

Git Branches & PRs

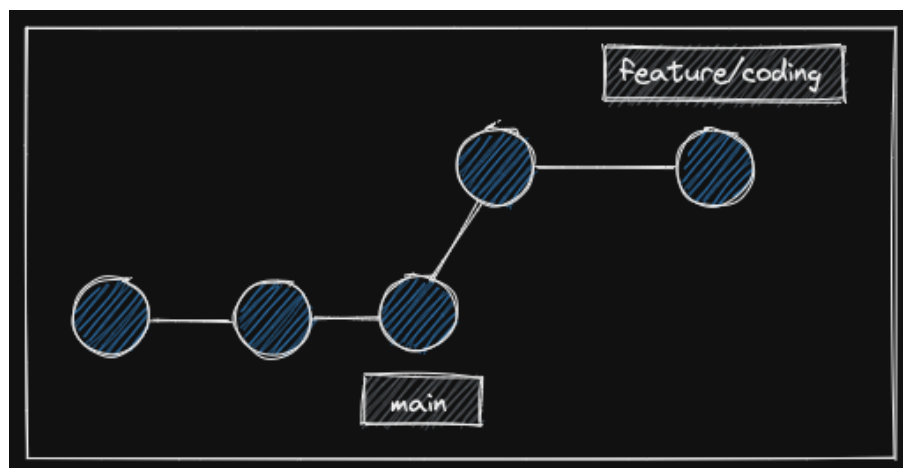
Learning Objectives

- Git branches and how to use them
 - pull requests and how to use them
 - how branches and pull requests facilitate collaboration
-

Git Branches

When working on a project, especially as a team, you want to work on features independently, so they never affect anyone else's work. Git offers us **branches** to keep our current work away from a teams common codebase until completion.

A branch lets you split from the main line of development. The new branch shares a part of its commit history with the main branch. At a certain commit the new branch branches off and the commit histories differ.



The teams common codebase is typically kept in the "main" branch. If you work on a new feature you:

1. create a new **feature branch** and work on that new branch.
2. commit your work on the new branch - the main branch is not effected.
3. finish work on the new feature, test the new functionality and have other developers review your work.
4. **merge** your feature branch into the main branch, so all your work is included in the main branch.

Naming branches

It is good practice to use short descriptive names for your feature branches, e.g. "contact-form". We recommend using hyphens as separators as they make the name more comfortable to read.

Git **branch** commands

command	functionality
---------	---------------

command	functionality
<code>git switch -c <branchname></code>	create a new branch and switch to it
<code>git switch <branchname></code>	switch branches
<code>git branch</code>	list your branches
<code>git branch -a</code>	list all branches (local and remote)
<code>git branch -d <branchname></code>	delete a branch

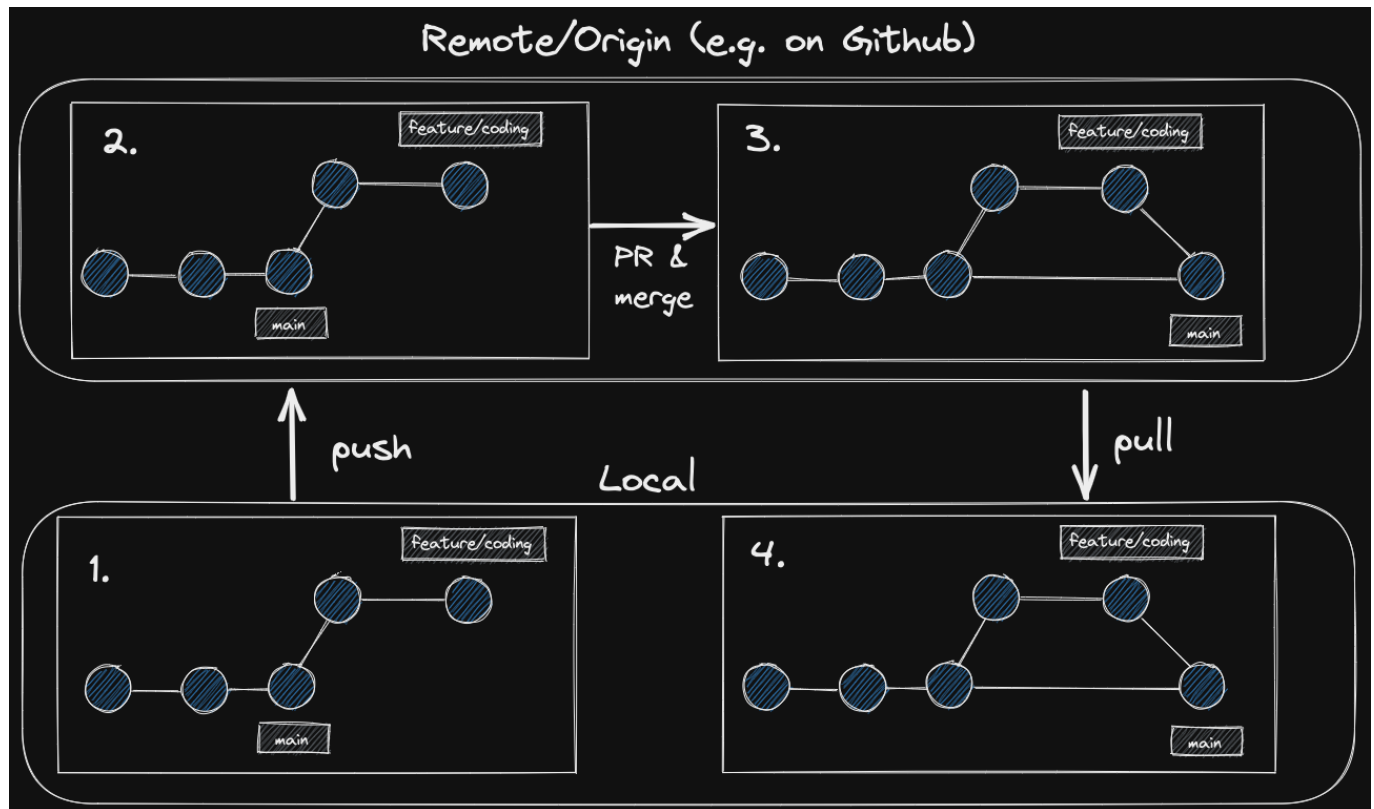
Git pull requests

GitHub offers us **pull requests** (PR) which we can use as a convenient way to request reviews of the work on a feature branch.

A pull request is a request to merge one branch into another branch. Other developers review the PR and suggest changes. If a pull request is approved we can merge the feature branch into the main branch.

Basic Workflow for a pull request

1. Create a new branch with `git switch -c <branchname>`.
2. Make changes to the code / write your code for the feature.
3. Push the changes and the new branch with `git push -u origin <branchname>` (after you have done this once you can use `git push` for this branch)
4. Create a pull request on GitHub from the new branch into main
5. Share the pull request with your team
6. Review the pull request, implement changes if needed, push again to update the pull request until it gets approved
7. Merge the pull request into main
8. Don't forget to `git pull` inside the main branch on your local machine
9. Delete the new branch on GitHub and locally



Resources

- [About branches](#)
- [Git Branching - Branches in a Nutshell](#)