

## Building and publishing your book

Once you've added content and configured your book, it's time to build the HTML for each **page** of your book. We'll use the `jupyter-book build` command-line tool for this. In the *next step*, we'll stitch these HTML pages into a book.

### Prerequisites

In order to build the HTML for each page, you should have followed the steps in [creating your Jupyter Book structure](#). You should have a Jupyter Book structure in a local folder on your computer, a collection of notebook/markdown files in your `content/` folder, a `_data/toc.yml` file that defines the structure of your book, and any configuration you'd like in the `config.yml` file.

### Build each page's HTML

Now that your book's content is in the `content/` folder and you've defined your book's structure in `_data/toc.yml`, you can build the HTML for each page of your book.

Do so by running the following command in the directory that contains your book's directory:

```
jupyter-book build mybookname/
```

or change directory into your book's directory, and run:

```
$ cd mybookname
$ jupyter-book build .
```

This will:

- Use the links specified in the `_data/toc.yml` file (pointing to files in `/content/`) and do the following:
  - Run `nbconvert` to turn the content files (e.g., `.ipynb`, `.md`, etc) files into HTML
  - Replace relative image file paths so that they work on your new built book
  - Place all these generated files in the `mybookname/_build/` directory.

After this step is finished, you should have a collection of HTML files in your `_build/` folder.

### Page HTML caching

By default, Jupyter Book will only build the HTML for pages that have been updated since the last time you built the book. This helps reduce the amount of unnecessary time needed to build your book. If you'd like to force Jupyter Book to re-build a particular page, you can either edit the corresponding file in the `content/` folder, or delete that page's HTML in the `_build/` folder. To rebuild all files, just delete the `_build` directory. All HTML files will be regenerated.

### Create an *online* repository for your book at GitHub

You've created your book on your own computer, but you haven't yet added it online. This section covers the steps to create your own GitHub repository, and how to add your book's content to it.

The simplest is to use GitHub-Pages to build the HTML for your book, and serve it to the web.

You just make your book's directory a **git** repository, edit all the content, run `$ jupyter-book build .` and use the GitHub Desktop tool to push all your files up to GitHub, using the master branch. GitHub will generate your web pages using Jekyll and serve them. If this doesn't make any sense, don't worry.

But if you've got a bibliography, GitHub doesn't know how to use the special Jekyll package for bibliographic references. You'll get a mysterious "Liquid tag" error. Don't panic. We'll build the HTML locally, using the special package, and then push it to GitHub, bypassing the GitHub machinery to build HTML. That's the `ghp-import` command described earlier, covered in [building and publishing your book](#).

### Set up GitHub

Get an account.

### Clean up this page and the next page (Tuesday, 8 Oct)

Get the application **GitHub Desktop**

1. First, log-in to GitHub, then go to the "create a new repository" page:

<https://github.com/new>

2. Next, add a name and description for your book. You can choose whatever initialization you'd like.

3. Now, clone the empty repository to your computer:

```
git clone https://github.com/<my-org>/<my-book-name>
```

4. Copy all of your book files and folders (what was created when you ran `jupyter-book create mybook` ) into the new repository. For example, if you created your book locally with `jupyter-book create mylocalbook` and your online repository is called `myonlinebook` , the command would be:

```
cp -r mylocalbook/* myonlinebook/
```

This will copy over the local book files into the online book folder.

5. Commit the new files to the repository in `myonlinebook/` :

```
cd myonlinebook
git add ./*
git commit -m "adding my first book!"
git push
```

That's it!

## Next step: build your book

Now that you've created the HTML for each page of your book, it's time to stitch them together into a book. That's covered in the next section.