

# Let's get this party started

This guide covers everything from forking the repo to making your first contribution in a step by step manner. I'm assuming you have already installed git and other project requirements like npm, react etc.

## 1. Fork the website repo:

- Go to [\\_CIPHER\(\)'s website](#) repo and hit the fork button.
- Type the following in git bash/linux terminal(whatever you're using) to clone your fork.
- Make sure to replace your-user-name with your actual GitHub username.

```
git clone https://github.com/your-user-name/cipher.git
cd cipher
git remote add upstream https://github.com/alphanoobie/cipher
```

This creates the directory cipher and connects your repository to the upstream (main project) repository.

## 2.

```
npm i
```

This command will install and update all the required modules.

- ## 3.
- Make a separate branch off master using the following command. DO NOT MAKE CHANGES ON THE MASTER BRANCH. It's a better idea to replace shiny-new-feature with your feature name like about-page, nav-bar etc.

```
git checkout -b shiny-new-feature master
```

- Changes in this branch should be specific to one feature/component/bug.
- Before starting a new branch ALWAYS make sure you are on the master branch i.e always branch off master and not off your feature branch.

#### 4. Adding your own code

```
git pull
```

- The first thing to do before writing your code is to check if someone else has added any code to your branch and update your local repository with that code. This can be done using the above command
- After that you can make whatever changes you want to the files.
- After making changes, use `git status` to see the files you've modified and use *git add <path-to -file> OR git add \** (\* adds all the files changed) to record the changes.
- You have to do this every time you make a change to any file. A simple rule of thumb is to make sure every file is green in the output of your `git status` before making commits (which is the next step).

- **Bonus Tip:**

Use the following command to fix indentation of your files. You can use it multiple times while writing the code. I would suggest you use this command every time before committing your code.

```
npx prettier --write .
```

- Once you are satisfied with the changes, use:

```
git commit -m "message"
```

- Replace message with a short description about what changes you made. Like added styling for about page etc.

## **5. Pushing your code and creating a PR (Whenever you feel like you've accomplished a milestone, you can push your code)**

- Typical workflow for current project(making club website) and future projects is going to be like this:-
  - Project leads will create issues and assign them to potential contributors.

- contributors will add code to address that issue and create a PR. Below is a guide on how to create a PR:-
- Push your changes. shiny-new-feature is the name of your feature branch which you used before.

```
git push origin shiny-new-feature
```

## **6. Creating a PR (Pull requests add your code to the main branch)**

- Navigate to your repository on GitHub
- Click on the Pull Request button
- You can then click on Commits and Files Changed to make sure everything looks okay one last time
- Click Send Pull Request.
- Your PR will then be sent to project leads who will review and suggest some changes if required. Once the changes are satisfactory, they will be merged into the master branch.