Q:When should you create a pull request on GitHub?

    S:After you've pushed commits to a remote branch

    S:Before starting work on a new feature

    S:To submit your final project code

    S:When you want to merge changes from one branch into another

    A:When you want to merge changes from one branch into another

Q:What is the correct command to push a local branch called "new-feature" to a remote called "origin"?

    S:git push new-feature origin

    S:git push -b new-feature origin

    S:git commit -am "push new branch"

    S:git push origin new-feature

    A:git push origin new-feature

Q:Which of the following are good practices when creating a pull request?

    S:Make the PR as large as possible to batch changes

    S:Include a detailed description of what was changed and why

    S:Don't add any details in the pull request description

    S:Add reviewers so your teammates can provide feedback

    A:Add reviewers so your teammates can provide feedback

    A:Include a detailed description of what was changed and why

Q:What will happen if you try to push your local branch test_branch to origin, but test_branch diverges from origin/test_branch due to new commits on origin/test_branch?

    S:Nothing, the push will succeed even with diverged histories

    S:The push will automatically merge the diverging histories

    S:A backup branch test_branch_old will be created on origin

    S:The push will fail unless you force it

    A:The push will fail unless you force it

Q:What is the recommended way to handle merge conflicts between your branch and master when submitting a pull request?

    S:Close the pull request and abandon your changes

    S:Locally merge master into your branch before pushing to resolve conflicts

S:Re-clone the repo in a different directory, and then copy-paste your changes into the new repo. Create a PR from that repo.

S:Force push your branch to override master

S:Let your teammate handle fixing the merge conflict

A:Locally merge master into your branch before pushing to resolve conflicts

Q:When reviewing a teammate's pull request, you notice their code is unnecessarily complex and could be simplified. What should you do?

S:Submit your own pull request with simplified code

S:Comment suggesting improvements and wait for their response

S:Reject the pull request until they simplify the code

S:Approve the pull request since the code still technically works as intended

A:Comment suggesting improvements and wait for their response