

Notes on Hugo

2019-11-10T11:39:23-05:00

As a Ph.D. student, we all need to build or website to show our research and to introduce ourselves to the world. This post is my note on how to use Hugo, a static website generator, to build your own website and host it on GitHub.

GitHub part

First of all, you need to sign up an account on GitHub. Details is on official website.

After signed up, you can press the icon on the top-right cornor and enter your profile.

Then create two new repository: **website** (or any name that store your raw content of your website) and **<username>.github.io** (the actuall repository to host your website). If you use other name for the repository to host your website, the website url would be **https://github.com/<username>/<repository_name>/**.

Use **git clone** to copy both repository to your local directories. I suggest it to be in two distinct directories. The url for repository can be found at the green buttom **Clone or download**. The code would be:

```
git clone <resitory_url>
```

Local file part

Now you need to install **hugo** and **git** to generate your static website and upload your website to github, respectively.

You can refer to the officien webste of hugo and git to know how to install both of them on your OS. If you use Arch, then you can install them by

```
sudo pacman -S hugo git
```

Enter or **cd** to your **website** directory, and generate your new website by

```
hugo new site [new-site-name]
```

Figure 1: GitHub profile

There will be a new directory in your **website** directory. Move all the content in the new directory to your **website**, and delete the new directory. The structure of the **website** directory will be

```
website
├── archetypes
│   └── default.md
├── config.toml
├── content
├── data
├── layouts
├── static
└── themes
```

6 directories, 2 files

Congratulation! You already built the skeleton of website!

Explanation on directory tree

The `default.md` in `archetypes` directory is the default format that generated by hugo.

`config.toml` is the configuration for the whole website. Should cooperate with `themes`

`themes` contains all the themes you downloaded. You can download themes from github using `git clone`.

You put all your posts in `content`.

Hugo command

`hugo new <directory>/<posts>` will create a new markdown file `posts` inside `./content/<directory>/`. For example, `hugo new post/HugoNote.md` will create a new posts called `HugoNote.md` inside `./content/post/`. If you want to post in `content` directory, you can use `hugo new content/<posts>`.

`hugo server` generates your website at `http://localhost:1313/`.

`hugo -d <directory>` outputs all the html and css files to the directory.

Host website on GitHub

Now we are going to transform your markdown content into html and css and host on github.

1. Use `hugo -d ../<username>.github.io/` to output your website.
2. Move to `<username>.github.io/` or `cd ../<username>.github.io/`.

3. In the `<username>.github.io/`, type `git status` to check the difference between local file and remote github file.
4. `git add --all` or `git add .` to add all difference into commit.
5. `git commit -m "<message>"` to commit all the difference with comment message.
6. `git push origin master` to push all the difference to remote github file.
7. Open browser and go to your `<username>.github.io` repository, go to **setting** and in the “GitHub Pages” section, choose “source” as **master branch**. Wait for a couple of minutes, and your website is established!

Include image in your blog

There are two ways to insert images into your posts:

1. The hugo-primer theme’s default image path is `./static/images/*`, i.e., every file in the `./static/images/` directory, so
 - Make a new directory called `images` in the static directory, i.e., `mkdir ./static/images`
 - Put all your images in the `images` directory.
 - Or, you can make a new directory based on your posts. For example, for this posts, I put all the images in the `./static/images/HugoNote/`.
2. Put image in your blog directory or any directory inside blog directory
 - Use `![text to display] (/blog/imagename)` when you put images in your blog directory.
 - Use `![text to display] (/blog/nameofdirectory/imagename)` when you put images in the new directory inside your blog directory.

Include pdf in your blog

Include pdf link in your blog

- Put your pdf files in either static directory, or just build a new directory that contains your markdown file and pdf file
- Use `[text to display] (path/to/your/pdffile)` to add a link to your pdf file.
 - For example, since static is the default path for files, if you put your pdf file in static directory, you can use `[text to display] (/nameofpdf.pdf)` as the path.
 - If you build a new directory, your path would be `[text to display] (/blog/nameofdirectory/nameofpdf.pdf)`.

Embed pdf in your blog

If you put the pdf file in the `/content/blog/` directory, then the corresponding url for it would be `yourwebsite.github.io/blog/nameofpdf.pdf`.

For example, if I put a file called `HugoNote.pdf` in the `/content/blog/` directory, then my url would be `https://huijunchen9260.github.io/blog/HugoNote.pdf`.

After figuring out the url of pdf file, we can use google drive pdf viewer to embed the pdf file. Use the url example above, the html code would be

```
<embed src="https://drive.google.com/viewerng/viewer?embedded=true&url=https://huijunchen9260.github.io/blog/HugoNote.pdf" width="100%" height="100%" />
```

Description of code: - `src` is the source of display. we use google drive viewer to show pdf file inside the webpage. - `url` is the url based on the website. - `width` and `height` can set using absolute measure (e.g. 500px) or relative measure (e.g. percentage of the webpage).