Quiz

Score: 15/15

1. What is Git merging?

- A process of combining the changes from different branches to a single branch
- A process of deleting branches in Git
- A process of creating new branches in Git
- A process of reverting to the previous commit in Git

Explanation

Git merging is the process of integrating changes from different branches into a single branch. It allows developers to combine the work done in separate branches and incorporate the changes into the main branch.

2. Which command is used to merge a branch into the current branch in Git?

- git push
- git commit
- git merge
- git branch

Explanation

The 'git merge' command is used to merge a branch into the current branch in Git. It integrates the changes from the specified branch into the current branch, creating a new commit.

3. What is a fast-forward merge in Git?

- A type of merge where the branch being merged has diverged and Git cannot automatically perform the merge
- A type of merge that updates the current branch to the target branch directly without creating a new commit
- A type of merge that discards all the changes from the branch being merged
- A type of merge that requires manual intervention to resolve conflicts

Explanation

A fast-forward merge in Git is a type of merge that updates the current branch to the target branch directly without creating a new commit. This occurs when the current branch's history has diverged from the target branch's history, and the target branch can be accessed with a simple linear history.

4. In Git, which command is used to abort a merge and return the working directory to the state before the merge?

- git drop
- git revert
- git merge --abort
- git reset

Explanation

The 'git merge --abort' command in Git is used to abort a merge and return the working directory to the state before the merge. It undoes the merge, resetting the state of the working directory and index to the state before the merge began.

5. What happens if there are conflicts during a merge in Git?

Git automatically resolves the conflicts

The merge fails and has to be restarted

Developers need to manually resolve the conflicts

Git discards the changes causing the conflicts

Explanation

When conflicts occur during a merge in Git, developers need to manually resolve the conflicts. Git will mark the conflicted files, and developers must review and modify the files to resolve the conflicts before finalizing the merge commit.

6. Which flag is used with the 'git merge' command to perform a squash merge in Git?

--squash

--fast-forward

--no-commit

--abort

Explanation

The '--squash' flag is used with the 'git merge' command to perform a squash merge in Git. This option allows developers to merge all the changes from a feature branch into the main branch as a single, new commit, effectively squashing the branch's history into one commit.

7. In Git, what is a three-way merge?

A merge involving three branches

A merge that involves three commit points

A merge that incorporates two divergent branches and

Explanation

A three-way merge in Git is a merge that incorporates two divergent branches and a common ancestor. It evaluates the differences introduced in each branch since the common ancestor

a common ancestor

A merge that requires three developers to review and approve the changes

and automatically merges the changes if possible. If conflicts occur, developers need to resolve them manually.

8. What is the purpose of the 'git rebase' command in Git?

To merge the changes from one branch into the current branch

To rewrite the commit history by moving or combining commits

To discard all changes in the current branch

To create a new branch from the current branch

Explanation

The 'git rebase' command in Git is used to rewrite the commit history by moving or combining commits. It allows developers to maintain a cleaner and more linear project history by integrating the changes from one branch onto another, effectively reordering and rewording the commits.

9. Which command is used to continue a merge after resolving conflicts in Git?

git continue

git merge --continue

git continue-merge

git resolve

Explanation

The 'git merge --continue' command is used to continue a merge after resolving conflicts in Git. Once conflicts have been resolved, this command allows Git to finalize the merge commit and complete the merge process.

10. In a pull request, if there are merge conflicts, who is responsible for resolving them?

The developer who submitted the pull request

The repository owner

Git automatically resolves the conflicts

The administrator of the Git server

Explanation

In a pull request, if there are merge conflicts, the responsibility for resolving them lies with the developer who submitted the pull request. They need to address the conflicts and ensure a successful merge before the changes are incorporated into the target branch.

11. What does the 'ours' strategy do during a Git merge?

Automatically resolves all conflicts in favor of the current branch

Automatically resolves all conflicts in favor of the incoming branch

Explanation

The 'ours' strategy during a Git merge automatically resolves all conflicts in favor of the current branch. This means that if there are conflicting changes, Git will prioritize the version from the current

Discards all changes from the current branch

Creates a new branch with the merged changes

branch, effectively ignoring any changes from the incoming branch.

12. What is a Git fast-forward merge?

A type of merge that discards all previous commits

A type of merge that abandons the current branch in favor of the incoming branch

A type of merge that simply moves the current branch pointer to the target branch without creating a new commit

A type of merge that reverts the changes from the incoming branch

Explanation

A Git fast-forward merge is a type of merge that simply moves the current branch pointer to the target branch without creating a new commit. This occurs when the current branch's history has not diverged from the target branch's history, allowing Git to incorporate the changes with a simple pointer movement.

13. What does a cherry-pick merge do in Git?

Creates a new branch with the merged changes

Resolves conflicts automatically during the merge

Selects a specific commit from another branch and applies it to the current branch

Explanation

A cherry-pick merge in Git selects a specific commit from another branch and applies it to the current branch. This allows developers to introduce individual commits from one branch into another, providing flexibility in incorporating specific changes without merging entire branches.

Reverts the changes from the specified commit

14. In a rebase operation, which branch's commits are moved and applied to the other branch?

Incoming branch

Current branch

Base branch

Feature branch

Explanation

In a rebase operation, the commits from the current branch are moved and applied to the other branch. This action effectively rewrites the commit history of the current branch, integrating its changes onto the target branch while maintaining a linear project history.

15. What is a Git squash merge?

A merge that discards all the changes from the feature branch

A merge that combines all the changes from a feature branch into a single, new commit

A merge that reverts the changes from the feature branch

A merge that automatically resolves conflicts

Explanation

A Git squash merge is a merge that combines all the changes from a feature branch into a single, new commit. This allows developers to streamline the commit history by consolidating multiple feature branch commits into a cohesive unit, providing a cleaner project history.