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 <u>public</u> repository for your project site and choose the name. This will determine the URL of your site. You have two choices:
- Name the repository `[username].github.io`, to create a site with the URL `[username].github.io`
- Name the repository anything else. If, for example, your account name is `georgewashington` and your project repository is `lottery-analytics`, the URL of the repository will be: `georgewashington.github.io/lottery-analytics`

<u>Do not</u> check any of the boxes to initialize the repository with template files:

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
    This is where you can write a long description for your project. Learn more.

☐ **Add .gitignore**
    Choose which files not to track from a list of templates. Learn more.

☐ **Choose a license**
    A license tells others what they can and can't do with your code. Learn more.

Next, at the bottom of the repository setup page, click the **Import Code** button, and import code from another site.

...or import code from another repository
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code]

3) Select a template, and copy the site template
One site (among others) that includes some free Github Pages theme templates is https://jekyllthemes.io/github-pages-themes - currently 8 of these themes are free.

Copy the github URL of the template of interest (for example, `https://github.com/mmistakes/minimal-mistakes`). The rest of this document assumes that you imported from `https://github.com/kerchner/beautiful-project`, but if you used a different template, you can usually find customization instructions on the main page (README.md) of your new repository (i.e. at

`https://github.com/`*`youraccount`*`/`*`yourproject`*). Paste the URL into the box for "Your old repository's clone URL"

Import your project to GitHub
Import all the files, including the revision history, from another version control system.
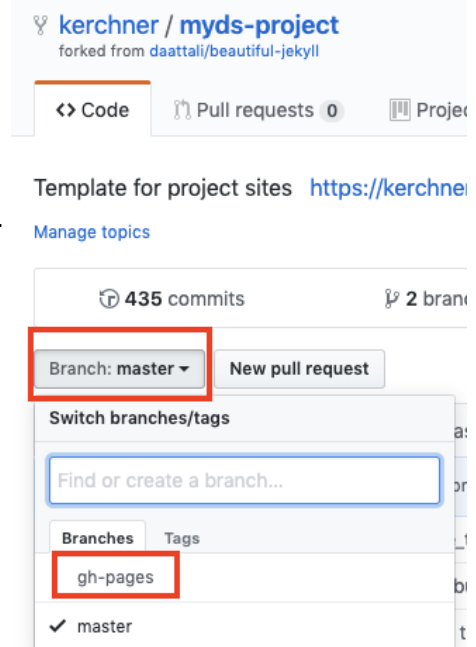
Your old repository's clone URL

    https://github.com/mmistakes/minimal-mistakes

Learn more about the types of supported VCS.

The import will take several minutes. After the import is complete, you should be able to visit `[username].github.io/[project-name]` and you will see your new site.
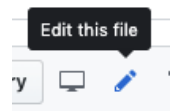
4) Personalize the template:
If the template you copied from has the site code in the **gh-pages** code branch, then switch to viewing the **gh-pages** code branch in GitHub. Many templates, however, have the site code in the **main** or **master** branch.

kerchner / **myds-project**
forked from daattali/beautiful-jekyll

<> Code   Pull requests 0   Proje

Template for project sites   https://kerchne

Manage topics

435 commits   2 bran

Branch: master ▾   New pull request

Switch branches/tags

Find or create a branch...

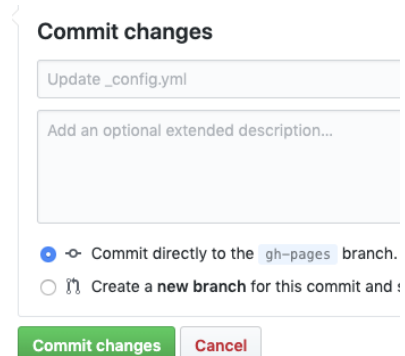Branches   Tags

gh-pages

✓ master

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:

Edit this file

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

Commit changes

Update _config.yml

Add an optional extended description...

Changes to make:
- title
- description

● ◦ Commit directly to the gh-pages branch.
○ Create a **new branch** for this commit and s

Commit changes   Cancel

- url
- author name, email

Edit a page

Try editing `md-example.md`, which is a text file that uses "Markdown" syntax to control styling and formatting. This file controls the content at [your site]/md-example/. Make some changes and commit the changes. You may need to wait a couple of minutes for the site to rebuild.

The home page is generated from `index.html`, which is an HTML file. If you are familiar with HTML, make some changes and commit the changes. Note that the lines with {{ }} and {% %} are "Liquid Templates" as used by Jekyll. For more information, see https://jekyllrb.com/docs/liquid/

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 `2021-11-04-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 (scroll down to the **Pages** tab on the left). 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.