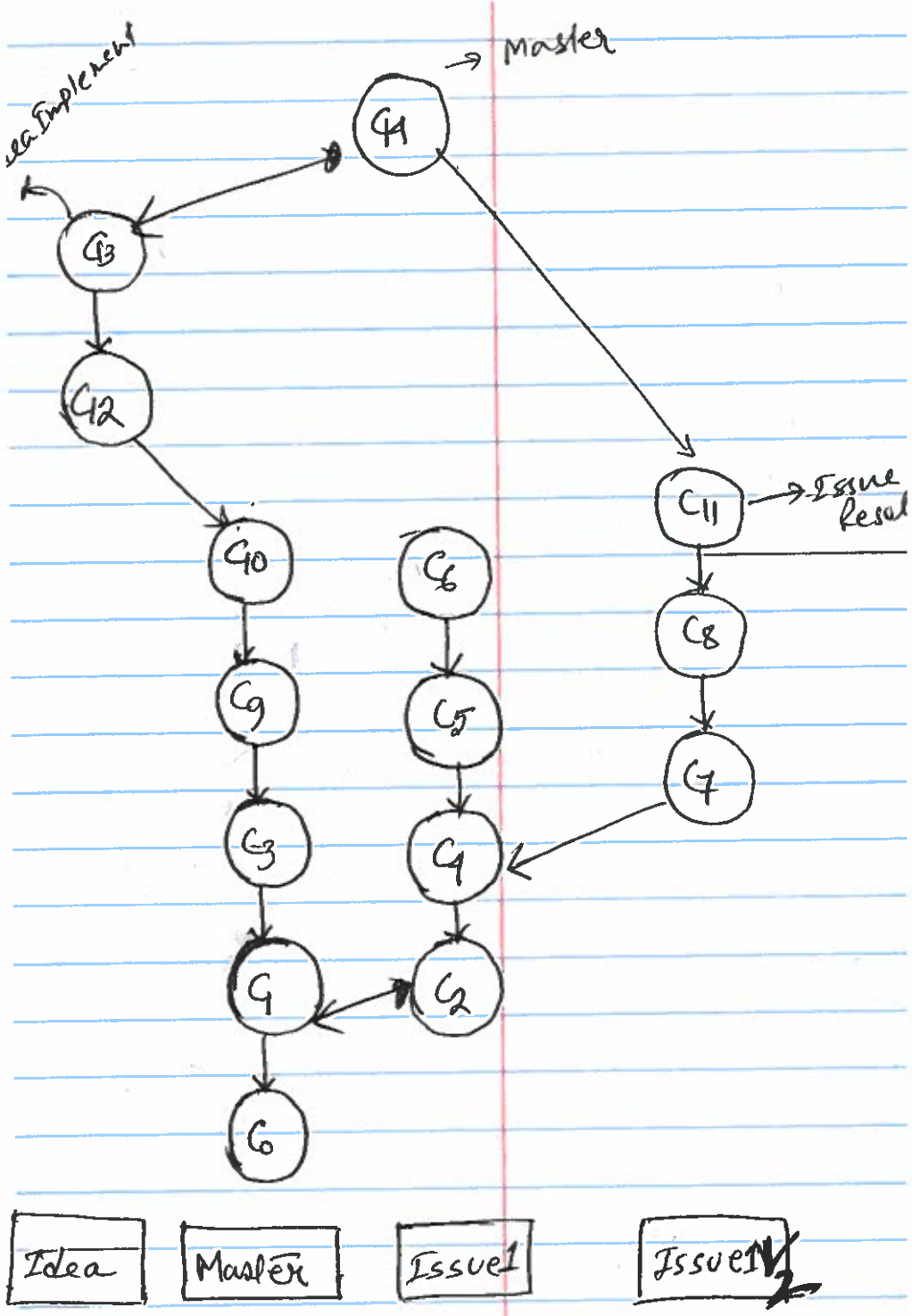
~~This~~ As the name said it runs until the project completed.
Other ~~way to~~ say it exist until the project exist.

Topic Branche

⇒ Replace text typed JS (TypedJsF

As the name s
it exists unt
the topic compl

When the topic completes the bra got deleted.


Master
C1
C3
C12
C10
C9
C3
C
C
C11
Issue
C8
C7
C6
C5
C4
C2
Idea
Master
Issue1
Issue

Pushing Git Branches To Remote Repos
Git Tutorials - #19

(i) git status (ii) git log (iii) git b
(iv) git checkout -b bugfix
(it will create a branch named "bugfix"
& at the same time it will bring me to th
path)
(v) git add i (vi) git commit -m "fixed the
(vii) git checkout master
1) (a) git remote (b) git remote -v

first there will be no git reposito
We have to go to git Website a
need to create a git repository.

→ create new → Repository Name →
Description → create Repository

Push an existing repository from th
command line:-

⇒ git remote add origin https://github.com/code
Harry / Demo repository.git (copy this co

paste it in git bash ~~console~~ console (shift + Insert)

git remote add origin https://github.com/codewithharry/DemoRepository.git

Now git remote and git remote -V will work.

git remote
origin

git remote -V
origin https://github.com/codewithHarry/DemoRepository.git (fetch)
origin https://github.com/codewithHarry/DemoRepository.git (push)

Now copy the second command in git bash shell

git push -u origin master

it will open a small window and ask for credential (user name and password)

Now if you go to github website you can see your master branch over there

Now if you want to push another branch (other than master) in Remote

Please swith to that branch :-

i) git checkout bugfix

ii) git push origin ~~master~~ bugfix

Now if you goto github website. I can see two branches!

if you want to compare then one "co button is availave in github web

You can also merge the file "Abl merge" option.

Recommendation :- if you want to pus some branch. First switch and go that branch.

Another command:-
git push origin bugfix : my bugfix

~~it will push local~~ "bugfix" branch
it will push local "bugfix" branch to remote and at the same time will change the name to "my bugfix".

but this is not good for change name in project.

What is the difference between:-
i) git push -u origin master
ii) git push origin master

~~How~~ How to delete branches from origin

(i) git push -d origin ~~bugfix~~ b
(remote branch)

(ii) git push -d origin mybugfix