

(Truly) Free and Professional Websites with Github and HTML5UP (Even if you don't yet know HTML!)

Sponsored by SVRN

Written by: Logan Pearce, 2022
svrn.organizers@gmail.com

By the end of this workshop you will have:

- A free website hosted on Github
- An introduction to Github hosting and HTML5UP website templates

I tried and tried and tried to find a TRULY FREE easy to use website for advertising myself and my contact info. Common free sites like Wix I've found will advertise that they are free but as soon as you get into it suddenly everything you really want to do is not free. Google Sites is truly free and it has a WYSIWYG editor, so it's a good choice, but the design is limited and isn't nearly as nice or flexible as HTML5UP. This workshop introduces you to HTML and hosting on GitHub to make nice, functional, flexible websites to advertise yourself well. It's worth it to put in a little bit of effort to learn enough HTML to modify the template and produce something you really like.

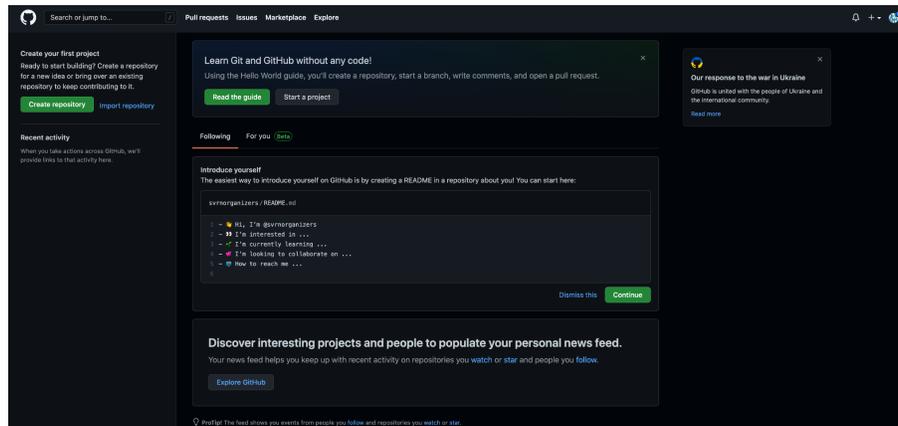
Preparation:

Before the start of the workshop, please accomplish the following:

1. Create an account on Github:

Go to www.github.com, click "sign up". Enter your email, create a username and password (I recommend using your name since this will be for a professional website, for example my

username is “logan-pearce”). Complete the verification procedures, and you should end with a screen that looks like this:



2. Decide broadly what you want your website to contain.

Do you want it to just be a single page with your photo, CV, and contact info? Do you want separate pages for your CV, bio, research, etc that are accessible via a menu? What is the purpose of your site and what do you want to convey about yourself? This can be changed later of course as your site grows and evolves, but it’s a good idea to have a sense of what you want going into the building process.

Check out some of these examples (taken from the astronomy field because that’s my field!) to get a sense for the different ways people structure their sites. Take note of what you like/don’t like, what works and doesn’t work, what might fit your design goals, etc.

<http://www.as.utexas.edu/~cmcasey/welcome.html>

<http://www.as.utexas.edu/~khawkins/>

<https://aibhleog.github.io/>

<https://saturnswings.github.io/>

<https://zjzhang42.github.io/>

<https://www.astro.utoronto.ca/~gilman/>

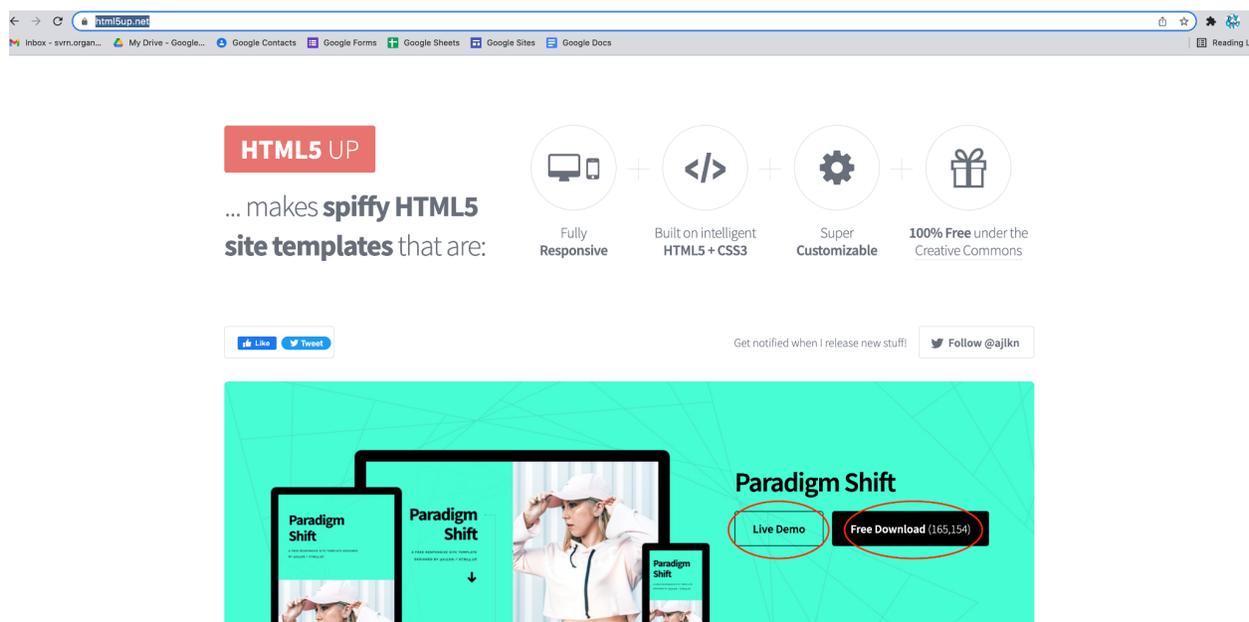
<https://www.stsci.edu/~sveash/#>

<http://www.ifa.hawaii.edu/users/kel/>
<https://lavinia.as.arizona.edu/~rehviding/public.html>
<https://caodonnell.github.io/>
<https://bedell.space/#>
<https://www.astro.ucla.edu/~sibirrer/>
<https://www.physics.mcgill.ca/~sievers/>
<http://www.loganpearcescience.com>

3. Find an HTML5UP website template you like and download the files

Go to <https://html5up.net/> . Check out the templates by clicking on the “Live Demo”. When you find one you like, and you think will work well with the structure you want for your site, click the download.

Unzip the file and put the folder in a convenient location on your computer.



Here are the templates I think would work best for a professional website:

Read Only, Prologue, Twenty, Alpha, Highlights, Spectral,
Solid State, Hyperspace, Forty, Stellar, Dimension, Helios,
Phantom

I recommend staying away from the portfolio, splash page, or blog type templates.

4. Gather materials

Collect the things you want to add to your site such as a headshot, your CV/Resume, any images you want to include, a little “about me” blurb, etc.

5. Brush up on HTML basics.

If you have never seen HTML before, I recommend spending at least an hour or so with an HTML introduction tutorial like https://www.w3schools.com/html/html_intro.asp or https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML , just to get the feel for how it works.

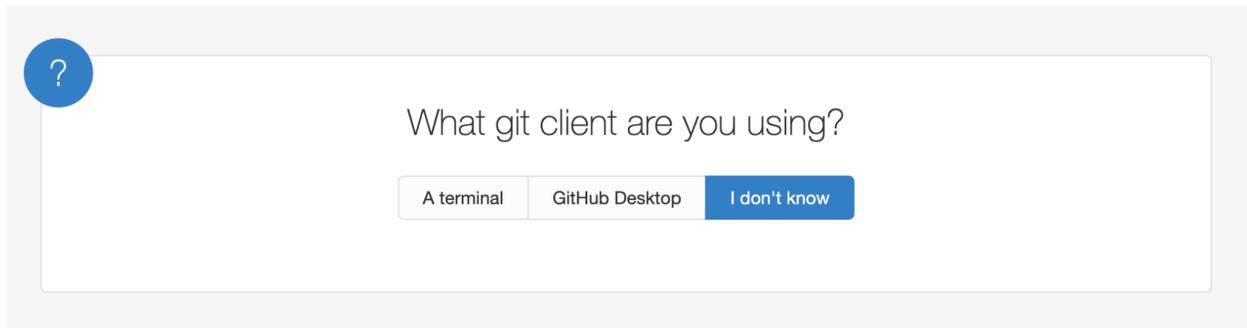
6. You will also need a text editor. I personally use Visual Studio Code. There’s also SublimeText, TextWrangler, BBEdit, Aquamacs, any many more. No microsoft word please.

Section 1: Set up Github Pages

Github is a code and software collaboration and development platform. It also provides users with one free website host. So even if you only even use Github for site hosting and nothing else, it’s worth learning a bit about it cause it’s free! Google Sites is another truly free option (that’s what SVRN is currently using, www.svrn.org), but it is limiting in the structure and look of your site.

To get Github Pages up and running, follow the instructions here:

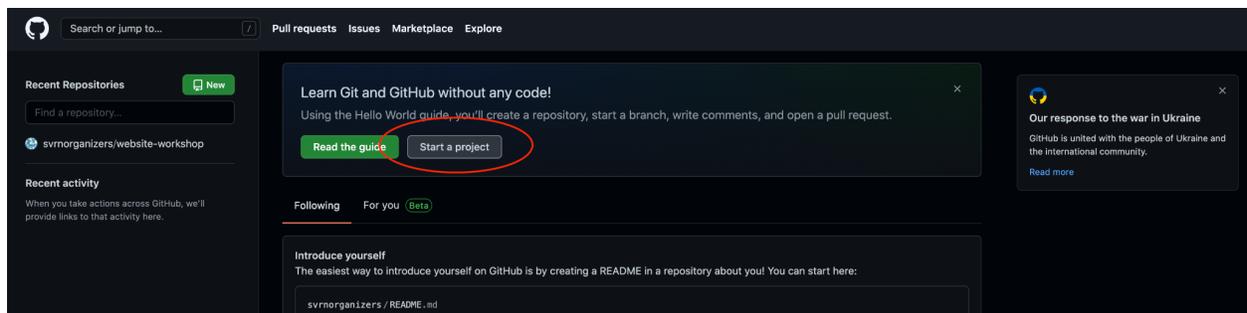
<https://pages.github.com/>



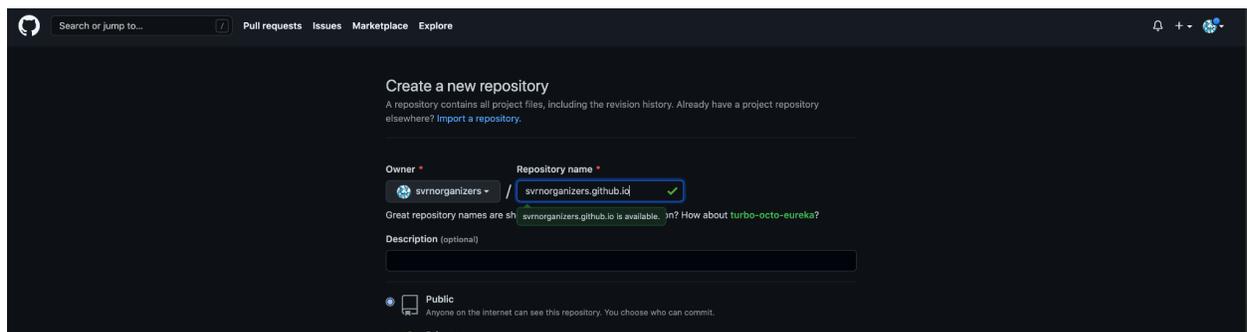
If you are comfortable using the terminal, you can follow the **terminal** instructions. You can also choose to download **Github Desktop** if you wish, which is a good idea. Follow the rest of the instructions.

But you can also just use the browser and the following instructions:

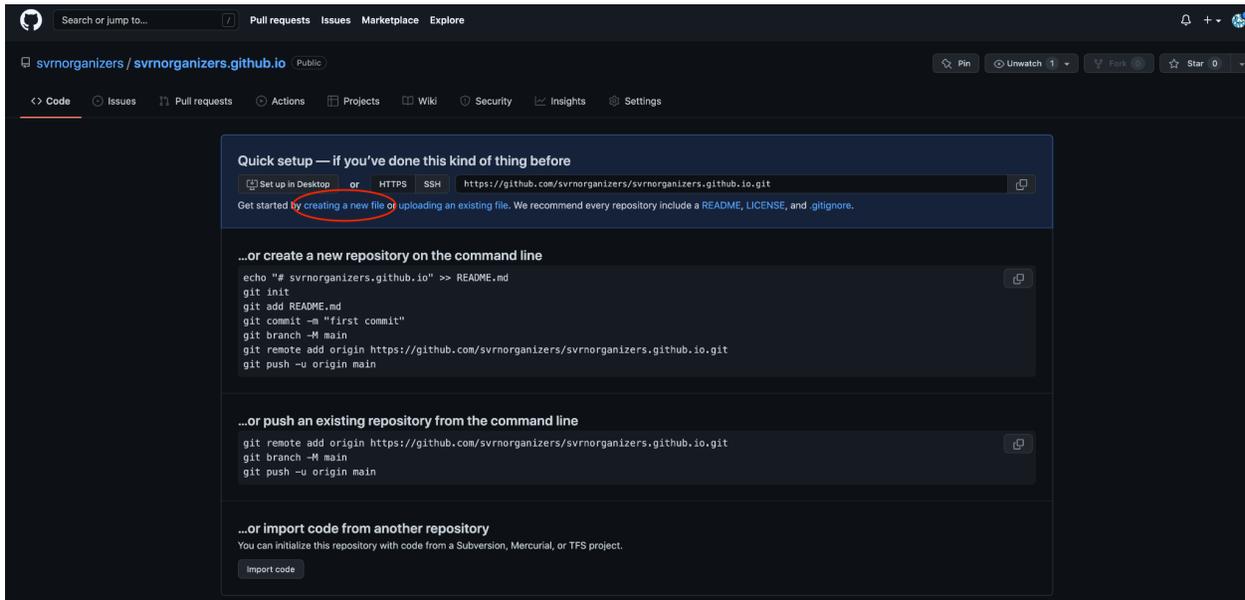
Log into Github and **create a new repository**:



The name of the repository MUST be **[username].github.io**:

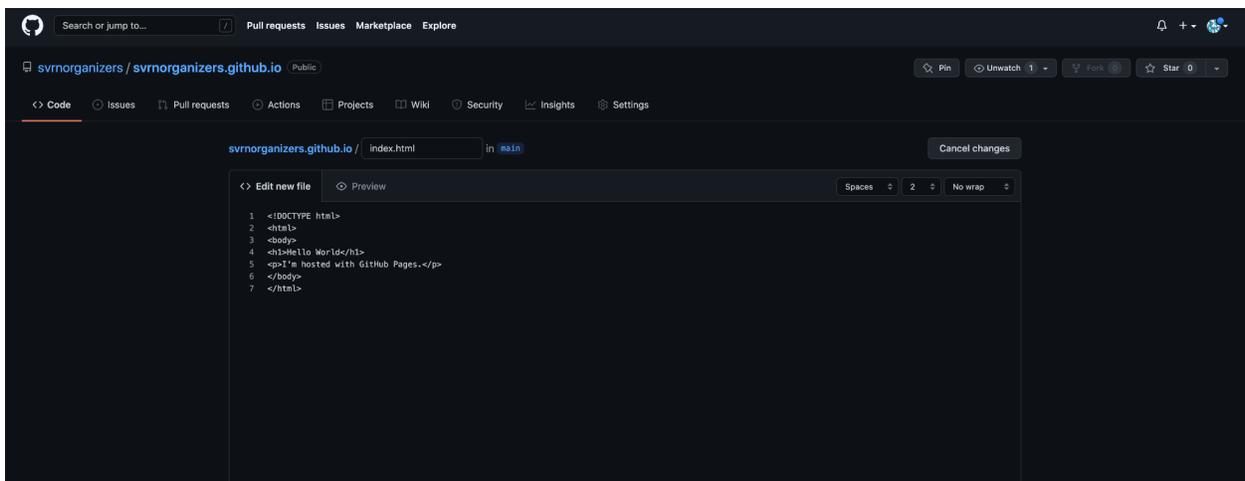


Initialize the repo by creating a new file:



Title the file **index.html**. The browser automatically displays the "index.html" file first as the default, so this document will contain the scripts for the first thing visitors see. You must have an index.html file in order to have a website.

You can copy and paste the code from Section 4 of the Github Pages tutorial:

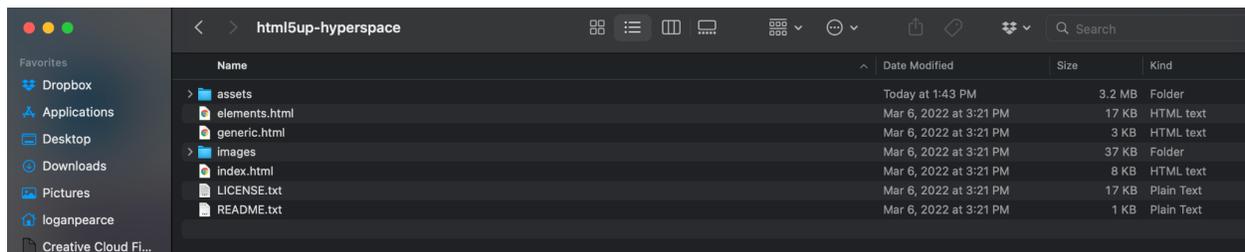


Scroll down, click commit, and you can see your brand new website by going to a browser and entering [username].github.io

Section 2: Dive into the HTML - Don't be scared, you can do this!

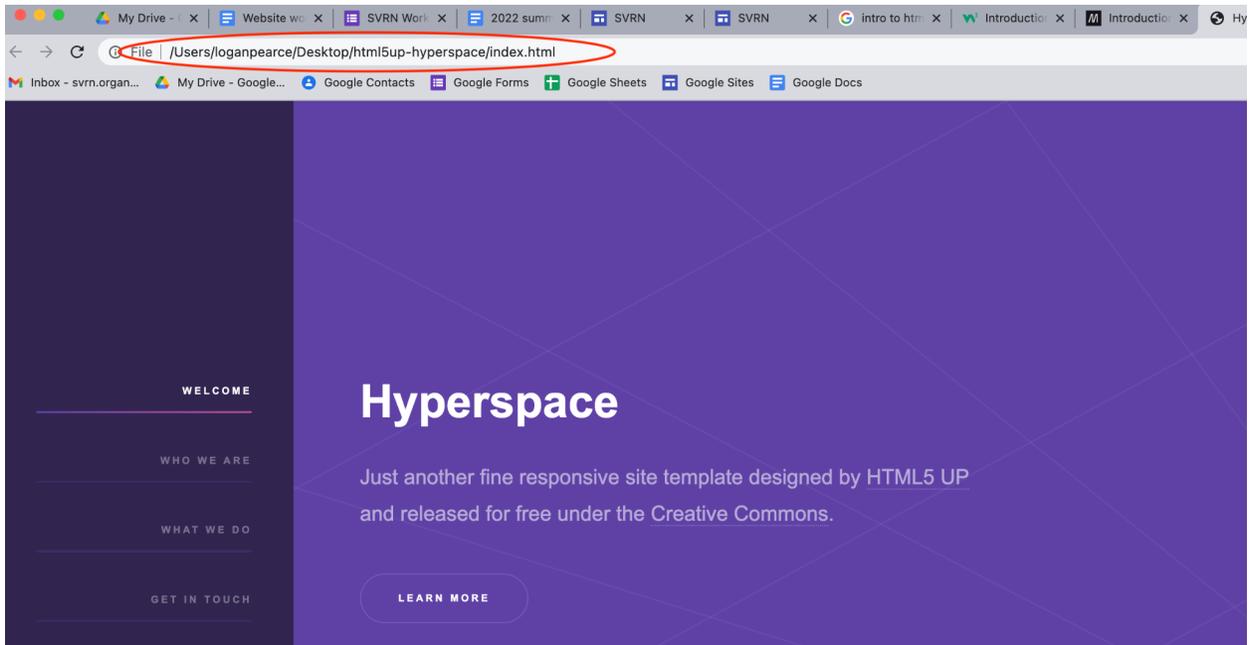
Set GitHub aside for a sec. We're going to do all of this section on our local machines. Find the materials you downloaded from HTML5UP and open the folder. You should see:

- two folders:
 - Assets: this folder holds all the CSS (cascading style sheets, which sets the look of each section of the website. Don't modify this folder or any of its contents unless you are familiar with CSS.
 - Images: this is where we will put any images we want to use for the site.



- Several .html files. Depending on the template you chose, there might be files with different names, but they will all have index.html

Double-click index.html. It should pop up in your browser, and look just like the online template did. But if you look in the address bar, we're looking at a local copy, not the online one. We will be editing the local copy and can see the changes here.



Open index.html in your text editor. Open index.html from inside your text editor to see the code. You'll see several sections delineated by indentations and `<!-- -->`. This is how you comment code in HTML, anything surrounded by these symbols won't be read by the browser.

1. **Title:** this is the very top title of your site that shows up in the browser. All you need to do here is to edit what's inside the `<title>` tag to match whatever title you want. (I noticed many folks in those example sites above neglected to do this!). Don't change anything else because this is telling browser where to look to get all the style information.
2. **Sidebar:** Or it might be called navigation or menu or something else depending on your template.

Let's dissect what the code is doing in this section, and then you can apply it to the further sections.

```
<!-- Sidebar -->
  <section id="sidebar">
    <div class="inner">
      <nav>
        <ul>
          <li><a href="#intro">Welcome</a></li>
          <li><a href="#one">Who we are</a></li>
          <li><a href="#two">What we do</a></li>
          <li><a href="#three">Get in touch</a></li>
        </ul>
      </nav>
    </div>
  </section>
```

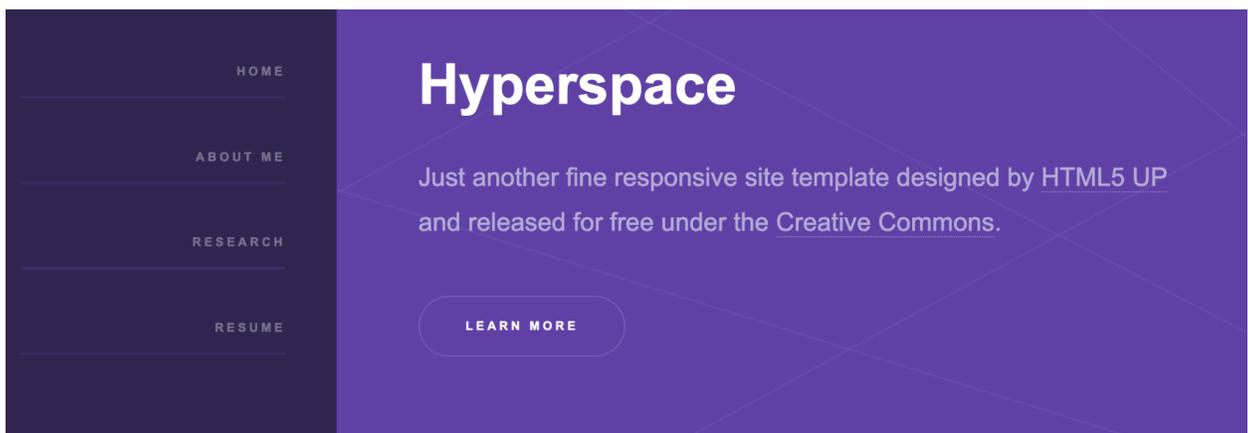
The first line is the comment telling you what section we're working on. Next is the code to open the section `<section id='sidebar'>`. That points to info in the CSS that sets how the sidebar will work and look. It tells the browser to look in the section of the CSS called 'sidebar', and apply the info that's there. Next line (notice they're all indented, keep it that way!) `<div class="inner">` defines a division of a section and again points to the CSS for "inner" divisions, however the template designer designed that. Next is `<nav>`, for navigation, then `` which sets up an unordered list.

Next is the line you actually want to edit. The `` tag defines a list item. Then inside that is `` This tag defines a link. This is your sidebar navigation, so you'll put the links to all your sections of your site. For now, go ahead and make a list item for each section of your site, and within the quotes put sectionname.html (we will actually build this file later). Then put the title of the section after the close bracket, then `` closes the link, and `` closes the list item. So if I were making a website with sections About Me, Research, and Resume, it would look like:

```
<!-- Sidebar -->
<section id="sidebar">
  <div class="inner">
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="aboutme.html">About Me</a></li>
        <li><a href="research.html">Research</a></li>
        <li><a href="resume.pdf">Resume</a></li>
      </ul>
    </nav>
  </div>
</section>
```

I also added a Home link so that users will be able to navigate back to the homepage. Then the following line closes out all the tags we opened above.

Once you've made those changes, save the file, go back to the local file on your browser and refresh it, and you should see those changes in the sidebar.



3. **Intro:** The next section should be the body of the site. In the Hyperspace template it's called "Intro", in my site (the Forty template) it's called "Main". Look for the header text. Here is what it looks like in Hyperspace:

```
<!-- Wrapper -->
<div id="wrapper">

  <!-- Intro -->
  <section id="intro" class="wrapper style1 fullscreen fade-up">
    <div class="inner">
      <h1>Hyperspace</h1>
      <p>Just another fine responsive site template designed by <a href="http://html5up.net">HTML5 UP</a><br />
      and released for free under the <a href="http://html5up.net/license">Creative Commons</a>.</p>
      <ul class="actions">
        <li><a href="#one" class="button scrolly">Learn more</a></li>
      </ul>
    </div>
  </section>
```

<h1> </h1> is the header tag, with 1 being the largest font size. <p> </p> is the paragraph tag, so write whatever text you want to say inside that tag. Notice there are some links embedded in the text surrounded by the tag.

Below that is a section starting with <ul class="actions"> That is a button that the designer has designed, in this case it's a type called "button scrolly". There are probably a few button styles defined in the CSS.

Your turn: Mess around with this section and check what the changes did by refreshing your browser.

Rules for coding! If you're not sure, do these two things before asking me:

- Google the things you're not sure about or questions you have
- Make a change and see what effect it had. Trial and error.

Another thing you can do is go to one of those example sites I gave you, right click in the site, and select "View Page Source". That will show you the html code the browser is reading. So if you're wondering how someone did something, check out their code and learn from what they did.

For example, here's what mine looks like on my website

<http://www.loganpearcescience.com> (ignore the `<center>` tags those are deprecated which is why they're red here):

```
!-- Main -->
<div id="main">
  <section style="background-color: #25293F">

    <div class="inner">
      <!--<span class="image"></span-->
      <center><a href="#" class="image"></a></center>
      <br>

    <div class="inner">
      <center><header class="major"><h1>Logan Pearce</h1></header>

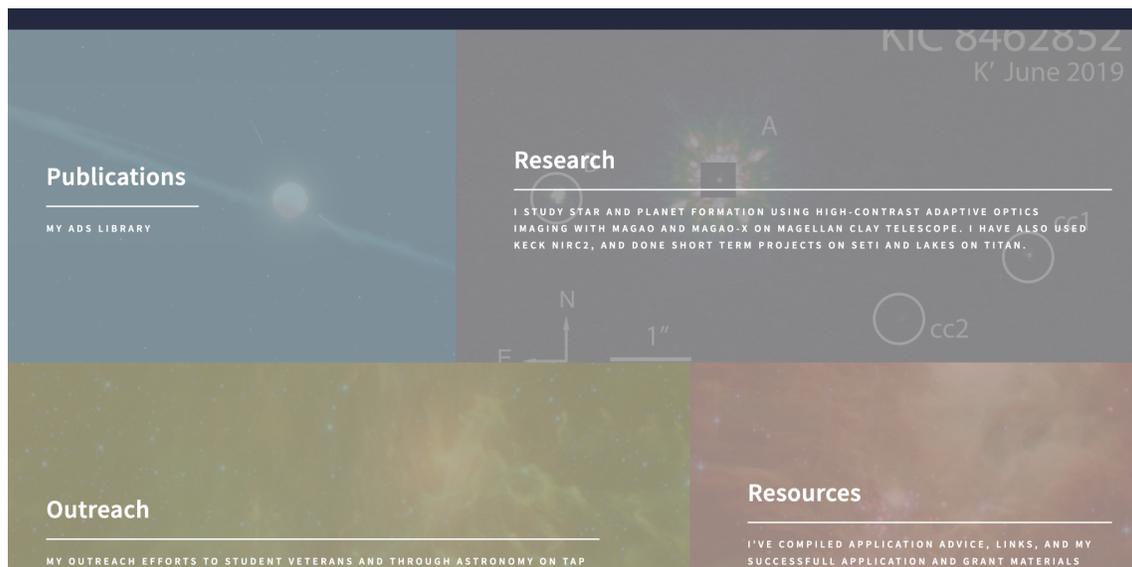
    <div class="content">
      <p><center>Astronomy PhD Candidate | Steward Observatory University of Arizona | NSF Graduate Fellow</p></div>
      <p><center>loganpearce1@email.arizona.edu
      <div itemscope itemtype="https://schema.org/Person"><a itemprop="sameAs" content="https://orcid.org/0000-0003-3904-7378" href="https://orcid.org/0000-0003-3904-7378" target="orcid.widget" rel="me noopener noreferrer" style="vertical-align: top;">https://orcid.org/0000-0003-3904-7378</a></div></p></center><br>
      <ul class="icons">
        <center>
          <li><a href="mailto:loganpearce1@email.arizona.edu" class="icon regular fa-paper-plane"><span class="label">Email</span></a></li>
          <li><a href="https://twitter.com/loganpearce" class="icon brands fa-twitter"><span class="label">Twitter</span></a></li>
          <li><a href="https://github.com/logan-pearce" class="icon brands fa-github"><span class="label">GitHub</span></a></li>
        </center>
      </ul>
      <!--<ul class="actions">
        <li><a href="#" class="button next scrolly">Get Started</a></li>
      </ul-->
    </div>
  </div>
</section>
```

I included a header of my name, brief info, my email address, and several links that appear below my name. I also included an image, which is given by the `` tag. Notice that it points to the images folder and then the image. So if you're ready to add an image, place your image in the folder and use that tag to point to it. Also notice that adding a link to your email begins with "mailto: email address". And remember that every tag you open with `<...>` must be closed at the end with `</...>`

4. Sections

Ok now we're getting into the body of the site. The next section probably begins with `<!-- One -->` and then has several sections below it. The mechanics of this part will change depending on the template you chose, but if you look at the template and the code you should be able to decipher what each section is doing.

For example, here is this section from my website. I used the Forty template, which has different colored boxes that include an image and a link to somewhere. Each box corresponds to each `<article>` tag in the code.



```
<!-- One -->
<section id="one" class="tiles">
  <article>
    <span class="image">
      
    </span>
    <header class="major">
      <h3><a href="https://ui.adsabs.harvard.edu/public-libraries/Bt9HQRu0SPq-6uqMm5lcMA" class="link" ta
      <p>My ADS library</p>
    </header>
  </article>
  <article>
    <span class="image">
      
    </span>
    <header class="major">
      <h3><a href="research.html" class="link">Research</a></h3>
      <p>I study star and planet formation using high-contrast adaptive optics imaging with MagAO and Mag
      Keck NIRC2, and done short term projects on SETI and lakes on Titan.
    </p>
    </header>
  </article>
```

Notice that links that go to other pages of my site just have `` while links that go to different sites include the full url.

Also, depending on your template, you may have different types of sections you can include (usually marked by `<!--Two-->` etc). Use them, delete them, whatever. Make it like you want it to look.

At the bottom of the front page you'll probably find a contact form. If you want visitors to use it go ahead and edit it, but I just removed mine because I don't really want it.

5. Footer

The last thing to edit in `index.html` is the footer. You probably want to put some kind of copyright, definitely should include the attribution that the template was made by HTML5UP. I put more links in the footer for my site:

```
<!-- Footer -->
<footer id="footer">
  <div class="inner">
    <ul class="icons">
      <center>
        <li><a href="mailto:loganpearce1@email.arizona.edu" class="icon regular fa-paper-plane"><span class="label">Email</span></a></li>
        <li><a href="https://twitter.com/loganpearce" class="icon brands fa-twitter"><span class="label">Twitter</span></a></li>
        <li><a href="https://github.com/logan-pearce" class="icon brands fa-github"><span class="label">GitHub</span></a></li>
      </center>
    </ul>
    <ul class="copyright">
      <li>&copy; Logan Pearce 2022. Page Template: <a href="https://html5up.net">HTML5 UP</a></li>
    </ul>
  </div>
</footer>
```

Below that is some scripts that tell the browser where to find the CSS, and also close out the body and html tags. Don't change those.

Once you're done, refresh your browser and see what it looks like!

Section 3: Subpages

You probably want some other pages on your site, like maybe an "about me" page or a description of research projects. Each page will need its own html file, such as

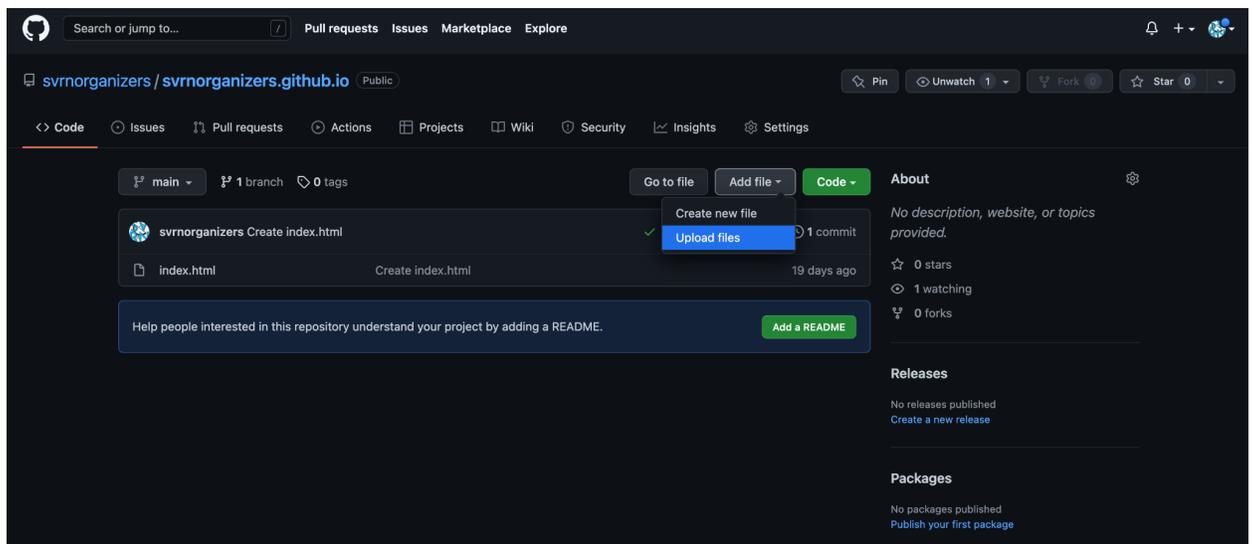
aboutme.html. In your template folder, you'll probably find a few other .html documents, which are examples of the kinds of subpages the template offers. Make copies and rename the file to the names of the subpages you want, such as aboutme.html. Open them in your text editor and make adjustments as you like.

I recommend copying the navigation and footer sections from index.html over to the subpages, so that they will all look and behave the same across all pages. Be sure to update the menu to include the new subpages as well.

Section 4: Publishing to GitHub

Once your site works the way you want locally, you need to push it to GitHub so it's accessible to the internet. If you're managing GitHub using the browser:

- Go to your profile on github.com and navigate to the [username].github.io repository. Click "add file", and upload everything in your website directory.



- Make sure the github repo contains everything that was on your local machine.
- Go to [username].github.io in a new browser to window to see your new site. It make take a few minutes for it to update, be patient.

If you're using the terminal or GitHub desktop, follow the directions at

<https://pages.github.com/>

You're done! Congrats!

Bonus: Custom URL

My website is www.loganpearcescience.com. You can purchase a custom domain name and point it to your GitHub page. You have to purchase a domain name with a domain name manager such as GoDaddy.

Instructions for tying your domain to GitHub pages are here:

<https://docs.github.com/en/pages/configuring-a-custom-domain-for-your-github-pages-site>