

[BACK TO MAIN PAGE](#)

GitHub Pages | Learning Markdown

1. Introduction

Follow along with the [introductory slides](#).

Presentations slides

Create a GitHub account

- Navigate to [Github](#) and sign up for an account if you don't already have one.
- After registering, sign in to your account.

2. Working with GitHub Pages

Create a new GitHub repository

- From your GitHub main page, click on the green **New Repository** button. This is the repository (storage location) where the content of your website will be hosted.

- Provide a name for your repository. The name you enter will determine the URL of your repository.
- i.e. `https://github.com/<your-github-username>/<your-repo-name>`
- e.g. `https://github.com/jasonbrodeur/DASH-githubpages`
- Check the box to **Initialize this repository with a README**
- Click the **Create repository** button
- Your browser will now open to the top-level page of the repository. Your repository will contain one file: **README.md**.

Create a folder and a text file

In this task, you'll create a folder that we'll need to initiate the webpage, and you'll learn how to create and edit files.

We want to create a folder named `docs` in the top level of the repository. This is where the webpage content will live. We also want to create a file in this folder called `index.md`. This will be the home page for your website. - In the top-level repository page, click **Create new file** - NOTE1: To create a file, simply enter the desired name and file extension. - NOTE2: To create a folder, you need to add a trailing slash (/) to the name, and you need to [create a file within the folder](#), since empty folders aren't saved. You can always delete this file later. - In the **Name your file...** box, enter `docs/index.md`. Click **Commit change**. You will now have a folder named `docs` in your repository with a file named `index.md`.

Edit your text file

- Click on the `index.md` file to open it in the file viewer page
 - On the file viewer page, click the edit button (pencil icon) to switch to editing mode
 - Add a bit of text to this file (anything is fine).
 - When finished editing, scroll to the bottom of the page to the **Commit changes** box.
 - NOTE: The **Commit changes** box is used to record information about changes before you commit them.
 - (optional) If interested, modify the comment from **Update index.md** to something more descriptive
 - (optional) Add a description, if desired
 - Click the **Commit changes** box to finalize your commit. Your changes will be represented in the file.

Viewing changes

- In the `index.md` file viewing page, explore the *Blame* and *History* buttons to better understand how GitHub keeps track of file changes.

3. Exploring Features and Settings | Preparing the webpage

In this step, we'll explore some of the functionality available in a GitHub repository and show you how to get started with [GitHub Pages](#) - Ensure that you've created the `docs` folder and the `index.md` file, as outlined in the previous step. - Explore with Jay the various Features (tabs) that are available within a repository - Open

the Settings Tab. Follow along with Jay as he explains the various pages - On the (topmost) **Options** page, scroll down to the **GitHub Pages** section - For **Source**, select **master branch /docs folder** (note you could also pick the *master branch* option, but not for this example.) - Click **Choose a theme** to select a theme for your webpage - Congratulations, your website is ready. The link will be provided in the GitHub Pages section. - You may initially receive a 404 error, but after a minute, your website will appear with the content from `index.md`. Remember: By default, GitHub Pages expects at least one file in the `docs` folder named `index.md`. This is the main (home) page for the webpage. **Now, we need to add some content to your webpage!**

4. Creating content with Markdown

One of the really nice things about GitHub and GitHub Pages is that you can use Markdown to format text on a webpage without the use (or knowledge!) of html coding. This makes it really easy to create content. In this section, you'll learn a bit more about Markdown and how to use it to create formatted text.

What is Markdown?

Borrowed shamelessly from Github's [Mastering Markdown](#) page:

Markdown is a way to style text on the web. You control the display of the document; formatting words as bold or italic, adding images, and creating lists are just a few of the things we can do with Markdown. Mostly, Markdown is just regular text with a few non-alphabetic characters thrown in, like `#` or `*`.

Markdown uses simple notation to apply simple formatting rules. Since it's pretty much just plain text, it's transferrable and much simpler than marked-up text like HTML or even Word or Google documents. It's also very readable in its plain text format, which is nice. For much of the writing that you do for the web, Markdown is good enough. Github uses Markdown for its documents (this document was created in markdown), as does a variety of other web platforms (Reddit and Trello, as examples).

Edit index.md

Using the Github editor, open `index.md` for editing and add some content to it. If you would like to add a title to your page, add the following text at the top:

```
---
title: <enter your title here>
---
```

Use the [Mastering Markdown guide](#) as a reference (or other guides on the web) to create a fictional document that contains most of the following elements: - Headings of a number of different levels - bolded, italicized text - insert an image from the web - insert an image that is hosted in your GitHub repository - An ordered list - A bulleted list - A link to another website - A snippet of code - A table - And finally, an emoji! **NOTE:** Use the **Preview changes** tab to see (mostly) how it will look on your page.

Commit your changes and enjoy the products of your hard work!

More information and references for Markdown: -[Mastering Markdown](#) - [Adam Pritchard's Markdown Cheatsheet](#) - [Daring Fireball's Markdown Introduction](#)

Create a second page

- Create another markdown page in the `docs` folder. Name it as you wish, but ensure that it ends in `.md`. Add a title and other elements as done earlier.

5. Customizing pages | What is jekyll?

You may have noticed by now that there is another file (`config.yml`) in your `docs` folder that you didn't create or put there. This is your configuration file, and it's written in a language called **YAML**. The `config.yml` file was created at the time that you turned on GitHub pages, and it provides instructions to a static site generator software called **jekyll**. GitHub pages uses jekyll in the background to convert your Markdown file(s) to formatted html to display as a webpages.

There is *a lot* that you can do with jekyll (within GitHub pages and as a standalone application on your computer) to generate customized webpages. While only the theme information is set in your file initially, there are many ways that you can **customize** it. Here, we'll make a few simple ones: - Use the GitHub editor to edit your `config.yml` file. - beneath the first (theme) line, add the following:

```
plugins:
  - jekyll-relative-links
relative_links:
  enabled: true
  collections: true
include:
  - <the name of your second page>.md

title: <your webpage title>
description: <description of your webpage>
```

What this does: - The `plugins` and `relative-links` lines make it easier for us to refer to files within our repository. - The include statement is used to indicate to jekyll which other Markdown pages should be converted to html pages. - The title and description statements allow you to override the repository name and description in the webpage title.

More about Jekyll

From jekyllrb.com

“Jekyll is a static site generator. You give it text written in your favorite markup language and it uses layouts to create a static website. You can tweak how you want the site URLs to look, what data gets displayed on the site, and more.”

GitHub pages uses jekyll to turn your markdown files (hosted in your GitHub repo) into html files and presents them on your desired website.

Jekyll [webpage](#) and Jekyll project [on GitHub](#).

6. Doing more

Embedding web content

The nice thing about jekyll is that it doesn't just accept Markdown—it also accepts html, meaning that you can insert things like embed codes to insert content. - Insert this workshop's slideshow into one of your webpages by inserting the following code into the .md file:

