**$ git config --global user.name "Utpal"**

**$ git config --global user.email "utpalbhargavsse@gmail.com"**

**$ git config --list**

To know the user details.

**$ git config user.name**

Utpal

**$ git config user.email**

[Utpalbhargavsse@gmail.com](mailto:Utpalbhargavsse@gmail.com)

Always check the status of the directory/folder that is it a Git repository before using it..

**$ git status**

It will give error if its not a git repository.

Now give below command to make the current directory a Git directory.

**$ git init**

You wil get below message and also create a .git folder in your directory. This folder will track all your files now:

Initialized empty Git repository in E:/Git/Git Tutorial/.git/

Also it will tell you that this directory is Master

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

Now check the status again

**$ git status**

Hit enter

On branch master

No commits yet

Untracked files:

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

Gittest1.txt

Gittest2.txt

Practice doc.docx

~$actice doc.docx

~WRL3300.tmp

nothing added to commit but untracked files present (use "git add" to track)

* This says that there is no file which is tracked now(red colour means files are not in staging area). So you have to add those files to staging area to track.. To add all the files to staging area use below command.

**$ git add --a**

- to check the status again

**$ git status**

* This will show status of the files. Green means files are in staging area. It says that files are not commited.

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: Gittest1.txt

new file: Gittest2.txt

new file: Practice doc.docx

new file: ~$actice doc.docx

new file: ~WRL3300.tmp

Changes not staged for commit:

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

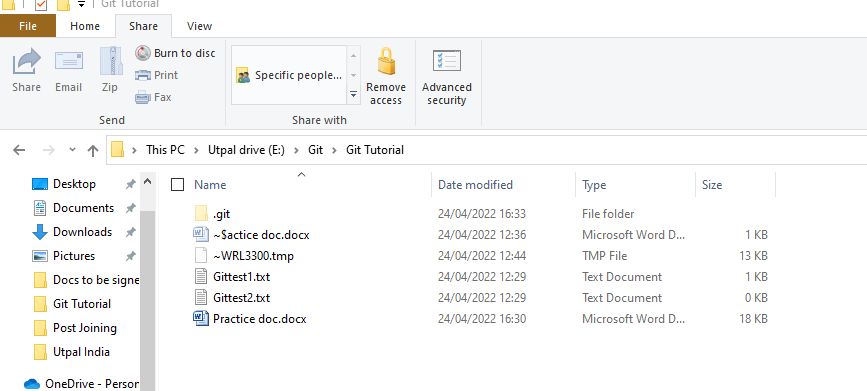
(use "git restore <file>..." to discard changes in working directory)

modified: Practice doc.docx

-Now we need to commit the files atleast first time so that we can have a snapshot and files will reach to git repository. To commit all the files, use below

**$ git commit -m "Initial Commit"**

-m is to provide the message. Below is the actual directory .



* Below is the message in git about committed files.

[master (root-commit) b48df90] Initial Commit

5 files changed, 3 insertions(+)

create mode 100644 Gittest1.txt

create mode 100644 Gittest2.txt

create mode 100644 Practice doc.docx

create mode 100644 ~$actice doc.docx

create mode 100644 ~WRL3300.tmp

-now if we check the status, it will say all ok. Nothing to commit.

**$ git status**

On branch master

nothing to commit, working tree clean

-we can change the file and commit one file also instead of committing all the files by giving the particular file name while committing or adding. Like below

**$ git commit "Practice doc.docx" -m "document is also changed for tutorial"**

[master 757e1f4] document is also changed for tutorial

1 file changed, 0 insertions(+), 0 deletions(-)

rewrite Practice doc.docx (77%)

* To check the log of what all committed so far.

**$ git log**

commit 5b246a8eee2d7950e11e9c0dab0222a640ed6ed5 (HEAD -> master)

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Sun Apr 24 16:47:19 2022 +0530

tutorial file moved to previous folder

commit 757e1f43fb32005d7c56b64343519c8d23e7301b

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Sun Apr 24 16:45:34 2022 +0530

document is also changed for tutorial

commit b48df90637549851141605d3ac5861112b8b1b89

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Sun Apr 24 16:33:05 2022 +0530

Initial Commit

* Above encrypted string after commit line is snapshot of each. We can use this to rollback the changes to that commit.

-what happes if we initialize a already git repository

**$ git init**

Reinitialized existing Git repository in E:/test git/.git

-nothing happens here.

-To remove the .git tracking from any folder, do below. Not recommended.

**$ rm -rf .git**

-Want to see the present working directory.

**$ pwd**

/e/Git/Git Tutorial

-Check what all files are there in the directory

**$ ls**

Gittest1.txt Gittest2.txt 'Practice doc.docx' Tutorial2/

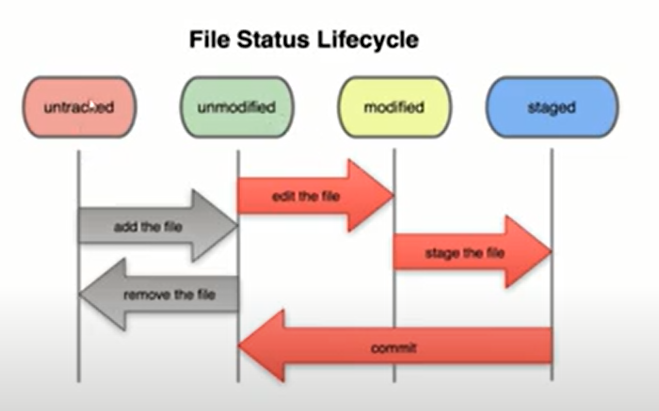
-if you want to go to that directory. It created Tutorial2 as a git direct by default as it was created under git tutorial folder which was a git directory.

**$ cd Tutorial2/**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial/Tutorial2 (master)

-what you want to clone a existing .git repository, use below to clone git tenserflow directory

**$ git clone https://github.com/tensorflow/tensorflow.git**



-To create a new file .

**$ touch error.log**

-to ignore the tracking of any file, create new .gitignore file and put the particular file name insite .gitignore

**$ touch .gitignore**

-Add error.log file into .gitignore file. It will not track it.

**$ git status**

On branch master

Untracked files:

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

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

-put the gitiignore file in .git repository in committed state. And now even if we change error.log file, it will not show it as changed file. We can ignore t=multiple files with same name using \*.log.

-if you want to untrack folder then give folder name with /. If we want to ignore last folder with same name then use / before the folder name like ‘/Test1/.

-If a file is already tracked and then we add that file into .gitignore then also git tracks it.. so we need to use different command to untrack the file. It will show that the file is deleted but it doesn’t delete but removes from the list of tracked file. See below when we change a file, untrack it, git status, then use command then status.

Added Gittest4.txt into .gitignore file. Commit it. And then change it.

$ git status

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: Gittest4.txt

-so use below command to remove it from track list

**$ git rm --cached Gittest4.txt**

-It will remove it from the list

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

deleted: Gittest4.txt

-commit the changes. Then change the file again and see the status:

**$ git commit -m "removed gittest4.txt from tracking list"**

[master 6e67d2d] removed gittest4.txt from tracking list

1 file changed, 0 insertions(+), 0 deletions(-)

delete mode 100644 Gittest4.txt

-now change it again and see the status, it will now track it.

-If folder is empty, it will not track it bydefault.

-Compare files in Staging Area with Working Directory, Use ‘Git Diff’ command.I changed the Gittest1.text file where removed one line and added one line.

**$ git diff**

diff --git a/Gittest1.txt b/Gittest1.txt

index 5c556d6..256afdb 100644

--- a/Gittest1.txt

+++ b/Gittest1.txt

@@ -1,3 +1,5 @@

Hi

This is my first file to practice Git.

-This is test file.

+

+

+Changing in working area to test the 'Git Diff' command

-if we want to compare the staging area with previous commit files then we can use below. To do that, we need to staged the changes and then run the command.

**$ git diff --staged**

diff --git a/Gittest1.txt b/Gittest1.txt

index 5c556d6..256afdb 100644

--- a/Gittest1.txt

+++ b/Gittest1.txt

@@ -1,3 +1,5 @@

Hi

This is my first file to practice Git.

-This is test file.

+

+

+Changing in working area to test the 'Git Diff' command

-if you want to skip the staging area and commit directly from working directory then use below command. It will not commit the files which are untracked means newly added. You will have to first put them in to stagin area. For ex, we added new file Gittest3.txt and changed the Gittest1.txt file. Below is git status

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: Gittest1.txt

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: Gittest1.txt

Untracked files:

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

GitTest3.txt

-Now, if we want to commit directly without staging the files then:

**$ git commit -a -m "Direct commit without stage the files"**

[master c7de81a] Direct commit without stage the files

1 file changed, 3 insertions(+), 1 deletion(-)

-now run the git status

**$ git status**

On branch master

Untracked files:

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

GitTest3.txt

nothing added to commit but untracked files present (use "git add" to track)

-Changing, renaming and removing the files

-Remove the file, it will remove the file and stage also. We can do it manually from window but then we will need to use the git to stage it as well.

**$ git rm GitTest3.txt**

-now see status

**$ git status**

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

deleted: GitTest3.txt

-we can commit the changes as BAU.

-if we want to rename the file from Gittest2.txt to Gittest4.txt then:

**$ git mv Gittest2.txt Gittest4.txt**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git status**

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

renamed: Gittest2.txt -> Gittest4.txt

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ ls**

Gittest1.txt Gittest4.txt 'Practice doc.docx' Tutorial2/ error.log

-To see the changes in last commit. It will show the difference of all commits.

**$ git log -p**

commit 6e67d2d8ffdfe5b54f077e49e2cbe26c66e9c709 (HEAD -> master)

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Mon Apr 25 11:12:23 2022 +0530

removed gittest4.txt from tracking list

**diff --git a/Gittest4.txt b/Gittest4.txt**

**deleted file mode 100644**

**index e69de29..0000000**

commit ab866fe7deca3a8fbe34c69120aca5043618a850

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Mon Apr 25 11:08:39 2022 +0530

added the gittest4.txt into gitignore

**diff --git a/.gitignore b/.gitignore**

**index c24da36..6c643a9 100644**

**--- a/.gitignore**

**+++ b/.gitignore**

@@ -1,2 +1,3 @@

test1/

error.log

+Gittest4.txt

\ No newline at end of file

commit 06dd5ffba0823a1bf16890d0550216c18be97740

Author: Utpal <utpalbhargavsse@gmail.com>

Date: Mon Apr 25 11:06:54 2022 +0530

commited the renamed file

**diff --git a/Gittest2.txt b/Gittest4.txt**

**similarity index 100%**

-to see last 2 or 3 commits in above list.

**$ git log -p -2**

-to see the statistic of log

**$ git log --stat**

-log status in one line

**$ git log --pretty=oneline**

6e67d2d8ffdfe5b54f077e49e2cbe26c66e9c709 (HEAD -> master) removed gittest4.txt from tracking list

ab866fe7deca3a8fbe34c69120aca5043618a850 added the gittest4.txt into gitignore

06dd5ffba0823a1bf16890d0550216c18be97740 commited the renamed file

8d81a76244eb058e690cce72008a34dc61359cfd deleted the GitTest3.txt file by rm command

f54011df0339a84913e5eae53f6fb15a35897583 added third file Gittest3.txt

c7de81ab506d107e5a07f151752626c283141040 Direct commit without stage the files

97cabe5af6265358340c604d2291e30c17e29d60 committed all

8eb2e839dd1aaaa60bc25b25a9d3d0fb0b07e1e5 aaded Test1 folder under Test1 folder

e4600c67739a6e60fef4f33369a3d0cf20c3b3d5 .gitignore file is added

5b246a8eee2d7950e11e9c0dab0222a640ed6ed5 tutorial file moved to previous folder

757e1f43fb32005d7c56b64343519c8d23e7301b document is also changed for tutorial

b48df90637549851141605d3ac5861112b8b1b89 Initial Commit

-to see short description of log

**$ git log --pretty=short**

**$ git log --pretty=full**

**$ git log --since=1day**

**$ git log --since=1month**

-call back the file from stage area to working directory.below files are in staged area, use git restore command to call back file.

**$ git status**

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: Gittest1.txt

new file: Gittest2.txt

**$ git restore --staged Gittest2.txt**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git status**

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: Gittest1.txt

Untracked files:

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

Gittest2.txt

-to remove the unwanted changes and restore the previous commit file

**$ git checkout -- Gittest1.txt**

-to remove all changes from staging area and refresh it from prevous commit

**$ git checkout -f**

-To create new branch from Master

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git checkout -b newbranch**

Switched to a new branch 'newbranch'

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (newbranch)

-to switch back to master

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (newbranch)

**$ git checkout master**

Switched to branch 'master'

D Gittest4.txt

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

-switch back to branch

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git checkout newbranch**

Switched to branch 'newbranch'

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (newbranch)

-to get the list of branchs

**$ git branch**

master

\* newbranch

-to get the commit history of each branch

**$ git branch -v**

-To show all merged branches

**$ git branch –merged**

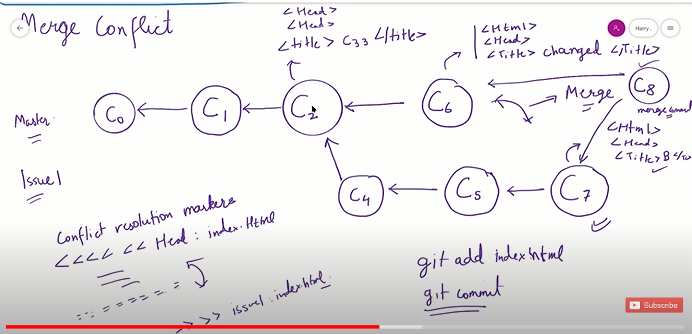
-to show all non merged branches

**$ git branch –no-merged**

-to delete the branch

**$ git branch –d newbranch**

--To merge and resolve conflict of branch



-first do the above.. create new file of cobol1 name and do above commits. Then crete new branch issue1 and do above. Then come to master and commit C6.. now try to merge, Here version of master and branch are different. It will show conflict

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git merge issue1**

Auto-merging Cobol1.txt

CONFLICT (content): Merge conflict in Cobol1.txt

Automatic merge failed; fix conflicts and then commit the result.

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master|MERGING)

-It changed the cobol1 text file and showed the conflict like below. We can accept any change and remove the unwanted one. Then use command Git Add cobol1.txt

Index

<<<<<<< HEAD

Message 'This is message 6'

=======

Message 'This is message 7'

>>>>>>> issue1

-removed the message 7 line from file and added it.

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master|MERGING)

**$ git status**

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

(use "git merge --abort" to abort the merge)

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: Cobol1.txt

no changes added to commit (use "git add" and/or "git commit -a")

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master|MERGING)

**$ git add Cobol1.txt**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master|MERGING)

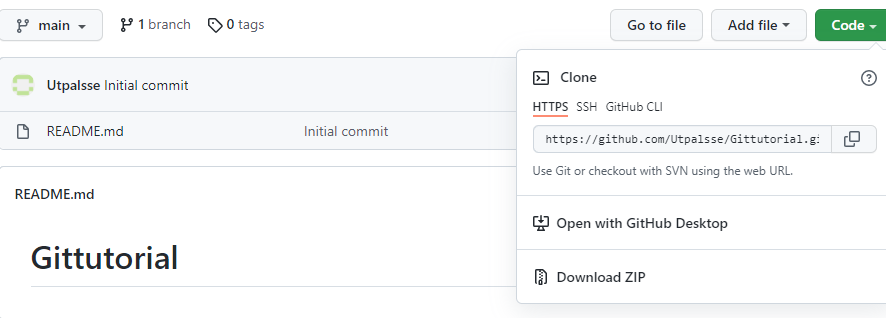
**$ git commit -m "resolved conflict"**

[master 6f1fb4b] resolved conflict

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

-Push from Git bash to Github

-Create account on Github, create repository, now get the HTTPS clone link from github for that repository, use it in Git Bash as mentioned below. Then use push command.. It will ask for userid and password then wil do automatically.



Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git remote**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git remote add origin https://github.com/Utpalsse/Gittutorial.git**

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git remote**

origin

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git remote -v**

origin https://github.com/Utpalsse/Gittutorial.git (fetch)

origin https://github.com/Utpalsse/Gittutorial.git (push)

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git push -u origin master**

Enumerating objects: 37, done.

Counting objects: 100% (37/37), done.

Delta compression using up to 4 threads

Compressing objects: 100% (21/21), done.

Writing objects: 100% (37/37), 58.68 KiB | 3.67 MiB/s, done.

Total 37 (delta 9), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (9/9), done.

remote:

remote: Create a pull request for 'master' on GitHub by visiting:

remote: https://github.com/Utpalsse/Gittutorial/pull/new/master

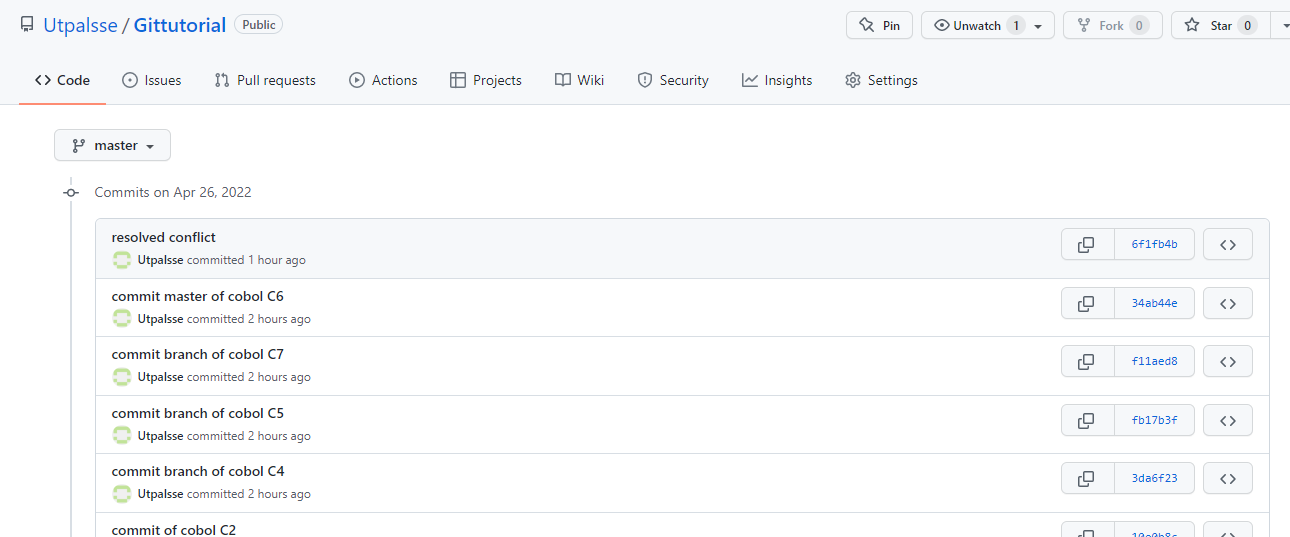
remote:

To https://github.com/Utpalsse/Gittutorial.git

\* [new branch] master -> master

branch 'master' set up to track 'origin/master'.

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)



-if you want to puch branches as well on Github, check what all branches are thre and push.

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git branch**

issue1

\* master

newbranch

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (master)

**$ git push origin issue1**

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

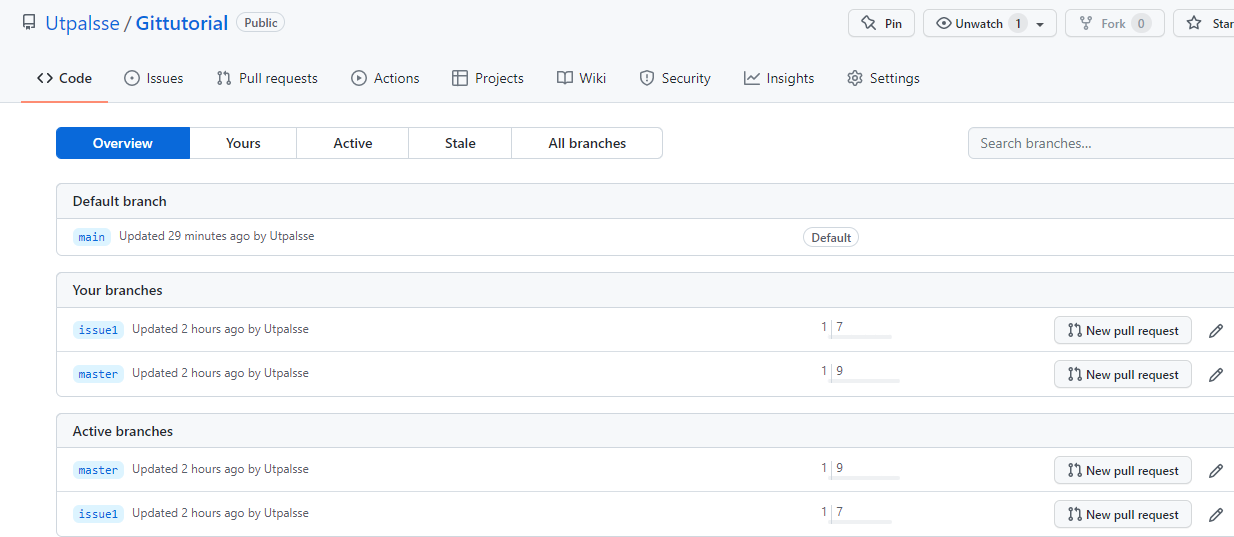
remote: Create a pull request for 'issue1' on GitHub by visiting:

remote: https://github.com/Utpalsse/Gittutorial/pull/new/issue1

remote:

To https://github.com/Utpalsse/Gittutorial.git

\* [new branch] issue1 -> issue1



-Delete a branch from github

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (newbranch)

**$ git push -d origin newbranch**

To https://github.com/Utpalsse/Gittutorial.git

- [deleted] newbranch

Mahendra@CSTiwari MINGW64 /e/Git/Git Tutorial (newbranch)

$