Git

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

What does Git do?

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

Working with Git

- Initialize Git on a folder, making it a **Repository**
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered **modified**
- You select the modified files you want to **Stage**
- The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

GitHub

- Git is not the same as GitHub.
- GItHub is hosting website. It is a website that host git repositories.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

GitHub Installation

Install the Git command line interface.

2. Verify your Git version.

3. Explore Git help.

4. Configure your user name, email address and default Git editor.

5. Then we will see important commands That we use a most like add, commit, log status, push ,pull, flags etc.

GitHub Installation

Open browser: git install

Click on the git SCM website

After installation:: open gitbash and check git is installed or not

Commands:

1. git
2. git --version

GitHub Installation

Initialize your directory with git, set your username and email:

Open gitbash

cd D: {go inside D folder}

dir {enter

mkdir int301git

cd int301git

a. Check status

git status

<not a git repository…. Git is not initialized on this directory>

GitHub Installation

Initialize this directory as git

<while being into 'int301git' directory, run command>

git init

Set global name and email in git config file:

Initialize your directory with git, set your username and email

git config --global user.name "manjot"

git config -- global user.email "manjot.28925@lpu.co.in"

GitHub Installation

To verify name and email added or not?

```
git config --global user.name
git config --global user.email
```

Edit config file and update name or email:

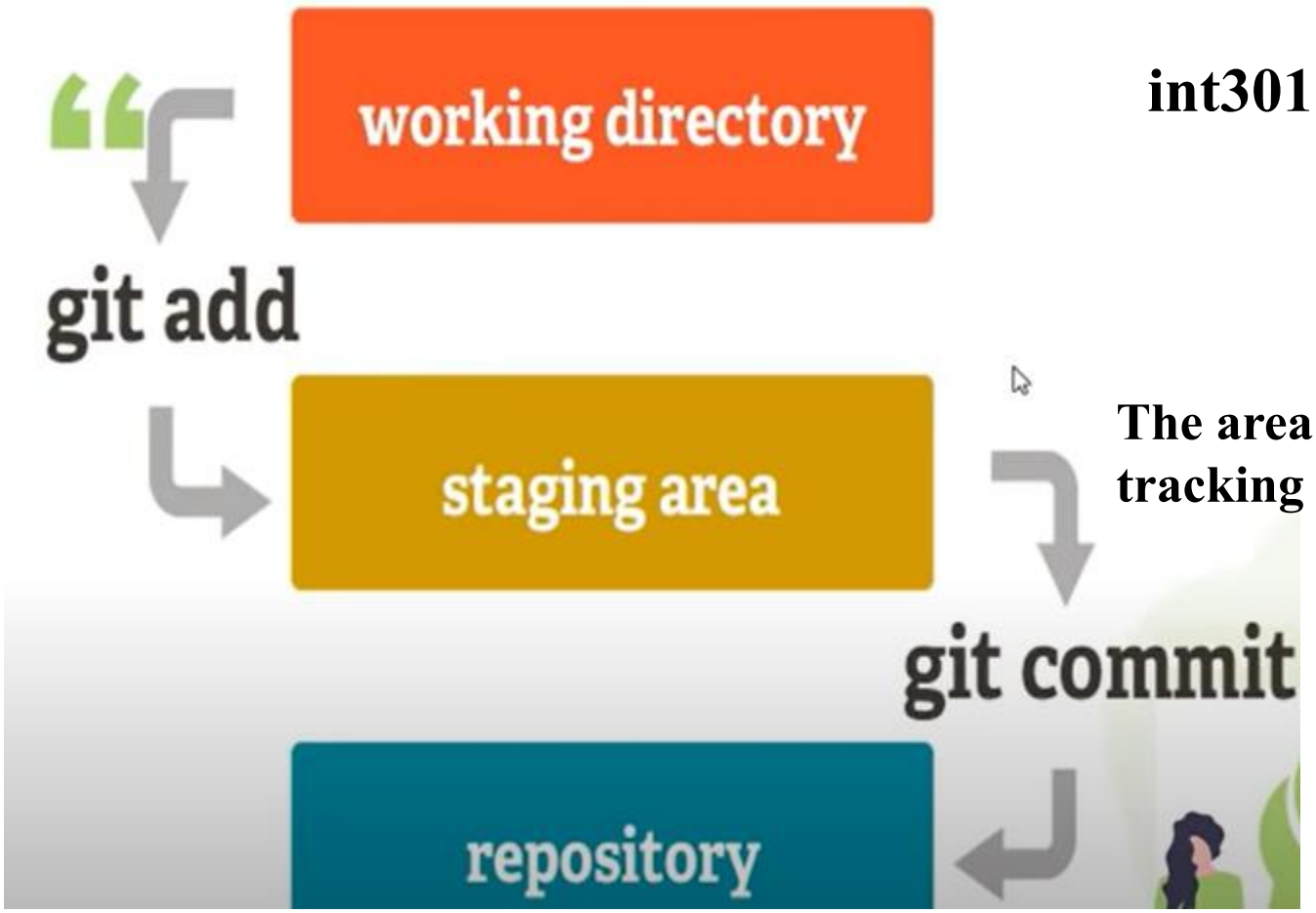
```
git config --global --edit
```

esc :wq

Help from git

```
git help
```

GitHub



int301git

The area where git starts tracking your directory

GitHub Installation

Open Gitbash

cd int301git

touch index.html

check by typing dir command

<write into index.html through bash>

notepad index.html {can use nano as well}

hello <ctrl+s save>

git status

< no commit yet because file is not in staging area yet >

GitHub Installation

Add the file

git add index.html

OR

git add . **<no need to add files individually, dot will cover all>**

git status

GitHub Installation

Commit

git commit -m ‘this is first commit’

Updates are now in repository.

git status <work area clean>

git log

or git log --oneline

GitHub

GitHub is code hosting platform for version control and collaboration.

It lets you and others work together on projects from anywhere.

Go to **github.com** and create your account there.

create a new repositor

GitHub

Git remote

git remote add origin <link >

git remote –v

git push –u origin master

refresh online

GitHub

Open file again

notepad index.html

add some text

git add .

git commit –m ‘second addition’

git log --oneline

git push –u origin master

GitHub Branch

Branch in git

A branch will allow making a "copy" while not affecting the original branch

The branch is a copy of the original file.

Open gitbash

cd to your PWD initialized as git

cd D:

cd int301git

git status

GitHub Branch

check for branch:

git branch

{suppose file has 10 lines of code }

keep first 2 line written in file and branch must delete line 3 onwards and add a new line, task is done.

GitHub Branch

Create a new branch

git branch local_branch

git branch

Access the branch , go to local branch

git checkout local_branch

notepad index.html and delete all fines from line 3 and add a new line...task is done by manjot

GitHub Branch

notepad index.html

git status

git add .

git status

git commit -m 'task done by manjot user'

git push -u origin local_branch

refresh online and check

GitHub Branch

To go from one branch to the other

```
git checkout master
cat index.html
```

Merge local code to the original code

```
git merge local_branch
```

```
git push -u origin master
```

refresh online

GitHub clone repository

Git clone: used to create a local copy of a remote repo.

When to use?

No local repository is there.

Suppose you want to access someone else's project, then you clone that project to get a local copy on your system.

→ If local folder is not initialized as git repository, then use git clone to access any project → then that project will be linked/put in that local directory.

→ Later you need not to initialize that folder as local repo.

GitHub clone repository

Open gitbash and

cd D:

mkdir new_clone_git

cd new_clone_git

git status

Now, goto github.com and open any project, ON RHS: clone button

Click and copy HTTPS link

GitHub clone repository

Back to gitbash and

cd new_git_clone

git clone <paste that https clone link here>

Now go to D drive on your machine and check file in new_git_clone folder.

goto this folder and run git status command

GitHub Undo changes

At gitbash

notepad index.html

Make some changes to the file.

git add .

Now run

git checkout -- index.html

It Undo uncommitted changes

GitHub Undo changes

If git checkout – doesn't work then use git revert

a. Get commit id by using 'git log' command

b. git revert <paste that git commit id here>