

GIT Commits

Score: 11/12

1. What is a Git tag?

A permanent branch

A pointer to a specific commit

A type of merge conflict

A branch for experimental changes

Explanation

A Git tag is used to mark a specific point in the project's history. It is commonly used to mark release points, such as v1.0, v2.0, etc. When a tag is created, it's like taking a snapshot of the project at that specific point in time.



2. What command is used to create a new branch in Git?

git checkout

git commit

Explanation

In Git, the command 'git branch <branch_name>' is used to create a new

git branch

branch. This allows developers to work on new features or fixes without affecting the main branch.

git merge



3. What is the purpose of branching in Git?

To merge all changes into the main branch

To isolate work in progress

To automatically resolve conflicts

To delete old commit history

Explanation

Branching in Git allows developers to diverge from the main line of development and continue work without affecting the main branch. It is commonly used to work on new features, bug fixes, or experiments without disrupting the stability of the main codebase.



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

git checkout

Explanation

git commit

git merge

git branch

The 'git merge <branch_name>' command is used to merge a branch into the current branch in Git. This integrates the changes from the specified branch into the current branch.



5. What is a merge conflict in Git?

A conflict between two different repositories

A conflict between two different commits

A conflict that occurs when merging branches

A conflict that occurs during repository cloning

Explanation

A merge conflict in Git occurs when two branches have diverged and the changes made to the same part of a file by both branches cannot be automatically merged. Git highlights the conflicting sections and requires manual intervention to resolve the differences.



6. In Git, what is the purpose of the 'git tag -a' command?

To delete a tag

To create a lightweight tag

To create an annotated tag

To merge two branches

Explanation

The 'git tag -a' command is used to create an annotated tag in Git. Annotated tags include a tagger name, email, date and time, and a tagging message, providing more information about the tag.



7. What are practical use cases of branching in Git?

Only for deleting old commit history

Only for merging all changes into the main branch

For working on new features

For cleaning up the repository

Explanation

Practical use cases of branching in Git include working on new features without interfering with the main codebase, fixing bugs in isolation, trying out experimental changes, and preparing for a release while still maintaining current state for bug fixes.



8. What is the correct order of commands to resolve a merge conflict in Git?

git add, git commit, git merge

git pull, git resolve, git push

Identify, Edit, Mark, Finalize

git status, edit files, git add, git commit

Explanation

To resolve a merge conflict in Git, the typical order of commands includes identifying the conflicting files, manually editing those files to resolve the conflicts, marking the conflicts as resolved, and then finalizing the merge.



9. What does the 'git fetch' command do in Git?

Merge changes into the current branch

Download changes from a remote repository

Create a new branch

Submit changes to a remote repository

Explanation

The 'git fetch' command in Git downloads objects and references from another repository, allowing the local repository to be updated with changes from the remote without merging them into the current branch. It is useful for reviewing changes before merging or working on a different branch based on the updated state.



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

To merge changes from one branch to another

To reset the current branch to a specific commit

To apply commits on top of another base commit

To delete all commit history

Explanation

The 'git rebase' command in Git is used to reapply commits on top of another base commit. It is often used to maintain a cleaner and more linear project history, by moving the current branch to the tip of another branch or rewriting the commit history to avoid merge commits.



11. When should you use the 'git merge' command over the 'git rebase' command in Git?

When maintaining a clean project history

When preserving original branch history is important

Explanation

The 'git merge' command is preferred if preserving the original branch history is important, or when merging code from many developers. On the other hand, 'git rebase' is useful for maintaining a linear, clean project

When squashing insignificant intermediate commits

When merging code from many developers

history, and for squashing insignificant intermediate commits.



12. In Git, what does 'HEAD' represent?

A specific commit

The latest commit in the repository

The currently checked out commit

A remote repository reference

Explanation

In Git, 'HEAD' is a reference to the currently checked out commit. It is essentially a pointer to the latest commit in the currently checked out branch. It is important for understanding which branch or commit is currently being worked on.