

Repository information

- [Forking](#)
- [Local development](#)
- [Managing branches and commits](#)
- [Initialise a Quarto project](#)
- [Building pages](#)
- [Push the changes](#)
- [Accessing the live website](#)

A brief guide on how to setup this website elsewhere and develop the wiki.

Forking

- If using a **Codeberg** account, fork [this repo](#).
- If using a **GitHub** account, fork [this repo](#).

Local development

It's recommended to use SSH and SSH keys when working with a remote git.

- Key generating instructions are found [here](#) (this is git-server agnostic).
- For **GitHub**, follow [this](#) to add your key.
- For **Codeberg**, it's similar, navigate to [this link](#) when logged in, and then **Add key** under the **Manage SSH Keys** section.

Clone the repo locally:

- For **Codeberg**:

```
git clone git@codeberg.org:<user>/msci-wiki.git
```

- For **GitHub**:

```
git clone git@github.com:<user>/msci-wiki.git
```

Managing branches and commits

Make a **new branch** to build your own website from with

```
git checkout -b <make_your_own_unique_name>
```

This branch will contain individual work + shared stuff. It should be the source of truth.

From now on, we will refer to this branch as **my_branch**.

When contributing to **shared** knowledge, switch to **shared** branch with `git checkout shared`. In this branch, there should be only one directory called **shared**. This is the root folder of all shared information. Any further directory structuring occurs within this, not outside of it.

- **No individual work goes into the shared branch to avoid conflicts with file-names!**

When merging new commits containing shared information from **shared** to **my_branch**, switch to **my_branch** with `git checkout my_branch`, and then perform a merge with

```
git merge shared
```

- **Don't merge from my_branch to shared!**

In the case there are problematic commits in **shared**, future merge can be done with cherry-pick. Say this is **shared**:

```
SHA0 (HEAD): I want this commit  
SHA1: I don't want this commit  
SHA2: I want this commit
```

Then, in **my_branch**:

```
git cherry-pick SHA2  
git cherry-pick SHA0
```

Initialise a Quarto project

In `my_branch`, make a new Quarto project with the website template:

```
quarto create project website
```

Take a look at `_quarto.yml` from branch `np` for an example of how to configure it.

Building pages

Quarto by default publishes to the branch `gh-pages`.

To build the pages, first, run the following:

```
quarto publish gh-pages
```

This will build and push to the `gh-pages` branch.

- If using **GitHub**, that's all you need to do.
- If using **Codeberg**, you can view it by going to <https://username.codeberg.page/@gh-pages>

Push the changes

Pushing changes to remote servers make rolling back very difficult, make sure everything looks correct first. Use `git log` and `git status` to ensure everything has been committed. To push everything to a mirror:

```
git push <remote> <branch>
```

Try to keep all branches up-to-date on all mirrors!

Accessing the live website

- For **Codeberg**, it's <https://username.codeberg.page/msci-wiki/@gh-pages> (more [information](#)).
- For **GitHub**, it's <https://username.github.io/msci-wiki> (more [information](#)).