Introduction to Git: A Beginner's Lesson

What is Git?

Git is a distributed version control system used to track changes in source code during software development. It allows multiple developers to work on a project simultaneously without interfering with each other's changes.

Why Use Git?

1. Collaboration: Multiple developers can work on the same project.
2. Version Control: Keeps track of every change made to the code.
3. Backup: Provides a reliable backup of your code.
4. Branching and Merging: Allows you to work on different features simultaneously.

Basic Git Terminology

Repository (Repo): A directory which contains your project work, including a .git folder, where Git stores all the metadata and object database.

Commit: A snapshot of your repository at a specific point in time.

Branch: A movable pointer to a commit. The default branch name in Git is main.

Clone: A copy of a repository.

Push: Sending your committed changes to a remote repository.

Pull: Fetching and merging changes from a remote repository to your local repository.

Merge: Combining changes from different branches.

Setting Up Git

1. Install Git: Download and install Git from https://git-scm.com/.

2. Configure Git:

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

Basic Git Commands

Initializing a Repository

```
git init
```

Cloning a Repository

```
git clone https://github.com/user/repo.git
```

Checking the Status

```
git status
```

Adding Changes

```
git add filename
```

or add all changes

```
git add .
```

Committing Changes

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

Introduction to Git: A Beginner's Lesson

Viewing Commit History

git log

Branching

git branch branch_name

git checkout branch_name

or create and switch

git checkout -b branch_name

Merging

git merge branch_name

Pushing Changes

git push origin branch_name

Pulling Changes

git pull origin branch_name

Basic Workflow

1. Clone the repository (if you haven't already):

   git clone https://github.com/user/repo.git

   cd repo

2. Create a new branch for your work:

   git checkout -b new-feature

3. Make changes to your files.

4. Check the status of your changes:

   git status

5. Add your changes to the staging area:

   git add .

6. Commit your changes:

   git commit -m "Add new feature"

7. Push your changes to the remote repository:

   git push origin new-feature

8. Create a pull request on the remote repository to merge your changes.

Conclusion

Git is a powerful tool that can greatly enhance your productivity and collaboration in software development. This lesson covers the basics, but there is much more to learn. As you become more comfortable with Git, explore more advanced features like rebasing, cherry-picking, and using hooks. Happy coding!