

# Hugo

[Hugo](#) is a fast, themeable static site generator written in Go. Here are some tips based on our class on 14 Feb.

1. Start by downloading the [latest release for your platform](#). That probably means the one labeled `macOS-64bit.tar.gz` or `Windows-64bit.zip`.
2. When you unzip it, there will be a README and a single executable file named `hugo` or `hugo.exe`. You can copy that executable to some directory in your PATH, or just specify its precise location each time you run it, such as:

```
$ ~/Downloads/hugo version
Hugo Static Site Generator v0.36 linux/amd64 BuildDate: 2018-02-05T15:22:28Z
```

The Windows instructions recommend unzipping to a new folder `C:\Hugo` and then adding `C:\Hugo\bin` to your PATH environment variable.

3. Next you want to use Hugo to create a skeleton for your site. First change to your `cs120` folder, something like this:

```
cd ~/Desktop/cs120
```

And then create the new folder like this:

```
hugo new site assn2
```

Depending on your installation in the previous step, you may have to replace `hugo` in that command with the full location, so that would be something like:

```
~/Downloads/hugo new site assn2
```

4. This part is optional, but I recommend making a snapshot of the pristine files before you begin to edit them:

```
git add assn2
git commit -m "hugo new site"
```

5. Next you want to add a theme. Themes can be downloaded and customized at any time. Browse the possibilities at <https://themes.gohugo.io/>. When you find one you like, click its “Download” button. It should take you to a GitHub page. Look for the green “Clone or download” drop-down button. I recommend using “Download ZIP”.

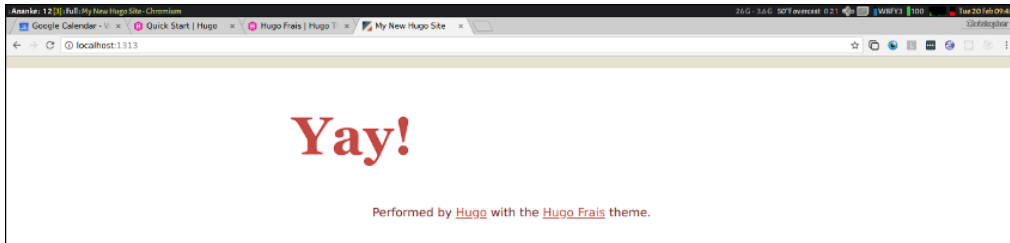


Figure 1:

6. You need to extract the theme's zip file *into* the new `assn2/themes` folder. It should look something like this, first make sure you're in the `assn2` folder:

```
$ cd assn2
$ ls themes
hugo-frais-master
```

This new folder, `hugo-frais-master`, is the unzipped version of [the theme that I chose](#).

7. Next, edit the file `assn2/config.toml`, adding a setting for the current theme. For me, that's:

```
theme = "hugo-frais-master"
```

But the setting should match the name of the subdirectory within the themes directory.

Again, I recommend making a snapshot that includes the pristine theme files:

```
git add .
git commit -m "hugo theme"
```

8. Now you should be able to ask Hugo to generate and serve your static site, like this:

```
$ hugo server -D
[...]
Watching for changes in ...
Serving pages from memory
Running in Fast Render Mode. For full rebuilds on change: ...
Web Server is available at http://localhost:1313/
(bind address 127.0.0.1)
```

And then you can access that address in your browser: <http://localhost:1313/>

9. Try editing `config.toml` again, changing the title setting:

```
title = "Really Great Hugo Site"
```

Save, and the still-running `hugo server` command will automatically incorporate that change and reload it in your browser. (The page title shows up in the tab bar.)

10. Next, you can try adding some content: posts or pages. The main command for creating a new page is:

```
hugo new posts/my-first-post.md
```

(You'll have to interrupt the running `hugo server` with a control-C, or you can open a separate terminal window – just make sure you're still in the `assn2` directory.)

11. The above command creates a file in `content/posts` directory called `my-first-post.md`, using Markdown (text) format. Open that in your editor, and it starts with some metadata:

```
---
title: "My First Post"
date: 2018-02-20T09:58:52-05:00
draft: true
---
```

You can edit those settings, and then add your content below the second `---` delimiter. Save it, and `hugo server -D` should generate it at <http://localhost:1313/posts/my-first-post/>

As you edit and save, that page will automatically refresh.

12. Finally, you should check the README for your theme to see what other capabilities and settings it has. You can also browse the documentation of Hugo itself at <https://gohugo.io/documentation/>.