# General GitHub workflow

In order to work on a repository, this is the workflow GitHub recommends:

[0. Pick an issue? So that we know who works on what]

1. Create a branch

2. Make changes

3. Create a pull request

4. Address review comments

5. Merge pull request

6. Delete branch (optional)

[7. Close the issue?] The issue should close automatically by connecting it to the pull request.

# Issues / Task Management

From the project board we are able to add a bunch of tasks under the column Todo. Initially, these are **drafts** of issues. You can convert them to an active issue by clicking on the **…** and selecting **Convert to issue**. If you open it up you can assign it to a team member (or to yourself).

If you go to the **issues** tab then you’ll see the new issue there. You can click **create a branch for this issue** (if you’re the one assigned to it) and a modal will pop up asking you to name your branch.

Make sure to follow the naming convention that we agree upon (see next section).

## Branch Naming

There are three kinds of branches:

1. Main branch

2. Development branch

3. Feature branch(es)

There is only one main branch and only one development branch, but there will likely be multiple feature branches at any given time. To easily differentiate them from one another while also keeping our commit history neat and readable, we need to agree on a branch naming convention.

One approach we could take is:

**[category]/[issue-number]-[brief-description]-[author]**

### Category

This begins by stating the category or type of work you’ll be doing with this branch. For example:

\* feat - feature implementation

\* style - code style/ clean up

\* fix - bug fix

\* test - test implementation (If we decide to include tests)

Starting with the category will make it super easy for others to understand the purpose of the branch.

### ID

This ID will correspond to the Github issue as well as the Github Projects board.

### Brief Description

Concise summary of the issue.

### Author

Ideally, I think we should keep this short by just using our initials.

This is just one idea. We can customise it however everybody likes. We can also use whatever separator everybody likes best: hyphen, slash or underscore.

# Making Changes

While you’re making changes to the code you want to make sure to make small (atomic) commits with clear and concise commit messages.

# Pull Requests

Clearly link your pull request to the corresponding issue by one of the following methods:

1. Create branch from issue screen in github repository.
2. Manually linking using gear icon in pull request screen.
3. Add a [linking keyword](https://docs.github.com/en/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue#linking-a-pull-request-to-an-issue-using-a-keyword) in the body of the pull request. For example, “Closes Issue#14 - Add a navbar”.

Code Review

Before anything can get merged into the development branch it needs to be reviewed by at least 2 other developers.

Before anything can get merged into the main branch it must be approved by the majority of the development team.

# Permissions

We should probably protect/lock the main branch so that we don’t accidentally mess it up. We can also protect the development branch too to make reviews required. There’s more info about protecting branches in the [Github docs](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/defining-the-mergeability-of-pull-requests/managing-a-branch-protection-rule).