HOW TO CREATE A WEBSITE USING JEKYLL AND MARKDOWN ON WINDOWS

Published by: Jordan Johnson

This guide describes the process of creating a basic website using Jekyll and hosting it for free online using GitHub pages.

Process Overview

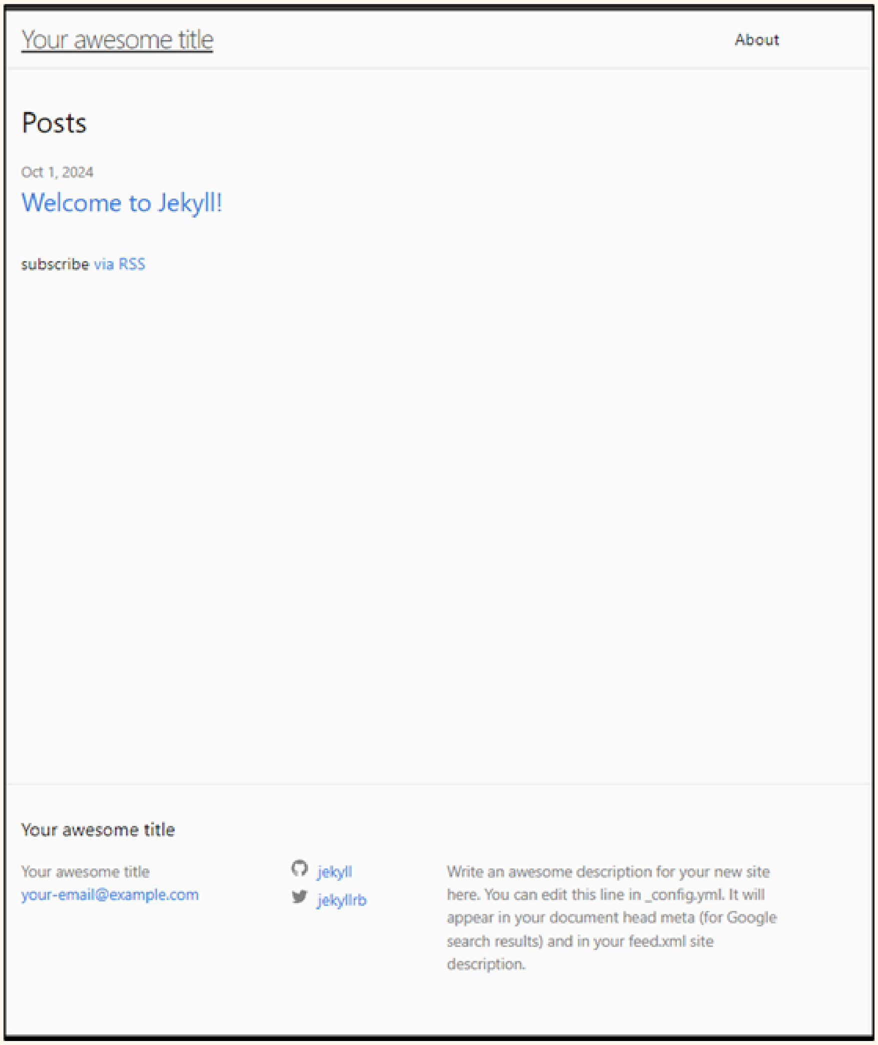
1. Install Jekyll.
2. Set the correct encoding value.
3. Create a skeleton website using the command line.

Install Jekyll on Windows

1. Download and install the latest Ruby Installer for Windows here: https://rubyinstaller.org/downloads/.
2. Run the installer using the default options.
3. Select **MSYS2 and MINGW development toolchain** when prompted in the installer.
4. Open a new terminal and run the command: **gem install Jekyll bundler**.
5. Verify that Jekyll has been installed properly by running: **jekyll -v**.
6. If you get an error stating that Jekyll was not installed properly, reboot your system and run **jekyll -v** again in a terminal. If the issue is still present, open a ticket with RubyInstaller.
7. To prevent any encoding issues in a terminal navigate to your Jekyll website directory and run the following command: **chcp 65001**.
    a. See the following page for more information on installing Jekyll on Windows: https://jekyllrb.com/docs/installation/windows/.

Create Your First Jekyll Website

1. In a Terminal navigate to the directory where you want to store your Jeyll website.
2. Run the following command:
   a. **gem install jekyll bundler**
3. Create a new Jekyll site called myblog by running:
   a. **jekyll new myblog**
4. Change into the newly created directory for your site by running:
   a. **cd myblog**
5. Build the site and make it available on your local server by running:
   a. **bundle exec jekyll serve**
6. View the newly created website by browsing to http://localhost:4000.



Your awesome title

About

Posts

Oct 1, 2024

Welcome to Jekyll!

subscribe via RSS

Your awesome title

Your awesome title
your-email@example.com

jekyll
jekyllrb

Write an awesome description for your new site here. You can edit this line in _config.yml. It will appear in your document head meta (for Google search results) and in your feed.xml site description.

Contents of Your Site Directory

Your blog folder will contain the following contents by default:

- **_Posts** – a folder for keeping your blog posts. You will likely use this folder the most. There is an example blog post created in this folder when you first build your site.
- **_Site** – a folder that holds files for the final output of your website. These are the files that you will use to deploy your final website to a web server. This site folder will automatically get updates as you build more blog posts and add more web pages. Do not manually update anything in this folder.
- **_config.yml** – a yaml file that allows you to configure settings for your Jekyll website.
- **Gemfile** – a file used with Ruby to store all dependencies for a website. You can specify plugins here.
- **About.markdown** – a markdown file for the **About** page on your website.
- **Index.markdown** - a markdown file for the **Home** page on your website.

Hosting Your Website on GitHub Pages

GitHub Pages is an option for hosting a static website for free.

Prerequisites:

1. You must have Git installed on your machine.
2. You must have a GitHub account.

Create GitHub Repository:

1. In GitHub, click **New Repository** and name it (e.g. my_blog).
   a. **Important:** Don’t initialize the repository with a README file.
2. Click **Create Repository**.

Update _config.yml:

1. In your Jekyll project, open **_config.yml**.
2. Update the **baseurl** variable to be the name of the repository you created in GitHub.
   a. For example: baseurl: “my_blog”
   b. **Note:** If you plan on using a purchased domain name, set the domain name to be the baseurl.
3. Save your changes.

Publish Your Jekyll Website and to GitHub Pages

1. Open a terminal and navigate to your Jekyll website directory.
2. Run **git init** to initialize the directory as a git repository.
    a. **Note:** When you create a GitHub Pages site, all the files for the site will go onto the **GitHub Pages** branch of your project.
3. Run **git checkout -b gh-pages** to checkout the GitHub Pages branch of your repository.
4. Run **git status**.
    a. Verify all the files in your Jekyll site show up and can be committed.
5. Run **git add .** to stage all your Jekyll files.
6. Run **git commit -m "initial commit"** to commit your changes.
7. Run **git remote add origin [Your GitHub repository link]** to link your local git repo with the git repo on GitHub.
8. Run **git push origin gh-pages** to push all your files to the remote GitHub repository.
9. In a browser, navigate to your repository in GitHub.
10. Verify all your Jekyll files are in your **gh-pages** branch.
11. Navigate to the settings tab in GitHub.
12. Scroll down to the GitHub Pages section.
13. Click the link where your site is published.

To Make Updates to Your Site

1. Commit and push all of your changes.
2. Ensure your changes are merged into the **gh-pages** branch.
3. Your page on GitHub Pages will update automatically.