# GitHUB Tutorial for Beginners - Pull Request

Pull requests (PRs) are a fundamental aspect of collaborative development workflows, especially in projects managed with version control systems like Git. Here's a tutorial on how to create and manage pull requests, along with scenarios and examples:

Step 1: Forking the Repository

1. Go to the repository you want to contribute to.
2. Click on the "Fork" button in the top-right corner to create a copy of the repository under your GitHub account.

Step 2: Cloning Your Forked Repository

1. Copy the URL of your forked repository.
2. Open your terminal or command prompt.
3. Navigate to the directory where you want to clone the repository.
4. Run the command:

git clone <your\_forked\_repository\_url>

1. Replace <your\_forked\_repository\_url> with the URL of your forked repository.

Step 3: Creating a New Branch

1. Navigate to the cloned repository directory.
2. Run the command to create a new branch:

git checkout -b feature/new-feature

1. This command creates a new branch named feature/new-feature and switches to it.

Step 4: Making Changes and Committing

1. Make the necessary changes to the codebase using your preferred code editor.
2. Run git status to see the changes.
3. Stage the changes with the command:

git add .

1. Commit the changes with the command:

git commit -m "Implement new feature"

Step 5: Pushing Changes to Your Fork

1. Push the changes to your forked repository:

git push origin feature/new-feature

Step 6: Creating a Pull Request

1. Go to your forked repository on GitHub.
2. Click on the "New pull request" button.
3. Select the base repository and branch you want to merge your changes into.
4. Review the changes and provide a title and description for your pull request.
5. Click on the "Create pull request" button to create the pull request.

Step 7: Reviewing and Discussing Changes

1. Project maintainers and collaborators can review the changes, add comments, and request modifications if needed.
2. You can respond to feedback, make additional changes if necessary, and push them to the same branch.

Step 8: Merging the Pull Request

1. Once the changes are approved, the project maintainer can merge the pull request into the main branch.
2. The merged changes are now part of the main project repository.

Step 9: Updating Your Local Repository

1. After the pull request is merged, update your local repository to include the latest changes:

git checkout main

git pull upstream main

1. Replace upstream with the original repository URL.

Scenarios and Examples:

1. Bug Fix:
   * Scenario: You discover a bug in the project and want to fix it.
   * Process: Fork the repository, create a new branch, make the fix, push the changes, and create a pull request for review.
2. New Feature:
   * Scenario: You have an idea for a new feature and want to implement it.
   * Process: Fork the repository, create a new branch, implement the feature, push the changes, and create a pull request for review and discussion.
3. Documentation Update:
   * Scenario: You notice outdated or missing documentation in the project.
   * Process: Fork the repository, create a new branch, update the documentation, push the changes, and create a pull request to propose the updates.
4. Code Refactoring:
   * Scenario: You identify areas of code that can be improved for better readability or performance.
   * Process: Fork the repository, create a new branch, refactor the code, push the changes, and create a pull request to suggest the improvements.

This tutorial provides a comprehensive guide on how to contribute to projects using pull requests, along with various scenarios and examples to illustrate real-world use cases. It empowers contributors to actively participate in open-source and collaborative development efforts.