

Approaches to operate git

1. Using git bash / command line approach
2. Using github to drag and drop files
3. Console usage

Introduction to Git

Git is a distributed version control system used to track changes in source code during software development

It helps developers:

- Track file changes
- Collaborate with teams
- Maintain version history
- Manage branches

Configure Git

```
git config --global user.name "Your name"  
git config --global user.email "your@gmail.com"
```

```
git config --list
```

Create a Local Repository

Step 1: Create project folder

```
mkdir myproject  
cd myproject
```

Step 2: Initialize git Repository

```
git init
```

This creates a hidden .git folder to track changes

Check status:

```
git status
```

Commit to Local Repository

Step 1: Create a file

```
touch app.py
```

Step 2: Add file to staging area

```
git add app.py
```

Or git add .

Step 3 : Commit changes

```
git commit -m "message"
```

Check commit history

```
git log
```

Create a Remote Repository

Remote repositories are typically on platforms like

- GitHub
- GitLab
- Bitbucket

Steps:

1. Login to GitHub
2. Click new Repository
3. Enter repository name
4. Click Create

Copy the repository URL

example:

<https://github.com/username/myproject.git>

Push to Remote Repository

Step 1: Connect local repo to remote

```
git remote add origin https://github.com/username/myproject.git
```

Check remote:

```
git remote -v
```

Step 2: Push Code

```
git push -u origin main
```

After first push next time onwards use

```
git push
```

Git References (important commands)

Command	Description
git status	Check current status
git add	Add file to staging
git commit	Save changes
git log	View commit history
git diff	View file changes
git branch	List branches
git checkout	Switch branch
git merge	merge branches
git pull	fetch + merge remote changes
git clone	copy remote repository

Branching and Merging

Branching allow parallel development without affecting the main branch

Create new branch:

```
git branch feature1
```

Switch to branch:

```
git checkout feature1
```

Or

```
git checkout -b feature1
```

merge branch to main

```
git checkout main  
git merge feature1
```

Delete branch

```
git branch -d feature1
```