How to Build Your own Web Site

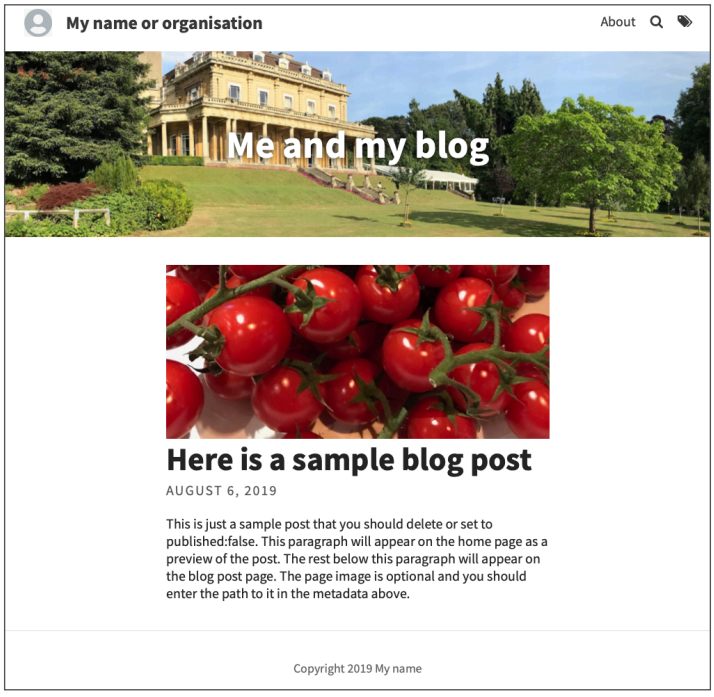
Freely hosted with GitHub Pages

Step 1. Getting the Software required	2
Step 2. Setup Github	3
Step 3. Getting a local copy	5
Step 4. Configuring your site	7
Step 5. Adding and Editing Content	9
Step 6. Getting the new content up to the web (making your web site live)	13
Step 7. Adding to and Editing your site Online	15
How does all of this work?	16
Clone, Push, Pull - What do these terms mean exactly?	17
Further reading	18
This is not the End	19

This document explains all of the steps to take when creating a static web site using the free hosting at Github Pages.

The site that we build can be a blog or just an information web site. We will use a template that styles the site making sure that it is responsive to different screen and device sizes.

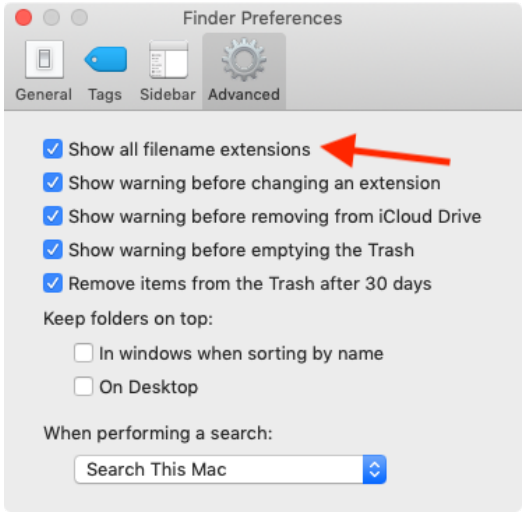
You will learn how to set the site up and configure with your own branding. Further, more complex styling can be achieved, but for this we need to have some knowledge of HTML and CSS.



Step 1. Getting the Software required

Before we begin:

It really helps if you can see the extension to a file.
Set this up in the Finder preferences.

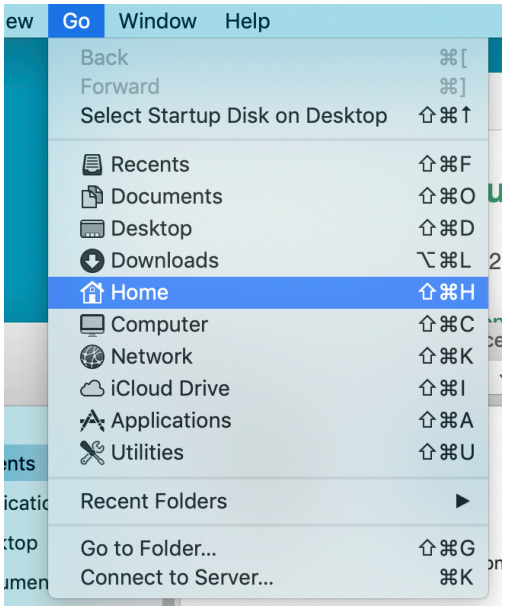


You will only need the application called *Atom* and to help you get the correct settings **I am providing you with a distribution** (`www.publisha.org/resources/atom.zip`) package.

This will create an `atom` folder which you need to place inside your `Home` folder.

Note: If you use Safari as your browser then the zip file will unpack into your downloads folder.

From the Finder use `Go` to find your Home folder. Read the `readme.txt` file that comes inside this zip file.



Installing Atom

1. In your Home folder (use `Go > Home`) create a new folder and name it `Applications` (note the capital **A**). - you may have already created an Applications folder.
2. Drag the `Atom.app` into this Applications folder.
3. Drag the `for_home_folder.zip` file into the home folder
4. Double click this file to unpack - this creates a **hidden** folder called `..atom` (but you won't see it).

You can view hidden files by using **SHIFT-CMD full-stop** on the keyboard.

The `Atom` program comes ready built with the following packages that will help you write posts for your blog.

- tool-bar
- markdown-writer
- toolbar-markdown-writer
- zen (this will provide a distraction free interface)

Now that you have the software installed you can sign up for the free web site at github.

Step 2. Setup Github

We are going to set up a free account on github.com

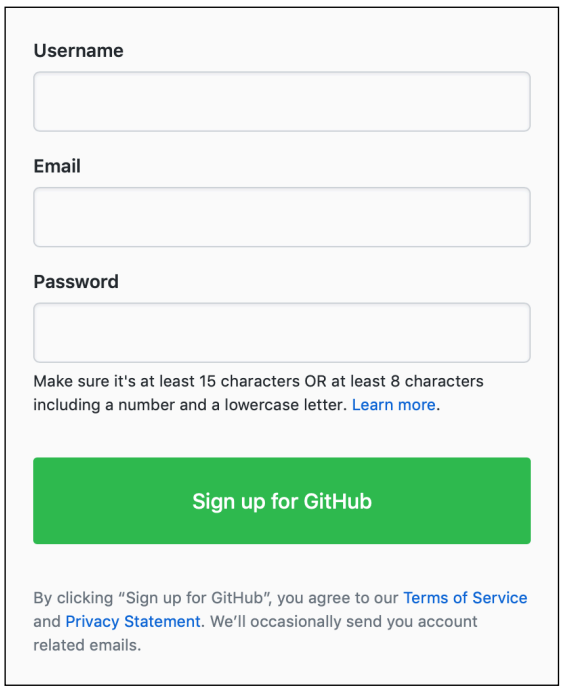
Naming the account (**your *username***) is important (although you can change later) because this will become part of your website URL.

For example, if you name the account `MyLoveofBooks` then the web site will become `myloveofbooks.github.io`. Not all names will be available and yours may be rejected if it is already taken.

Please remember your username and password or keep somewhere safe.

Choose a **Free** account and go through the various steps although you can skip the questions about your interests. You will also need to verify your email address. You can use your Brookes email address but you don't need to and - you can change this at any time, after all, this web site is your forever even after you leave Brookes.

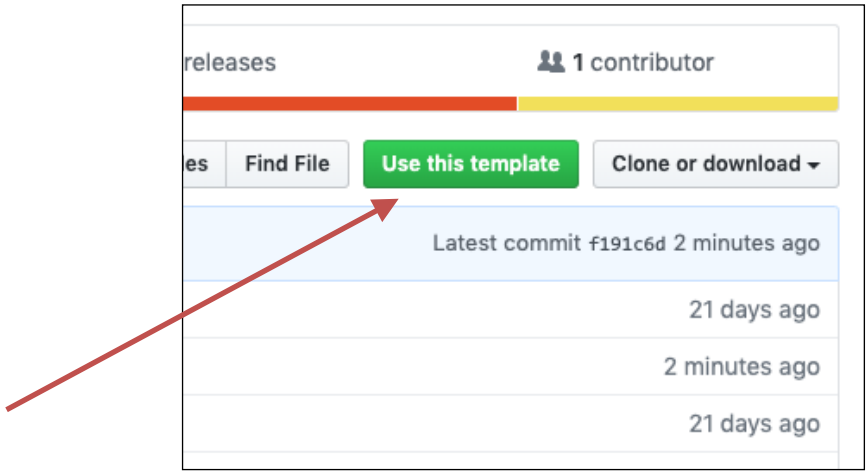
Optionally you can edit your profile and add an avatar image.



Now you are signed in to GitHub

Once you are signed in to Github, go to the following URL:

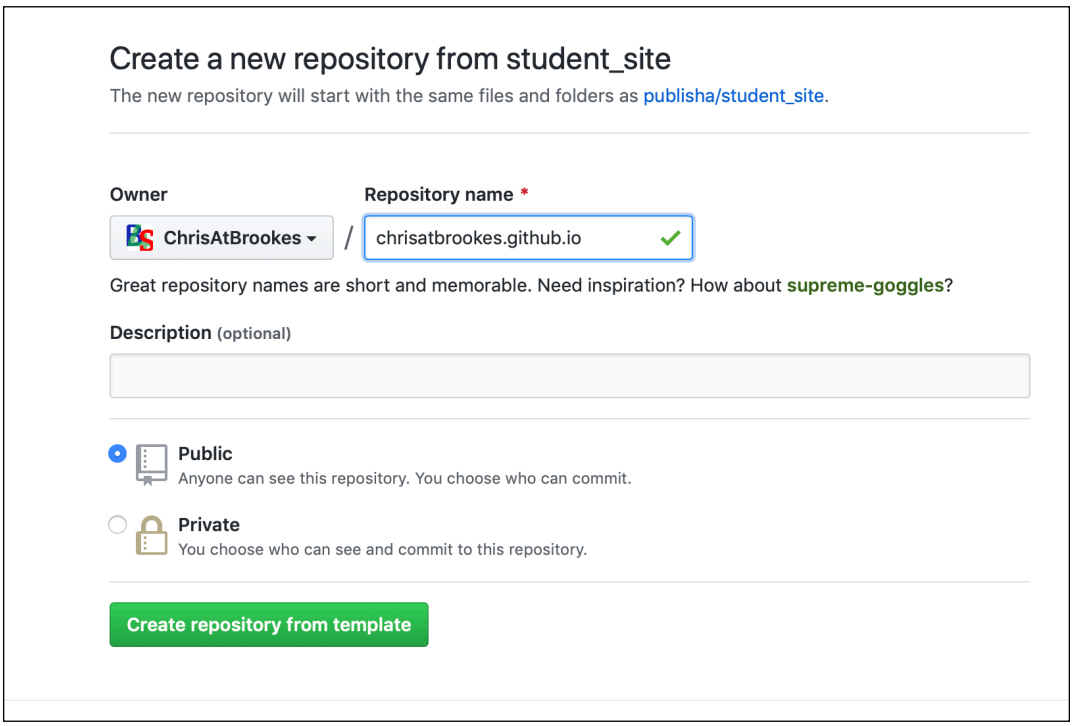
`https://github.com/publisha/student_site/`



Use the template link to receive the repository in your own github account. Make sure that you choose the `Public` option. You will be asked to name the repository.

Naming Your Repository

Name the repository using lowercase letters but use the same as your GitHub owner name



This is important: You need to name the repository `username.github.io` where username is the name of your username or owner name but in lowercase letters.

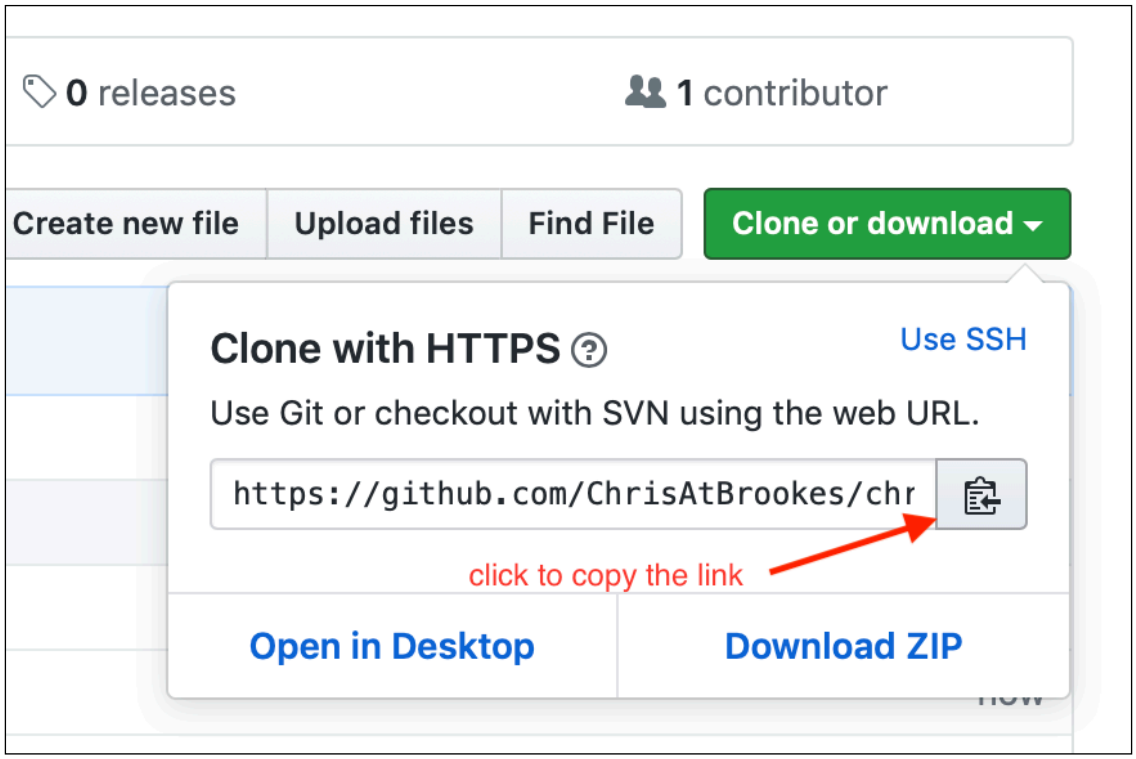
You should now have a copy of the template web site in your github account.

This will be the URL of your site.

Although it is always possible to edit the files directly on the github site, there is a much better way! We can get a local copy to be edited on our computer.

Step 3. Getting a local copy

Select the repository just created and locate the button labelled `Clone or download`. Once you click this, you need to copy the URL (the button to the right of the URL field will copy this for you).

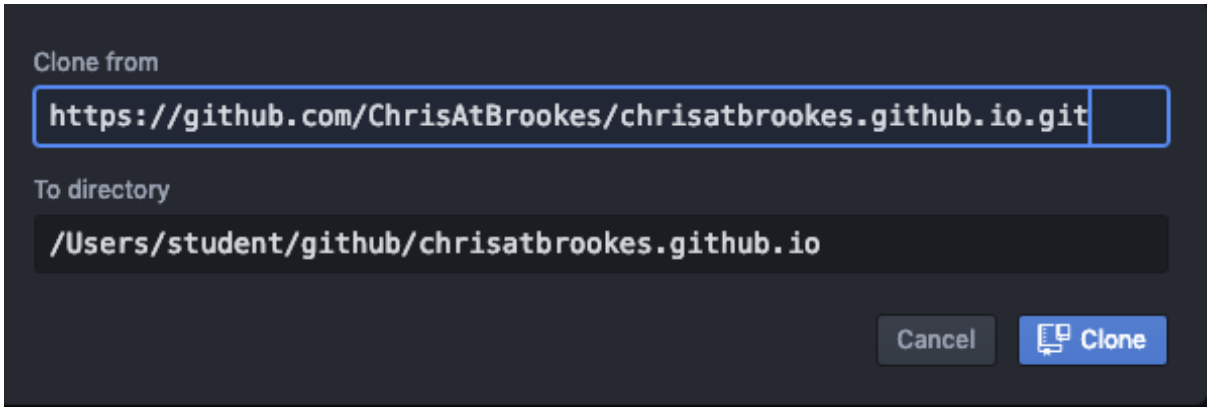


Now open Atom

You should see a set of icons at the bottom of the window.

The icon at the far right end looks like this:

Click this and a panel appears for us to paste in the URL. This operation is known as cloning the repository..



Here is the clone window. Paste the contents of your clipboard (you just copied the link right?) into the top box.

In the 'To directory' box the default will be to place the copy into a github folder inside your home folder.

Note: Before you finish setting up, you should move the complete web site folder out of the `github` folder to your `Creative Cloud` folder. That way, you can find it on any of the computers that you are signed into.

You must take the whole folder named ***yoursite*.github.io**

Close the Atom window and locate the github folder inside your Home folder. Now drag the folder called `yoursite.github.io` to a location on your **Google Drive** or your **Creative Cloud** space.

Step 4. Configuring your site

Before you make your web site live and post new articles, you need to edit the configuration file with your own details. The file you must edit is:

`_config.yml`

The file itself has comments so it should be self explanatory but here follows the important changes that you must make:

Site configuration

Please make the following changes before attempting to use the student blog.

Configure as your own website in the file called `_config.yml`:

Change `username` to the username you chose for GitHub. In other words, the url will be the repository name prefixed with `https://`

Meta and Branding

Meta variables hold basic information about your site which will be used throughout and as meta properties for search engines, browsers, and the site's RSS feed.

Change these variables in `_config.yml`

Variable	Placeholder	Make yours
url:	https://yoursite.github.io	Change this to your url
title:	My name or organisation	Name of website will appear top left on your pages
description:	A website with blog posts and pages	Short description, primarily used by search engines
avatar:	assets/img/face.jpg	This can be changed to your own picture or a logo optionally
favicon	assets/favicon.ico	Change to reflect your brand
header_text	Me and my blog	Yours to change
header_feature_image:	/images/hhh.jpg	You should create a different image to replace the picture of Headington Hill Hall. This image needs to be **no** smaller than 1800 pixels wide.
footer_text	Copyright 2019 *My name*	Put your name or organisation in here

You can also add your social media links such as Instagram, Twitter, Facebook etc.

There are other settings in the `_config.yml` file such as the configuration of prose. This is used if you use `prose.io` to edit and post to your site (see the section '**Step 7. Adding and Editing your site online**).

You will need to edit the `siteurl` in the **prose** section of the `_config.yml` file.

You also need to edit the placeholder **tags ur**l in the file called `_mdwriter.cson`. This is at **line 34** in that file. This will make it easier to select and re-use tags for your posts.

```
urlForTags: 'https://yoursite.github.io/tags.json
```

Step 5. Adding and Editing Content

Writing to your blog involves using the Markdown language and this will then be automatically be converted to HTML for the web site. You can learn about Markdown by viewing the Markdown Guide on the web. (see Further Reading)

There are many writing apps that use Markdown but we are using Atom with a user-friendly toolbar that makes structuring your content easy..

There are 2 types of pages in your site:

Posts

You can post articles or blog posts and these will be dated so that they appear in date order, on the home page, the latest at the top. You will find a sample post in the _posts folder and you can duplicate this or click on the icon at the bottom left of the Atom window.

Each post has it's own page but the extract will appear on the home page. The extract is the first paragraph of your text. Placeholders for the page metadata will be included for you to edit.

The number of post extracts that appear on the home page can be configured in the `_config.yml` file (the default is 5 posts with a link to the previous 5 etc.)

The posts themselves are files that will automatically be generated inside the `_posts` folder.

Pages

You can create other pages for special (non-dated) content and, as long as you save them inside the pages folder, they will automatically generate a menu item at the top right of your site.

Apart from the `About` page described below, you can also create any page that you like to add to your site and, thus, a menu item. Examples could be a CV, or a Project page, or even a gallery of your photos.

The About page

This is the one page ready for you to edit. This page needs to be edited to contain information about yourself or what you want this site to do!

Editing this page will be a good way for you to get used to the way to edit and create pages using `Markdown`.

If you want to create another page, you can duplicate the *About* page so that you can see what the structure needs to be.

Markdown

Markdown is a way to write for a web site that uses simple structure and markup that will then be converted to **HTML**. It is useful to understand how **Markdown** works, but actually, when we use Atom, the elements can be selected from the toolbar.

Here are some examples of what the Markdown syntax means.

```
## Heading level 2
### Heading level 3
This is **bold** text.
This is *italic* text
```

YAML metadata

At the top of each page or post there will always need to be some metadata to control certain aspects of the page when it is converted to `HTML` and then included in your site. When you create a post with Atom, the metadata is added automatically, but you need to edit this/

This metadata is enclosed within the 2 groups of hyphens. Here is an example.

```
---
layout: post
title: Here is a sample blog post
date: 2019-10-19
published: true
header_feature_image: images/2019/07/tomatoes.jpg
caption: "Juicy Tomatoes"
tags:
  - Journalism
  - Life
  - Food
---
```

Notice that the header_feature_image does NOT have the leading slash.

Some of this metadata is generated automatically when you create a new post (the date for example).

You will also need to edit this metadata to confirm that the post is ready to be published; change from **`published:false`** to **`published:true`**. When the page is set to `published:false` it won't appear on your live site.

Page image

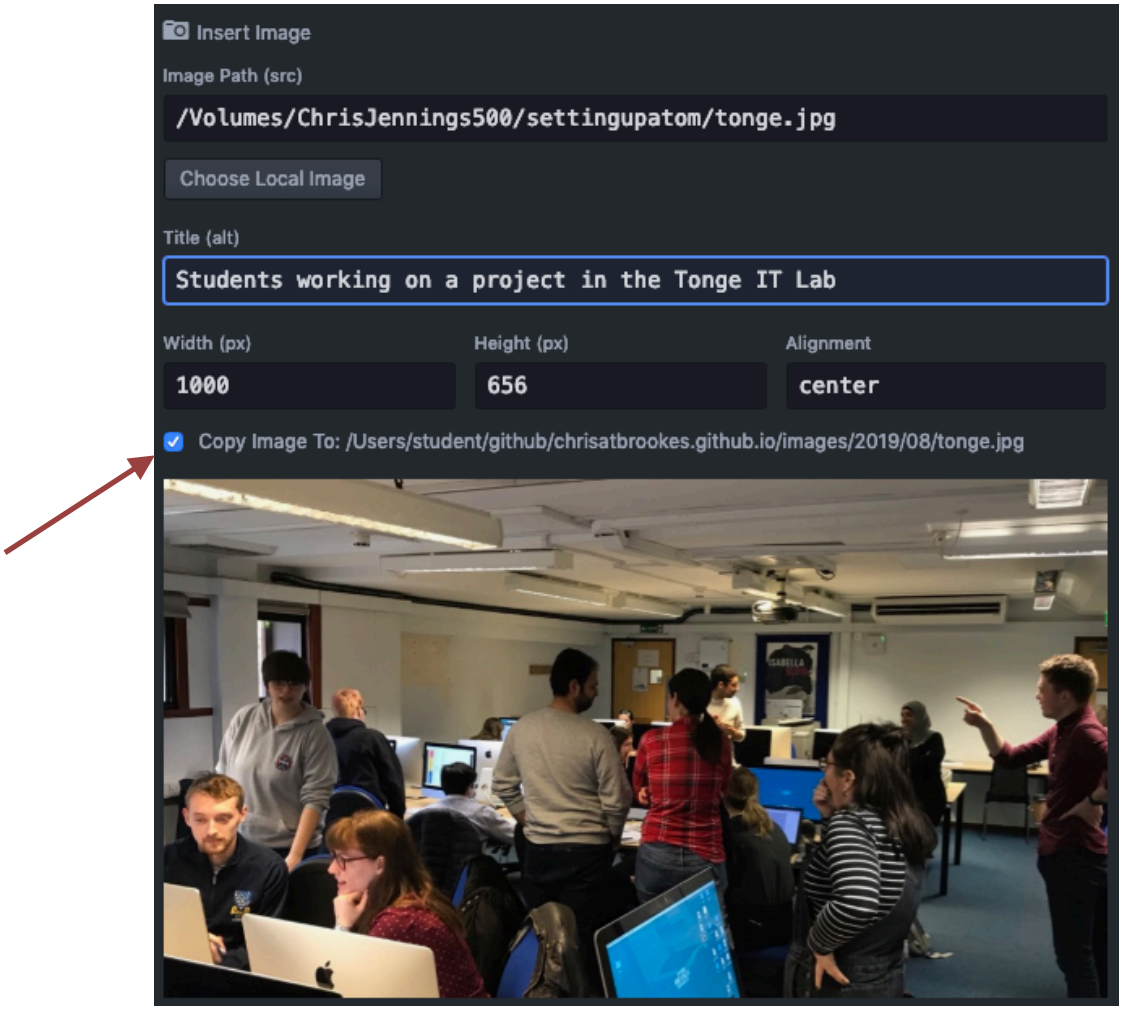
We will describe how you can add images to your posts and pages later, but all content can also have a **`header_feature_image`** and this image will appear at the top of the page, with the post heading set against it. You will need to place this image into your **`images`** folder and then provide the link to it in your metadata. The header_feature_image does NOT have the leading slash.

Tip you can drag images into the images folder (make sure that they do *not* have spaces in the file name) and then right-click over the image and copy the path. Then paste and edit this path into the metadata for header_feature_image: pathtoimage.

Adding images to a page

This is done through the image icon on the **`markdown`** toolbar at the bottom of the active Atom window.

You will get a popup window and you need to select and image from somewhere on your system. Make sure to click the box **`Copy to …`**

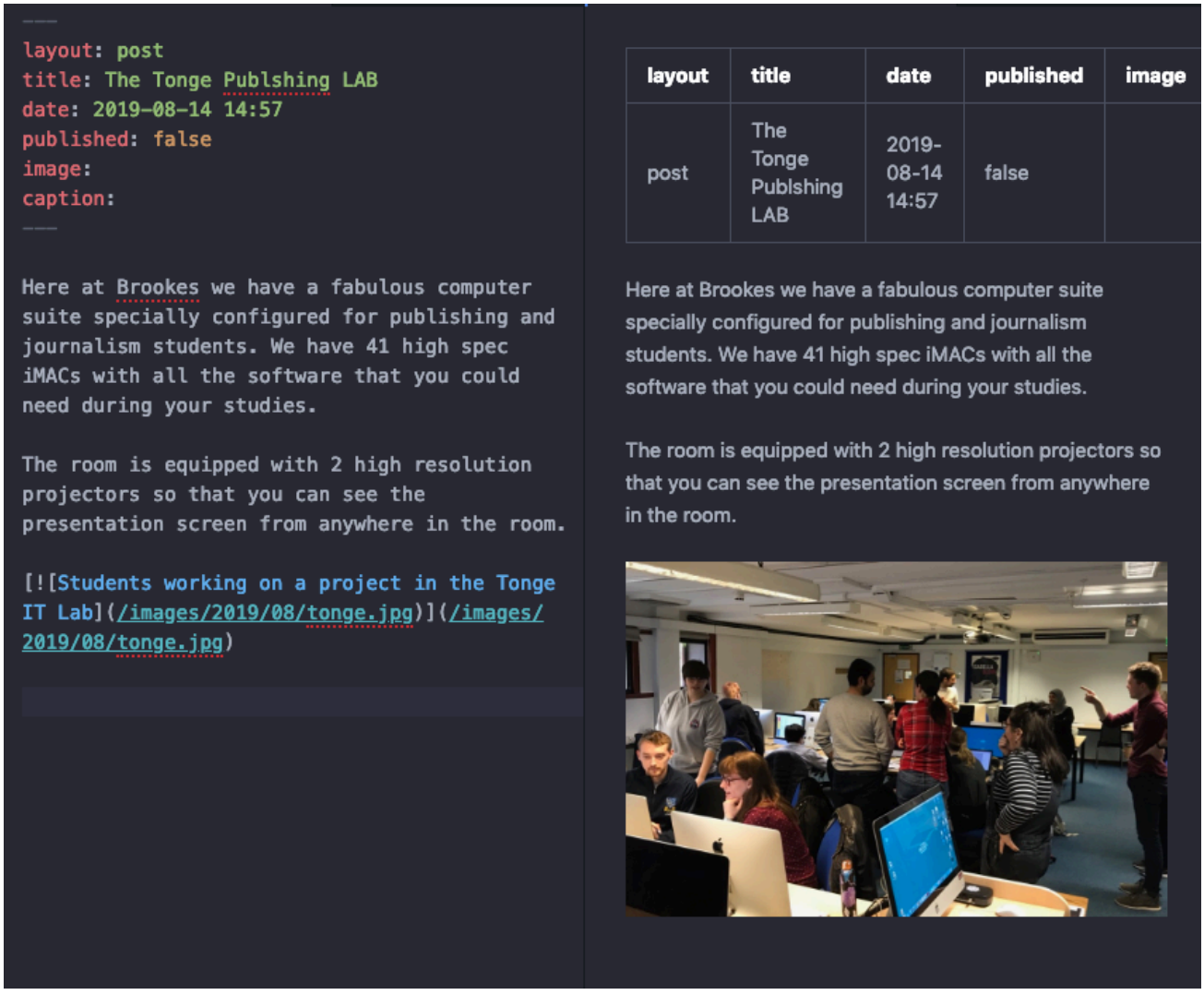


Previewing your page

You can preview the appearance of your page by clicking the icon near to the bottom left.

The icon appears like this

And we then see the preview pane on the right.:



```
---
layout: post
title: The Tonge Publshing LAB
date: 2019-08-14 14:57
published: false
image:
caption:
---

Here at Brookes we have a fabulous computer
suite specially configured for publishing and
journalism students. We have 41 high spec
iMACs with all the software that you could
need during your studies.

The room is equipped with 2 high resolution
projectors so that you can see the
presentation screen from anywhere in the room.

[![Students working on a project in the Tonge
IT Lab](/images/2019/08/tonge.jpg)](/images/
2019/08/tonge.jpg)
```

layout	title	date	published	image
post	The Tonge Publshing LAB	2019-08-14 14:57	false	

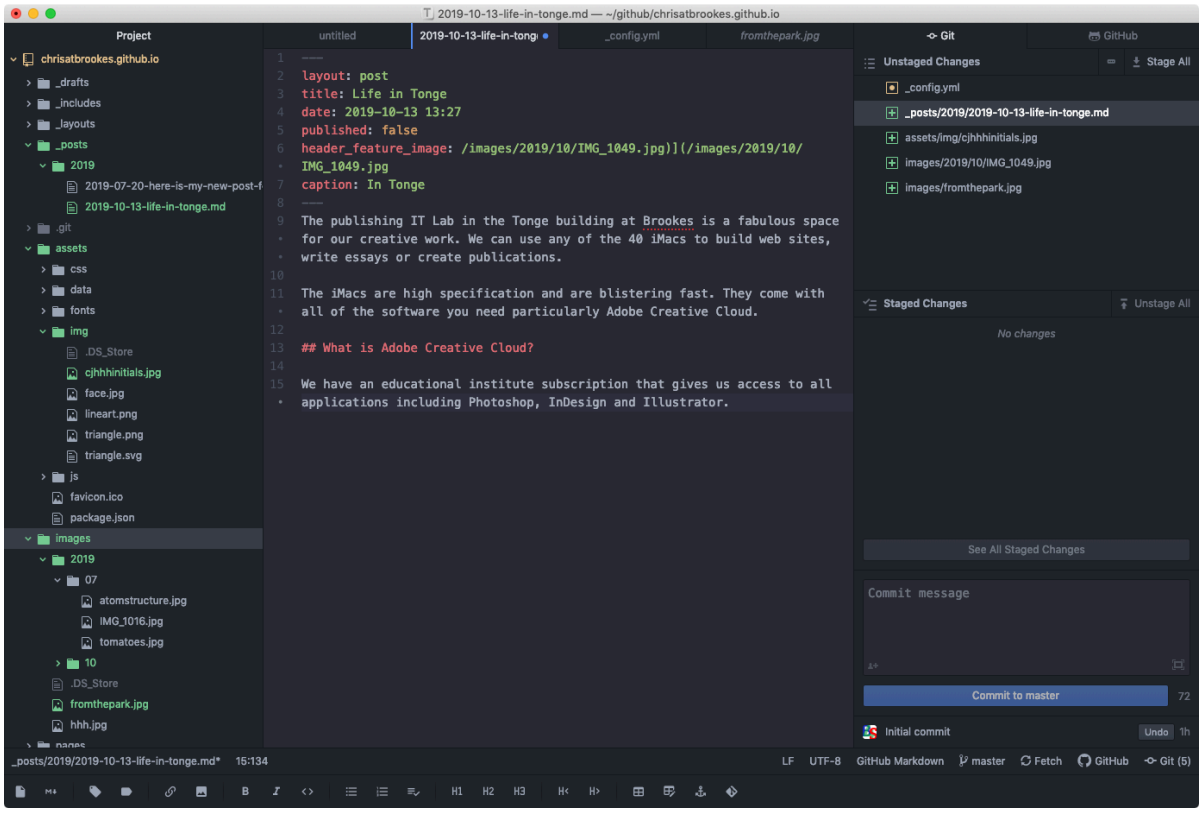
Here at Brookes we have a fabulous computer suite specially configured for publishing and journalism students. We have 41 high spec iMACs with all the software that you could need during your studies.

The room is equipped with 2 high resolution projectors so that you can see the presentation screen from anywhere in the room.

This isn't an absolutely perfect rendition of your web page, but it simply confirms that the structure is correct and the **markdown** is converted to **HTML** correctly.

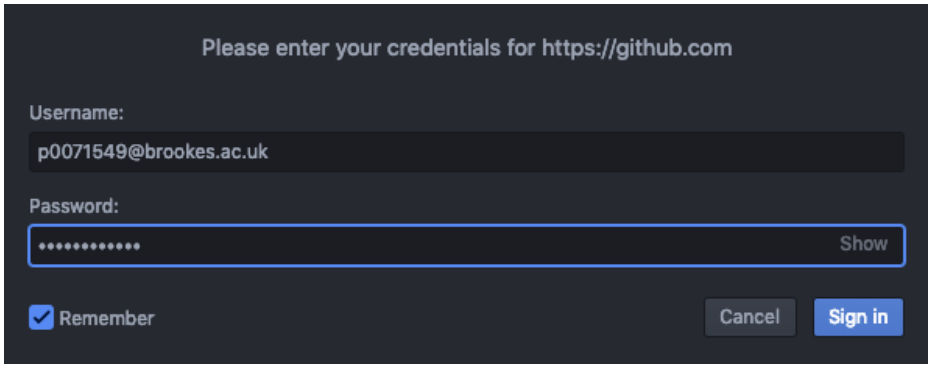
Step 6. Getting the new content up to the web (making your web site live)

Atom interacts with your **`github`** site, so once you have made changes or added a post, you will need to open the **`Git`** pane (bottom right) and then notice the changed or added files up at the right. These are u**`nstaged Changes`**.



Now click **`Stage All`** and these will move down to the **`Staged Changes`** pane. You need to write something in the commit box, so that the changes will have a meaning to you later.

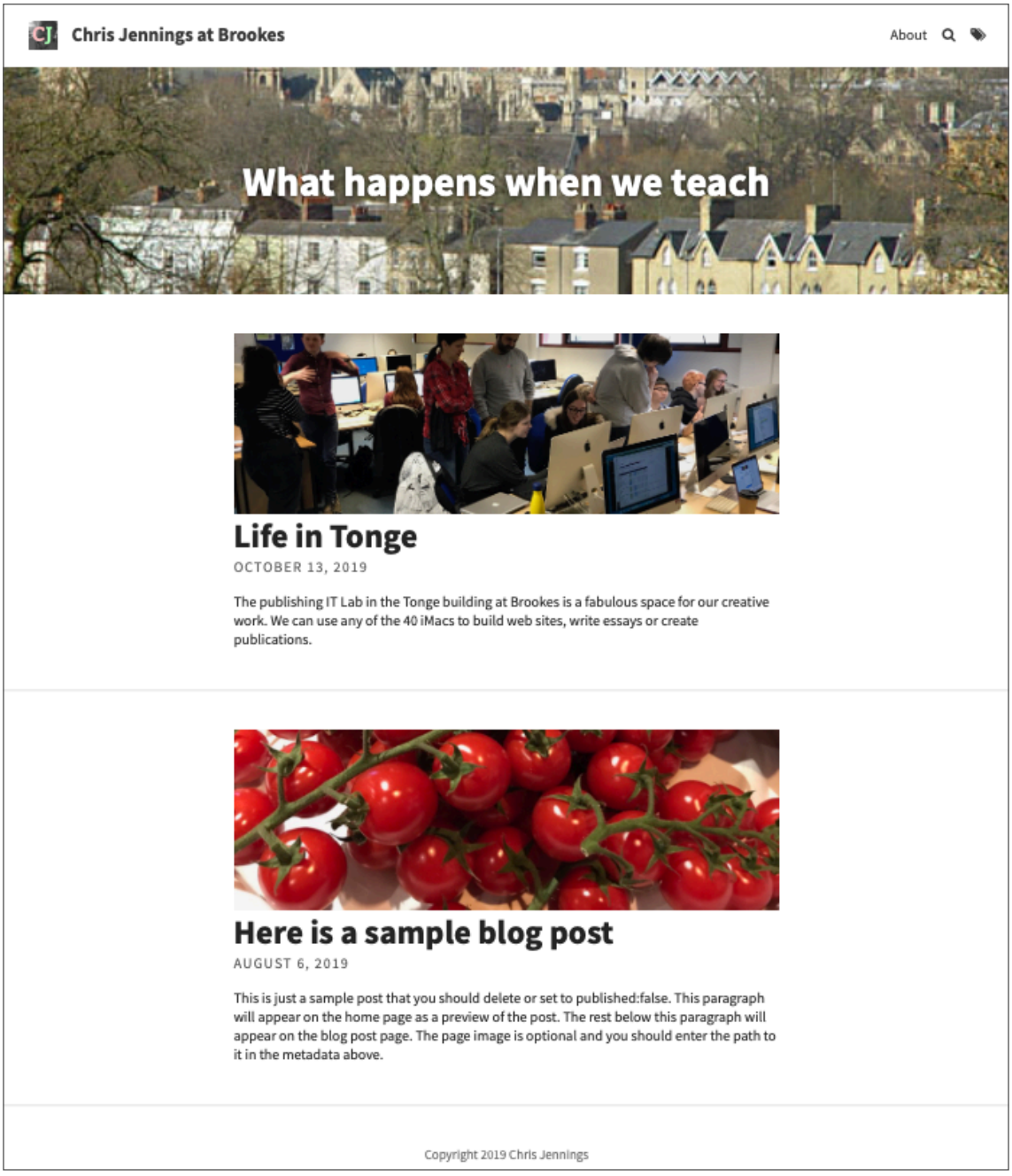
You can now **`Push`** these changes to **`Github`** but for the first time you will need to add your credentials (please tick the **remember box** so that you only do this for the first time).



Take a first look at your new site

Your site is available now at **`username.github.io`**. Change **`username`** of course.

You should now be able to write further blog posts and and add further pages. You can customise your site by changing the title, the main image and the avatar through the `_config` file.



For further changes to the appearance you will need to make changes to the CSS. This is the subject of future advanced sessions.

Step 7. Adding to and Editing your site Online

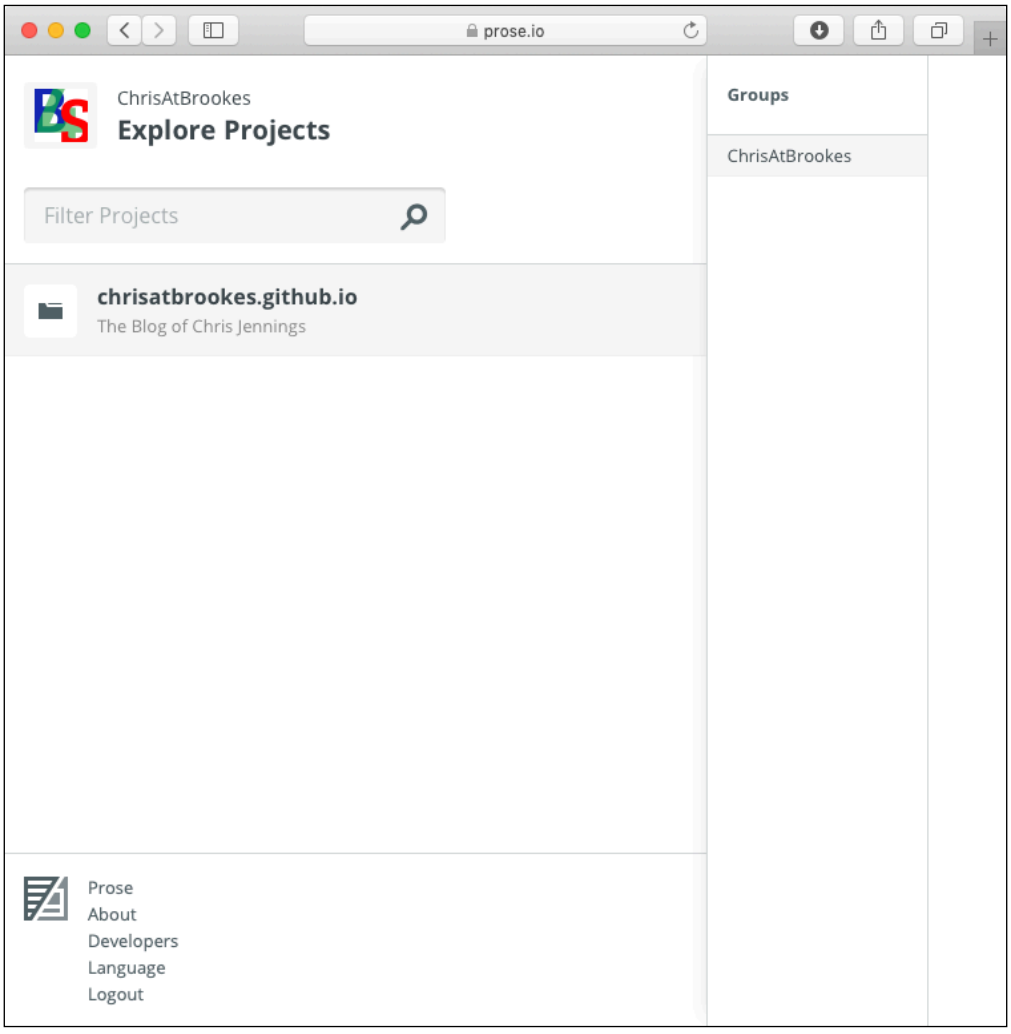
Although we recommend using **Atom** as the way to write posts and edit your site (you certainly need to use it for editing the config file and changing the appearance of your site), you can optionally use an online tool from anywhere with an internet connection to post articles.

Using Prose

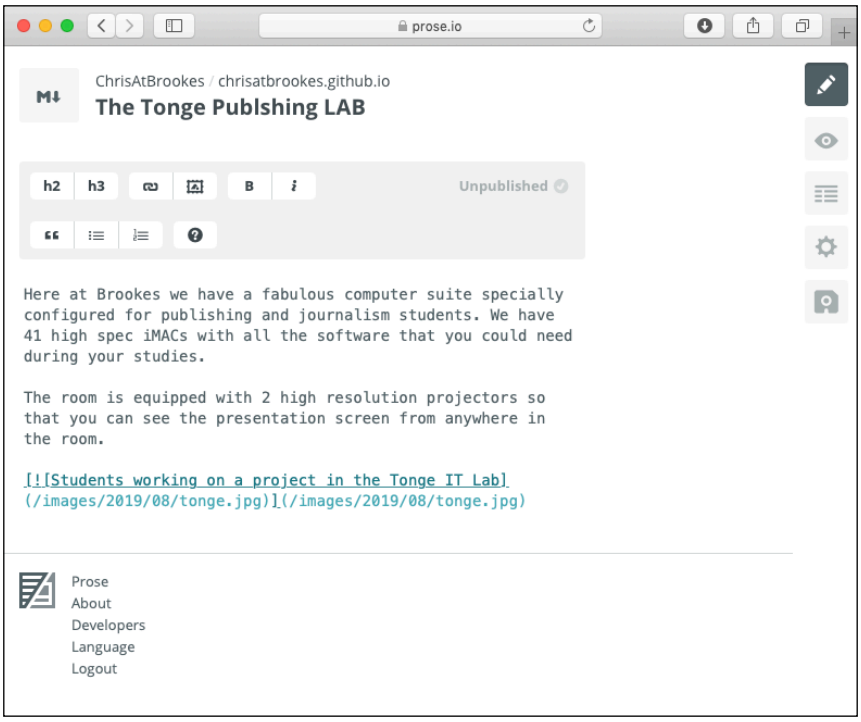
Make sure that you change the **siteurl** under the **prose** section of the **_config.yml** file.

Go to the web site **prose.io** and authorize with your GitHub credentials.

You should now see your repository available.

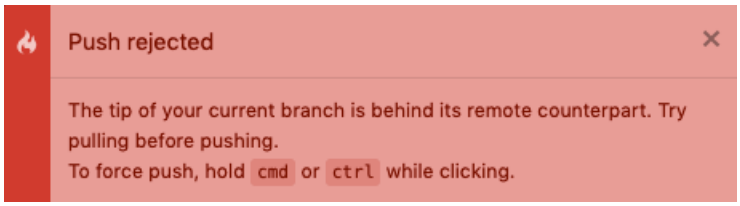


Once you go into the repository, you can edit the posts or create new posts directly.



Synchronising

Be aware that if you use **`prose.io`** to edit your site then your local copy (edited with Atom) will not be the same. If this is the case then you will need to find the **`Fetch`** link at the bottom right of the Atom window to **`pull`** the changes to your local space.



How does all of this work?

Github Pages

GitHub provides for free github pages.

You can read more about this here: (https://pages.github.com).

Basically anyone can host web pages here but we are using a static site generation system called Jekyll which is also supported in Github pages.

Jekyll

This software operates your site, converting the markdown files into HTML when you *push* your files up to the GitHub repository.

You can read more about this here: (https://jekyllrb.com).

The Jekyll site that you have created from the provided template includes all of the files necessary to build the live pages including the stylesheets (CSS) and some javascript. There is also a system of logic that builds the menus and pagination.

Liquid

The logic behind the web site that is used by Jekyll is called liquid. You don't need to have knowledge of this coding language but if you are interested then you can explore further here. `https://shopify.github.io/liquid/`

Clone, Push, Pull - What do these terms mean exactly?

When you `clone` the site, you are taking a copy of the complete repository from GitHub to your local computer.

When you use `Push`, you are sending to github the changed files.In other words, you will deliver only files that have changed on your local copy in comparison to the live web site. This might mean new files of course

When you use `Pull`, you are only receiving the changed files.In other words, you will receive only files that have changed on github.

`Push` and `Pull` are found under the item `Fetch` at the bottom right of the Atom window.

Making a Commit

When you make a change to your site or post a new article or page, you are advised to write a short commit message. That way, you can go through your history to see what changes have been made. This also means that you can revert changes back to an earlier version.

Storing in the Cloud?

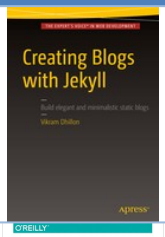
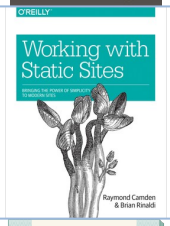
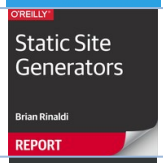
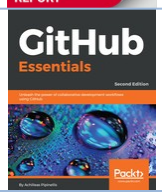
If you save your local repository in the cloud (remember - we moved the cloned site from our github folder to Google Drive or Creative Cloud -), then you can edit the site with Atom on any computer where you have signed in to that cloud service (Google drive or Creative Cloud). Just drag the complete folder onto the Atom icon to open the complete project. You just need to be sure to `Push` the changes up to your live GitHub site.

Not storing in the cloud?

If you want to edit a local copy of your site **not** stored in the cloud and you want to edit in **more than one** place, you will need to, first `Clone` the GitHub repository (use Atom as instructed above) and then, to keep these local copies in synch, you will need to use `Pull` to get the latest version changes on GitHub.

Although this can seem a bit complicated, Atom, will warn you if the current live web site has more recent changes than your local copy. You will be asked to `Pull` first before trying to `Push`.

Further reading

	Title	Details
	Creating Blogs with Jekyll	Vikram Dhillon, 2016, APRESS
	Working with Static Sites	Raymond Rinaldi, 2017, O'Reilly Press
	The Atom Flight Manual	Website: https://flight-manual.atom.io
	The Markdown User Guide	Website: https://www.markdownguide.org
	Static Site Generators	Brian Rinaldi, 2015, O'Reilly Press
	GitHub Essentials	Achillieas Pipinellis, Packt Publishing, 2018

All of the above books can be found on Safari Books Online - available through Oxford Brookes University library.

For a further help on using **`prose.io`** , go to Todd Riley's web site here:

`https://www.websbytodd.com/documentation/using-prose/`

Also checkout the Wiki for **prose.io** here:

`https://github.com/prose/prose/wiki/Getting-Started`

This is not the End

Once you have everything set up as outlined above you should be able to post to your site without difficulty.

But what about...

I want to use my other computer

As pointed out above, you can use `prose.io` to post to your site from anywhere with an internet connection. However, if you want to use Atom on a different MAC computer then follow the instructions set out in **Step 1**. To do the same on a Windows PC, you will need to download `Atom` yourself (`https://atom.io`) and then add the following packages:

- tool-bar
- markdown-writer
- toolbar-markdown-writer
- zen (this will provide a distraction free interface)

Once you have Atom running then you will need to follow **Step 3**.

I want my site to look different

The CSS (stylesheet) that controls the appearance in the CSS folder inside the assets folder. You can edit the CSS with Atom however, you won't see the results until you push the changes up to your live web site. This can be a bit frustrating and there are 2 ways that you could make this easier:

Get your Jekyll site running locally and the instructions are here:

`https://help.github.com/en/articles/testing-your-github-pages-site-locally-with-jekyll`.

Unfortunately, we can't do this on the Tonge MACs at the time of writing this.

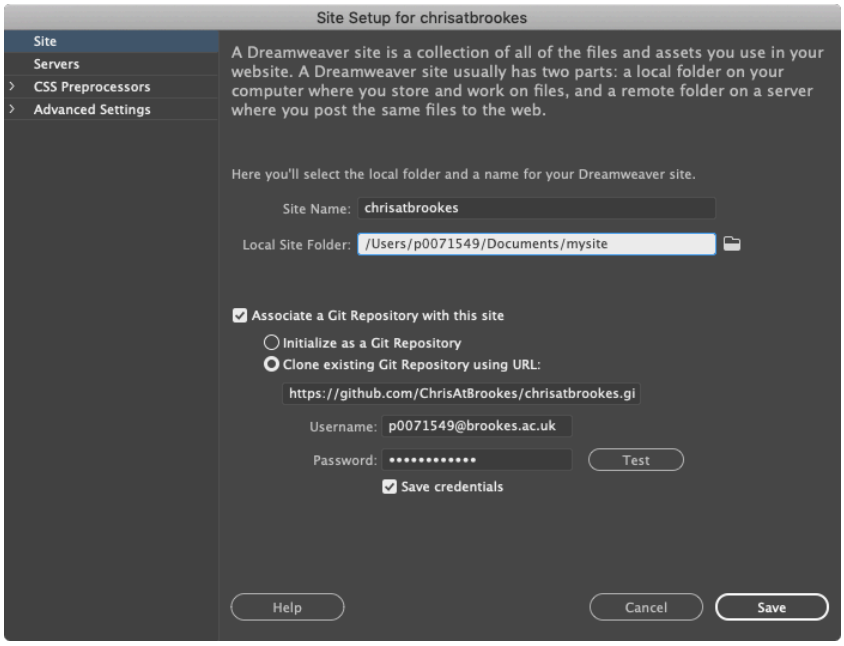
Another way to check your CSS changes is to use the 2 sample HTML pages that use the same CSS. Just open these in your browser and then as you edit the CSS, you can refresh the pages to see the effect.The 2 files are in the root of your project (`samplehome.html` and `samplepost.html`)

I want to use Dreamweaver

If you are familiar with Adobe Dreamweaver and you would like to use this to edit the CSS of the site, you can!

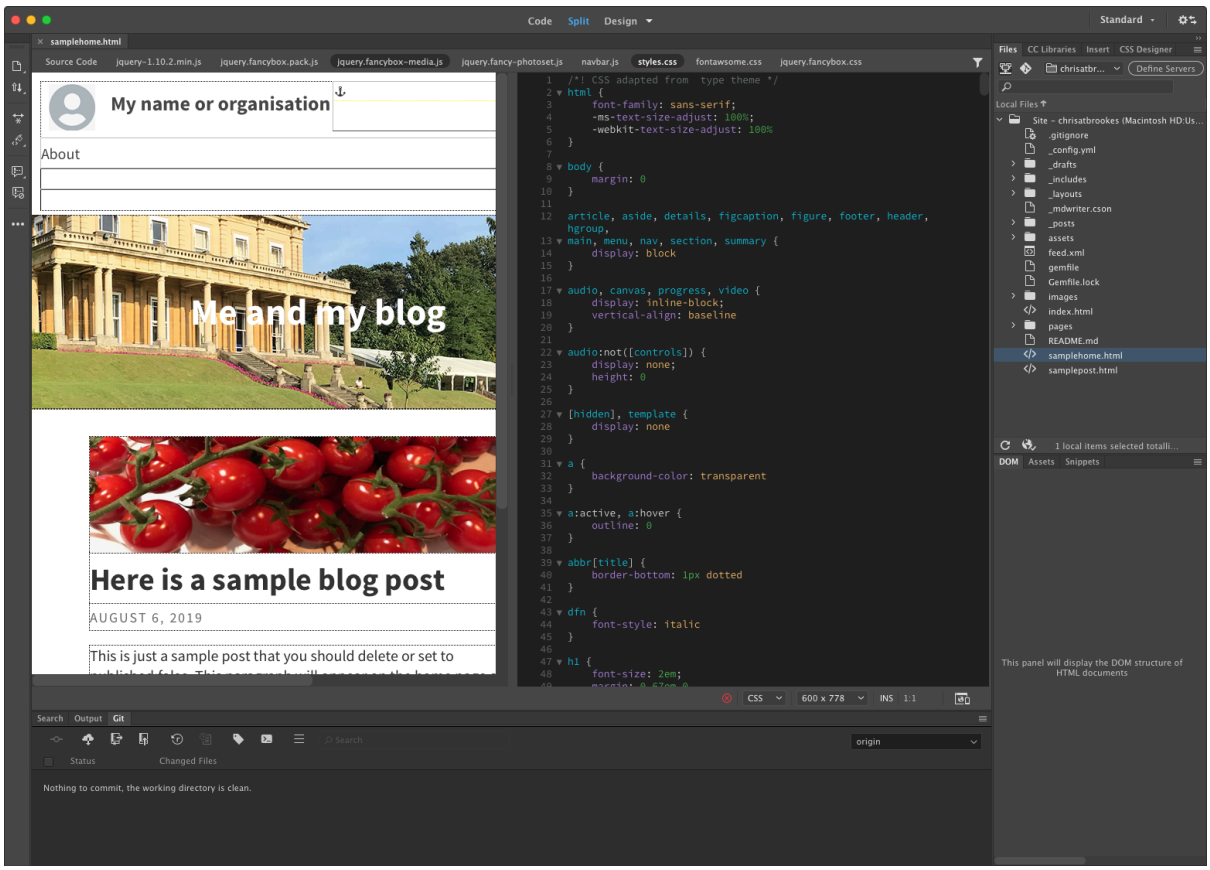
Dreamweaver can interact with the GitHub repository but you do need to edit the CSS locally.

Once you have opened Dreamweaver go to `Site > New site`.on the menu.



Clone an existing Git repository.

Put in your repository location and credentials (see **Step 3** above). The complete site will then arrive at the location you provide and you can edit the site and even post new content.



Chris Jennings 2019