* What is artificial intelligence?

🡺Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are

programmed to think and learn like humans. It is a broad field of study that encompasses

various subfields, including machine learning, natural language processing, computer vision,

robotics, and expert systems.

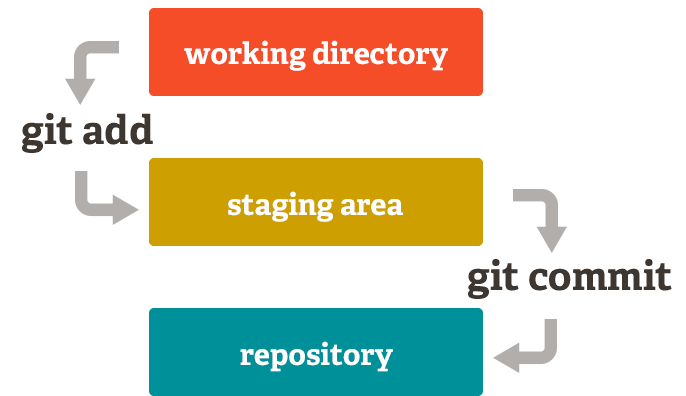
* Why Do We Need a Version Control System?

🡺 Version control is important for keeping track of changes to code, files, and other digital assets. you should use the version control software on development project where multiple team members works on the same project with collaboration.

* What is Git?

🡺Git is a version control system used to track changes in computer files. it’s a tool to manage our code and file history while co-ordinating work remotely on those files with others.

***Working of git***



**Git installation and setup**

install and set up Git, you can follow these steps:

🡺download Git: <https://git-scm.com/download>

🡺Run the installer

🡺Complete the installation steps

🡺 Verify the Installation: open the command prompt and run the given command to view what version the git is : **git –version**

* The git—version command shows the version of the git .

🡺 Configure Git: open the git bash to configure our name and e-mail address. Type the command

**git config --global user.name "Your Name"**

**git config --global user. email** [**your\_email@example.com**](mailto:your_email@example.com)

***Basic local Git operations:***

* Create a project folder with your name in the directory.

**mkdir directory\_name**

* switch to that directory where you want to create the repository using the Git Bash.

**cd directory\_name**

* to initialize a new Git repository:

**git init**

* “git init” creates an empty Git repository in the current directory.
* Editing the content of the file:

**Vim file1.txt**

* View the status of the files:

**Git status**

* The git status command shows the status of the files i.e., whether the files are tracked or not.
* The red colour files denote that those files are untracked and the green colour files denotes those files are tracked.
* Staging (adding)and Committing Changes to the files:
* Before committing we need to stage the changes:

**git add filename**

* To stage multiple file at once, you can use the following command:

**git add file1\_name file2\_name**

* To stage all files at once, You can use the following command:

**git add .(. indicates all)**

* Commit changes: to commit the staged files to the local repository:

**git commit -m "Commit\_message"**

* To view history of the changes made we use log command:

**Git log**

* Undoing Changes*: Undoing changes to a file before staging:*

**git checkout – filename**

Cloning a Repository: Cloning a local Git repository involves creating a copy of an existing repository stored on our local machine.

* Run the following command to clone the local repository:

**git clone path-to-source-directory**

Git Branching and Merging

Branching: In a git hub master is a main branch all the source code is stored in the master branch after review by the team leader.

* With the help of branches we can parallely develop the code for different software.

Creating and Switching to New Branches:

• Run the following command to see the currently active branch**. git branch**

• The active branch will be indicated with an (\*) .

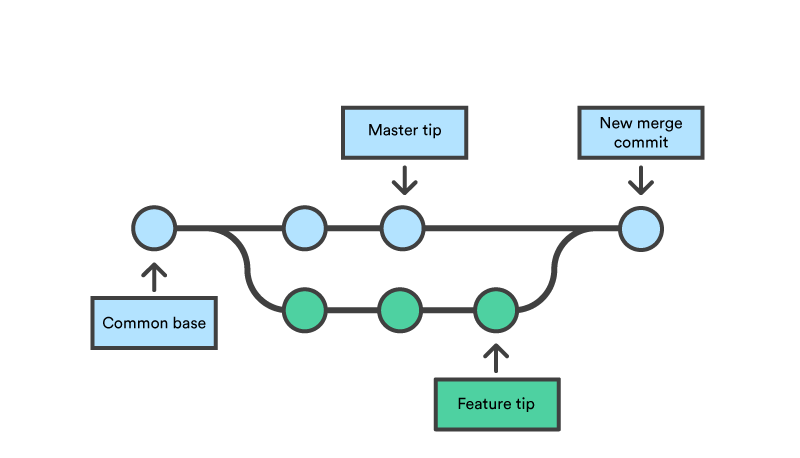
• Create a new branch: Use the following command to create a new branch**:**

**git branch** **branch-name**

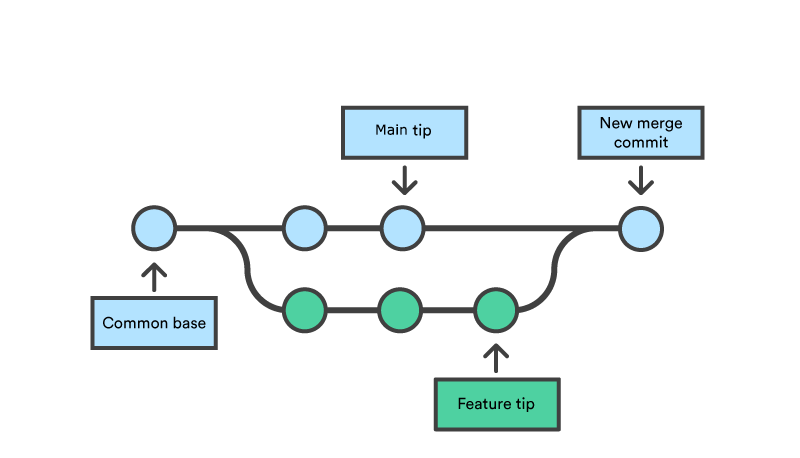
• To switch to the newly created branch: **git checkout-branch\_name**

Merging:

There are two ways of merging:

* Fast-forward merge
* Three-way merge
* Fast-forward merge: changes are happened in the child branch only.

* Three way merge: changes are happened in the parent and child branches both. So there is chances of conflicts.

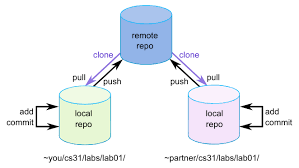
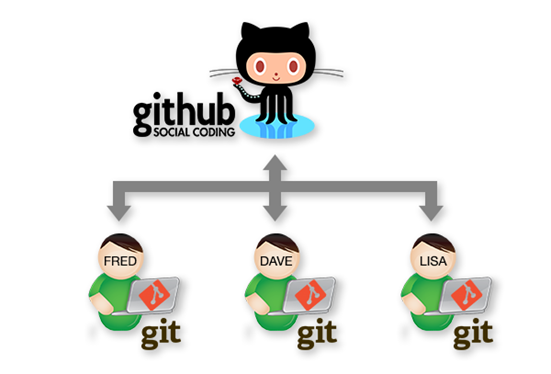


* To Merge the branches Use the following command to merge another branch into the currently active branch: **git merge branch-name**
* to merge a branch named feature-branch into the main branch and run

**git merge feature**

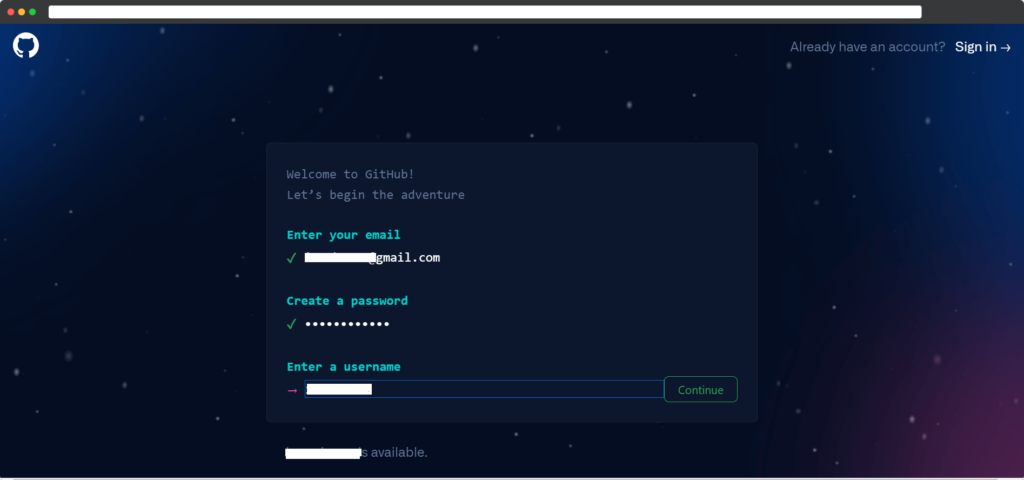
git-hub:

git-hub is a hosting service for git repository. git is the tool, where git-hub is the server to use git.

Creating a git-hub account and configuring it:

* open google chrome in a search list type “sign in git-hub” press enter.
* Click on sign-up button to create a new account in git hub.
* Specify the username ,email address ,password. After specifying the e-mail, a confirmation mail will be sent to the e-mail account.
* Open your email account, open confirmation mail and confirm it by clicking on a link present in a mail.
* After creating a account click on sign in and provide username and password ,click on sign in button.



Create and push to repositories: (This code you will get from Git hub repository )

To create file: **touch file1.txt**

To push the files or repository to the remote repository, use the given command: **git push origin master**

To update the contents in a local repository: **git pull origin master**

Collaboration **:**

Using the collaboration concept for a single project we can invite number of users so that we can work together.

Migration: Migration in GitHub generally refers to the process of moving repositories, projects, or data from one location or platform to another. This can include migrating code from one version control system to Git, transferring repositories between GitHub accounts, or moving from another version control platform to GitHub.