



Quarto Website Update Workflow

1. Make Your Edits

- Edit `.qmd` files for page content (`index.qmd`, `research.qmd`, `teaching.qmd`, `cv.qmd`, `contact.qmd`).
 - Edit `styles.css` for styling changes.
 - Edit `_quarto.yml` for site structure or navbar changes.
 - Add files (PDFs, images) to your project folder if needed.
-

2. Preview Locally (Optional)

Run:

```
bash
CopyEdit
quarto preview
```

- Opens your site in a local browser.
 - Updates live as you save changes.
 - Stop with `CTRL + C`.
-

3. Render the Site

Build the updated HTML files:

```
bash
CopyEdit
quarto render
```

- This updates the `docs/` folder (your GitHub Pages output directory).
-

4. Commit Changes to Git

```
bash
CopyEdit
git add .
git commit -m "Update website content"
```

5. Push to GitHub

```
bash
CopyEdit
git push origin main
```

(Replace `main` with your branch name if different.)

6. Wait for GitHub Pages to Update

- GitHub will automatically rebuild your site using the `docs/` folder.
 - Usually live within **1–2 minutes**.
 - Check via: <https://YOUR-USERNAME.github.io/YOUR-REPO-NAME/>
-

7. Test the Live Site

- Click through pages to check formatting.
 - Test the **Contact Form** to ensure Formspree still works.
-



Pro Tips

One-command publish (optional)

If you prefer skipping manual GitHub setup:

bash

CopyEdit

```
quarto publish gh-pages
```

- This will push to a special `gh-pages` branch and publish immediately.
- **Keep backups**
Always commit changes before major edits so you can roll back.
- **Optimize images**
Large images slow down your site — resize before uploading.
- **Version control content**
Use commit messages like "Update Research page with new figure" for clarity.