

Hobart Hackerspace static website

hobarthackerspace.github.io

A [GitHub Pages](#) space for the Hackerspace.

This repository serves as the source for the HHS public web site: <https://hobarthackerspace.org.au/>.

The repository exists under a [GitHub Free for Organizations](#) account. Essentially that means that we get it for free, subject to all content being publicly available. The account login ID is ‘hobarthackerspace’ with a registered email address of ‘github@hobarthackerspace.org.au’. For the password, see our [Bitwarden](#) repository.

Email addressed to the github@hobarthackerspace.org.au address is sent to a small set of members - those who have been updating the site. If you’d like to be added to that group, just ask Brian, Leo or Shane.

Site maintenance

Software platform

The site is built with Jekyll, using the [minimal-mistakes](#) theme.

File repository

The site files live in a [git repository on GitHub](#). (It’s also the place in which this README resides.) See below for some brief notes on working with `git` to work with this if you need refreshing on `git`.

Previews

If you install Ruby & Jekyll on your local machine, you can review site changes before committing to your working copy (let alone pushing up to GitHub) by executing the command `bundle exec jekyll serve`. This will present you with a view of the site at <http://localhost:4000>.

To install Ruby & Jekyll, follow the appropriate guide from [here](#). Don’t bother to install Ruby first (unless you already use it for something else) - it’s included in the above links and following those ensures a compatible version. If you already use it, you probably know about `chruby` or similar to give a suitable virtual environment.

Adding pages

To add a page to the site, put the appropriate markdown document in the `_pages` directory, with suitable “front matter” (use an existing page for a model for new front matter). See [the minimal-mistakes guide](#) for more details.

If the page is long(-ish), add a table of contents as in the [About](#) page front matter.

If you want your page to show up in the left-hand side navigation menu, you also need to add it to the navigation configuration file `_data/navigation.yml`. This is a *YAML* file; add your new page to the tree-structured list under “**side:**”, following the examples there. (Full details of the sidebar navigation configuration are described in the relevant section of [the minimal-mistakes guide](#)).

Add pictures to your content with the [basic markdown syntax](#) `! [Title] (URL)`. For example: `! [Beambox-safety] (/assets/ima`
More sophisticated picture use can be done using the [Figure layout](#)

File/directory structure of the repo

To keep things maintainable with a number of us maintaining the site, keep the “_pages” directory and filenames structured to mirror the site structure. (So, if the URL your page is `/example/content/`, call the page file `content.md` and place it in the directory `_page/example/`, creating that directory if necessary.)

Working with git on the site

The “live” version of the website is the set of files hosted on GitHub pages within the branch named “main”. The process of pushing “main” from a remote copy to GitHub triggers an inbuilt GitHub workflow to generate and publish the appropriate pages.

Working copy

You can get a working copy of the site by cloning the [repository](#)

Create your own branch to make changes to that working copy (see above about using Ruby & Jekyll for previews). It's convenient for all of us if we each call our branch by our name.

Publishing

When you have some changes to publish, commit them to your branch and then merge that branch back into the “main” branch. (“git checkout main” followed by “git merge ” will do that.)

Then push your merged copy up to GitHub with “git push”. That will trigger re-publishing the site (it takes about 30 seconds to a few minutes).

Finally, don't forget to checkout your branch again before you make any more local changes.

Domain Name

Domain name registration and DNS record hosting.

Our [hobarthackerspace.org.au](#) domain is registered through Shane Dalglish's NISS company. There are two aspects - registration and domain record hosting.

- Registration is done via [niss.partnerconsole.net](#) under the name of “The Committee” and with emails sent to the [committee address](#). Login details are in the Bitwarden account under “Hackerspace Domain Name Registration NISS”
- Domain records hosting is accessed via [https://hosting.niss.net.au:8443](#). Once again the login details are held in the Bitwarden account under “Hackerspace Hosting & DNS”.

Note re Bitwarden: Our shared Bitwarden password store is at [vault.bitwarden.com](#) under the account “bitwarden@hobarthackerspace.org.au”.

DNS records for the website

There are three specific record types relating to our website. These are constructed according to [the quite specific rules from GitHub Docs](#) about domain name redirection.

In particular, that means:

- We have a CNAME record for ‘www.hobarthackerspace.org.au’ which points to [hobarthackerspace.github.io](#);
- We have four A records which point to IPv4 addresses of GitHub Pages (185.199.108.153, 185.199.109.153, 185.199.110.153 & 185.199.111.153); and
- We have four AAAA records which point to IPv6 addresses of GitHub Pages (2606:50c0:8000::153, 2606:50c0:8001::153, 2606:50c0:8002::153 & 2606:50c0:8003::153).

Add-on tools

- Slideshow Carousel
 - “Jekyll without plugins” / Slider/Carousel
- Contact Us page

- This uses a very simple form POSTing to an account on the [HeroTofu service](#). This free account is good for 1000 submissions/month.
- Queries are forwarded to the Committee. Account credentials are in the password safe.

Site search facility

Tools

We have a free subscription to [algolia](#) under the account name `github@hobarthackerspace.org.au`. (The password is in the password safe.) This provides a searchable index and delivers search results via a `jeekyll/minimal-mistakes` theme page. Note that the free account is limited to a total of 10k records and a monthly aggregate of 10K calls; this shouldn't limit us in normal use, but may be an issue if we get some kind of bot attack.

Guides

- Advice for setting up Algolia using GitHub & Jekyll is [here](#).
- Excellent specific advice for the minimal-mistakes theme is [here](#).

Indexing

The index needs to be re-created each time an update occurs to the site. This is (still to be) done using a subscription to the continuous integration tool [Travis-CI](#), which hooks to the GitHub repo each time a push is made.

If you are previewing locally, the index still works. Use the bash command `ALGOLIA_API_KEY=secret-key bundle exec jeekyll algolia` to re-index the site (the api key secret-key is in the safe with the algolia account password). Note that there is only the one index, used by both the live site and any local preview, so that updating the index from your local machine will update the live index, too.

Results page

The results page is controlled by a file in the theme `_layouts/search.htm`.