

Version Control with Git

Version Control with Git

1. What is Version Control?

- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

2. What is Git?

- Git is a distributed version control system that allows multiple developers to collaborate on a project simultaneously.

3. Key Concepts of Git:

- **Repository (Repo):** A folder or directory where your project files are stored.
- **Commit:** A snapshot of your project at a specific point in time.
- **Branch:** A parallel version of the code within the repository.
- **Merge:** Combining changes from different branches.
- **Pull Request:** Requesting changes made in one branch to be merged into another branch.

4. Basic Git Workflow:

- **Initialize a Repository:**

- Use the command `git init` to start a new repository.

- **Add and Commit Changes:**

- Use `git add <file>` to stage changes and `git commit -m "message"` to commit them.

- **Create and Switch Branches:**

- Use `git branch <branch_name>` to create a new branch and `git checkout <branch_name>` to switch to it.

- **Merge Changes:**

- Use `git merge <branch_name>` to merge changes from one branch into another.

Version Control with Git

- **Resolve Conflicts:**

- Conflicts may arise when merging branches. Use `git status` to identify conflicts and `git mergetool` to resolve them.

5. Best Practices with Git:

- **Commit Often:** Make small, frequent commits to track changes effectively.
- **Use Branches:** Separate work into branches to avoid conflicts and test new features.
- **Write Descriptive Commit Messages:** Explain the changes made in each commit clearly.
- **Collaborate:** Utilize features like pull requests to review and merge code changes.

6. Resources for Learning Git:

- Online tutorials, Git documentation, and interactive platforms like GitHub and GitLab offer resources for learning and mastering Git.

7. Conclusion:

- Git is a powerful tool for managing project versions and collaborating with other developers. Understanding its key concepts and best practices can help streamline development processes and improve code quality.