<https://github.com/Kunena/Kunena-Forum/wiki/Create-a-new-branch-with-git-and-manage-branches>

<https://www.freecodecamp.org/forum/t/push-a-new-local-branch-to-a-remote-git-repository-and-track-it-too/13222>

<https://git-scm.com/docs/git-push/1.6.2>

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

What is the difference between Git and GitHub?

Git is a distributed version control tool that can manage a development project's source code history, while GitHub is a cloud based platform built around the Git tool. Git is a tool a developer installs locally on their computer, while GitHub is an online service that stores code pushed to it from computers running the Git tool. The key difference between Git and GitHub is that Git is an open-source tool developers install locally to manage source code, while GitHub is an online service to which developers who use Git can connect and upload or download resources.

==========================================================

How to config git in our local machine using gitbash??

s

gaura@LAPTOP-PPU5V8NV MINGW64 /d/study/git/training

$ git config --global user.name "8440Gau"

gaura@LAPTOP-PPU5V8NV MINGW64 /d/study/git/training

$ git config --global user.email "8440gaurav@gmail.com"

Finding location of files??

git config --list --show-origin

file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt

file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl

file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain

file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true

file:C:/Program Files/Git/etc/gitconfig core.fscache=true

file:C:/Program Files/Git/etc/gitconfig core.symlinks=false

file:C:/Program Files/Git/etc/gitconfig credential.helper=manager

file:C:/Users/gaura/.gitconfig user.name=8440Gau

file:C:/Users/gaura/.gitconfig user.email=8440gaurav@gmail.com

file:C:/Users/gaura/.gitconfig core.editor=notepad++

Editing config file ??

$ git config core.editor "notepad++ -multiInst -nosession"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

How to create a directory in Git??

1. git init "Name of Directory"

2. when a directory is created it always been having new branch which has been created and that will be master branch.

3. to check list of file which are in directory we use

$ ls -la

total 8

drwxr-xr-x 1 gaura 197609 0 Nov 6 22:02 ./

drwxr-xr-x 1 gaura 197609 0 Nov 6 20:57 ../

-rw-r--r-- 1 gaura 197609 0 Nov 6 22:02 backupofgitconfig.txt

-rw-r--r-- 1 gaura 197609 1245 Nov 6 22:06 test.txt

4. to check the status of repository :

git status

5.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Repositories are building block of Github.

git remote add origin

1.Repositories are like ”Folder” for your project .Repository are 2 types private and public.

2. ReadMe file contain information and instruction about the code and repository. It will be in .md

Format.

3. Contributing and contributors are files contain information about coder and coding and organization

Which have contributed in repository.

4. ChangeLog file contain information about list of changes different version of project.

5. Support this file contain about the people from whom you can get help.

6. Code of conduct file it contain rule by which we have to do coding and follow it.

Cloning to local Machine

**$ git clone https://github.com/8440Gau/training.git**

Cloning into 'training'...

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (10/10), done.

remote: Total 13 (delta 1), reused 12 (delta 0), pack-reused 0

Unpacking objects: 100% (13/13), done.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2 (master)

$ cd training

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ ls –la

total 23

drwxr-xr-x 1 GASINGH 1049089 0 Nov 7 15:21 ./

drwxr-xr-x 1 GASINGH 1049089 0 Nov 7 15:21 ../

drwxr-xr-x 1 GASINGH 1049089 0 Nov 7 15:21 .git/

-rw-r--r-- 1 GASINGH 1049089 6982 Nov 7 15:21 command.txt

-rw-r--r-- 1 GASINGH 1049089 17 Nov 7 15:21 README.md

drwxr-xr-x 1 GASINGH 1049089 0 Nov 7 15:21 ssh/

-rw-r--r-- 1 GASINGH 1049089 46 Nov 7 15:21 test2.js

-rw-r--r-- 1 GASINGH 1049089 704 Nov 7 15:21 Timer.js

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

Adding

Adding New files in Local Repository.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2 (master)

$ cd training

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Untracked files:

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

assets/

favicon.ico

images/

index.html

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

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git add .

warning: LF will be replaced by CRLF in assets/css/font-awesome.min.css.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in assets/css/main.css.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in assets/fonts/fontawesome-webfont.svg.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in assets/js/jquery.min.js.

The file will have its original line endings in your working directory

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git commit -m "Added website files"

[master f89cd6e] Added website files

14 files changed, 3905 insertions(+)

create mode 100644 assets/css/font-awesome.min.css

create mode 100644 assets/css/main.css

create mode 100644 assets/fonts/FontAwesome.otf

create mode 100644 assets/fonts/fontawesome-webfont.eot

create mode 100644 assets/fonts/fontawesome-webfont.svg

create mode 100644 assets/fonts/fontawesome-webfont.ttf

create mode 100644 assets/fonts/fontawesome-webfont.woff

create mode 100644 assets/fonts/fontawesome-webfont.woff2

create mode 100644 assets/js/jquery.min.js

create mode 100644 favicon.ico

create mode 100644 images/bg.jpg

create mode 100644 images/gitreadmefile.PNG

create mode 100644 images/pic02.jpg

create mode 100644 index.html

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git push origin master

Enumerating objects: 22, done.

Counting objects: 100% (22/22), done.

Delta compression using up to 4 threads

Compressing objects: 100% (20/20), done.

Writing objects: 100% (21/21), 5.79 MiB | 503.00 KiB/s, done.

Total 21 (delta 0), reused 0 (delta 0)

To https://github.com/8440Gau/training.git

a54b291..f89cd6e master -> master

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

**git fetch** is a primary command used to download contents from a remote repository. **git fetch** is used in conjunction with **git** remote , **git** branch , **git** checkout , and **git** reset to update a local repository to the state of a remote. The **git fetch** command is a critical piece of collaborative **git** work flows.

Git fetch download the changes from the Remote Repository and update it in the local git repo.

After that we want to merge things we can do it manually.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git fetch

remote: Enumerating objects: 8, done.

remote: Counting objects: 100% (8/8), done.

remote: Compressing objects: 100% (5/5), done.

remote: Total 6 (delta 3), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (6/6), done.

From https://github.com/8440Gau/training

7511a92..267549c master -> origin/master

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git status

On branch master

Your branch is behind 'origin/master' by 3 commits, and can be fast-forwarded.

(use "git pull" to update your local branch)

nothing to commit, working tree clean

**git pull** is a Git command used to update the local version of a repository from a remote.

It is one of the four commands that prompts network interaction by Git. By default, git pull does two things.

1. Updates the current local working branch (currently checked out branch)
2. Updates the remote tracking branches for all other branches.

git pull fetches (git fetch) the new commits and merges [(git merge)](https://guide.freecodecamp.org/git/git-merge) these into your local branch

git pull perform 2 things fetch from the remote repository and merge.

The merge notice that we do not have conflicting chages.

Example: we have made change in Index.html directly in Github Traning repository.

But locally we have not change in index.html and instruction.txt is a new file.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git pull

Updating f89cd6e..267549c

Fast-forward

Instruction.txt | 1 +

index.html | 4 ++--

2 files changed, 3 insertions(+), 2 deletions(-)

create mode 100644 Instruction.txt

When we made change locally as well as remotely same time.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ notepad++ "index.html"

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git add .

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git commit -m "changed index.html"

[master d66da29] changed index.html

1 file changed, 1 insertion(+), 1 deletion(-)

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git push origin master

To https://github.com/8440Gau/training.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/8440Gau/training.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git pull

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/8440Gau/training

d66da29..de4f157 master -> origin/master

Merge made by the 'recursive' strategy.

Instruction.txt | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git push origin master

Enumerating objects: 9, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 4 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (5/5), 565 bytes | 188.00 KiB/s, done.

Total 5 (delta 3), reused 0 (delta 0)

**Branch in github**

Use a branch to isolate development work without affecting other branches in the repository. Each repository has one default branch, and can have multiple other branches. You can merge a branch into another branch using a pull request.

You can use branches to:

* Develop features
* Fix bugs
* Safely experiment with new ideas

Working with local git branch command:

1. Git branch “branch Name” 🡺 use to create a new git branch
2. Git checkout “branchName”🡺 use to switch the git branch.
3. Git push –u origin branch

-u(update)🡺 it will specific the origin of code.

-u 🡺 this parameter is going to be used to create a tracking relation between what is locally my master and my origin have.

Note : Working with Pull request.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

**Create the branch on your local machine and switch in this branch**

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git checkout -b "add-installation"

Switched to a new branch 'add-installation'

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ notepad++ install.txt

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git add .

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git commit -m "new file"

[add-installation 4adb672] new file

1 file changed, 279 insertions(+)

create mode 100644 install.txt

**Push the branch on github :**

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git push -u origin install

error: src refspec install does not match any

error: failed to push some refs to 'https://github.com/8440Gau/training.git'

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git push -u origin install add-installation

error: src refspec install does not match any

error: failed to push some refs to 'https://github.com/8440Gau/training.git'

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git push -u origin add-installation

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 4 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 2.92 KiB | 997.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

remote:

remote: Create a pull request for 'add-installation' on GitHub by visiting:

remote: https://github.com/8440Gau/training/pull/new/add-installation

remote:

To https://github.com/8440Gau/training.git

\* [new branch] add-installation -> add-installation

Branch 'add-installation' set up to track remote branch 'add-installation' from 'origin'.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation

**You can see all the branches created by using :**

$ git branch –a

**Add a new remote for your branch :**

$ git remote add [name\_of\_your\_remote] [name\_of\_your\_new\_branch]

**Push changes from your commit into your branch :**

$ git push [name\_of\_your\_new\_remote] [url]

# [How to change the URI (URL) for a remote Git repository?](https://stackoverflow.com/questions/2432764/how-to-change-the-uri-url-for-a-remote-git-repository)

git remote set-url origin new.git.url/here

**Update your branch when the original branch from official repository has been updated :**

$ git fetch [name\_of\_your\_remote]

**Then you need to apply to merge changes if your branch is derivated from develop you need to do :**

$ git merge [name\_of\_your\_remote]/develop

**Delete a branch on your local filesystem :**

$ git branch -d [name\_of\_your\_new\_branch]

**To force the deletion of local branch on your filesystem :**

$ git branch -D [name\_of\_your\_new\_branch]

**Delete the branch on github :**

$ git push origin :[name\_of\_your\_new\_branch]

**If you want create a new branch:**

$ git branch <name\_of\_your\_new\_branch>

below is used to a add a new remote:

git remote add origin git@github.com:User/UserRepo.git

below is used to change the url of an existing remote repository:

git remote set-url origin git@github.com:User/UserRepo.git

below will push your code to the master branch of the remote repository defined with origin and -u let you point your current local branch to the remote master branch:

git push -u origin master

**If you want to push newly created branch from local git to github**

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git branch local

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git checkout local

Switched to branch 'local'

M Command.docx

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (local)

$ git push -u origin local

Total 0 (delta 0), reused 0 (delta 0)

remote: This repository moved. Please use the new location:

remote: https://github.com/8440Gau/training.git

remote:

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

remote: https://github.com/8440Gau/training/pull/new/local

remote:

To https://github.com/8440Gau/Training.git

\* [new branch] local -> local

Branch 'local' set up to track remote branch 'local' from 'origin'.

**Creating a new Branch and switching it to new branch and pushing the branch from local to the github repo.**

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git branch local

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git checkout local

Switched to branch 'local'

M Command.docx

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (local)

$ git push -u origin local

Total 0 (delta 0), reused 0 (delta 0)

remote: This repository moved. Please use the new location:

remote: https://github.com/8440Gau/training.git

remote:

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

remote: https://github.com/8440Gau/training/pull/new/local

remote:

To https://github.com/8440Gau/Training.git

\* [new branch] local -> local

Branch 'local' set up to track remote branch 'local' from 'origin'.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (local)

$ git checkout master

Switched to branch 'master'

M Command.docx

Your branch is up to date with 'origin/master'.

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git push -u origin master

Everything up-to-date

Branch 'master' set up to track remote branch 'master' from 'origin'.

Tags

1. throught tagging we can highlight certain point have certain

value.

 Git has the ability to tag specific points in a repository’s history as being important. Typically, people use this functionality to mark release points (v1.0, v2.0 and so on). In this section, you’ll learn how to list existing tags, how to create and delete tags, and what the different types of tags are.

2 ways to create tag

* 1. Lightweight
  2. Annotated (message,checksum)

Tagging command:

git tag 🡺 for tagging

git log 🡺 for log

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git log --oneline --graph --decorate –all

\* 4adb672 (HEAD ->add-installation, origin/add-installation) new file

\* d424b89 (tag: stable, origin/master, origin/HEAD, master) Merge branch 'master' of https://github.com/8440Gau/training

|\

| \* de4f157 updating

\* |d358058 changed index.html again

|/

\* d66da29 changed index.html

\* 267549c update again

\* 69662e3 creating New

\* 7511a92 updating it

\* f89cd6e Added website files

\* a54b291 command txt sending

\* 7cf9590 other files

\* 9d2f454 Initial Commit

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git tag -a v0.1 -m "0.1 release" f89cd6e

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git tag

stable

v0.1

**How to push git tag in remote repository ?**

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (add-installation)

$ git push --tags

Enumerating objects: 1, done.

Counting objects: 100% (1/1), done.

Writing objects: 100% (1/1), 159 bytes | 31.00 KiB/s, done.

Total 1 (delta 0), reused 0 (delta 0)

To https://github.com/8440Gau/training.git

\* [new tag] stable -> stable

\* [new tag] v0.1 -> v0.1

**How to delete git tag in local repository ?**

git tag –d “name of tag” (local)

git push origin :name of tag

Working With Forks

1. Forks are copy of repository.
2. Doesnot impact original repository.
3. Change

**Changing a remote's URL**

The git remote set-url command changes an existing remote repository URL.

The git remote set-url command takes two arguments:

* An existing remote name. For example, origin or upstream are two common choices.
* A new URL for the remote. For example:
  + If you're updating to use HTTPS, your URL might look like:

https://github.com/*USERNAME*/*REPOSITORY*.git

* + If you're updating to use SSH, your URL might look like:

[git@github.com:*USERNAME*/*REPOSITORY*.git](mailto:git@github.com:USERNAME/REPOSITORY.git)

### [**Switching remote URLs from SSH to HTTPS**](https://help.github.com/en/github/using-git/changing-a-remotes-url#switching-remote-urls-from-ssh-to-https)

1. Open Git Bash.
2. Change the current working directory to your local project.
3. List your existing remotes in order to get the name of the remote you want to change.
4. $ git remote -v
5. > origin git@github.com:USERNAME/REPOSITORY.git (fetch)

> origin git@github.com:USERNAME/REPOSITORY.git (push)

1. Change your remote's URL from SSH to HTTPS with the git remote set-url command.

$ git remote set-url origin https://github.com/USERNAME/REPOSITORY.git

1. Verify that the remote URL has changed.
2. $ git remote -v
3. # Verify new remote URL
4. > origin https://github.com/USERNAME/REPOSITORY.git (fetch)

> origin https://github.com/USERNAME/REPOSITORY.git (push)

The next time you git fetch, git pull, or git push to the remote repository, you'll be asked for your GitHub username and password.

* If you have [two-factor authentication](https://help.github.com/en/articles/securing-your-account-with-two-factor-authentication-2fa) enabled, you must [create a personal access token](https://help.github.com/en/articles/creating-a-personal-access-token-for-the-command-line) to use instead of your GitHub password.
* You can [use a credential helper](https://help.github.com/en/articles/caching-your-github-password-in-git) so Git will remember your GitHub username and password every time it talks to GitHub.

Example :

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git remote set-url origin https://github.com/8440Gau/Learning.git

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ remote -v

bash: remote: command not found

GASINGH@DIN16005503 MINGW64 ~/Desktop/git/Learning/DemoApp2/training (master)

$ git remote -v

origin https://github.com/8440Gau/Learning.git (fetch)

origin https://github.com/8440Gau/Learning.git (push)

### [**Switching remote URLs from HTTPS to SSH**](https://help.github.com/en/github/using-git/changing-a-remotes-url#switching-remote-urls-from-https-to-ssh)

1. Open Git Bash.
2. Change the current working directory to your local project.
3. List your existing remotes in order to get the name of the remote you want to change.
4. $ git remote -v
5. > origin https://github.com/USERNAME/REPOSITORY.git (fetch)

> origin https://github.com/USERNAME/REPOSITORY.git (push)

1. Change your remote's URL from HTTPS to SSH with the git remote set-url command.

$ git remote set-url origin git@github.com:USERNAME/REPOSITORY.git

1. Verify that the remote URL has changed.
2. $ git remote -v
3. # Verify new remote URL
4. > origin git@github.com:USERNAME/REPOSITORY.git (fetch)

> origin git@github.com:USERNAME/REPOSITORY.git (push)