

An important part of getting yourself setup for the job market is to have a website to call your own. It's a bit intimidating to set one up yourself, and so I thought I'd write a few words to help you out.

## Do Not Use Google Sites or Dropbox for Storage

It is very tempting to use Google Sites alone for your academic website. DO NOT DO THIS. Many large employers block access to Google Drive and Dropbox at work, and so people will be unable to download your PDFs (like your CV and JMP) if you put them here.

## So what do I do?

You essentially have three options based on your time/money tradeoffs and your experience with the command line:

- No time: Use one of Squarespace, Wix, or Wordpress for ~\$150/yr. I'd recommend Wix.
- Time and facility with git: Use GitHub pages
- Time and no facility with git: Use Google Sites but host your files on GitHub Pages

### No Time All Money: Wix

Steps to working:

- Go to Wix.com
- Choose a template. They have a whole section for portfolios and CVs  
<https://www.wix.com/website/templates/html/portfolio-cv>
- Buy a domain name using Wix when you upgrade to a premium plan
- Voila

You can probably get away with the Combo plan on Wix (\$11/month), but they have a \$14/month plan if you think you're going to get a ton of traffic.

Alternatively, Squarespace and Wordpress are also great, but they don't have the best themes for portfolios / academics.

### All Time No Money (and I can't use git): Google Sites + GitHub Pages

Google Sites is great. You can even add a [custom URL](#) (which you should). However, if you try to upload PDFs, it will give your guests Google Drive links, which is not good.

Instead, we're going to use GitHub Pages to serve your papers and Google Sites to be your webpage.

Sign up for GitHub

Start a new GitHub account. Go to <https://github.com> and sign up.

## Create a new repository

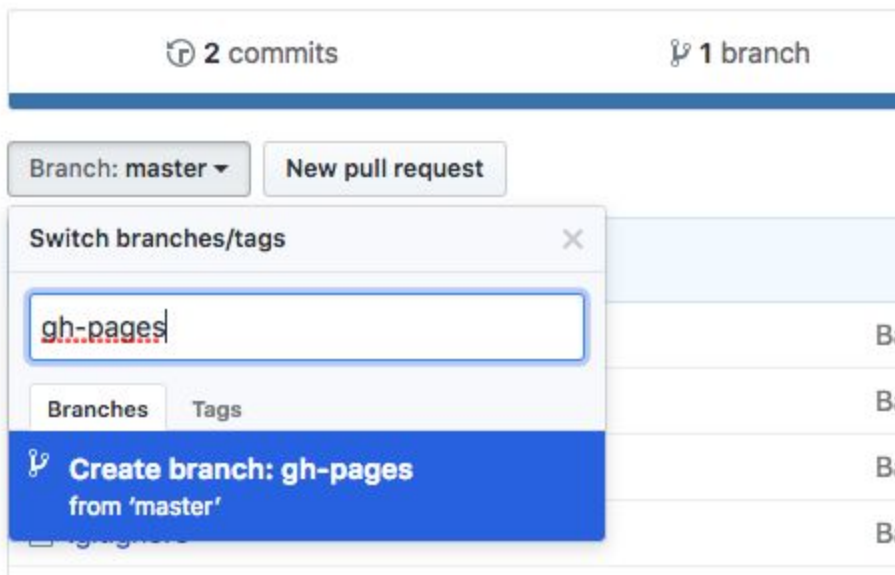
Click the + button in the top right corner and click “New Repository.” Give it a name like MyWebsite or kevinhayeswilson.com. Something memorable.

I recommend clicking the “Initialize repository with a README” button. Note that your code is public (unless you pay for GitHub), but your website will be too so this isn’t so crazy.

## Create a gh-pages branch

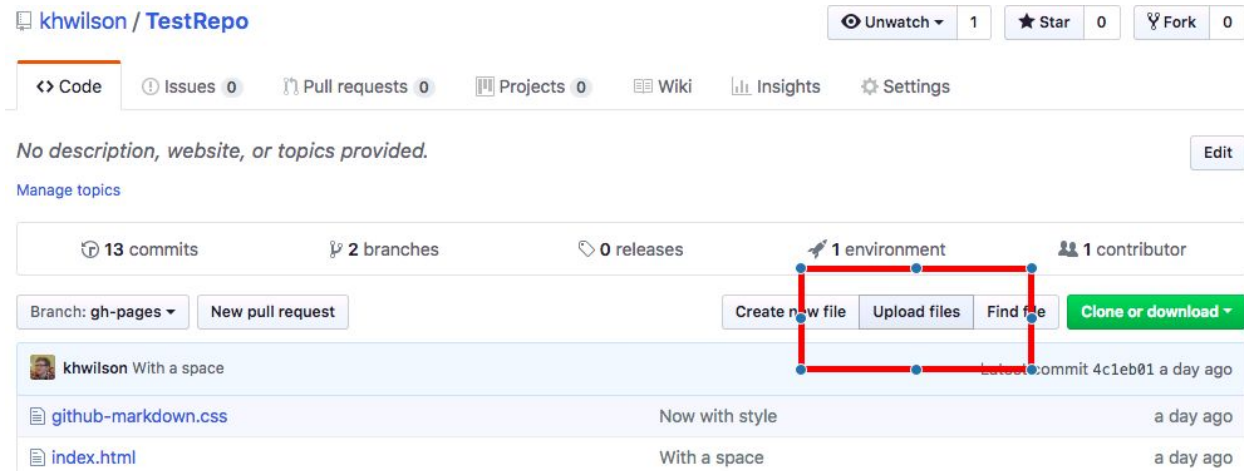
Click the “Branch” button on the page of your new repository. It’s in the upper left of the file listing.

Type “gh-pages” in the box and click “Create branch: gh-pages.” *You must spell this correctly.*



## Upload your papers

Click “Upload Files” and follow the instructions on the next screen. When you’re done adding files, click the big green “Commit changes” button.



Make sure your papers are there

Go to [https://yourusername.github.io/reponame/name\\_of\\_paper.pdf](https://yourusername.github.io/reponame/name_of_paper.pdf). Replace yourusername with your user name and reponame with whatever you called your repo. So, for instance, my username is khwilson and the repo above is TestRepo, so I would go to <https://khwilson.github.io/TestRepo/d4.pdf>. (This is case sensitive!)

Note that sometimes this takes a few minutes to propagate.

Setup your Google Sites as usual

Follow the usual instructions to setup a Google Site. But when you want to reference a paper, don't upload it to Google Drive. Instead, create a link to the github.io link above.

Add more papers!

When you go back to your GitHub repository, you'll probably be on the "master" branch. Make sure the branch button says "gh-pages." If it does not, then click it and choose gh-pages first.

## All Time No Money (and I can use git): GitHub Pages

GitHub will do web hosting for you! It even let's you host at a personal domain name. Here's how you get it setup!

Sign up for GitHub

Start a new GitHub account. Go to <https://github.com> and sign up.

Create a new repository

Click the + button in the top right corner and click "New Repository." Give it a name like MyWebsite or kevinhayeswilson.com. Something memorable.

I recommend clicking the “Initialize repository with a README” button. Note that your code is public (unless you pay for GitHub), but your website will be too so this isn’t so crazy.

### Moving to the command line

In your repository there will be a big green “Clone or Download” button. Click that, copy the URL, and in the terminal, type ``git clone [that url]``.

Once you have your repository, then you can do ``git checkout -b gh-pages``. Anything you put on the gh-pages branch will be published as a webpage.

### Upload papers

Create a folder called ``papers`` in the repository, and put all your pdfs in it. Then run ``git add papers`` and ``git push origin gh-pages``.

### Upload an image of yourself

Take a nice professional shot and add it to the repository.

### Design your webpage!

I can’t help you here, but here are the steps to get a basic one:

- Go to <https://khwilson.github.io/TestRepo/>
- Go to File > Save and save this file with the name ``index.html`` in your repository. Make sure you’re saving the “HTML Only.”
- Go to <https://khwilson.github.io/TestRepo/github-markdown.css>
- Go to File > Save and save this file with the name ``github-markdown.css`` in your repository.
- Edit the file ``index.html`` removing references to me and putting in references to yourself.

Note that this file is HTML, and so it’s pretty structured. Follow the patterns that are present in the file already and you’ll be fine.

Of course, you can also get fancier if you desire.

### Push to your repo

`git add . && git commit -m “Initial commit” && git push origin gh-pages`

### You should be up and running

Your website should be at <https://yourusername.github.io/reponame>.

### Custom Domain? Part 1

Get a custom domain. I use [name.com](https://name.com). You’ll need to create a CNAME record in your DNS entries that points `www` to `yourusername.github.io` and an ANAME record that points your apex

domain to yourusername.github.io. For instance, here is what my name.com dashboard looks like for kevinhayeswilson.com.

The screenshot shows the Name.com DNS management interface for the domain **kevinhayeswilson.com**. On the left is a sidebar with navigation options: Details, Contacts, Advanced Security, Nameservers, **DNS Records** (highlighted), URL Forwarding, Branded URL Shortener, Email Forwarding, NS Registration, Account Transfer, Web Hosting, Name.com Email, and Website Builder. The main content area is titled "Edit DNS Records: kevinhayeswilson.com" and includes buttons for "Show DNS Templates" and "Delete Selected Records", along with a link to "Export DNS Records (CSV)".

Type	Host	Answer	TTL	Prio	Actions
<input type="checkbox"/> ANAME	kevinhayeswilson.com	khwilson.github.io	300	N/A	<a href="#">Edit</a> <a href="#">Delete</a> <small>Created: 2018-10-30</small>
<input type="checkbox"/> CNAME	www.kevinhayeswilson.com	khwilson.github.io	300	N/A	<a href="#">Edit</a> <a href="#">Delete</a> <small>Created: 2018-10-30</small>

Below the table is a form to add a new record with fields for Type (A), Host (.kevinhayeswilson.com), Answer, TTL (300), and Prio (N/A), and an "Add Record" button.

**A Records**

An A record points a hostname to an IP address. To point just kevinhayeswilson.com to an IP address, leave the "Record Host" field blank. To point www.kevinhayeswilson.com to an IP address enter www into the "Record Host" field. To point all sub domains for kevinhayeswilson.com (that do not already have a host record) to an IP address enter \* into the "Record Host" field. You can add multiple A records for a host name.

## Custom Domain? Part 2

Back in your git repository on the gh-pages branch, add a file called `CNAME` whose only contents are the name of your website. For instance, see this file:

<https://github.com/khwilson/kevinhayeswilson.com/blob/gh-pages/CNAME>

Add more stuff!

You can customize your website to the nines. There are innumerable tutorials on how to get jekyll pages going. If you really want to be fancy, feel free to copy my website, including its automatic CircleCI build. <https://github.com/khwilson/kevinhayeswilson.com>.