**Importance of Git and GitHub (BLP Session ALKA)**

Pathway: <https://degreed.com/pathway/08qdegrlpr/pathway?path=git>  
  
1) [Blended Learning Program Git & GitHub Mentoring Session 1 - 9th August 2023 - 02\_00 PM TO 05\_00 PM - 23071928RCTechNext-20230809\_140636-Meeting Recording.mp4](https://capgemini-my.sharepoint.com/:v:/p/alka_jhanwar/EV8w7zaCpdNPkt2IMSicUioB0L9kNomM_yeXEbB0ZW5E8Q?isSPOFile=1)

2) [Blended Learning Program Git & GitHub Mentoring Session 2 - 16th August 2023 - 02\_00 PM TO 05\_00 PM - 23071928RCTechNext-20230816\_140508-Meeting Recording.mp4](https://capgemini-my.sharepoint.com/:v:/p/alka_jhanwar/ERhLJ-Fmuw5JjVVoWcN_064BtLAPMuIiIBCc0Yz6o4KgUg?e=masJGb)

3) [Blended Learning Program Git & GitHub Mentoring Session 3 - 21st August 2023 - 02\_00 PM TO 05\_00 PM - 23071928RCTechNext-20230821\_140543-Meeting Recording.mp4](https://capgemini-my.sharepoint.com/:v:/p/alka_jhanwar/EZ4yT2hJvDpGm2mKWLIUz7gBjZ2UZJi4DKuJWXluYvhPQg)

1) What is Git? and why we use git

-> \* Git is a popular source controller

\* Distributed system

\* Free and open source

\* Git is a repository, where we commit our objects

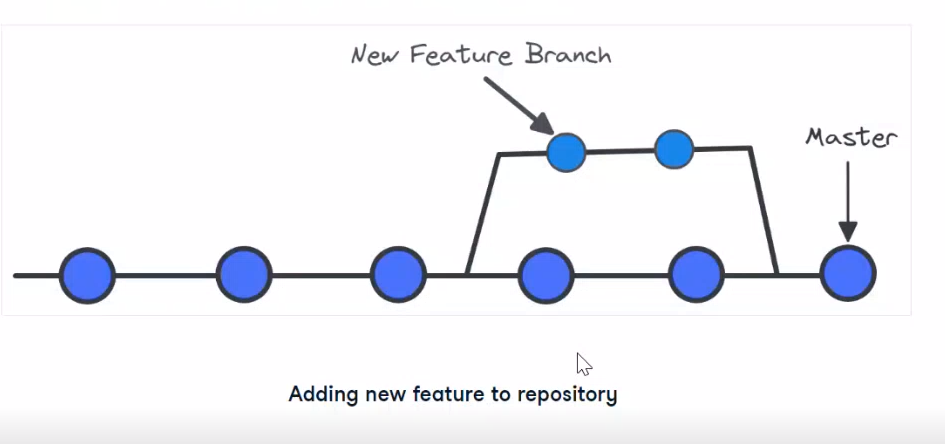
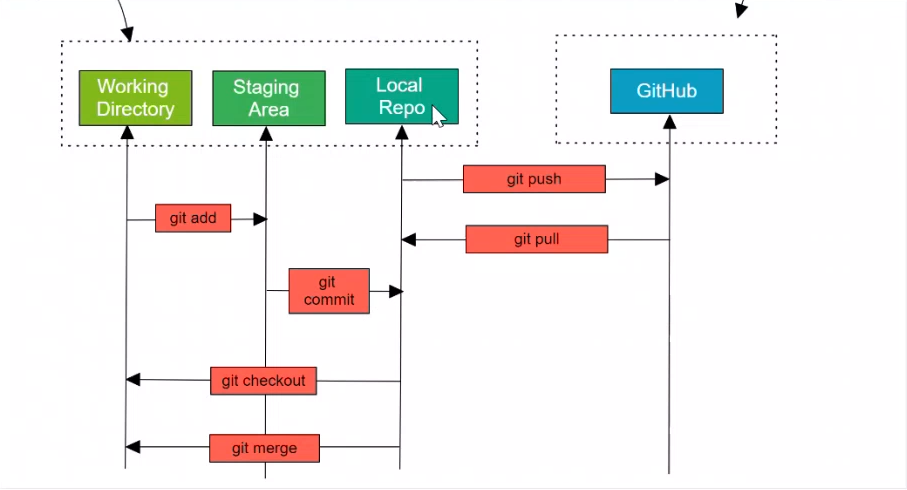
\* This is for control to track and maintain the developer changes

\* So, Git is version control system

\* Meaning of version control is managing of our source codes

\* It can help us for multiple users can work together and we can handle the larger projects efficiently using git and GitHub

2) Difference of git and GitHub?  
-> **GIT**: Git is for managing our repository in locally.  
 **GitHub**: We using GitHub create a centralized repository or remote repository.

**some Linux commands are:**

\* Check the present working directory pwd

use (pwd) command ex: it shows /c/Users/varunkm

\* Us (ls) command list all our data

\* (git --version) it will display git version

\* Configure our username and email

git config --global user.name "varun kumar"

git config --global user.email varunkumarkm44@gmail.com

\* **git init**: it is initializing a git repository

\* **git status:** It shows a status

\* **git add . or git add filename**

\* **git commit -m "initial commit" add the commit message**

\* We want to check which user is configure and which email and username used

git config --list

\* We want to check the username

git config --global user.name

\* Same way we want to check email

git config --global user. email

\* Once we add, we need to remove the staging

git restore --staged file name

\* We want to open the folder in vscode

code .

\* How to add a new commit message

git commit --named -m ""

\* We can check the logs of our commit

git log

\* We want to check the online log

git log --online

For every commit one id is created

\* Committing a modified file (This command is only work for when we modified the file) it’s adding all the files which we modified

git commit -a -m "file name"

\* it's showing a difference of modified file and committed file, showing what new things are added

git diff

after git add . use this command, the check git diff it’s not showing anything

after we check using this command (git diff --staged)

\* If we want to create a new file, we use this command

touch file name ex: testing.html

\*Sometimes we working we don't want to add our local repository

we create one file (.gitignore) on our IDE

after we specify a file name which we do not add to stage, then that file is not showing to commit

we can write any number of log file on .gitignore file we use on more command \*.log

when we not adding an any specified name we use /filename

\* How we delete our local repository

rm -rf .git

\* Remove file for the IDE

rm filename

\* We want to remove the committed file

git rm file name

\* We want to rename the file name

git mv initialname newname

**Branches:**

\* Why we created the branches

Because when we are working on the project multiple people can working, all people are not working on the master branch

what we can do, we create a new branch form the existing version then the person can work on that branch and then

if the task is completed it can be merge with the master branch.

\* For each people can working different branches, and whenever the task is completed then finally merge with the master branch

\* We check the how many branches are there

git branch

\* How to create a branch

git branch branchname

ex: git branch development

\* How we switch to the development branch

git checkout development

\* git checkout command is used to switch the one branch to another

\* I wanted to merge with master branch to our development branch

git merge development (But current branch is master)

\* Merge all the development files, we come out to initial page use escape and: wq

\* Another way to create a branch name

git checkout -b testing

Instead of we are using two commands here we use only one command

\* We need to check how many branches are merged on master branch

git branch --merged

\* We need to check not merged branches

git branch --no-merged