



4

## Create a Free Personal Academic Website with Hugo

9 October 2016 · 11 min read · 4 Responses ([https://georgecushen.com/create-your-website-with-hugo/#disqus\\_thread](https://georgecushen.com/create-your-website-with-hugo/#disqus_thread))

I recently moved my personal/academic website from an expensive and complicated shared hosting plan to the simple GitHub Pages (<https://pages.github.com/>) platform for free. At the same time, I also changed the engine from Jekyll to Hugo (<https://gohugo.io/>).

In this guide, I will discuss why you should also use Hugo and Github for your personal or academic website and cover how to install and use Hugo. This will enable you to configure a static site, create content (such as blog posts or academic publications), and publish your site for free with Github.

Like Jekyll, Hugo also requires a basic understanding of using the command line in the Terminal (Mac/Linux) or Command Prompt (Windows) app on your computer. If you are not interested in this, then I can highly recommend the following simpler alternatives:

- GitHub Automatic Page Generator (<https://pages.github.com/>) - free and ideal for a very simple one page site. To get started:
  1. Signup for Github (<https://github.com/>), preferably using a username based on your real name since your website URL will be in the form `https://YOUR-USERNAME.github.io` (<https://YOUR-USERNAME.github.io>)
  2. Login, click the "+" in the top right corner and create a new repository with the name `YOUR-USERNAME.github.io`
  3. From the main page of your new repository, click *Settings* and then *Automatic Page Generator*
  4. Follow the wizard to create your new website - help is available here (<https://help.github.com/articles/creating-pages-with-the-automatic-generator/>)
- SquareSpace (<https://www.squarespace.com/>) - ideal if you need more features and customization. It is non-free (currently 12USD/month).
- WordPress (<https://wordpress.com/>) - ideal for a highly customizable multi-page site. Although it has a free plan, I recommend paying the small fee (currently 2.99USD/month) for the *Personal* plan so that WordPress adverts are not shown on your site.

Whichever tools you decide to use to build your website, you can further personalize your website with your own domain name. I can highly recommend Namecheap (<https://www.namecheap.com/?aff=105828>) for registering your domain, as it provides great value for money whilst providing fast support. To find a good and available domain, try a mix of your first and last names or initials, with either a `.com` or `.me` ending. Later in this article, we'll look at how to integrate a new domain name with Github.

### Why Hugo and Github?

While there are an enormous amount of website hosting and creation services available, a Hugo based website hosted on Github can have significant advantages over other options for a student or academic:

- **It's free.** Students will definitely appreciate this! Public repositories are free on Github. Hugo is free and open-source.
- **Simple editing.** Instead of using HTML to write your content, you can write using popular Markdown and LaTeX math expressions (<https://sourcethemes.com/academic/post/writing-markdown-latex/>).
- **Academic theme/framework** (<https://sourcethemes.com/academic/>). Manage your biography, publications, talks, news, teaching resources, projects, etc.
- **No dependencies.** The beauty of Hugo is that it is just one standard binary program with no dependencies, so it's dead simple to maintain a site made with Hugo. Jekyll and WordPress on the other hand have complex Ruby and PHP dependencies, respectively.
- **It's multiplatform.** Build your site from a Mac, Linux, or Windows computer.
- **Lightning fast to generate.** Hugo builds my site, piecing together all the various parts and converting the Markdown content to HTML in less than a second. Jekyll took significantly longer and occasionally stopped working due to dependencies becoming deprecated or packages requiring to be manually updated.
- **Lightning fast to access.** The inherent advantage of using a static site generator is that each page and asset can immediately be served to the client without any dynamic processing. Coupled with Github's content distribution network, this means that wherever the client is in the world, they can instantly see your website.
- **Secure TLS (HTTPS).** TLS helps guarantee that the content of your website has not been interfered with. Google recently began ranking secure HTTPS sites above insecure HTTP sites, and HTTP is being deprecated by major web browsers such as Chrome and Firefox. TLS is supported by Github and it's the way forward.
- **Excellent support.** Hugo has excellent community support forums (<https://discuss.gohugo.io/>) and a team of very active and friendly developers. Whereas Github support is omnipresent on Google and elsewhere like StackExchange (<http://stackoverflow.com/questions/tagged/github>).
- **It's independent.** Github is independent from your job, so it is portable. Whereas if your university affiliation changes, you'll lose your university account and have to re-create a university hosted website with a new URL.
- **Solid roadmap.** I have heard from a number of colleagues about universities dropping their own website hosting services and replacing them with other services or just not maintaining them adequately. Hugo and Github will not be disappearing any time soon.
- **Professional URLs.** Github Pages URLs look professional (i.e. *your-name.github.io*) and there is also the possibility to use your own domain name with Github Pages if you wish.

## Hugo vs Jekyll

In addition to the above, here are some specific reasons to use Hugo for your website if you are choosing between Hugo and Jekyll or considering migrating from Jekyll:

- Hugo can generate a site with 1000 pages in only 0.1 seconds, whereas Jekyll takes several minutes for the same site
- Hugo is a standard binary program, so it has no Ruby dependencies to maintain when testing/building the site locally on your computer
- Paginations, redirection, and live reload work right out of the box in Hugo without having to setup plugins

- In Hugo, you can just add `draft = true` in the front matter of the post to save an unpublished draft, rather than having to move it to another folder
- Plus Hugo has numerous other benefits such as the content is organised in perhaps a more intuitive structure and has multilingual features built-in

Note that this is not an exhaustive list and the benefit-cost(time) ratio will depend on the size of any existing site and your skills and preferences.

## Prerequisites

I will assume that you have basic experience using the command line, as I will be demonstrating commands for the Linux/Mac Terminal.

Windows users may possibly need to adapt some commands for Windows command prompt. Note that Windows command prompt can be opened from **Start** menu > **All apps** > **Windows System** > **Command Prompt**.

## Installing Hugo

For **Windows** users,

1. Download the latest zipped Hugo executable (e.g. `hugo_<version>_Windows-32bit.zip`) from the Hugo Releases (<https://github.com/spf13/hugo/releases>) page.
2. Create a *Hugo* and *bin* folder: `C:\Hugo\bin\`
3. Move the downloaded ZIP file into your `C:\Hugo\bin\` folder.
4. Extract the ZIP file such that `hugo.exe` is extracted in `C:\Hugo\bin\`
5. Add Hugo to your Windows PATH:
  1. Click **Start** > **System** > **Advanced System Settings** > **Environment Variables**
  2. Double-click on **PATH**
  3. Click **New**, type `C:\Hugo\bin`, and press **Enter**
  4. Click **OK** to save and close the windows

For **Mac** users, assuming you have Homebrew (<http://brew.sh/>) installed, run the following in your Terminal app:

```
brew update && brew install hugo
```

For **Ubuntu Linux** users,

1. In your *Terminal* app, type `arch`. If the output is `x86_64` then you'll want to download the 64-bit file ending in `-64bit.deb` in the next step, otherwise download `-32bit.deb`.
2. Download the latest Hugo package for your system (as noted in the previous step) from the Hugo Releases (<https://github.com/spf13/hugo/releases>) page and install by double clicking it, or via command line. For example:

```
cd ~
wget https://github.com/spf13/hugo/releases/download/v0.17/hugo_0.17-64bit.deb ★1
(https://github.com/spf13/hugo/releases/download/v0.17/hugo_0.17-64bit.deb)
sudo dpkg -i hugo*.deb
```

Note that further installation guides are available in the Hugo documentation (<https://gohugo.io/overview/installing/>).

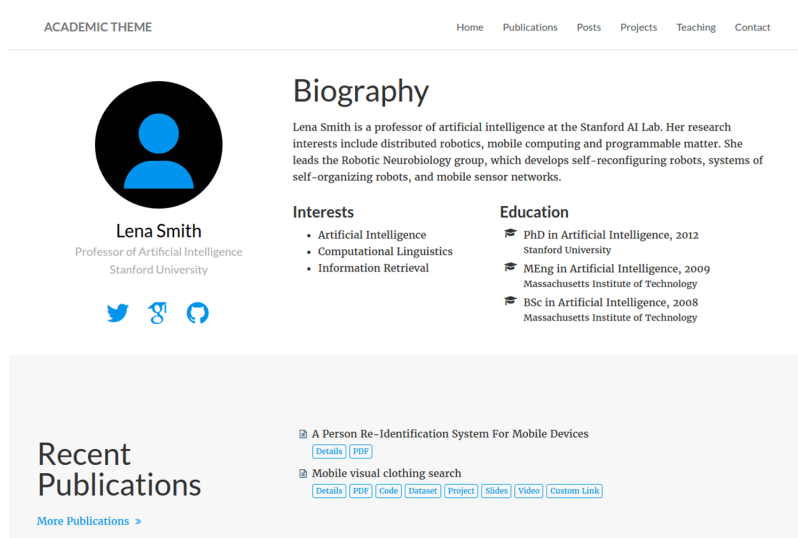
## Getting started with Hugo

When I first started using Hugo, there were no website themes or frameworks available for academic staff or students. Hence, I went about creating one, namely Hugo Academic (<https://sourcethemes.com/academic/>) - as seen in the image below. The key features are:

- Designed for academic staff, students, or general personal use
- Includes Biography, Publications, Projects, News/Blog, Teaching, and Contact widgets

- Write in Markdown (<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>) for easy formatting and code highlighting, with LaTeX (<https://en.wikibooks.org/wiki/LaTeX/Mathematics>) for mathematical expressions
- Academic linking (Scholar etc.), Google Analytics (<https://analytics.google.com/>), and Disqus (<https://disqus.com/>) comments
- Responsive and mobile friendly
- Simple and refreshing one page design
- Easy to customize

The theme/framework rapidly gained the attention of academics and has now become very popular with thousands of users worldwide. Since the theme is highly customizable, letting you pick and choose which features you want, it's possible to generate many different types of website, even including general personal or freelance websites.



**FIGURE 1:** Creating a personal academic website with Hugo Academic.

To create your Hugo website with the Academic theme/framework:

1. Open your Terminal/Command Prompt app, navigate to your desired location using the `cd` command, and create a new website:

```
cd ~
hugo new site my_website
cd my_website
```

2. Install the Academic theme with git (<https://help.github.com/articles/set-up-git/>):

```
git clone https://github.com/gcushen/hugo-academic.git (https://github.com/gcushen/hugo-academic.git) themes/academic
```

Or alternatively, download Academic (<https://github.com/gcushen/hugo-academic/archive/master.zip>) and extract it into a `themes/academic` folder within your Hugo website.

3. Copy the contents of the `exampleSite` folder to your website root folder, overwriting existing files if necessary. The `exampleSite` folder contains an example config file and content to help you get started. The following command can be used to copy the folder via the command line:

```
cp -av themes/academic/exampleSite/* .
```

4. Start the Hugo server from your website root folder:

```
hugo server --watch
```

Now you can go to `localhost:1313` (<http://localhost:1313>) and your new Academic themed website should appear. Note that you can press the `Ctrl + C` buttons in the terminal when you wish to stop viewing your local website using the Hugo server and be able to type other commands again.

5. Customize your website (<https://sourcethemes.com/academic/post/getting-started/>)
6. Add your own content (<https://sourcethemes.com/academic/post/managing-content/>) (homepage widgets, news/blog posts, academic publications, projects etc.)
7. Build it by running the `hugo` command. This will generate a folder called `public/` that contains all the HTML code and assets for your website.
8. Deploy it using Github, as discussed in the next section. Or alternatively, by copying the `public/` directory (by FTP, Rsync, etc.) to your production web server (such as your university's hosting service).

## Getting started with Github

Go to Github (<http://www.github.com/>) and register for an account if you have not done so already. Github encourage using your real name for your account username, and this can help your Github URL (which you will be assigned later) to look professional.

Once you have created your Github account and visited your profile, we will create two new repositories (often abbreviated as *repos*) on Github with the following names:

- `website` or any other name you like - we will save your Hugo content to this repo
- `USERNAME.github.io` where `USERNAME` is your Github username - we will save the generated HTML website to this repo

To create each repository, click the "+" icon in the top right corner and then choose "New Repository". For the `USERNAME.github.io` repository, ensure to check the *Initialize this repository with a README* box, as this will make the next step easier. Enter a brief description if you like and then "Create repository" for each one.

In your Hugo `config.toml` file, set `baseUrl = "https://USERNAME.github.io/"` (`https://USERNAME.github.io/`), where `USERNAME` is your Github username. Stop Hugo if it's running and delete the `public` directory (if it exists) with:

```
rm -r public/
```

Install git (<https://help.github.com/articles/set-up-git/>) if it's not already present on your system. You can check by running `git --version`.

Whilst your terminal is still in the root folder of your website, initialize a git repository and add the remote Github repository that you created:

```
git init
git remote add origin https://github.com/YOUR-USERNAME/website.git (http://github.com/YOUR-USERNAME/website.git)
```

Add your `USERNAME.github.io` repository into a submodule in a folder named `public`:

```
git submodule add https://github.com/YOUR-USERNAME/YOUR-USERNAME.github.io.git (http://github.com/YOUR-USERNAME/YOUR-USERNAME.github.io.git) public
```

Add everything to your local git repository and push it up to your remote repository on GitHub:

```
git add .
git commit -m "Initial commit"
git push -u origin master
```

Whilst running the above commands you may be prompted for your Github username and password. Next, regenerate your website's HTML code by running Hugo and upload the *public* submodule to GitHub:

```
hugo
cd public
git add .
git commit -m "Build website"
git push -u origin master
```

Once Git has finished uploading your site to Github, you can open your new <https://USERNAME.github.io> (<https://USERNAME.github.io>) website in your browser, substituting *USERNAME* with your Github username.

## A practical example

As discussed, Hugo Academic includes an example website for you to use as a template for your website. You can view the code for this online (<https://github.com/gcushen/hugo-academic/tree/master/exampleSite>). My personal website is based on this example site.

## Custom domains

You can use your own domain name with Github Pages if you wish. You will need to register a domain, point it to Github, and create a `CNAME` file in the `static` folder of your website, so that Github knows your intentions. For more information, check out the domains guide by Github (<https://help.github.com/articles/setting-up-a-custom-domain-with-github-pages/>).

Remember that after you have setup a custom domain, you will need to wait approximately 24-48 hours for the DNS to propagate and then you'll need to update `baseurl` in your Hugo `config.toml` to your new URL, regenerate the site with the new urls, and redeploy.

## Automating deployment

If you are feeling more adventurous, you can consider automating deployment such that when a change is pushed to your Hugo `website` repository, your `USERNAME.github.io` repository is automatically re-built. ★<sup>1</sup>

You can *fork* the code (<https://github.com/gcushen/hugo-academic-demo>) for the live demonstration website (<https://sourcethemes.com/academic/>), which is based on the example code above. However, note that this demo repository is setup with project-based Github Pages as a branch named `gh-pages`, rather than with user-based Github Pages as a separate repository to the Hugo content. You'll want to use user-based Github Pages which requires creating two separate repositories, as previously described. Also, the Academic theme is setup as a git submodule so that theme updates can easily be applied. Wercker (<http://wercker.com/>) is utilized for continuous deployment. Hence, you'll need to connect Wercker with your Github account and add a Wercker webhook to your `website` repository on Github. If there is interest, I will consider expanding on this setup in a future post.

## Conclusion

Hugo Academic is a great way to get a personal or academic website up and running quickly and easily. Static websites, such as Hugo Academic, load significantly faster than traditional content management system (CMS) sites whilst being much easier to manage and much harder to hack. Individuals do not usually need dynamic database-driven content or the advanced features that content management systems offer. With Hugo Academic, you can focus your energy on creating great content instead of administering a complex system.

Happy creating! Feel free to leave your comments or questions below!

[Hugo \(/tags/hugo/\)](/tags/hugo/)[Tutorial \(/tags/tutorial/\)](/tags/tutorial/)**WRITTEN BY GEORGE CUSHEN**[\(https://twitter.com/GeorgeCushen\)](https://twitter.com/GeorgeCushen)

I'm a data scientist and vision engineer passionate about machine learning, augmented reality, and internet of things. Follow me on twitter to be notified of new content.

---

**Related**

- [Wake Up with Spotify Alarm Clock for Raspberry Pi \(/spotify-alarm-clock-raspberry-pi/\)](/spotify-alarm-clock-raspberry-pi/)
- [Wake Up with Spotify Alarm Clock for Ubuntu Linux \(/spotify-alarm-clock-ubuntu-linux/\)](/spotify-alarm-clock-ubuntu-linux/)
- [Wake Up with Spotify Alarm Clock for Raspberry Pi and Ubuntu Linux \(/spotify-alarm-clock-raspberry-pi-ubuntu-linux/\)](/spotify-alarm-clock-raspberry-pi-ubuntu-linux/)

4 Comments

George Cushen

Login

Recommend

Share

Sort by Best



Join the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS ?

Name

**disqus\_MHNZErTnKb** • 15 days ago

great material george!  
where do I go to get rid of the social media buttons?

^ | v • Reply • Share ›

**Bapi Loya** • 2 months ago

Great article! It was helpful to build my own page. In the website repo, we added a submodule([username.github.io](https://github.com/username.github.io)) under public. Do we need to make it private by adding it to .gitignore?

^ | v • Reply • Share ›

**Sander W. van der Laan** • 3 months ago

Really great tutorial and ditto theme. Just one thing: what is the best way to update the website as you make changes to it?

^ | v • Reply • Share ›

**Yida Wang** • 3 months ago

Nice Template! Thank you George, I'm building my own website according to your theme. It might be much better if hugo can parse Latex codes for some CV and paper sharing

^ | v • Reply • Share ›

ALSO ON GEORGE CUSHEN

### Create a Free Personal Academic Website with Hugo

13 comments • a year ago



**Matthew Barbour** — I think I'm having the same problem you had. How did you get it to work? It seems that once I publish the website on ...

### Wake Up with Spotify Alarm Clock for Ubuntu Linux

2 comments • 2 years ago



**misikir** — Hi, I am about to finalize my personal website using the academic template. Thanks for taking your time to program this simple yet ...

### Wake Up with Spotify Alarm Clock for Raspberry Pi and Ubuntu Linux

5 comments • 2 years ago



**Krisztián Sereg** — I also found the mopydy workaround, but so far your tutorial is the only one which actually writing about the alarm ...

### Wake Up with Spotify Alarm Clock for Raspberry Pi

7 comments • 2 years ago



**Nicolas** — Excuse me, but where's the second part please? :D

Subscribe Add Disqus to your siteAdd DisqusAdd Privacy

DISQUS

