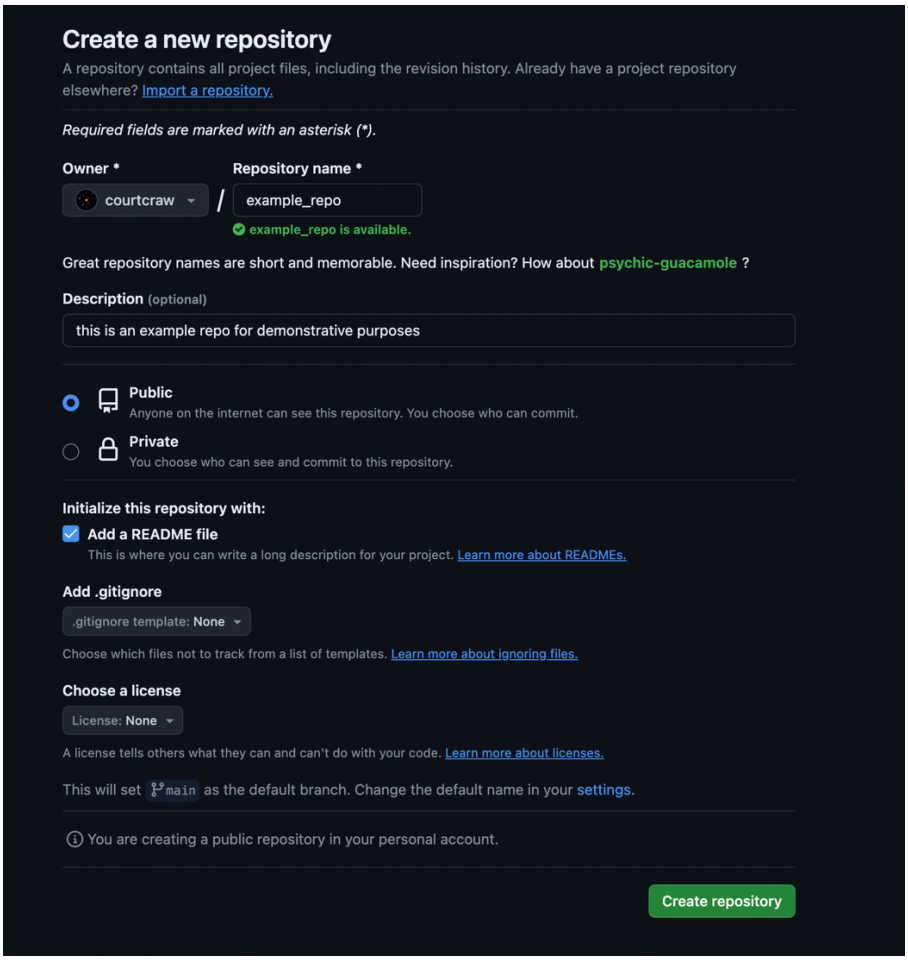
Courtney's Super Simple Guide to Making a Website

This will entirely be hosted on Github Pages, which is extremely simple to use and set up, but has the capacity to get more complicated if you like. You will need a github account to get started.

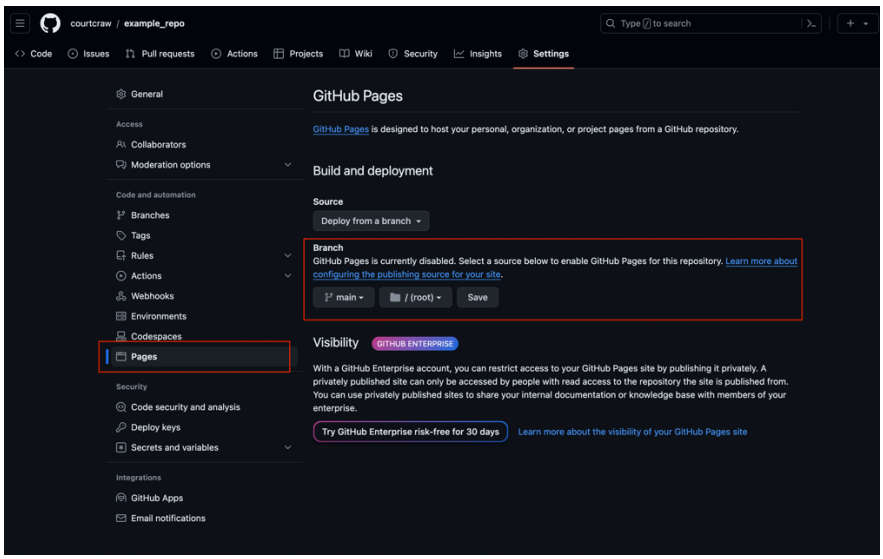
Github allows you to host websites through their service. Each account is allowed 1 personal website and infinite project websites. Most online guides will help you set up a personal website, and I do not recommend doing so for something like MESA workshops. You'll want to set these up as project websites.

Here are the minimal steps to get set up on using Github Pages:

1. Set up a new repository on github with your desired URL extension. You can't really choose this much, it will always be hosted on `[your_github_username].github.io/[your_repo_name]` so choose this repo name wisely.
   Write yourself a description and initialize with a README which will be very minimal to start.

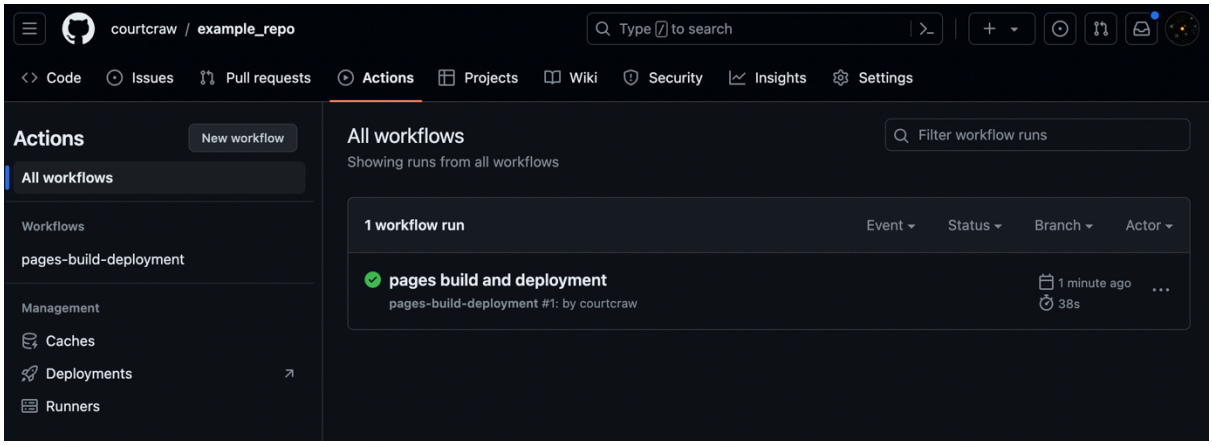


2. Go into settings in this new repo. Head to the Pages tab on the left-hand side. Under Branch, change None to main, and then set to root (for now). You can change it to launch from a folder called docs/ later if you prefer, as that's convention. Save the change.



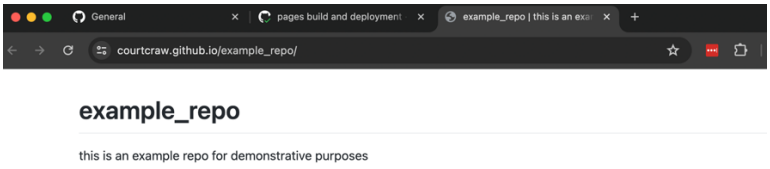
Saving will add a few more options to this page. You can ignore them.

3. Now, you can go to the Actions panel in the github repo to confirm that the initialization worked. You should see a page like this



The green check will indicate that everything worked fine. If you eventually start messing with HTML, check this page first to make sure you didn't break something. Github will automatically run build and deployment every time you apply a new commit, so you don't need to worry about doing this in general. If you click on the build, it will give you a link where the website is being hosted at, or you can simply navigate there yourself.

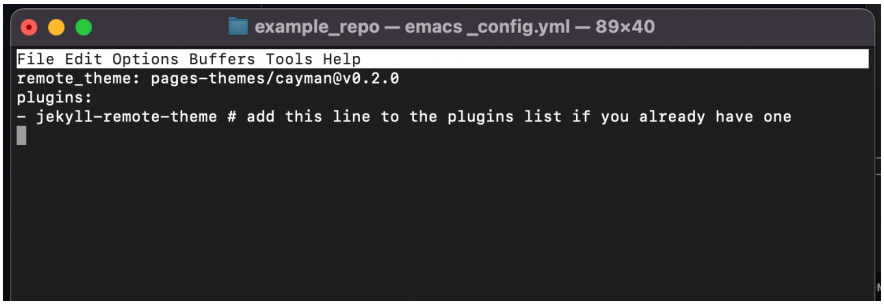
4. This is what the website will look like at this point, it's not the most exciting thing ever.

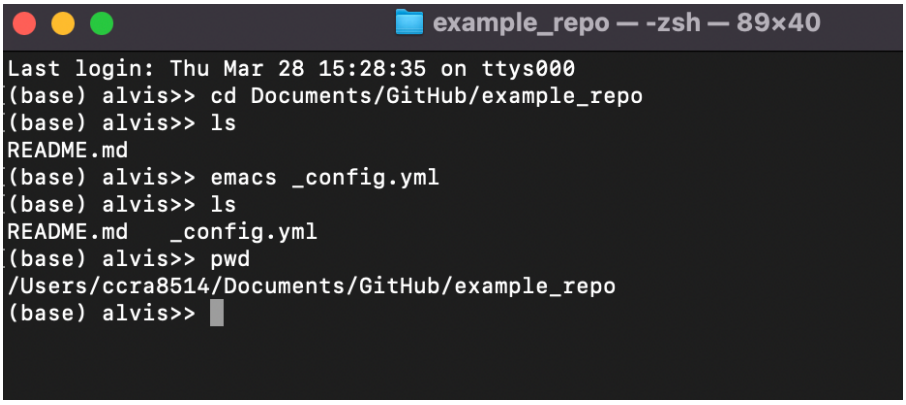


to make it look nicer we can use one of the Github pages recommended default Jekyll styles. There are non-default templates you could use if you wanted to, but you'll have to look that up on your own if you're interested.

5. Choose one of the Github pages default themes for your site. They are listed at https://pages.github.com/themes/
I'll choose Cayman for this example. https://github.com/pages-themes/cayman

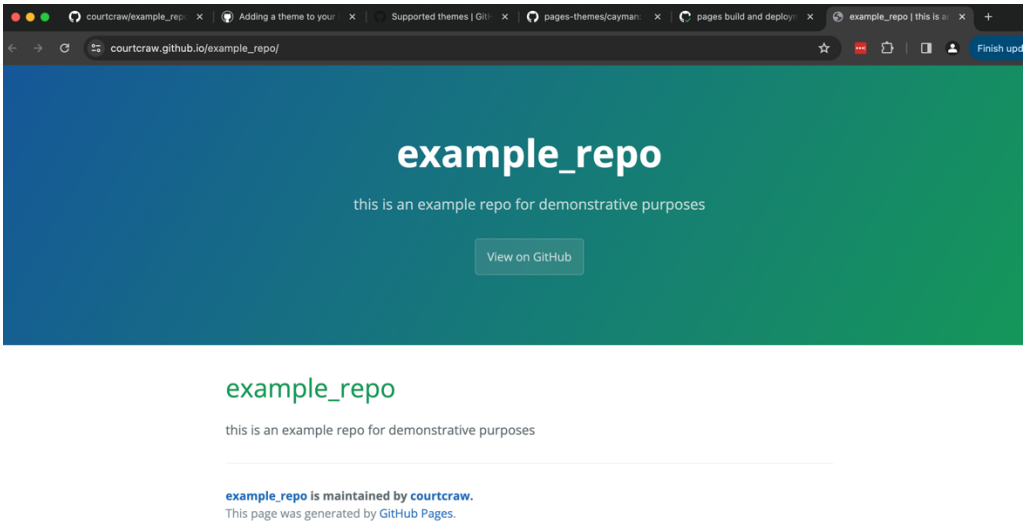
As it says in the usage section, you'll just need to add those two lines to your `_config.yml` file. This doesn't exist yet, you simply need to add it yourself. Clone your repo to your local machine like you normally would with github, create a file called `_config.yml`, then add the lines from your theme Usage page to the file.





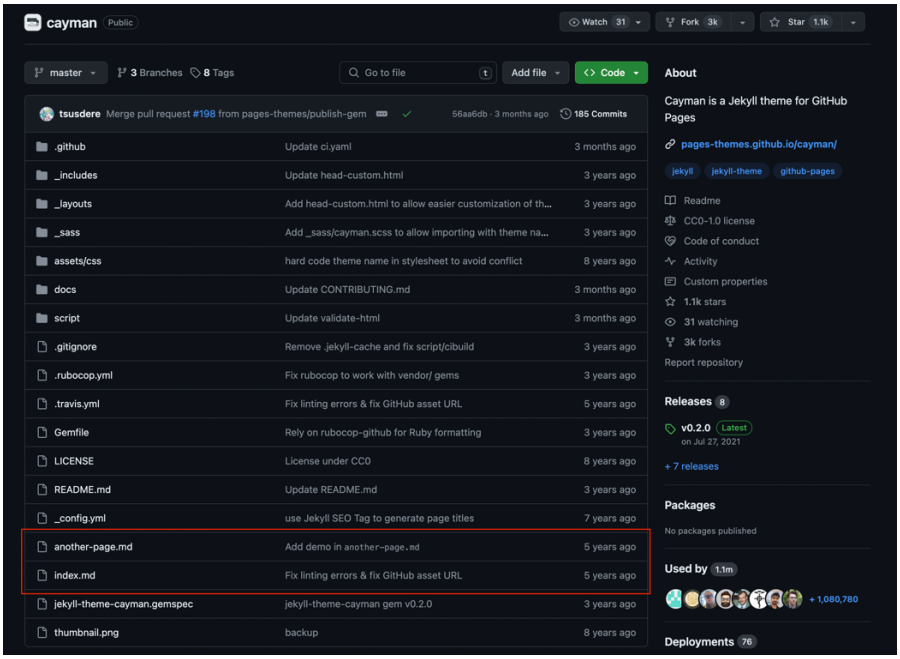
Now commit and push your changes to your github repo. I use Github Desktop which is a nice GUI for github, but many people will prefer to actually learn github and do everything from the command line.

6. You'll need to wait a minute or two for the deployment process to finish, but then check your website again and it will have updated to a new theme.

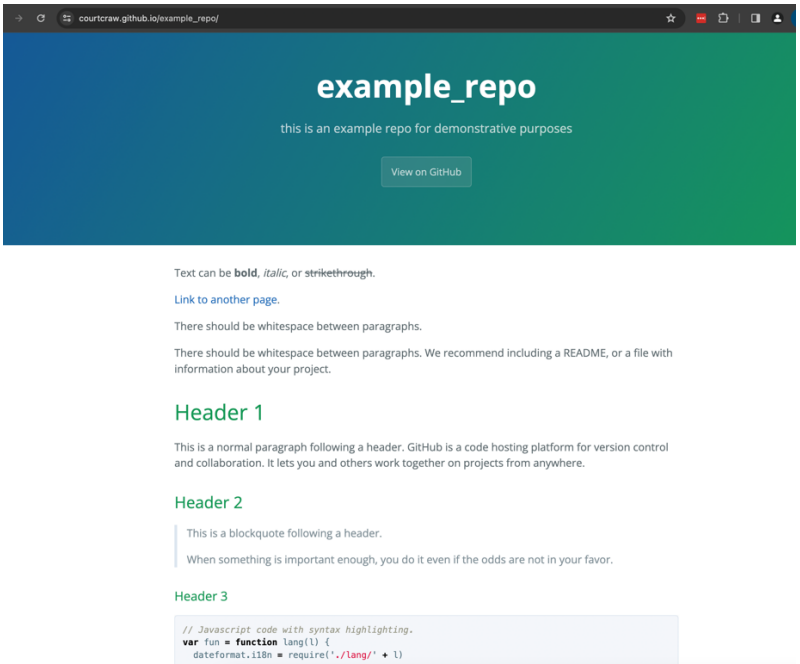


cool! Now we only need some markdown files.

7. Generally speaking, the landing page is reading your README.md as the homepage. You can leave it as this if you like, but if you want an actual functioning README on github then this might not be the best option for you. Most people's landing pages are written on a file called `index.md` . In my opinion one of the easiest things to do is to copy the `index.md` from your chosen theme's template and put it right into your own repo. These are great because they contain examples on how to use markdown which are usually enough to get you started. They also usually have an `another-page.md` that you can copy over as well.



Simply copy these files into your own repo, commit and push, and then refresh your website.



Great! That should in general be enough to get you started. Adding a new page is as simple as making a new markdown file, and linking to it from your homepage, or wherever you want the link to be.

In general this is enough to get you a pretty nice simple website. If you want to mess with the CSS or the HTML style, you can do so, I simply recommend looking up a tutorial. The template READMEs are also quite helpful for just getting started. You can also view my website repo as an example if you want https://github.com/courtcraw/mesadu_wdbinaries . Feel free to ask me if you have questions about something specific I did. HTML and CSS aren't too hard to learn, and there are some good tutorials online if you want.