GITHUB FORK

**1. Fork this project**

Click on the ***Fork*** button at the top right corner of this page. Forking means that you're copying this entire project and all its files into your account. Do not click on the ***Create fork*** button on the next page yet.

**2. Rename the repository to YOURUSERNAME.github.io**

You'll see the word "repository" used a lot in GitHub - it simply means "project". Under ***Repository name*** you should see the name beautiful-jekyll, this is where you need to rename your project to YOURUSERNAME.github.io (replace YOURUSERNAME with your GitHub user name). It's important to use this exact name so that GitHub will recognize it and automatically create a website for this project.

Tip: If you want to use a different URL for your website, check out the [FAQ](https://beautifuljekyll.com/faq/#custom-domain)

**3. Customize your website settings**

Edit the \_config.yml file to change any settings you want. To edit the file, first click on it to view the file, and on the next page click on the pencil icon to edit it (watch the video tutorial above if you're confused). The settings in the file are self-explanatory and there are comments inside the file to help you understand what each setting does. Any line that begins with a hashtag (#) is a comment, and the other lines are actual settings. After changing the settings, click the green ***Commit changes*** button to save these edits.

Note: In the video above, only one setting in the \_config.yml file is edited, but you should go through the rest of the settings as well.

**4. Congratulations! You have a website!**

If you named your project correctly and made an edit to the config file, your website should be ready in a minute or two at https://YOURUSERNAME.github.io. Every time you make a change to any file, your website will get rebuilt and should be updated in about a minute or so. Your website will be initialized with several sample blog posts and a couple other pages.

**The harder way (for advanced users)**

The instructions above explain how to use Beautiful Jekyll in the easiest way: by forking on GitHub. There are more [advanced installation methods](https://beautifuljekyll.com/getstarted/#install-steps-hard) that include either using GitHub Pages with remote themes, or using Ruby gems. They provide you with more control, but are only intended for advanced users.

Note: Beautiful Jekyll was primarily designed to be used as a GitHub theme, so you will not get any support if you use this theme via Ruby gems.

**Plans**

Beautiful Jekyll is, and always will be, free. But if you want to remove the Beautiful Jekyll ad from your website, use a Dark Mode skin, access office hours, or simply support the development efforts, [check out the different plans](https://beautifuljekyll.com/plans).

**Add your own content**

To add pages to your site, you can either write a markdown file (.md) or you can write an HTML file. It's much easier to write markdown than HTML, so that's the recommended approach ([here's a great tutorial](https://markdowntutorial.com/) if you need to learn markdown in 5 minutes).

To see an example of a markdown file, click on any file that ends in .md, for example [aboutme.md](https://github.com/daattali/beautiful-jekyll/blob/master/aboutme.md). On that page you can see some nicely formatted text (there's a word in bold, a link, a few bullet points), and if you click on the pencil icon to edit the file, you'll see the markdown code that generated the pretty text. Very easy!

In contrast, look at [tags.html](https://github.com/daattali/beautiful-jekyll/blob/master/tags.html). That's how your write HTML - not as pretty. So stick with markdown if you don't know HTML.

Any markdown or HTML file that you create will be available on your website under https://<yourusername>.github.io/<pagename>. For example, if you create a file about.md (or about.html) then it'll exist at https://<yourusername>.github.io/about.

Files you create inside the [\_posts](https://github.com/daattali/beautiful-jekyll/blob/master/_posts) directory will be treated as blog entries. You can look at the existing files there to get an idea of how to write blog posts. Note the format of the blog post files - they must follow the naming convention of YEAR-MONTH-DAY-title.md. After you successfully add your own post, you can delete the existing files inside [\_posts](https://github.com/daattali/beautiful-jekyll/blob/master/_posts) to remove the sample posts, as those are just demo posts to help you learn.

**Customizing parameters for each page**

**One last important thing**: In order to have your new pages use this template and not just be plain HTML pages, **you must add**[**YAML front matter**](https://jekyllrb.com/docs/front-matter/)**to the top of each page**:

---

---

This is where you'll be able to give each page some extra parameters (such as a title, a subtitle, an image, etc - [below is a list of all parameters](https://github.com/daattali/beautiful-jekyll#supported-parameters)). Add any parameters you want between these two dashed lines, for example:

---

title: Contact me

subtitle: Here you'll find all the ways to get in touch with me

---

If you don't want to use any parameters on a page, you still need to use the two dashed lines. If you don't, then your file will be shown as-is without the Beautiful Jekyll template.

You can look at the top of [aboutme.md](https://raw.githubusercontent.com/daattali/beautiful-jekyll/master/aboutme.md) as an example.

**Important takeaway: ALWAYS add the YAML front matter, which is two lines of three dashes, to EVERY page. If you have any parameters, they go between the two lines.**

**Supported parameters**

Below is a list of the parameters that Beautiful Jekyll supports (any of these can be added to the YAML front matter of any page). Remember to also look in the \_config.yml file to see additional site-wide settings. If there's a parameter that you want to apply to the entire site instead of one specific page, check out [this FAQ answer](https://beautifuljekyll.com/faq/#default-params).

**Main parameters**

These are the basic YAML parameters that you are most likely to use on most pages.

| **Parameter** | **Description** |
| --- | --- |
| title | Page or blog post title |
| subtitle | Short description of page or blog post that goes under the title |
| tags | List of tags to categorize the post. Separate the tags with commas and place them inside square brackets. Example: [personal, analysis, finance] |
| cover-img | Include a large full-width image at the top of the page. You can either provide the path to a single image (eg. "/path/to/img") , or a list of images to cycle through (eg. ["/path/img1", "/path/img2"]). If you want to add a caption to an image, then you must use the list notation (use [] even if you have only one image), and each image should be provided as "/path/to/img" : "Caption of image". |
| thumbnail-img | For blog posts, if you want to add a thumbnail that will show up in the feed, use thumbnail-img: /path/to/image. If no thumbnail is provided, then cover-img will be used as the thumbnail. You can use thumbnail-img: "" to disable a thumbnail. |
| comments | If you want do add comments to a specific page, use comments: true. Comments only work if you enable one of the comments providers (Facebook, disqus, staticman, utterances, giscus, CommentBox) in \_config.yml file. Comments are automatically enabled on blog posts but not on other pages; to turn comments off for a specific post, use comments: false. |
| mathjax | If you want to use LaTeX formulas, you need to enable MathJax. Note that in MathJax you need to use $$ and \\( to start and end expressions |

**Parameters for SEO and social media sharing**

These parameters let you control what information shows up when a page is shown in a search engine (such as Google) or gets shared on social media (such as Twitter/Facebook).

| **Parameter** | **Description** |
| --- | --- |
| share-title | A title for the page. If not provided, then title will be used, and if that's missing then the site title (from \_config.yml) is used. |
| share-description | A brief description of the page. If not provided, then subtitle will be used, and if that's missing then an excerpt from the page content is used. |
| share-img | The image to show. If not provided, then cover-img or thumbnail-img will be used if one of them is provided. |

**Less commonly used parameters**

These are parameters that you may not use often, but can come in handy sometimes.

| **Parameter** | **Description** |
| --- | --- |
| author | Specify the author of a blog post (useful if a website has multiple authors). |
| readtime | If you want a post to show how many minutes it will take to read it, use readtime: true. |
| show-avatar | If you have an avatar configured in the \_config.yml but you want to turn it off on a specific page, use show-avatar: false. |
| social-share | By default, every blog post has buttons to share the page on social media. If you want to turn this feature off, use social-share: false. |
| nav-short | By default, the navigation bar gets shorter after scrolling down the page. If you want the navigation bar to always be short on a certain page, use nav-short: true |
| gh-repo | If you want to show GitHub buttons at the top of a post, this sets the GitHub repo name (eg. daattali/beautiful-jekyll). You must also use the gh-badge parameter to specify what buttons to show. |
| gh-badge | Select which GitHub buttons to display. Available options are: [star, watch, fork, follow]. You must also use the gh-repo parameter to specify the GitHub repo. |
| last-updated | If you want to show that a blog post was updated after it was originally released, you can specify an "Updated on" date. |
| layout | What type of page this is (default is post for blog posts and page for other pages). See *Page types* section below for more information. |

**Advanced parameters**

These are advanced parameters that are only useful for people who need very fine control over their website.

| **Parameter** | **Description** |
| --- | --- |
| footer-extra | If you want to include extra content below the social media icons in the footer, create an HTML file in the \_includes/ folder (for example \_includes/myinfo.html) and set footer-extra to the name of the file (for example footer-extra: myinfo.html). Accepts a single file or a list of files. |
| before-content | Similar to footer-extra, but used for including HTML before the main content of the page (below the title). |
| after-content | Similar to footer-extra, but used for including HTML after the main content of the page (above the footer). |
| head-extra | Similar to footer-extra, but used if you have any HTML code that needs to be included in the <head> tag of the page. |
| language | HTML language code to be set on the page's <html> element. |
| full-width | By default, page content is constrained to a standard width. Use full-width: true to allow the content to span the entire width of the window. |
| js | List of local JavaScript files to include in the page (eg. /assets/js/mypage.js) |
| ext-js | List of external JavaScript files to include in the page (eg. //cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.2/underscore-min.js). External JavaScript files that support [Subresource Integrity (SRI)](https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity) can be specified using the href and sri parameters eg. href: "//code.jquery.com/jquery-3.1.1.min.js" sri: "sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" |
| css | List of local CSS files to include in the page |
| ext-css | List of external CSS files to include in the page. External CSS files using SRI (see ext-js parameter) are also supported. |

**Page types**

* **post** - To write a blog post, add a markdown or HTML file in the \_posts folder. As long as you give it YAML front matter (the two lines of three dashes), it will automatically be rendered like a blog post. Look at the existing blog post files to see examples of how to use YAML parameters in blog posts.
* **page** - Any page outside the \_posts folder that uses YAML front matter will have a very similar style to blog posts.
* **home** - The home layout is meant to act as the homepage of your blog posts - it will display all your blog posts, sorted from newest to oldest. A file using the home layout must be named index.html (not index.md or anything else!).
* **minimal** - If you want to create a page with minimal styling (ie. without the bulky navigation bar and footer), assign layout: minimal to the YAML front matter.
* If you want to completely bypass the template engine and just write your own HTML page, simply omit the YAML front matter. Only do this if you know how to write HTML!

**Featured users (success stories!)**

Visit the [Official website](http://beautifuljekyll.com/examples) to see sample websites using Beautiful Jekyll.

If you'd like to showcase yourself and join this list, [upgrading to the Individual plan](https://github.com/sponsors/daattali/sponsorships?&tier_id=7362) will give you that publicity plus some other rewards!

**Next steps**

Congratulations on making it this far! You now have all the tools to easily build a beautiful website for free.

* After you get comfortable with the basics of writing in markdown, I suggest taking a look at this [sample post](https://beautifuljekyll.com/2020-02-28-sample-markdown/) and [the code that created it](https://raw.githubusercontent.com/daattali/beautiful-jekyll/master/_posts/2020-02-28-sample-markdown.md) to learn some more advanced tips about markdown.
* I **highly recommend** going over the [*Frequently Asked Questions*](https://beautifuljekyll.com/faq/) to find out answers to questions you may not even know you have. Every few months I suggest checking the [*What's New?*](https://beautifuljekyll.com/updates/) page to see if there are new features, and learn [how to update your site to the newest version](https://beautifuljekyll.com/faq/#updating) when it's time.
* You can also check out the [advanced installation methods](https://beautifuljekyll.com/getstarted/#install-steps-hard) that give you a little more control but are harder to use. Keep in mind that Beautiful Jekyll was primarily designed to be used as a GitHub theme, so you will not get any support if you choose one of the Ruby installation methods.
* Remember that Beautiful Jekyll is built on top of Jekyll. This means that reading through the [Jekyll documentation](https://jekyllrb.com/) will introduce you to many more features that you might find useful!

**Getting help**

Visit the [FAQ page](https://beautifuljekyll.com/faq) for answers to commonly asked questions.

**If you choose to**[**become a sponsor**](https://beautifuljekyll.com/plans/)**, you'll have access to my**[**office hours**](https://beautifuljekyll.com/officehours/)**where you can ask for help.** You can also use the [Discussions](https://github.com/daattali/beautiful-jekyll/discussions) area to try and get help from the community.

Beautiful Jekyll is used by 50,000+ people with wildly varying degrees of web skills, so it's impossible to answer all the questions that may arise. For any question that's not specifically related to Beautiful Jekyll and is more about Jekyll or web development in general, the answer can often be found on Google, in the [Jekyll documentation](https://jekyllrb.com/), or on the [Jekyll support forum](https://talk.jekyllrb.com/).

**Contributions**

Thank you to [all past contributors](https://github.com/daattali/beautiful-jekyll/graphs/contributors). If you find any problems or would like to contribute in any way, feel free to create a pull request/open an issue/send me a message.

You can also contribute by becoming an [official sponsor](https://github.com/sponsors/daattali/sponsorships?tier_id=39856) to help keep Beautiful Jekyll well-maintained!

**Credits**

This template was not made *entirely* from scratch. I'd like to give special thanks to [Jekyll Now](https://github.com/barryclark/jekyll-now) and [Bootstrap Clean Blog](https://github.com/IronSummitMedia/startbootstrap-clean-blog), from whom I've taken several ideas initially.

I'd also like to thank [Dr. Jekyll's Themes](https://drjekyllthemes.github.io/), [Jekyll Themes](http://jekyllthemes.org/), and another [Jekyll Themes](http://jekyllrc.github.io/jekyllthemes/) for featuring Beautiful Jekyll in their Jekyll theme directories.