Getting Started with Git

→ Go to GitHub Account and signup

→ Make New Repository

→ Now Follow the steps which is provided by git

```
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin <origin https url>
```

→ To Check current origin url

```
git remote -v
```

→ To get the latest code inside local machine we need to use this command

```
git pull
```

Now to push new updates on GitHub we need to follow this steps

- `git status` To check current changes

- `git add <filename>` or `.` → if we mention filename it will only add that file and if we enter `.` (dot) it will include all the files inside the current folder
- `git commit -m “message”` → Commit is used to track project at every level (version)
- `git push origin main` or other `branchName` → it will push the changes to that branch

⇒ Merge conflicts

▼ What is Merge Conflicts

A Git merge conflict occurs when two branches have changes to the same part of a file, and Git cannot automatically resolve those differences during a merge operation. This typically happens when:

1. Two branches modify the same line of a file.
2. One branch deletes a file while the other branch modifies it

→ Now from GitHub we are going to make some changes and also some changes from our code editor so before pushing that local file changes we need to get the current state or changes of our repository

→ command

```
git pull
```

→ Git pull is used to get the latest changes from GitHub

⇒ Getting back to a certain history of application by commit

```
git log
```

→ This code will give us the history of commits

```
git reset --hard <commit hash>
```

▼ Example Image of commit hash

⇛ Adding New Branch

→ Checking in which branch I'm currently in

```
git branch
```

→ Making new branch and navigating inside of it

```
git checkout -b <newBranchName>
```

→ Switching to the branch

```
git checkout <branchName>
```

→ Merging branches

```
git merge <branchName>
```

→ Now In a new Branch getting changes of other branch

```
git pull origin <branchName>
```

→ Now after pulling the new changes from another branch, In order to reflect that changes to a new branch we need to pull that another branch changes and push it to the current branch

```
git pull origin <branchName>
```

→ This line of command will pull the changes of that branch

```
git push origin <currentBranchName>
```