ONE PAGE SUMMARY

IT & Software › Git & GitHub › Merging and Rebasing

Introduction to Merging and Rebasing

Mastering Merging and Rebasing in Git Simplifies Collaboration.

DEFINITION

> Merging and rebasing are two methods in Git for integrating changes from one branch into another. Merging creates a new commit, while rebasing rewrites commit history.

KEY CONCEPTS

Merging

Combines changes from different branches into a single branch, preserving history.

Rebasing

Moves or combines a sequence of commits to a new base commit, creating a linear history.

Fast-Forward Merge

Occurs when the current branch's HEAD is directly behind the branch being merged, allowing a simple pointer update.

Conflict Resolution

Happens when merging or rebasing changes that cannot be automatically reconciled, requiring manual intervention.

Interactive Rebase

Allows users to edit, reorder, or squash commits during the rebasing process for cleaner history.

EXAMPLES

→ git merge feature-branch

→ git rebase main

→ git rebase -i HEAD~3

MEMORY TIPS

★ Think of merging as 'joining' and rebasing as 'rewriting' history.

★ Remember 'M' for Merge and 'R' for Rebase to differentiate their functions.

★ Visualize merging as a fork in the road, while rebasing is a straight path.

COMMON MISTAKES

✗ Forgetting to resolve conflicts before completing a merge or rebase.

✗ Using rebase on shared branches, which can confuse collaborators.

✗ Neglecting to back up branches before performing a rebase.

QUICK RECAP

Merging combines branches while preserving history, whereas rebasing rewrites commit history for a cleaner look. Both are essential for effective collaboration in Git.