Building an Academic Website in 60 Minutes

Thompson, Jonathan
VanIwaarden, River

University of Colorado at Colorado Springs

AMS Graduate Chapter Website Workshop
May 02, 2022

Prerequisites

Here's what we'll need for this workshop:

1. Git: `https://git-scm.com/downloads`

Prerequisites

Here's what we'll need for this workshop:

1. Git: `https://git-scm.com/downloads`
    - Why? Git is a version control system used to manage any kind of source code (C, MATLAB, LaTeX, etc.).

Prerequisites

Here's what we'll need for this workshop:

1. Git: `https://git-scm.com/downloads`
   - Why? Git is a version control system used to manage any kind of source code (C, MATLAB, LaTeX, etc.).

2. That's it!

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
    - HTML describes the **content** of a webpage.

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
    - HTML describes the **content** of a webpage.

2. CSS: **C**ascading **S**tyle **S**heets

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
    - HTML describes the **content** of a webpage.

2. CSS: **C**ascading **S**tyle **S**heets
    - CSS describes the **style** of a webpage.

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
   - HTML describes the **content** of a webpage.

2. CSS: **C**ascading **S**tyle **S**heets
   - CSS describes the **style** of a webpage.

3. JavaScript

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
   - HTML describes the **content** of a webpage.

2. CSS: **C**ascading **S**tyle **S**heets
   - CSS describes the **style** of a webpage.

3. JavaScript
   - JavaScript describes **how elements interact** on a webpage.

Background

What comprises a website?

1. HTML: **H**yper**T**ext **M**arkup **L**anguage
    - HTML describes the **content** of a webpage.

2. CSS: **C**ascading **S**tyle **S**heets
    - CSS describes the **style** of a webpage.

3. JavaScript
    - JavaScript describes **how elements interact** on a webpage.

Building websites from scratch takes a lot of work!

Introduction to Hugo

Building a website from scratch is a bit like building a PDF document from scratch:

- Doing so provides the most flexibility, but is also excessive for most purposes.

Introduction to Hugo

Building a website from scratch is a bit like building a PDF document from scratch:

- Doing so provides the most flexibility, but is also excessive for most purposes.

- There exist tools to create websites visually, but these tools tend to be specialized and difficult to collaborate with.

Introduction to Hugo

Building a website from scratch is a bit like building a PDF document from scratch:

- Doing so provides the most flexibility, but is also excessive for most purposes.
- There exist tools to create websites visually, but these tools tend to be specialized and difficult to collaborate with.
- For our purposes, we often care more about content than we do about customization.

Introduction to Hugo

Building a website from scratch is a bit like building a PDF document from scratch:

- Doing so provides the most flexibility, but is also excessive for most purposes.
- There exist tools to create websites visually, but these tools tend to be specialized and difficult to collaborate with.
- For our purposes, we often care more about content than we do about customization.

This is where Hugo comes in!

Markdown

Markdown is a simple markup language for creating **formatted text** designed to be easily readable in its source code form.

Markdown

Markdown is a simple markup language for creating **formatted text** designed to be easily readable in its source code form.

Hugo takes Markdown...

```
---
title: "Hello World!"
date: 2022-05-01T00:35:03-06:00
draft: true
---

# Hello, world!

### Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.

> Ut enim ad minim veniam, quis nostrud exercitation ullamco
> laboris nisi ut aliquip ex ea commodo consequat.

- enim neque volutpat ac tincidunt
- tincidunt ornare massa eget egestas
- sit amet justo donec enim
```

Markdown

Markdown is a simple markup language for creating **formatted text** designed to be easily readable in its source code form.

Hugo takes Markdown...

```
---
title: "Hello World!"
date: 2022-05-01T00:35:03-06:00
draft: true
---

# Hello, world!

### Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.

> Ut enim ad minim veniam, quis nostrud exercitation ullamco
> laboris nisi ut aliquip ex ea commodo consequat.

- enim neque volutpat ac tincidunt
- tincidunt ornare massa eget egestas
- sit amet justo donec enim
```

→

...and turns it into a webpage:

Markdown

Markdown is a simple markup language for creating **formatted text** designed to be easily readable in its source code form.

Hugo takes Markdown...

```
---
title: "Hello World!"
date: 2022-05-01T00:35:03-06:00
draft: true
---

# Hello, world!

### Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.

> Ut enim ad minim veniam, quis nostrud exercitation ullamco
> laboris nisi ut aliquip ex ea commodo consequat.

- enim neque volutpat ac tincidunt
- tincidunt ornare massa eget egestas
- sit amet justo donec enim
```

→

...and turns it into a webpage:

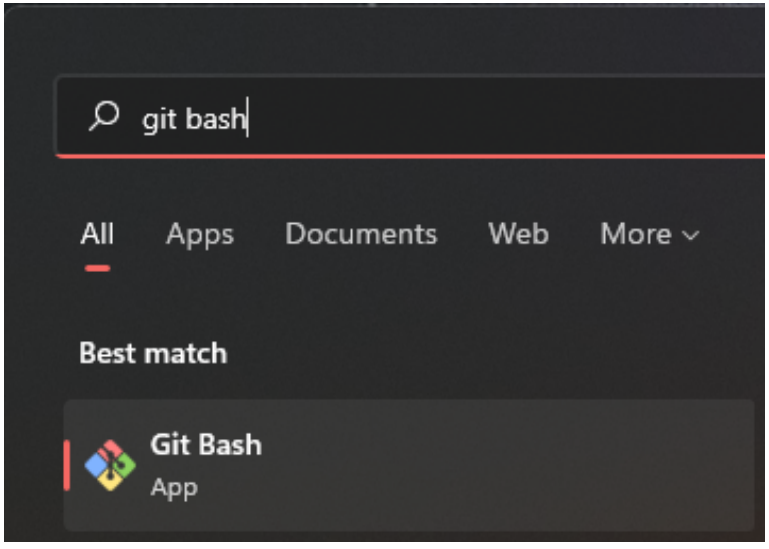
In many ways, Hugo does for webpages what LaTeX does for documents.

Getting Started

First, we need to open a command line. The command line (also known as a terminal or console) is an invaluable tool for executing text commands on a computer.

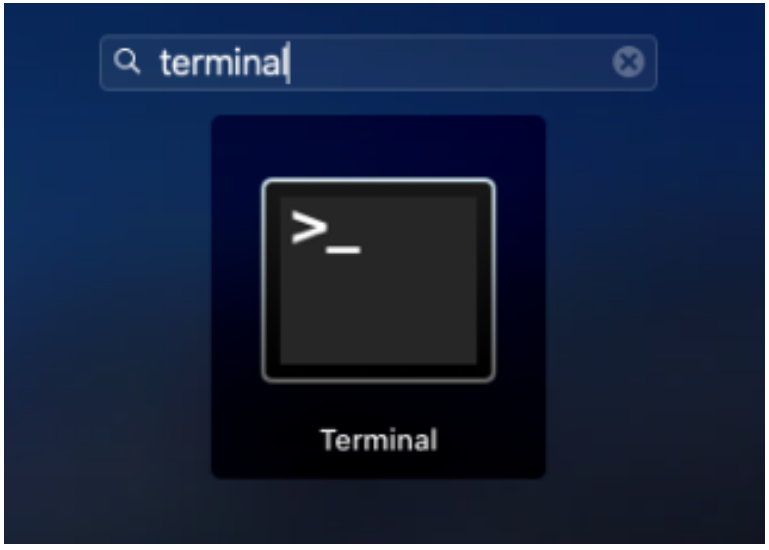
Windows

From the Start Menu, open the `Git Bash` application:



MacOS

Open the `Terminal` application from Launchpad:



Getting Started

Now we'll change directories (`cd`) to the `Desktop` folder. Type the following into your terminal (**without** the "$" character[1]), and then hit the `[Enter]` key:

```
$ cd Desktop
```

Next, we'll use Git to clone (download) all the necessary code for this project. Type the following command into your terminal, and then hit the `[Enter]` key:

```
$ git clone https://github.com/uccs-ams/website-workshop.git --recurse
```

[1] It is customary to add this character in technical documentation to indicate that the proceeding command is to be entered into a terminal.

Getting Started

Let's change directories again to the newly-created `website-workshop` folder:

```
$ cd website-workshop
```

The next command we need to run depends on our operating system. We're going to check out slightly different versions of the code to account for operating system differences:

Windows

```
$ git checkout windows
```

MacOS

```
$ git checkout macos
```

MacOS [M1]

```
$ git checkout macos-arm64
```

Start Hugo

We're finally ready to build our website! We first need to start our Hugo server, which lets us see our website in a browser and rebuilds the site automatically whenever we modify any project files:

```
$ ./hugo -D serve
```

After running this command, we can see our website by navigating to `http://localhost:1313/` in a web browser. It'll be blank for now, but we're about to fix that.

Website Scaffolding

Basic information about our website (title, theme, website URL, etc.) is contained in the `config.toml` configuration file inside the `website-workshop` folder on our Desktop.

Let's open that file with a text editor and replace the existing content with the following to enable the `researcher` Hugo theme:

config.toml

```
1  baseURL = "http://example.org/" # If you have your own domain name, add it here.
2  title = "Mathy McMathFace"
3  theme = "researcher"
```

We're going to add quite a few more things to this file before we're done.

Website Scaffolding

Now we'll add to our website a footer and a link to our CV by adding a handful of lines to our `config.toml` file.[2]

Be sure to pay attention to spacing!

config.toml

```
4
5   [params]
6     [params.footer]
7       text = "Copyright 2022 Mathy McMathFace"
8
9   [menu]
10    [[menu.main]]
11      name = "Curriculum Vitae"
12      url = "/cv.pdf"
13      weight = 1
```

Let's see what our website looks like so far by navigating to `http://localhost:1313/` in a web browser.

[2] The file we're linking to in this workshop can be found in our project folder at `content/cv.pdf`.

Main Page

Our **index** (main) page is looking pretty barren - let's change that! Our first task is to create a new Markdown file for Hugo to turn into a webpage, and we'll use Hugo to do so.

Open a new terminal, and then run the following command to navigate to our project directory:

```
$ cd Desktop/website-workshop
```

From here, we'll instruct Hugo to generate a new Markdown file for us to modify:[3]

```
$ ./hugo new _index.md
```

This command generates a new Markdown file at `website-workshop/content/_index.md`.

[3] The `_index.md` file name holds special significance in Hugo projects; the contents of this file are rendered to the main page.

Main Page

This is our first encounter with actual Markdown code. Let's open our newly-generated file in a text editor and add some information in order to introduce ourselves to the world:

content/_index.md

```
---
title: "Mathy McMathFace"
date: 2022-05-01T17:53:23-06:00  # This is auto-generated - no need to modify!
#draft: true
---

# About

## PhD Student - University of Colorado at Colorado Springs

 
**Mathy McMathface** is currently a graduate student studying Mathematics
at the University of Colorado at Colorado Springs (UCCS) and is an officer for
the American Mathematical Society Graduate Student Chapter at UCCS. Their
research interests include inverse reactive currents for use in unilateral
phase detractors, automatic cardinal grammeter synchronization, and how these
two may be combined in various ways to create turbo encabulators.
```

Contact Page

Now that the world knows who we are, let's create a contact page so that they know how to reach us. Like before, we'll use Hugo to generate a new page via the terminal:

```
$ ./hugo new contact.md
```

We also need to add a link to this page from our index page. We can add links to pages just by modifying our `config.toml` configuration file like we did before for our resume:

config.toml

```
8   ...
9   [menu]
10    [[menu.main]]
11      name = "Resume"
12      url = "/resume.pdf"
13      weight = 1
14    [[menu.main]]
15      name = "Contact"
16      url = "/contact"
17      weight = 2
```

Contact Page

This page gives us the opportunity to see some new Markdown syntax for quote blocks and hyperlinks:

content/contact.md

```
---
title: "Contact"
date: 2022-05-01T18:22:37-06:00
#draft: true
---

# Contact

 
## E-mail
Feel free to e-mail me at [mmcmath@uccs.edu](mailto:mmcmath@uccs.edu).

 
## Mailing Address
> University of Colorado at Colorado Springs
>
> Department of Mathematics
>
> 1420 Austin Bluffs Parkway
>
> Colorado Springs, CO 80918
```

Posts Page

We can also use Hugo to easily create and manage blog posts:

```
$ ./hugo new posts/hello-world.md
```

As before, we'll add a link to our new `Posts` page:

config.toml

```
...
[menu]
  [[menu.main]]
    name = "Resume"
    url = "/resume.pdf"
    weight = 1
  [[menu.main]]
    name = "Contact"
    url = "/contact"
    weight = 2
  [[menu.main]]
    name = "Posts"
    url = "/posts"
    weight = 3
```

Posts Page

We've seen up to this point that Markdown provides some pretty basic formatting out of the box. What if we want LaTeX in our blog posts?

content/posts/hello-world.md

```
---
title: "Hello World"
date: 2022-05-01T15:58:31-06:00
#draft: true
math: true
---

# Hello, world!

This is my first blog post. In this post, I will do things like:

- Make lists
- _Emphasize_ text
- Make text **bold**
- Show off some inline $\LaTeX$
```

Back to the Beginning

At this point, we have created a basic website with Hugo by creating several Markdown pages. We now need to convert these files into HTML, CSS, and JavaScript so that our website can be hosted on the public internet.

To convert our markdown files into a website, we simply run the following command in our terminal:[4]

```
$ ./hugo -D
```

The compiled website files can then be found in the `public` folder in our project directory.

[4] The `-D` flag instructs Hugo to include Markdown files with `draft: true` in their header in the final compiled website.

Conclusion

We conclude with some helpful links for reference:

- Markdown Cheatsheet:
  `https://www.markdownguide.org/cheat-sheet/`
- Hugo Documentation:
  `https://gohugo.io/documentation/`
- Hugo Hosting Overview:
  `https://gohugo.io/hosting-and-deployment/`
- Researcher Hugo Template:
  `https://themes.gohugo.io/themes/hugo-researcher/`

Questions?

Appendix: Hosting

Now that we've built a basic website, we need somewhere to host it so that it is accessible on the public internet. There are two steps to this process:

1. Choosing a domain name (e.g., example.org)
2. Choosing a hosting service

Choosing a hosting service is outside the scope of this workshop, but some good places to start are:

- Bluehost
- Namecheap
- HostGator

Each of these services provide both domain registration and website hosting services.

Appendix: Hugo Themes

Hugo's support for various site themes makes it easy to change the look and feel of our website. In this workshop, we used the excellent Researcher theme for its academic look and feel, but a full list of themes can be found at `https://themes.gohugo.io/`.

To change a theme (e.g., to the Ananke theme), we first clone (download) the theme to our project by running the following from our terminal (as before, the entire command should be on a single line):

```
$ git submodule add https://github.com/theNewDynamic/gohugo-theme-ananke.git
    themes/ananke
```

Appendix: Hugo Themes

Next, we update the active theme in our project's `config.toml` configuration file:

config.toml

```
1  baseURL = "http://example.org/" # If you have your own domain name, add it here.
2  title = "Mathy McMathFace"
3  theme = "ananke"
4  ...
```

In general, configuration options for a particular theme (e.g., to change background or text color) can usually be found in the corresponding example configuration file for that theme at `themes/<theme name>/exampleSite/config.toml`.