

# What is GitHub Pages?

 [docs.microsoft.com/en-us/learn/modules/create-host-web-sites-github-pages/2-what-is-github-pages](https://docs.microsoft.com/en-us/learn/modules/create-host-web-sites-github-pages/2-what-is-github-pages)

Here, we discuss the process of creating and maintaining a GitHub Pages web site.

GitHub Pages are static sites hosted directly from your GitHub repository. But they're more than just a collection of static files. By making use of site generation technologies like Jekyll and Liquid, developers define dynamic templates that are processed into complete static web sites. Every time a change is committed to the source branch associated with the site, it is re-generated using the latest updates and automatically published to the target URL.

Learn more about [Publishing sources for GitHub Pages sites](#).

## Enabling GitHub Pages

The first step in using GitHub Pages is to enable it from your repository's **Settings** tab. You can opt to use the **master** branch, or specify the **docs** folder within it. If you ever want to disable GitHub Pages, you can do so here.

### GitHub Pages

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

✓ Your site is published at <https://contoso.github.io/github-pages-with-jekyll/>

#### Source

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

master branch ▼

Select source

✓ master branch

Use the master branch for GitHub Pages.

master branch /docs folder

Use only the /docs folder for GitHub Pages.

None

Disable GitHub Pages.

Save

## Choosing a theme with Jekyll

**Jekyll** is the static site generator used by GitHub to build your web site from the contents of your repository. In addition to providing great content convenience, it also conforms to a standard design convention. This style standardization allows for swappable themes, which you can select from the **GitHub Pages** configuration.

## GitHub Pages

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

✓ Your site is published at <https://contoso.github.io/github-pages-with-jekyll/>

**Source**

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

master branch ▾

**Theme Chooser**

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

Your site is currently using the Minima theme.

Change theme

There are a variety of themes provided by GitHub. There is also an array of commercial and open source themes available from the Jekyll community.

←

Cayman

Slate

MERLOT

Time Machine

MINIMAL

Leap Day

→

# Cayman theme

Cayman is a clean, responsive theme for GitHub Pages.

[View on GitHub](#)[Download .zip](#)[Download .tar.gz](#)

Text can be **bold**, *italic*, or ~~strikethrough~~.

[Link to another page.](#)

There should be whitespace between paragraphs.

There should be whitespace between paragraphs. We recommend including a README, or a file with information about your project.

## Header 1

This is a normal paragraph following a header. GitHub is a code hosting platform for version control

Learn more about [Jekyll Themes](#).

## Using YAML front matter

---

The term *front matter* refers to YAML metadata that prepends the content of a file. For Jekyll, this includes generator instructions to indicate the layout style of a Markdown page ( `post` , `page` , and so on). It may also include page metadata, such as the document title, or page content variables, such as a blog post's author.

Below is a simple example that would use the `post` layout. This assumes there is a `_layouts/post.html` file that defines the container HTML. Other layout options may be offered by adding their respective HTML files in the `_layouts` folder.

yaml

```
---
layout: post
title: This is set as the document title.
---
```

This is visible body content, which may use Markdown, HTML, and Liquid templating.

Learn more about [Front Matter](#).

---

## Customizing your site

Once your site is up and running, you can customize details about your site via `_config.yml` . This file includes virtually all site-wide configuration options, including site metadata, navigation menus, theme colors, compiler options, and more.

Learn more about [\\_config.yml Configuration](#).

---

## Creating and editing content

Creating and editing pages on your site follows the standard GitHub experience. The files you use for your GitHub Pages web site enjoy all of the same benefits as other files in your GitHub repository, so you can edit them with any tool, create and merge branches, and link with issues or pull requests.

In addition to Markdown and HTML, Jekyll supports the **Liquid** template language syntax. Liquid provides the ability for users to dynamically insert variables and basic logic flow constructs into their content files. When compiled, the final product is standard HTML.

The example below shows a combination of `for` looping and variable insertion.

markdown

```
<ul>
  {% for post in site.posts %}
    <li>
      <h2><a href="{{ post.url }}">{{ post.title }}</a></h2>
      {{ post.excerpt }}
    </li>
  {% endfor %}
</ul>
```

Learn more about [Liquid template language](#).

## Working with blog posts

---

Despite not having a database to work with, Jekyll still supports the concept of blogging using a specific convention: `_posts/2020-06-25-blog-post-name.md`. As you can likely infer, all blog posts are stored in the `_posts` folder and use the date and name convention as shown. During compilation, Jekyll processes the files in this folder to produce a list of HTML blog posts.

The example below illustrates the structure of a simple blog post. It includes metadata for `subtitle`, `tags`, and `comments`, which may or may not be supported by the theme you choose.

markdown

```
---
layout: post
title: Blog post title rendered by theme
subtitle: Blog post subtitle rendered by theme
tags: welcoming
comments: true
---
```

This is the first line of rendered content in the post.

Learn more about [Adding content to your GitHub Pages site using Jekyll](#).

---

## Next unit: Exercise - Enable, create, and update a GitHub Pages web site

---

[Continue](#)

Need help? See our [troubleshooting guide](#) or provide specific feedback by [reporting an issue](#).