

CODELAND 2018 | BY MONICA POWELL

CREATE A BLOG WITH JEKYLL



ABOUT US!

Monica Powell, long-time #CodeNewbie! I currently lead e-mail marketing at Jopwell and periodically write for FreeCodeCamp, Hacker Noon and Code Like A Girl.

👋 Twitter: [@waterproofheart](https://twitter.com/waterproofheart)

🔗 www.aboutmonica.com/

Nia Murrell is a PLACEHOLDER TEXT. Remember to ask Nia for her bio.

🔗 www.niamurrell.com/

TODAY WE WILL CREATE A
STATIC BLOG USING
JEKYLL.

SOME OF THE TOPICS WE WILL COVER

- ▶ Static vs. Dynamic Sites
- ▶ Installing Jekyll
- ▶ Jekyll Project Structure
- ▶ Editing Sitewide Variables
- ▶ Creating a Post
- ▶ Adding images
- ▶ Adding Custom Overlay Colors

STATIC VS DYNAMIC WEBSITES

- ▶ Static websites unlike dynamic websites:
- ▶ Don't have to maintain or secure databases
- ▶ Display the same information for all readers
- ▶ Less expensive to host
- ▶ Load faster
- ▶ Less prone to hacking
- ▶ Usually don't have CMS

WHAT IS JEKYL

A “Jekyll website” is a “static (plain HTML) website that has been created using Jekyll. Jekyll is software that creates websites. Jekyll isn’t actually “running” the live website; rather, Jekyll is a “static site generator”: it helps you create the static site files, which you then host just as you would any other HTML website.

Programming Historian

WEBSITES POWERED BY JEKYL

- ▶ Documentation Sites

- ▶ Ruby <https://www.ruby-lang.org/en/>

- ▶ Bootstrap <https://getbootstrap.com/>

- ▶ Blogs/Personal Websites

- ▶ <https://zachholman.com>

- ▶ www.datalogues.com (Monica's blog!)

- ▶ <https://www.chenhuijing.com/>

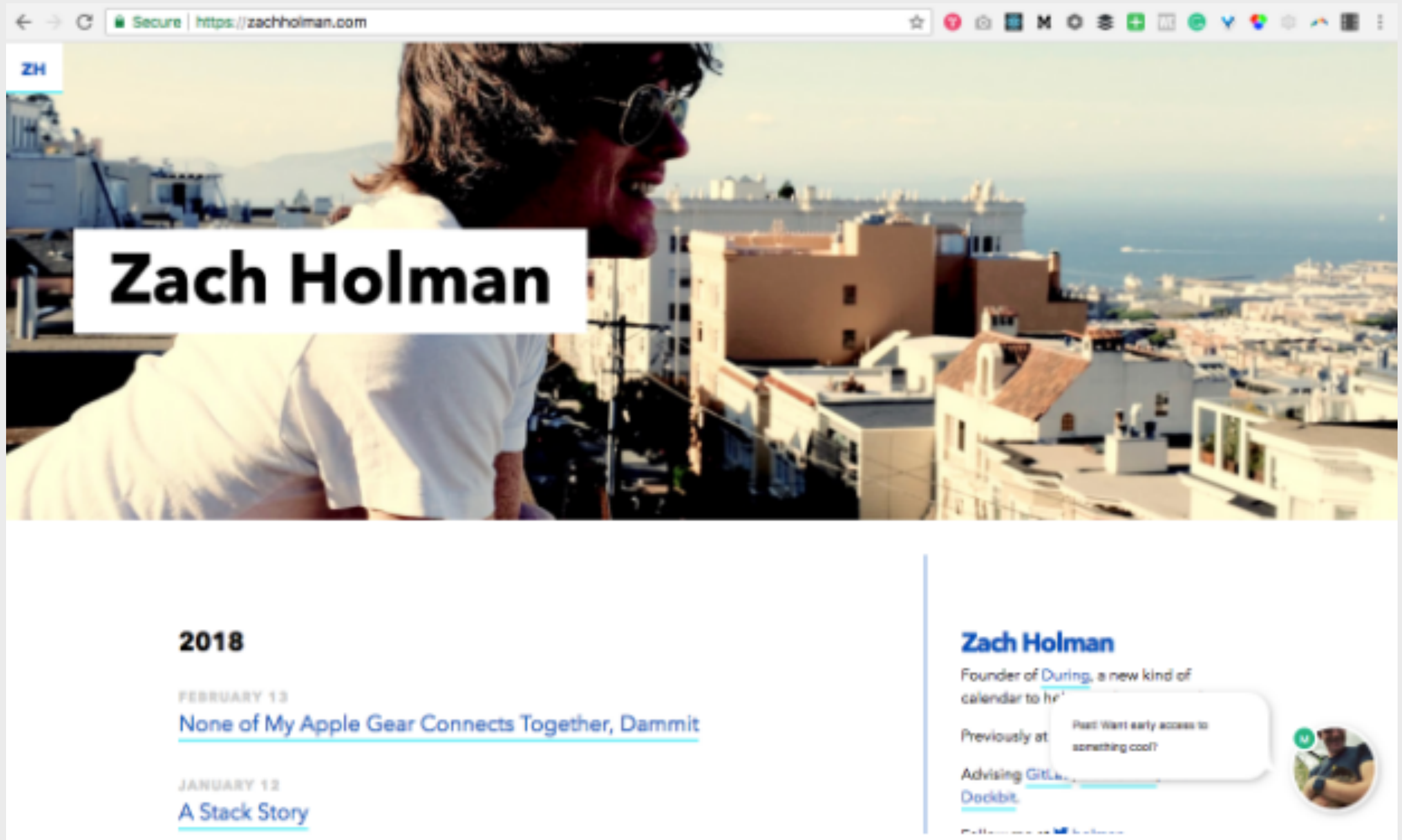
WEBSITES POWERED BY JEKYL



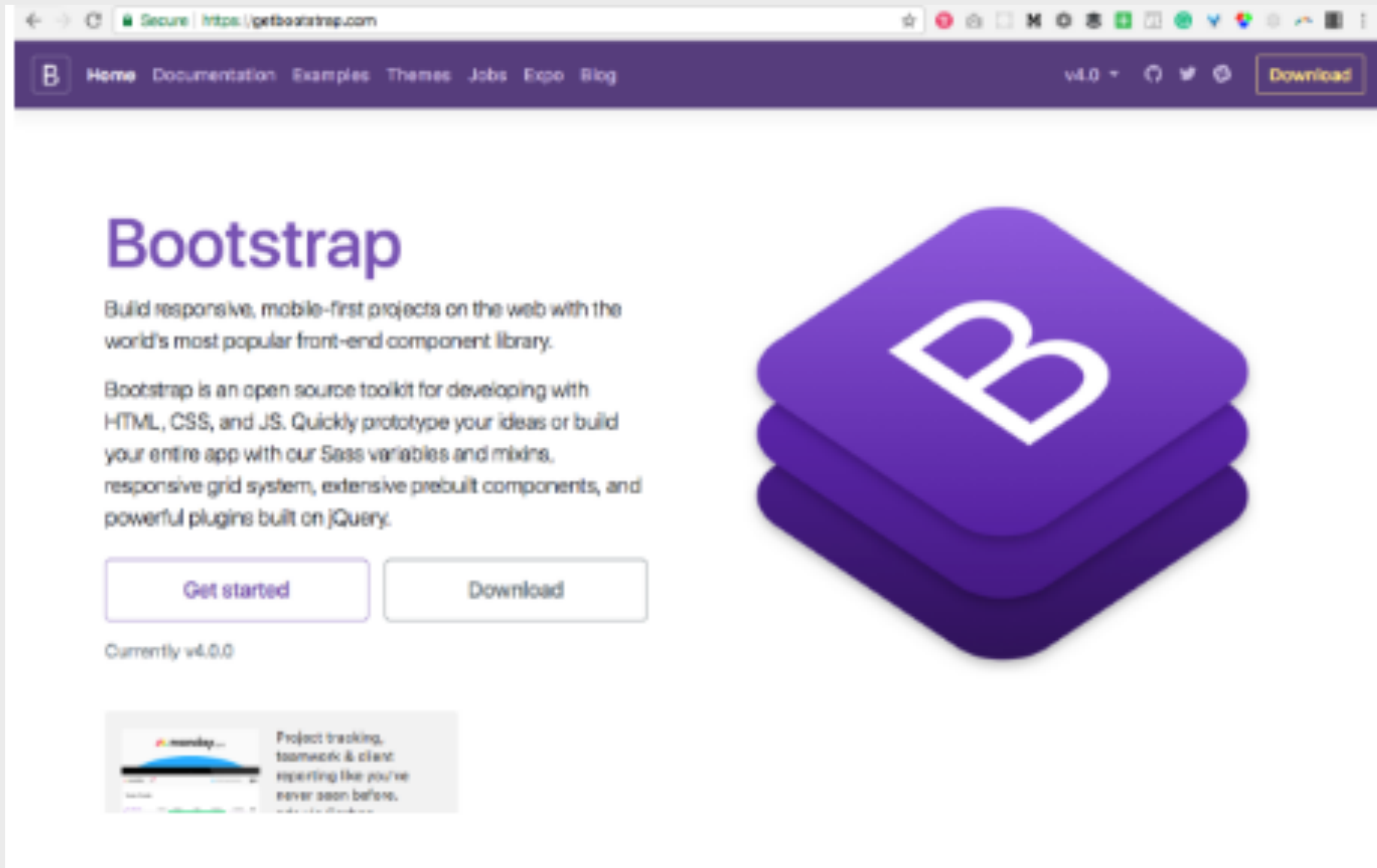
WEBSITES POWERED BY JEKYL



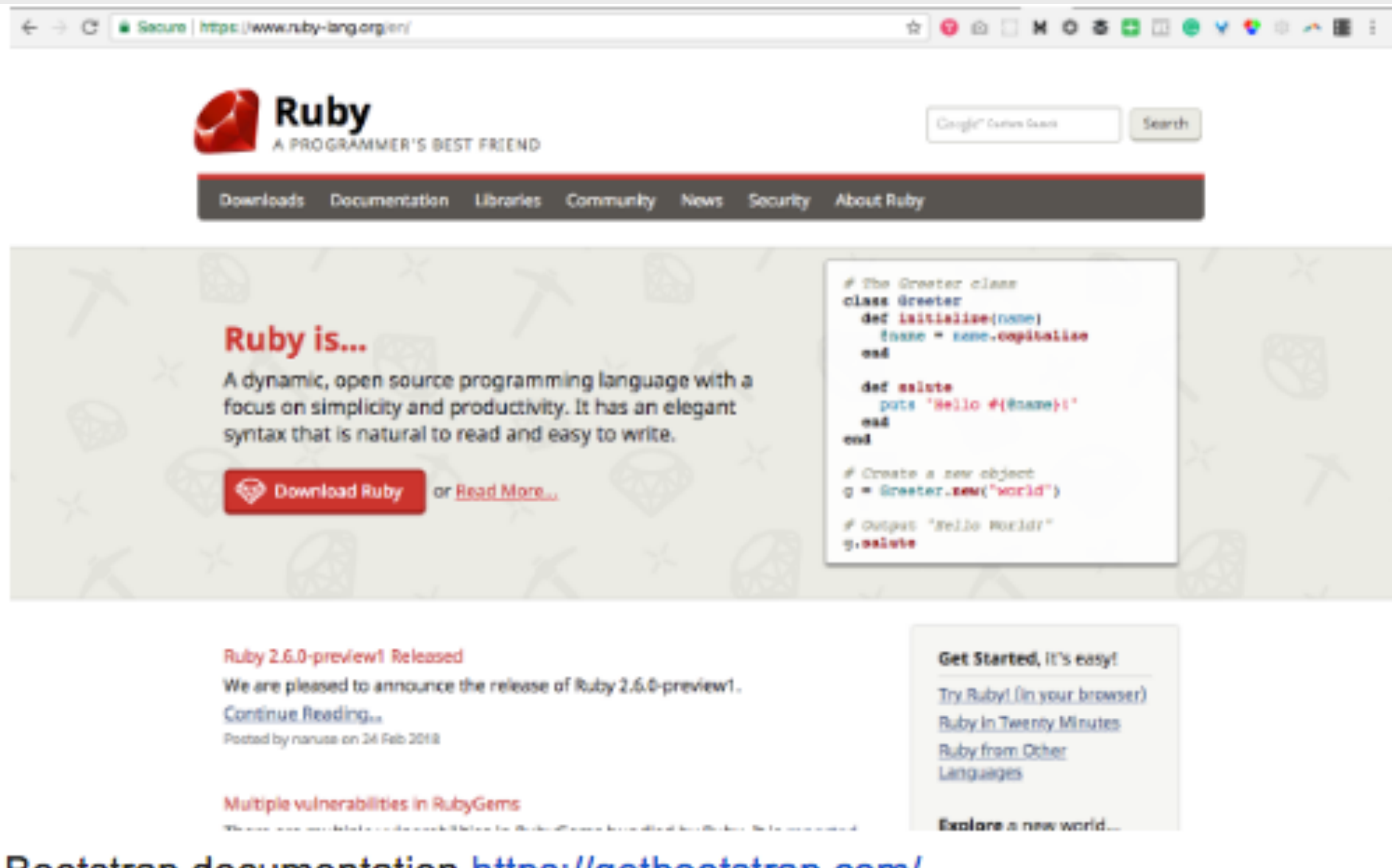
WEBSITES POWERED BY JEKYL



WEBSITES POWERED BY JEKYL



WEBSITES POWERED BY JEKYLL



The screenshot shows the Ruby website homepage. At the top, there's a navigation bar with links: Downloads, Documentation, Libraries, Community, News, Security, and About Ruby. Below this, a large banner features the Ruby logo and the text "Ruby is... A dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write." To the right of the banner is a code snippet for a Ruby class. Below the banner, there are two news items: "Ruby 2.6.0-preview1 Released" and "Multiple vulnerabilities in RubyGems". On the right side, there's a sidebar with links: "Get Started, It's easy!", "Try Ruby! (in your browser)", "Ruby in Twenty Minutes", "Ruby from Other Languages", and "Explore a new world...".

Ruby
A PROGRAMMER'S BEST FRIEND

Google Custom Search Search

Downloads Documentation Libraries Community News Security About Ruby

Ruby is...
A dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.

[Download Ruby](#) or [Read More...](#)

```
# The Greeter class
class Greeter
  def initialise(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end

# Create a new object
g = Greeter.new("world")

# Output: "Hello World!"
g.salute
```

Ruby 2.6.0-preview1 Released
We are pleased to announce the release of Ruby 2.6.0-preview1.
[Continue Reading...](#)
Posted by nanase on 24 Feb 2018

Multiple vulnerabilities in RubyGems
We have been notified of several vulnerabilities in the RubyGems library. We have released

Get Started, It's easy!
[Try Ruby! \(in your browser\)](#)
[Ruby in Twenty Minutes](#)
[Ruby from Other Languages](#)
[Explore a new world...](#)

Bootstrap documentation <https://getbootstrap.com/>

WEBSITES POWERED BY JEKYL

- ▶ Documentation Sites

- ▶ Ruby <https://www.ruby-lang.org/en/>

- ▶ Bootstrap <https://getbootstrap.com/>

- ▶ Blogs/Personal Websites

- ▶ <https://zachholman.com>

- ▶ www.datalogues.com (Monica's blog!)

- ▶ <https://www.chenhuijing.com/>



INSTALLATION FEST

"To start, press any key."
Where's the "any" key?

INSTALLATION FEST

1. Open Command Line
2. Install git from <https://git-scm.com/downloads>
3. Confirm installation by running:
 - ▶ `$ git -version`

INSTALLATION FEST

- ▶ Configure git with **your** name and e-mail address.

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.com
```

Use the same e-mail that is associated with your GitHub account.

INSTALLATION FEST

- ▶ Install Ruby
- ▶ On Windows:
 - ▶ Use [RubyInstaller](#)
- ▶ On Mac:
 - ▶ By default Ruby may be installed depending on OS. If not:

```
$ brew install ruby
```

INSTALLATION FEST

- ▶ On [GitHub.com](https://github.com/Monica/dactl) fork this repository <https://github.com/Monica/dactl>
- ▶ On command line:
 - ▶ Git clone https://github.com/YOUR_USERNAME/dactl
 - ▶ Note: GitHub may prompt you to login with your credentials.

INSTALLATION FEST

1. FORK MY REPOSITORY AT:
[HTTPS://GITHUB.COM/MONICA/DACL](https://github.com/Monica/dactl)

MOnica / dactl
forked from melangue/dactl

Code

Pull requests 0

Projects 0

Wiki

Insights

Settings

Fork 53

Modern, fast and configurable Jekyll theme with some tricks up it's sleeve. <http://melangue.github.io/dactl>

Edit

Add topics

10 commits

1 branch

0 releases

3 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is even with melangue/dactl

V and V Code blocks fix

.bundle

_includes

_layouts

2. NAVIGATE TO THE REPOSITORY YOU JUST FORKED. TYPE 'GIT CLONE' IN YOUR TERMINAL FOLLOWED BY THIS URL. COMMAND SHOULD BE 'GIT CLONE URL'

Clone with HTTPS

Use SSH

Use Git or checkout with SVN using the web URL.

<https://github.com/Monica/dactl.git>

Open in Desktop

Download ZIP

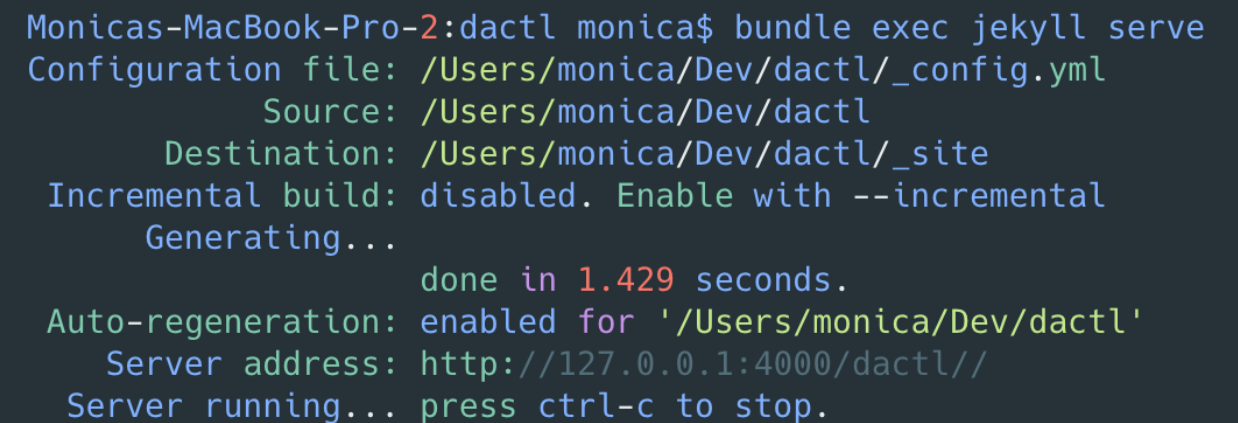
INSTALLATION FEST

- ▶ On [GitHub.com](https://github.com/Monica/dactl) fork this repository <https://github.com/Monica/dactl>
- ▶ On command line:
 - ▶ Git clone https://github.com/YOUR_USERNAME/dactl
 - ▶ Note: GitHub may prompt you to login with your credentials.

INSTALLATION FEST

- ▶ Install dependencies and then view Jekyll site on local server with the following:

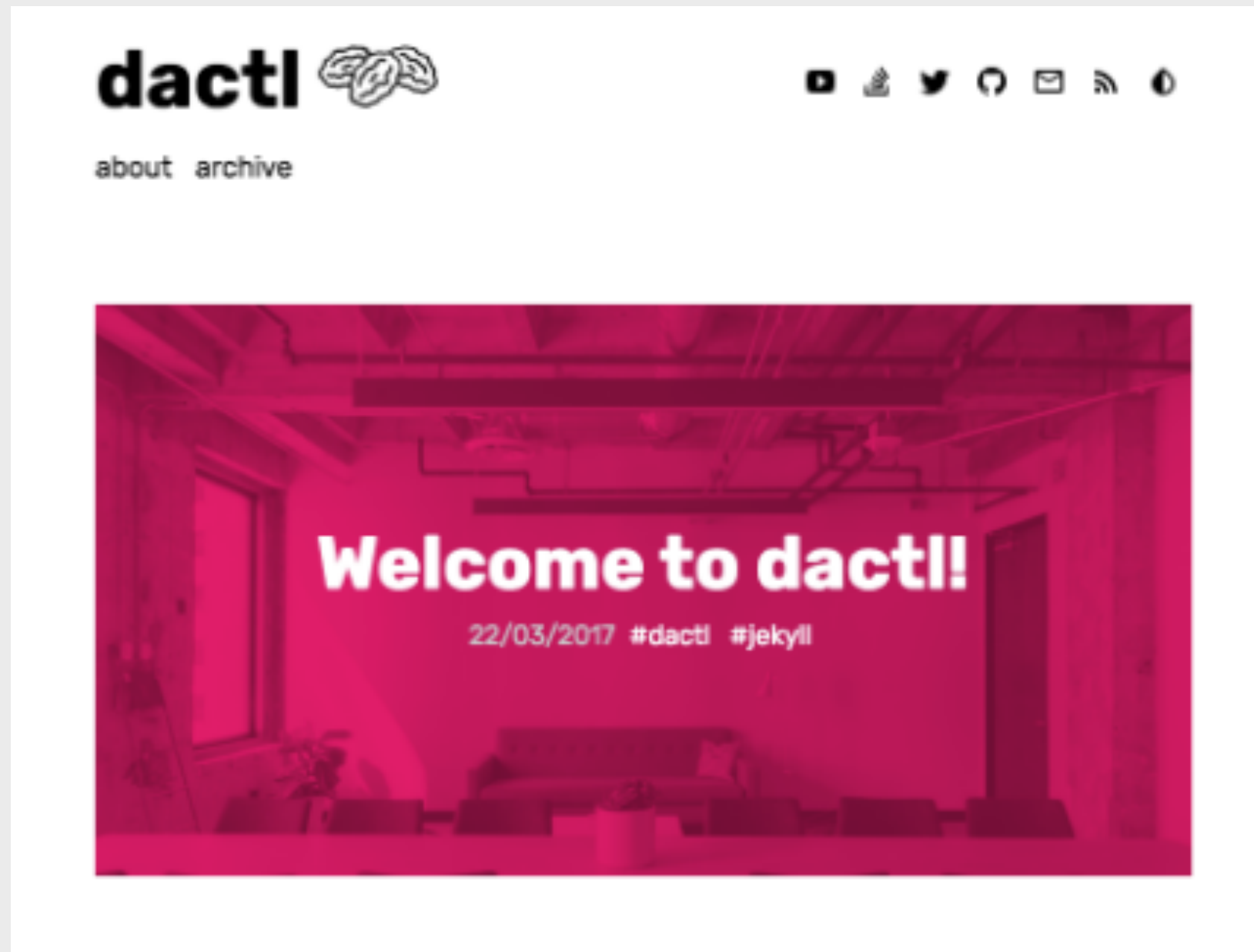
```
$ cd dactl
$ bundle install
$ bundle exec jekyll serve
```

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays the output of the command 'bundle exec jekyll serve'. It shows the configuration file path, source and destination directories, incremental build status, generation time, auto-regeneration status, and the server address.

```
Monicas-MacBook-Pro-2:dactl monica$ bundle exec jekyll serve
Configuration file: /Users/monica/Dev/dactl/_config.yml
      Source: /Users/monica/Dev/dactl
      Destination: /Users/monica/Dev/dactl/_site
Incremental build: disabled. Enable with --incremental
      Generating...
                                done in 1.429 seconds.
Auto-regeneration: enabled for '/Users/monica/Dev/dactl'
      Server address: http://127.0.0.1:4000/dactl//
      Server running... press ctrl-c to stop.
```

INSTALLATION FEST

- ▶ **Success!** If you visit <http://127.0.0.1:4000/dactl/> a page similar to this.





EDITING POSTS IN JEKYLL

"To start, press any key."
Where's the "any" key?

EDITING JEKYL FILES

- ▶ We will primarily be editing various files that Jekyll converts into HTML and CSS for the final site.
- ▶ Note that any files placed in the “_site” folder will be deleted/overridden when the website is generated. Therefore make sure you are placing your files in the correct folder.

YAML

- ▶ YAML stands for Yet Another Markup Language. YAML was created to be easy for humans to write and computers to read.
- ▶ YAML is used in Jekyll projects to define site-wide variables (in `_config.yml`) and to tell Jekyll how to process various pages (in **front matter** of posts).

YAML _CONFIG.YML

- ▶ **Activity:** Edit at least 3 site-wide variables including title, description, and anything else you want to update in the _config.YML file.

YAML_CONFIG.YML

title: The title of your website, as you want it to appear in the header of the webpage.

email: Your email address.

description: A description of your website that will be used in search engine results and the site's RSS feed.

baseurl: Fill in the quotation marks with a forward slash followed by the name of your website folder (e.g. `"/JekyllDemo"`) to help locate the site at the correct URL.

author:

fullname: Your full name

twitter: Your Twitter username (do not include @ symbol).

github: Your GitHub username.

YAML_CONFIG.YML

Let's make changes to the site in the _config.YML and view them by restarting our server.

In the terminal:



```
ctrl + C <!-- to stop the server if it is running-->  
bundle exec jekyll serve
```

NOTABLE JEKYL FILES

- ▶ `_config.yml` provides basic settings information about your site, such as the site's title and additional possibilities we won't cover here, like how to structure links to posts (e.g. should they follow the pattern `MySite.com/year/month/day/post-title?`).
- ▶ `_includes` folder has files that get included on all or certain pages (e.g. code to make the header contain your site title and main menu on every page of the site)

NOTABLE JEKYL FILES

- ▶ `_layouts` folder contains code that controls how the pages on your site look (default.html), as well as customizations of that code to further style blog posts (post.html) and pages (page.html)
- ▶ `_posts` folder holds the individual files that each represent a blog post on your website. Adding a new post to this folder will make a new blog post appear on your website, in reverse chronological order (newest post to oldest).

NOTABLE Jekyll FILES

- ▶ `_sass` folder holds SCSS files that control the visual design of the site
- ▶ `_site` folder is where the HTML pages that appear on the web are generated and stored (e.g. you'll write and save posts as Markdown files, but Jekyll will convert these to HTML for display in a web browser)

NOTABLE JEKYL FILES

- ▶ index.html is where the layout of the homepage is defined
- ▶ about.md is an example of a Jekyll page. It's already linked in the header of your website, and you can customize its text by opening and writing in that file.
- ▶ css folder holds CSS converted from SCSS that controls the visual design of the site

NOTABLE JEKYLL FILES

- ▶ feed.xml lets people follow the RSS feed of your blog posts
- ▶ index.html controls the structuring of content on your site's homepage



WRITE FIRST BLOG POST

WRITE FIRST BLOG POST

- ▶ Create a file in the `_posts` directory.
- ▶ “How you name files in this folder is important. Jekyll requires blog post files to be named according to the following format:
 - ▶ YEAR-MONTH-DAY-title.MARKUP
- ▶ The following are examples of valid post filenames:
 - ▶ 2018-05-04-code-land-is-awesome.md
 - ▶ 2018-01-01-hello-world.md

YAML FRONT MATTER

"Any file that contains a [YAML](#) front matter block will be processed by Jekyll as a special file. The front matter must be the first thing in the file and must take the form of valid **YAML** set between triple-dashed lines."*

(Source: [Jekyll Documentation](#))

YAML FRONT MATTER

[View This Post](#)



```
---
```

```
layout: post
```

```
title: YAML front matter in posts
```

```
tags:
```

- jekyll
- dactl
- howto

```
hero: https://source.unsplash.com/collection/345758/
```

```
overlay: orange
```

```
---
```

YAML FRONT MATTER

- ▶ Let's create the front matter for our new post and view it on our local server by saving the file and then refreshing the local version of our site.
- ▶ The server needs to be reset for changes to `_config.yml` to appear however, changes to bother files will generally appear with a refresh.

YAML FRONT MATTER

- ▶ Paste this into the top of your empty blog post file.

```
---  
layout: post  
title:  "Welcome to dactl!"  
tags:  
  - dactl  
  - jekyll  
hero: https://source.unsplash.com/collection/430471/  
overlay: red  
published: true
```

YAML FRONT MATTER

- ▶ View new post on your site.
- ▶ Check-in:
 1. Change overlay from red to blue, green, purple or orange.
 2. Change title of the post.

ADDING CONTENT

- ▶ Below the 3 dashes associated with YAML Front Matter we can begin writing our content!
- ▶ We will be writing our content in Markdown.

MARKDOWN REVIEW

← → ↺

Secure

https://markdown-it.github.io

☆

markdown-it demo

☐ html ☐ xhtmlOut ☐ breaks ☒ linkify ☒ typographer ☒ highlight

language-

☐ CommonMark strict

clear permalink

html source debug

Advertisement :)_

- __[pica](https://nodeca.github.io/pica/demo/)_ - high quality and fast image resize in browser.

- __[babelfish](https://github.com/nodeca/babelfish/)_ - developer friendly i18n with plurals support and easy syntax.

You will like those projects!

h1 Heading 8-)

h2 Heading

h3 Heading

h4 Heading

h5 Heading

h6 Heading

Horizontal Rules

Advertisement 😊

- [pica](#) - high quality and fast image resize in browser.
- [babelfish](#) - developer friendly i18n with plurals support and easy syntax.

You will like those projects!

Heading 😎

Heading

Heading

Heading

Heading

Heading

Horizontal Rules

MARKDOWN REVIEW

- ▶ <https://markdown-it.github.io/>
- ▶ Play around on Markdown-it and create a sample blog post (keep it short and sweet).

ADDING CONTENT – BLOG POST

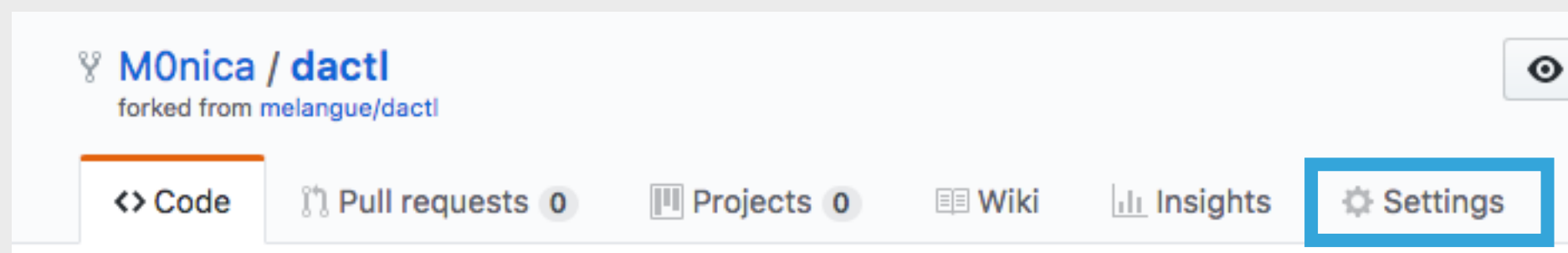
- ▶ **Activity: Write a post about something(s) you've learned or enjoyed at Codeland so far!**
- ▶ Either write your post in Markdown-it and paste it into your file or write Markdown from scratch into our blog post file (under the YAML)
- ▶ Save post

PUBLISH CONTENT – GITHUB PAGES

- ▶ **Activity: Publish our work on Github!**
- ▶ Commit work to GitHub via command line:

```
git status  
git add .  
git commit -m "commit message"  
git push
```

- ▶ After successfully committing to GitHub, go to your repository's settings page:



PUBLISH CONTENT – GITHUB PAGES

- ▶ Scroll down to the GitHub Pages section of settings and select master branch as the source.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://m0nica.github.io/dactl/>.

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

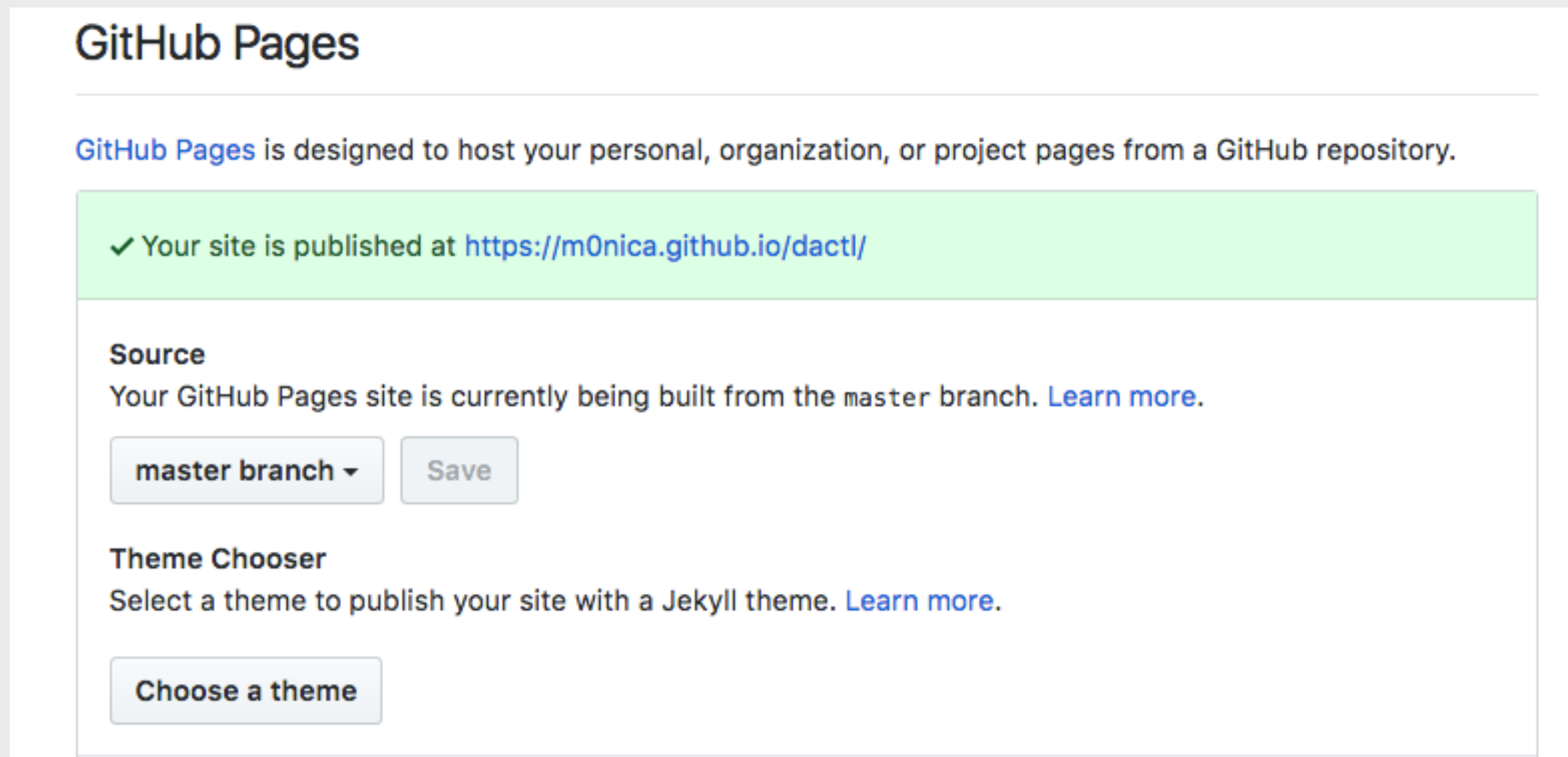
Choose a theme

Custom domain

Custom domains allow you to serve your site from a domain other than `m0nica.github.io`. [Learn more.](#)

PUBLISH CONTENT – GITHUB PAGES

- ▶ If all went well then you should see a green message appear after refreshing the page. You can click the link to view your blog on GitHub.



ADDING CONTENT – ABOUT PAGE

- ▶ **Activity: Edit About page to better reflect YOU!**
- ▶ Edit About.md
- ▶ Save file
- ▶ Refresh browser and see your new about page by clicking on the "About" link in the nav!

ADDING CONTENT – ABOUT PAGE

- ▶ **Activity: Edit About page to better reflect YOU!**
- ▶ Either write your post in Markdown-it and paste it into your file or write Markdown from scratch into our blog post file (under the YAML)
- ▶ Save post
- ▶ Refresh browser and see your new post!

ADDING CONTENT – IMAGES

Workshop 

[about](#) [archive](#)



About



ADDING CONTENT – IMAGES

- ▶ Currently our about.md file is using our author photos which are defined in the config file.

ADDING CONTENT – IMAGES

```
# Author info
author:
  fullname      : Monica Powell
  rss           : true # generate RSS feed and show it's icon in header
  mail          : monica@aboutmonica.com # change to your e-mail address
  twitter       : waterproofheart
  github        : m0nica
  youtube       : # youtube-user-name
  stackoverflow : # stackoverflow-user-name
  Disqus        : # dactl # your Disqus site name
  google_analytics : # 'UA-XXXXXXXX-X'
  photo         : "uploads/me2.png"
  photo2x       : "uploads/me.png"
```

```
<div class="page" markdown="1">

{% capture page_subtitle %}

{% endcapture %}
```

ADDING CONTENT – IMAGES

- ▶ Update author photo by uploading an image of yourself or your favorite animal/character to the uploads folder.
- ▶ Name the file firstname.png (or whatever file extension)
- ▶ then in your config file change line 38 from
- ▶ `photo` : "uploads/me2.png" to
- ▶ `photo` : "uploads/firstname.png"

ADDING CONTENT – IMAGES

- ▶ Restart server to see the changes we made to config file reflected on the about page.



```
ctrl + C <!-- to stop the server if it is running-->  
bundle exec jekyll serve
```

ADDING CONTENT – IMAGES

Workshop 

[about](#) [archive](#)



About



ADDING CONTENT – IMAGES

- ▶ To change the image on a blog post go to <http://www.unsplash.com>
- ▶ Find an image and then copy paste it into the hero: part of the YAML for the post where you want to update the image. You should replace the old URL entirely.

ADDING PARTIAL – SOCIAL ICONS

- ▶ Partial layouts can be found in the `/_includes` folder of our project. These are HTML templates with re-usable elements that are found on multiple pages such as the header, footer, etc.
- ▶ **Activity: Add social icons to all of our posts.**

Share "Welcome to My Workshop!!":



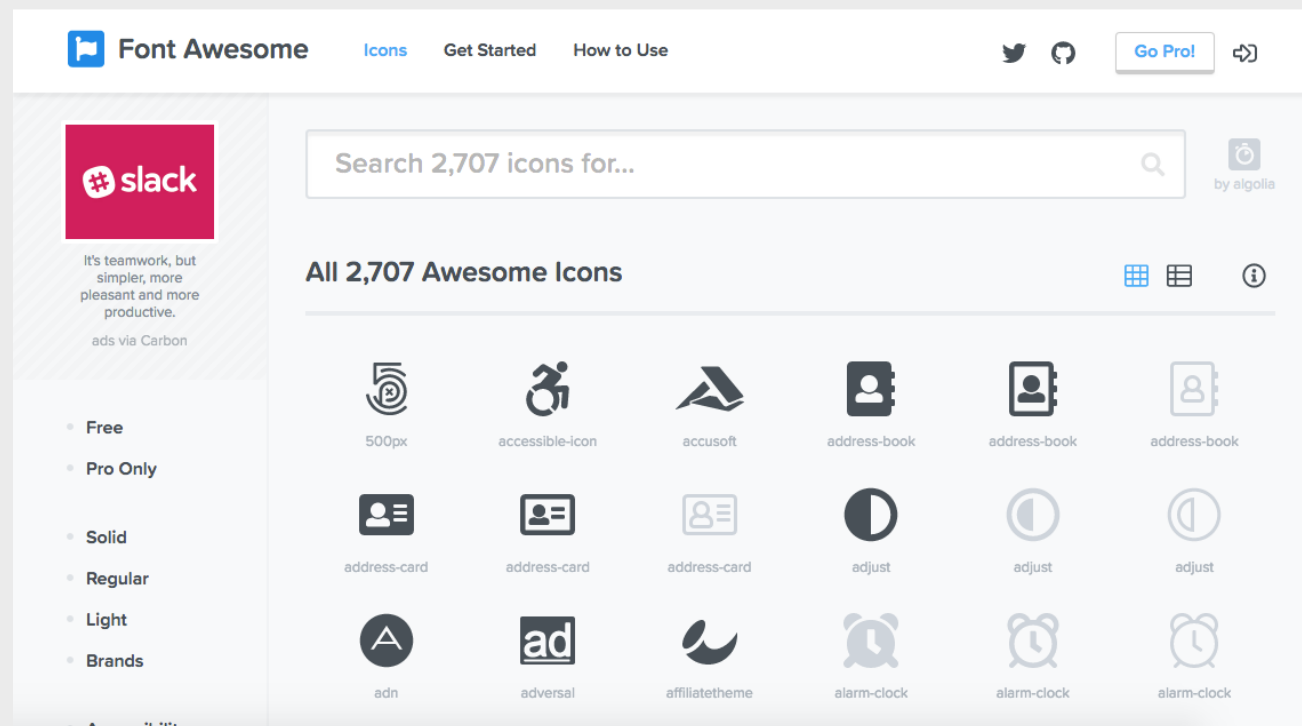
ADDING PARTIAL – SOCIAL ICONS

- ▶ We will be using [Font Awesome](#)'s free social icons.

- ▶ Already referenced in our head.html:

```
<!-- Font Awesome -->
```

```
<link rel="stylesheet"
href="https://use.fontawesome.com/
releases/v5.0.9/css/all.css"
integrity="sha384-5S0iIsAziJl6AWe0
HWRKTXlfcSHKmYV4RBF18PPJ173Kzn7jzM
yFuTtk8JA7QQG1"
crossorigin="anonymous">
```



ADDING PARTIAL – SOCIAL ICONS

- ▶ Create a file named `share_icons.html` in the `post` folder within the `_includes` folder.
- ▶ Add a link to the partial file we just created (code below) in our post layout in `_layouts/post.html`

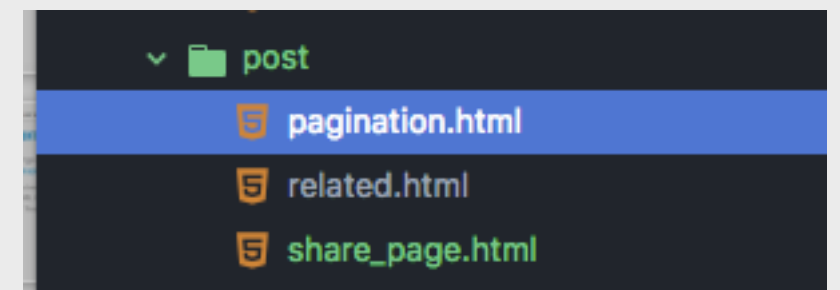
```
---  
layout: default  
---
```

```
{% include post.html post=page %}
```

```
<!-- The following line links to the partial we created for our social-share-icons -->  
{% include post/share_icons.html %}
```

```
{% if site.author.disqus %}  
  {% include externals/disqus.html %}  
{% endif %}
```

```
{% include post/related.html %}
```



ADDING PARTIAL – SOCIAL ICONS

► Add HTML for social icons to share_icons.html

```
<div class="share-page">
<h3>Share "{{ page.title }}"</h3>
```

```
<ul class="icons">
```

```
  <li><a href="https://twitter.com/intent/tweet?text={{ page.title }}&url={{ site.url }}{{ page.url }}
&via={{ site.author.twitter }}&related={{ site.author.twitter }}" rel="nofollow" target="_blank" title="Share on
Twitter"><i class="fab fa-twitter"></i></a></li>
```

```
  <li><a href="https://facebook.com/sharer.php?u={{ site.url }}{{ page.url }}" rel="nofollow" target="_blank"
title="Share on Facebook"><i class="fab fa-facebook-f">&nbsp;</i></a></li>
```

```
  <li><a href="https://www.linkedin.com/shareArticle?mini=true&url={{ site.url }}{{ site.baseurl }}{{ page.url }}
&title={{ page.title }}&summary={{ page.description }}&source={{ site.title }}"
onclick="window.open(this.href, 'pop-up', 'left=20,top=20,width=500,height=500,toolbar=1,resizable=0'); return
false;"
title="Share on LinkedIn"><i class="fab fa-linkedin-in"></i></a></li>
</ul>
```

```
</div>
{% include post/related.html %}
```

ADDING PARTIAL – SOCIAL ICONS

- ▶ Add styling for social icons in the `_icons.scss` file in our `_sass` folder.

```
// /* Style all font awesome icons */  
.fab {  
  padding: 20px;  
  width: 100px;  
  text-align: center;  
  text-decoration: none;  
  color: white;  
}  
  
ul.icons {  
  -webkit-padding-start: 0px;  
}  
ul.icons li {  
  display: inline-block;  
  margin: 10px;  
}
```

ADDING PARTIAL – SOCIAL ICONS

- ▶ Add background-color for social icons in the `_icons.scss` file in our `_sass` folder. According to the [brands colors](#).

```
// /* Style all font awesome icons */  
  
i.fab.fa-twitter {  
  
    background-color: #1da1f2;  
  
}  
i.fab.fa-linkedin-in {  
    background-color: #0077b5;  
  
}  
i.fab.fa-facebook-f {  
    background-color: #3b5998;  
  
}
```

EDIT CSS – UPDATE HEADER COLOR

- ▶ Edit css in `_SASS/_variables.scss`

```
$bg-color: #eee;
$text-color: #000;
$text-color-light: #828282;
$link-color: #c34f4a;
$metadata-color: #999;
$post-title: #313131;
$code-border: #f5f5f5;
$code-bg-color: #fff;
$table-border-color: #e5e5e5;
$table-header-color: #fefefe;
$shadow-color: rgba(0, 0, 0, 0.1);
$invert-logo-color: 'invert(0)';
```

EDIT CSS – UPDATE HEADER COLOR

- ▶ Change the hex color in line 24:

```
$link-color: #8BC34A;
```

- ▶ Updated to:

```
$link-color: #c34f4a;
```

Feel free to pick your own color!

Workshop 

about archive

Workshop 

about archive

CREATE NEW POST OVERLAY

- ▶ Did you try to change the post overlay color to a color other than red, blue, green, purple or orange?
- ▶ If so! That shouldn't have worked. Let's add a new custom color by navigating to `_includes > utils > hero.html`
- ▶ **Activity: Create a custom overlay color to better fit your ideal blog aesthetic.**

CREATE NEW POST OVERLAY

- ▶ Currently in hero.html only red, blue, green, purple or orange are defined and the fallback is a white color.

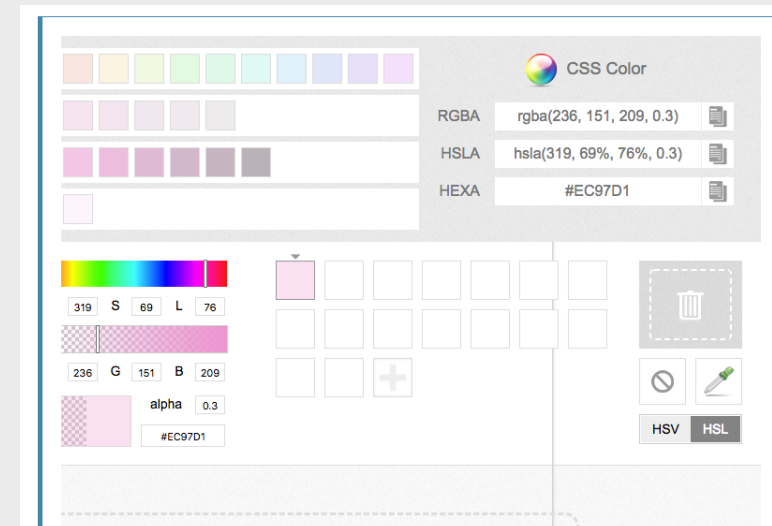
```
{% if post.overlay != nil %}

{% if post.overlay == "red" %}{% assign overlay = "rgba(255,0,100," %}{% endif %}
{% if post.overlay == "blue" %}{% assign overlay = "rgba(0,100,255," %}{% endif %}
{% if post.overlay == "green" %}{% assign overlay = "rgba(100,255,0," %}{% endif %}
{% if post.overlay == "purple" %}{% assign overlay = "rgba(100,0,255," %}{% endif %}
{% if post.overlay == "orange" %}{% assign overlay = "rgba(255,100,0," %}{% endif %}

{% else %}
{% assign overlay = "rgba(0,0,0,.8)" %}
{% endif %}
```

CREATE NEW POST OVERLAY

- ▶ Create bubblegum color



```
{% if post.overlay == "bubblegum" %}{% assign  
overlay="rgba(236, 151, 209," %}{%endif %}
```

```
{% if post.overlay == "YOUR CUSTOM COLOR" %}{%  
assign overlay="rgba(R, G, B," %}{%endif %}
```

- ▶ Choose your own color at: [https://
developer.mozilla.org/en-US/docs/Web/CSS/
CSS_Colors/Color_picker_tool](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool)

CREATE NEW POST OVERLAY

- ▶ Change front matter:
 - ▶ overlay: bubblegum
- ▶ Refresh blog and see new bubble gum colored post!

