



# Git

**Open Source Distributed Version  
Control System**

Term	Meaning
Control System	Git is a content tracker. Git can be used to store content. It is mostly used to store code due to the other features it provides.
Open Source	Open-source software is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software.
Version Control System	The code which is stored in Git keeps changing as more code is added. Also, many developers can add code in parallel. Version Control System helps in handling this by maintaining a history of what changes have happened.
Commit	An operation which sends the latest changes of the source code to the repository, making these changes part of the head revision of the repository.
Branch	A branch is nothing but a pointer to the latest commit in the Git repository.
Local Branch	A local branch is a branch that only you (the local user) can see. It exists only on your local machine.
Remote Branch	A remote branch is a branch on a remote location (which is usually a web server). You can push the newly created local branch to origin .

# Download Git

<https://git-scm.com/download/win>



# Verify Git Installation

```
git --version
```

```
C:\Users\MuhammadDaniyal>git --version  
git version 2.38.1.windows.1
```

```
C:\Users\MuhammadDaniyal>
```

# Creating Local Repository

- In your computer, create a folder for your project. Let's call the project folder `simple-git-demo`
- Go into your project folder and add a local Git repository to the project using the following commands: `git init`
- The `git init` command adds a local Git repository to the project.

# Demo Changes

Create a file  
called `demo.txt` in the  
project folder and add  
the any text into it



# Staging and Committing the code

- Committing is the process in which the code is added to the **local repository**. Before committing the code, it has to be in the **staging area**. The staging area is there to keep track of all the files which are to be committed.
- Any file which is not added to the staging area will not be committed. This gives the developer control over which files need to be committed.

# Staging

- Command for staging the file: `git add demo.txt`
- Add multiple files: `git add file1 file2 file3`
- Add all the files inside your project folder to the staging area: `git add .`



# Committing

- Command to commit the file

```
git commit -m "Initial Commit"
```

- “*Initial Commit*” is the commit message here. Enter a relevant commit message to indicate what code changes were done in that particular commit.

# Status

- Modify the demo.txt file and add the following: *Initial Content Adding more Content*
- Use `git status` to find out information regarding what files are modified and what files are there in the staging area

# Branch

- Create a new branch called ***test*** using the following command: `git branch test`
- This command creates the test branch.
- We are still in the context of the master branch. In order to switch to the test branch. use the following command: `git checkout test`
- You can list out all the branches in local using the following command: `git branch`

## Some Commits in the New Branch

- Modify `demo.txt` by adding the following snippet: *Initial Content Adding more Content Adding some Content from test Branch*
- stage and commit using the following commands: `git add demo.txt git commit -m "Test Branch Commit"`

# Merging

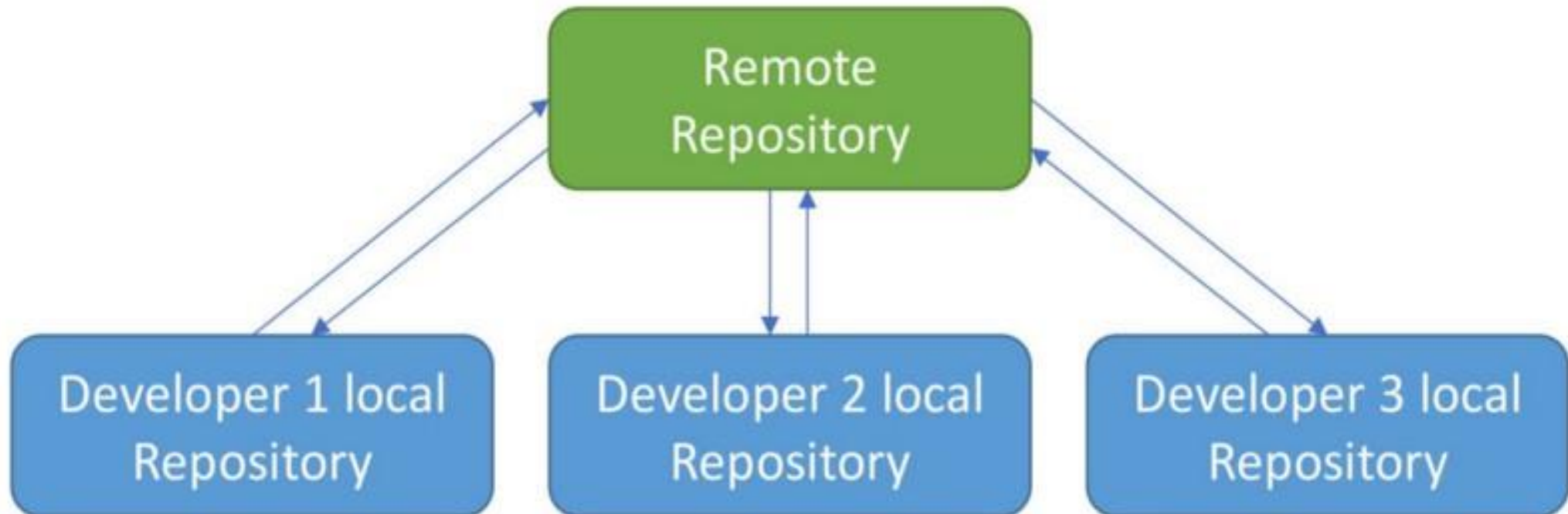
- Test Branch is ahead of the Master by 1 commit. Let's say that now we want all the code in the Test Branch to be brought back to the Master Branch.
- This is where `git merge` is very useful.

# Merging

- Back to the master branch: `git checkout master`
- Run the merge command: `git merge test`

# Remote Repository

- Each developer will work in their local repository but eventually, they will push the code into a remote repository. Once the code is in the remote repository, other developers can see and modify that code.



# Github

- We will be using GitHub for the remote repository.
- Go to <https://github.com/> and create an account.
- After registering in the GitHub homepage, click on **Start a Project** to create a new Git repository. Give the repository a name and click “Create Repository”
- Give the name as `git-codeline-demo`.
- This will create a remote repository in GitHub, and when you open the repository




# Repository


 daniyalmehmood / **git-codeline-demo** Public


 Pin


 Unwatch **1** ▼

 Fork **0** ▼





 **Code**

 Issues


 Pull requests

 Actions

 Projects


 Wiki

 Security

 Insights

 Settings

## Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

git@github.com:daniyalmehmood/git-codeline-demo.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# git-codeline-demo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:daniyalmehmood/git-codeline-demo.git
git push -u origin main
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:daniyalmehmood/git-codeline-demo.git
git branch -M main
git push -u origin main
```



# Getting remote repository

```
git remote add origin [repository url]
```

# Git Push

- In order to push all the code from the local repository into the remote repository, use the following command:

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

# Git Pull

- `git pull` is used to pull the latest changes from the remote repository into the local repository. The remote repository code is updated continuously by various developers, hence `git pull` is necessary.

```
git pull origin master
```

# Git Clone

- `git clone` is used to clone an existing remote repository into your computer.

```
git clone [repository url]
```