

Build a Project Site Using GitHub Pages

1) Set up a GitHub account

Create your own GitHub account by going to <http://github.com>. Your username will be part of your site's URL, so choose wisely. You will be prompted to choose an account type: choose the Free account option.

2) Create a repository for your site

Create a new public repository for your project site. For example, if your account name is georgewashington and your project repository is lottery-analytics, the name of the repository will be: `georgewashington.github.io/lottery-analytics`

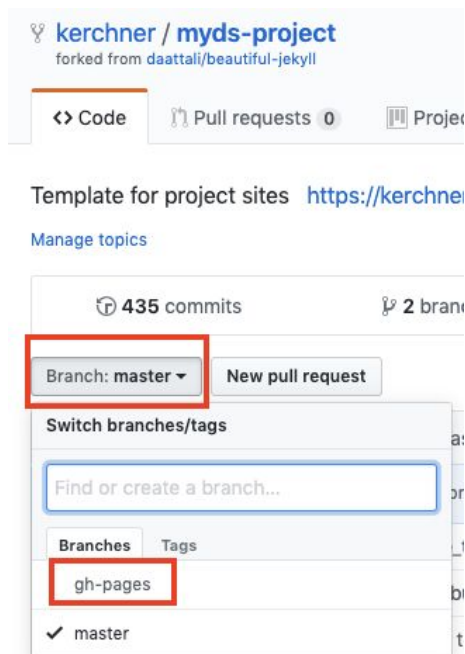
3) Copy the site template

At the bottom of the repository setup page, click the Import Code button. Import code from: <https://github.com/kerchner/beautiful-project>

After the import is complete, you should be able to visit `[username].github.io/[project-name]` and you will see your new web site.

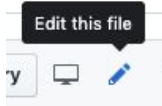
4) Personalize the template:

First, switch to viewing the **gh-pages** code branch in GitHub:

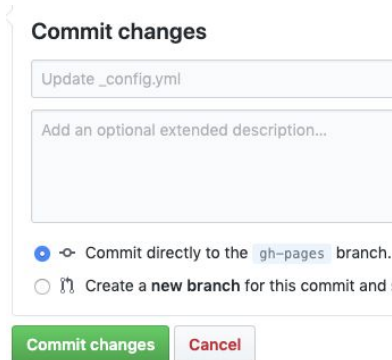


Edit _config.yml

This file has settings for your site, and uses "YAML" syntax. Make sure to indent with spaces, not tabs! Edit the file by clicking the pencil icon:



Then, make your changes, and scroll down to press the "Commit changes" button (optionally, add a brief commit message to note what you changed):

A screenshot of a "Commit changes" form. At the top, it says "Commit changes". Below that is a text input field containing "Update _config.yml". Underneath is a larger text area with the placeholder text "Add an optional extended description...". At the bottom, there are two radio buttons. The first is selected and labeled "Commit directly to the gh-pages branch.". The second is labeled "Create a new branch for this commit and :". At the very bottom, there are two buttons: a green "Commit changes" button and a red "Cancel" button.

Changes to make:

- title
- description
- url
- author name, email

Edit index.md

The home page is generated from index.md, which is a text file that uses "Markdown" syntax to control styling and formatting. Make some changes and commit the changes.

Here are some links for pages briefly listing Markdown syntax:

<https://kramdown.gettalong.org/quickref.html>

Add a new page

Copy the text of **md-example.md** (use the "Raw" button) and create a new page (using the "Create a new file" button) - make sure the new page's file name ends in ".md"

Now edit **_config.yml** and add a new menu item (or sub-menu item) in the **navbar-links** section that links to your new page.

Add an image

Browse to the `/img` folder. Use the "Upload files" button to upload a new image file.

Add a reference to the image in one of the pages or blog posts (in `/_posts`). You might want to follow the example in `md-example.md`.

Add a blog post

Notice how all of the blog posts in the `_posts` folder have file names that incorporate a date, like `2015-02-26-a-post-with-an-image.md`. Create a new blog post (by copying an existing one) and make sure to follow the same file naming scheme. The date controls the order the posts will appear.

Change a color

In the `/css` folder, colors, fonts, etc. are controlled in the `main.css` stylesheet. You can find a color picker online to generate hexadecimal color codes.

Include an interactive visualization (or any other HTML)

- Export your visualization as HTML.
- Upload your HTML file to the `_includes` folder
- Add the following to one of your pages where you'd like to embed the visualization. The page can be either Markdown or HTML (substitute the name of your HTML file):

```
{% include my_plot.html %}
```

Troubleshooting

- If there is a syntax error in one of your files that's preventing the site from building successfully, you can view the error message in your GitHub repository's Settings page (on the Options tab, scroll down to the GitHub Pages section). Usually GitHub will also send you an email with information about the error, including which line in which file is causing the error.
- Make sure you're changing files in the **gh-pages** branch of your repository. GitHub pages builds your site from this branch, not from master (unless you specifically configure it that way)
- If you're not seeing your change show up after a few minutes, and everything else looks good, it's possible that your browser is caching the page. You may need to clear your browser's cache to see the change.