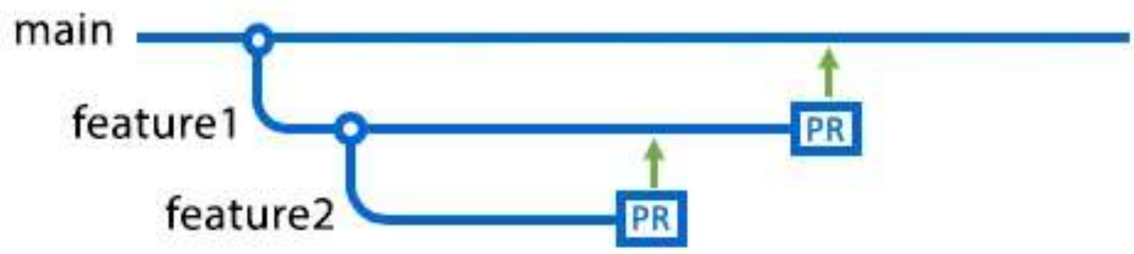
Branches:

- Branches are a way to develop your code and safely experiment in a contained environment to protect the original code base.
- It is very important when multiple branches are created to understand which is the base branch (this will become important when merging changes).
- Your current branch is sometimes referred to as the head branch.
- The branch that was the source is referred to as the base branch.
- It is important to ensure that the base branch is synced prior to accepting pull requests into it.
    - This ensures that the changes could successfully be passed along, should they need to be.

Example:



- In this diagram, feature1 is a feature branch of main.
- feature2 is a feature branch of feature1.
- There are pull requests open from both branches (feature1 and feature2).
- The arrows indicate the base branch of each pull request (main is the base for the PR from feature1, feature1 is the base of the PR from feature2).
- Ideally, feature2 should be merged to feature1 and then merged into main (if there are no conflicts).
- If feature1's PR were to be merged first, then feature1 could be deleted and feature2's changes would be merged directly into main (not best practice).

Deleting a Branch:

- Before GitHub will allow you to delete a branch, it will check for any open pull requests that have the deleted branch as their base branch.

Creating a new branch in GitHub (the simplest way):

Branch drop down
↓
Type in name of new branch
↓
Select "Create branch "..." from "*current branch*"

Development Branches:

- The same logic could be applied to creating a test/development branch.
- Having a branch one “layer” above the main branch that would act as the final test environment is a common practice.
- This keeps you base code (main) protected, as this is the code that the current release is likely running on.
- In this workflow, features would be written and developed in their respective feature branches before being brought all in together in the development or test branch to fully test the functionality.
- Then when it is time for release, the test environment is committed to main.

Sources:
https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches
https://www.youtube.com/watch?v=Wbz8zM_5iCc
https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-and-deleting-branches-within-your-repository
https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow

Additional Resources:
https://youtu.be/KScwEeYwJJk?si=BF_noK_VVLCsXc-5
https://youtu.be/e2IbNHi4uCI?si=xiNyBxc6ILjvm8MO
https://youtu.be/Q1kHG842HoI?si=uhFF_vjpd1lBJjqK
https://github.com/etsuDummy/KinserPedia/blob/main/GitHub%20Made%20Simple.pdf