Git History Commands

Git History Commands

Understanding Git history is crucial for managing changes and collaborating effectively in a version control environment. Here are some commonly used Git commands for viewing and manipulating the commit history:

Viewing Commit History

1. git log

   Basic log of commits.

   ```
   git log
   ```

2. git log --oneline

   Condensed log, displaying one commit per line.

   ```
   git log --oneline
   ```

3. git log --graph

   Visual graph representation of the commit history.

   ```
   git log --graph
   ```

4. git log --author="[author name]"

   Show commits by a specific author.

   ```
   git log --author="John Doe"
   ```

5. git log --since="[date]" --until="[date]"

Show commits within a date range.

```
git log --since="2023-01-01" --until="2023-12-31"
```

6. git log -p

Show patch (diff) introduced in each commit.

```
git log -p
```

7. git log --stat

Show statistics for files modified in each commit.

```
git log --stat
```

Viewing Specific Commit Details

1. git show [commit-hash]

Show details of a specific commit.

```
git show abc1234
```

Visualizing History with Tools

1. gitk

Graphical history viewer.

```
gitk
```

2. gitg

Another graphical history viewer.

Git History Commands

```
gitg
```

Comparing Commits

1. git diff [commit1] [commit2]

   Show changes between two commits.

   ```
   git diff abc1234 def5678
   ```

2. git diff [branch1]...[branch2]

   Show changes between two branches.

   ```
   git diff main...feature-branch
   ```

Undoing Commits

1. git revert [commit]

   Create a new commit that undoes changes made by a specific commit.

   ```
   git revert abc1234
   ```

2. git reset [commit]

   Move the current branch to a specific commit.

   ```
   git reset --hard abc1234
   ```

3. git reset HEAD~1

   Undo the last commit, keeping changes in the working directory.

   ```
   git reset HEAD~1
   ```

Git History Commands

Additional Useful Commands

1. git blame [file]

   Show who modified each line in a file.

   ```
   git blame filename.txt
   ```

2. git reflog

   Show a log of all references in the repository, including branch movements.

   ```
   git reflog
   ```

These commands provide a robust toolkit for navigating and managing your Git commit history.