

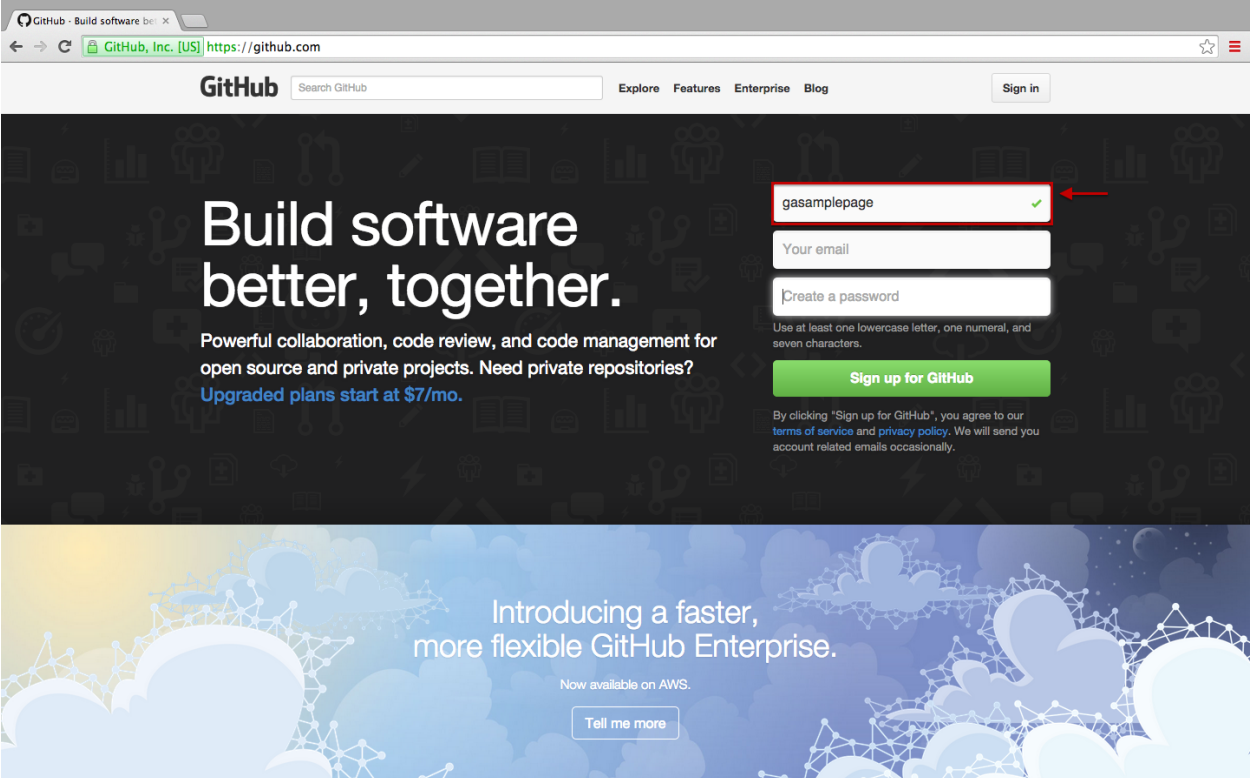
Getting Started with GitHub

GitHub enables you to host and share website files online, similar to the way Dropbox or Google Drive let you store and share other types of files. GitHub hosts the files in a remote server in the cloud and gives you a working URL so that you can share your finished site with other people.

Let's walk through this process:

Step 1: Create a GitHub account

To start, let's head over to GitHub.com. The first thing we'll do is create a username by clicking on the field at the top right of the screen.



The screenshot shows the GitHub website's sign-up interface. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Features', 'Enterprise', and 'Blog'. A 'Sign in' button is on the right. The main section has a dark background with the text 'Build software better, together.' and a sub-header: 'Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo.' To the right is a sign-up form with three input fields: 'Username' (containing 'gasamplepage' with a green checkmark), 'Your email', and 'Create a password'. Below the password field is a note: 'Use at least one lowercase letter, one numeral, and seven characters.' and a green 'Sign up for GitHub' button. A red arrow points to the username field. At the bottom, there's a light blue banner for 'Introducing a faster, more flexible GitHub Enterprise.' with the text 'Now available on AWS.' and a 'Tell me more' button.

The username you enter here will be part of the github URL for every site you host there, so choose something simple and easy to remember. Most people use their name, a pen name, or their company's name. To simplify things, try making your username one word and all lowercase.

For this guide, we're going to use the name "gasamplepage." Please pick whatever name you like, for example "johnsmith123." After selecting our username, we'll add our email and make up a password. Then we'll click "Sign up for GitHub."

Clicking that button brings us to the following page:

The screenshot shows the GitHub 'Welcome to GitHub' page for a user named @gasamplepage. The page is divided into three steps: Step 1 (Completed: Set up a personal account), Step 2 (Choose your plan), and Step 3 (Go to your dashboard). Under 'Choose your personal plan', there is a table with five plans: Large (\$50/month, 50 private repos), Medium (\$22/month, 20 private repos), Small (\$12/month, 10 private repos), Micro (\$7/month, 5 private repos), and Free (\$0/month, 0 private repos). The 'Free' plan is highlighted with a blue background and a red box around the 'Chosen' button. To the right of the table, a box titled 'Each plan includes:' lists features: Unlimited collaborators, Unlimited public repositories, Free setup, SSL Protection, Email support, Wikis, Issues, Pages, & more. Below the table, there is a note about charges in US Dollars and a checkbox for 'Help me set up an organization next'. At the bottom, there is a green 'Finish sign up' button with a red arrow pointing to it.

Plan	Cost	Private repos	
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	<button>Chosen</button>

Charges to your account will be made in US Dollars. Converted prices are provided as a convenience and are only an estimate based on current exchange rates. Local prices will change as the exchange rate fluctuates. Don't worry, you can cancel or upgrade at any time.

☐ Help me set up an organization next
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees. [Learn more about organizations.](#)

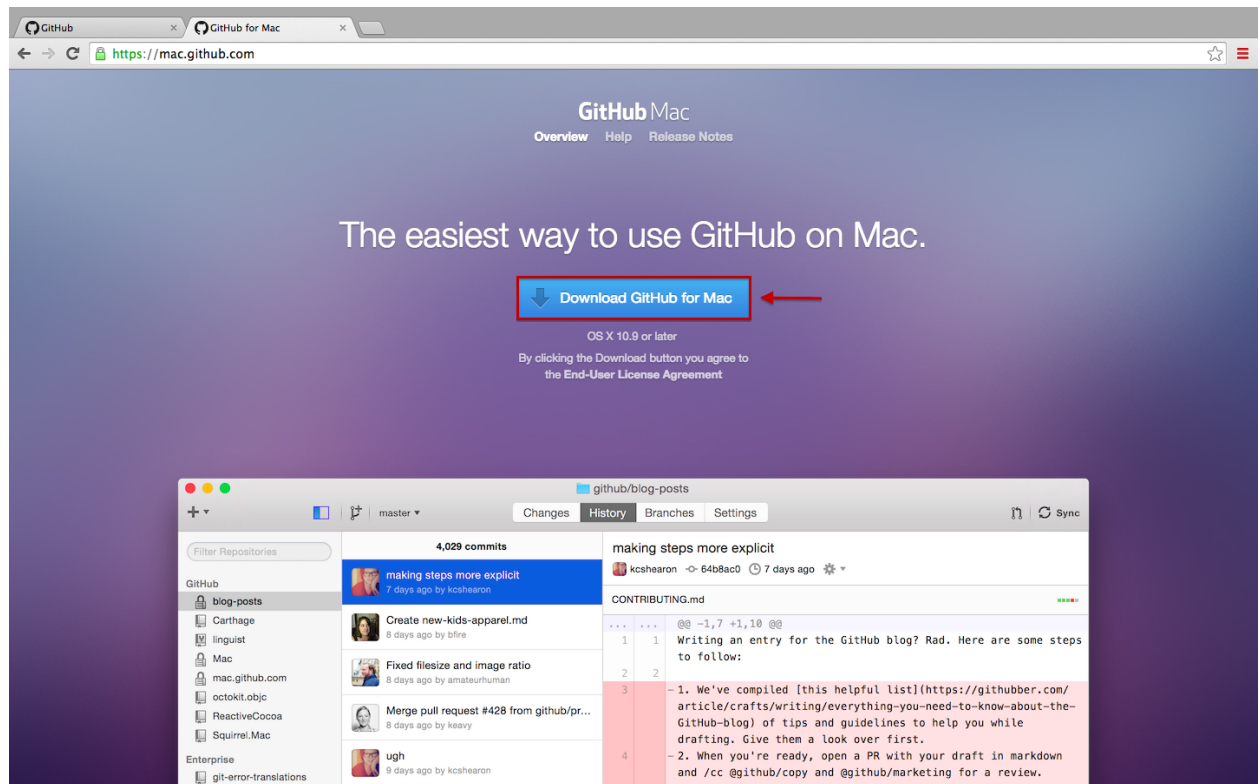
Finish sign up

All we need to do on this page is make sure that “Free” is chosen as our personal plan. It should be pre-selected; however, if it’s not, make sure to click on it. Then, press the “Finish Sign Up” button.

GitHub will send you an email asking you to verify your account. Take a moment to do this now. If you do not verify your email address, your page will not work.

Step 2: Download the GitHub App

Now that we've finished creating a GitHub account, we'll need to download the GitHub app. We're using a Mac, so we'll go to mac.github.com. If you're using a PC, go to windows.github.com.



From here, click on Download GitHub. Once it's finished downloading, add it to the Applications folder. Then, we'll open our new GitHub app.

Step 3: Logging into the GitHub App

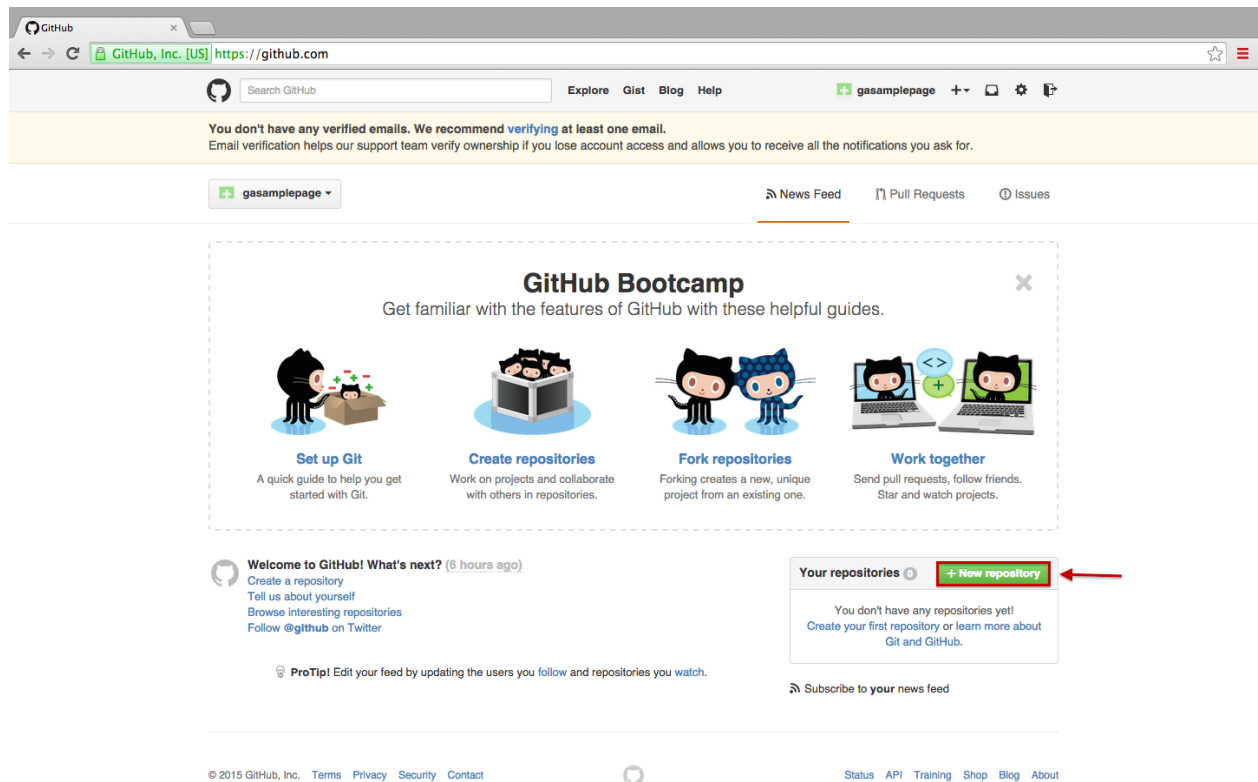
Let's login to the app with our new GitHub account. If this is the first time you've had GitHub on your computer, a login screen will automatically pop up. You'll log in and click "continue." Two additional pages will pop up, but you can ignore them for now. Just click "continue" again, then click "done."

If the computer you're using has had GitHub before, you'll have to manually log in. Just click on the "GitHub" tab, then click on "Preferences." Click on "Accounts," then log in.

But, before we can do anything in the GitHub app, we'll first need to go back to github.com and create a new folder to house all our files.

Step 4: Creating a New Repository

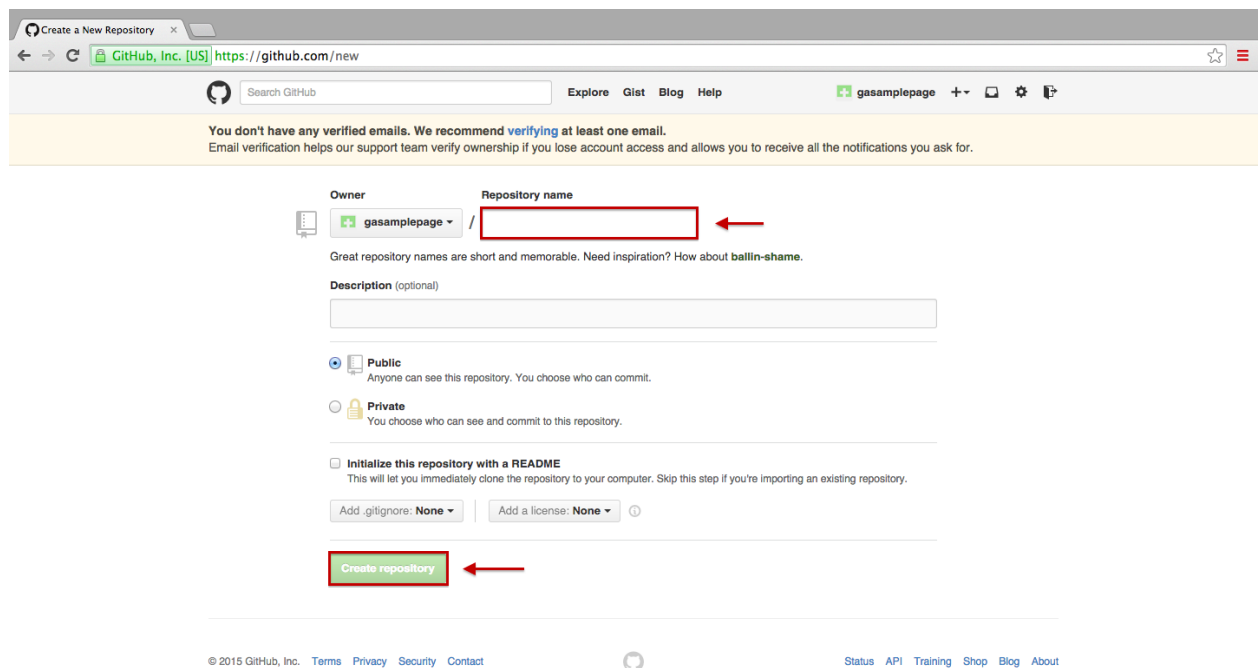
In GitHub, the public folder in the cloud where you can store, update, and share all the files that make up your website is called a “repository”. In order to create a “repo” (as the cool kids say), click on this green button that says “New Repository.”



Step 5: The Really Important Step

Clicking that button should bring you to the new repository creation screen, which should look something like the following image.

The first step we'll take on this page is to name our new repository. **This is a very crucial step in the process, so please don't rush ahead.**



Create a New Repository

Search GitHub

Explore Gist Blog Help

gasamplepage

You don't have any verified emails. We recommend [verifying](#) at least one email. Email verification helps our support team verify ownership if you lose account access and allows you to receive all the notifications you ask for.

Owner: gasamplepage / Repository name:

Great repository names are short and memorable. Need inspiration? How about [ballin-shame](#).

Description (optional):

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** Add a license: **None**

Create repository

© 2015 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

The first part of your repository name needs to be **exactly the same** as the username you created earlier. Then, you **must** follow that name with “.github.io”.

Since we used “gasamplepage” as our username for this written guide, our repository **must** be written **exactly** as follows, or our page will not load properly later.



Owner: gasamplepage / Repository name: gasamplepage.github.io ✓

So, if your username was “johnsmith123” you would want to name your repository “johnsmith123.github.io”.

Double-check everything to make sure it's accurate. After naming our new repository, we don't need to adjust any other settings, so we'll just click “Create Repository.”

Step 6: Glad We're Done with Step 5!

Yay! We made a new repository. Our next big step is to add our website's files to it. To get started, click on the "Set up in Desktop" button. This brings us back to the GitHub App we downloaded earlier.

The screenshot shows the GitHub repository page for `gasamplepage/gasamplepage.github.io`. The browser address bar shows the URL `https://github.com/gasamplepage/gasamplepage.github.io`. The repository name is `gasamplepage / gasamplepage.github.io`. The page displays a yellow banner about email verification. Below the repository name, there are buttons for 'Unwatch' and 'Star'. The main content area shows 'Quick setup — if you've done this kind of thing before' with a red arrow pointing to the 'Set up in Desktop' button. Other options include 'HTTPS', 'SSH', and 'Import code'. The 'Set up in Desktop' button is highlighted with a red arrow. Below this, there are instructions for creating a new repository on the command line and pushing an existing repository from the command line. The 'Import code' button is also visible. On the right side, there are links for 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'.

gasamplepage/gasamplepage: x
← → ↻ GitHub, Inc. [US] https://github.com/gasamplepage/gasamplepage.github.io

This repository Search Explore Gist Blog Help gasamplepage + - ⚙️ 🔑

You don't have any verified emails. We recommend [verifying](#) at least one email.
Email verification helps our support team verify ownership if you lose account access and allows you to receive all the notifications you ask for.

gasamplepage / gasamplepage.github.io Unwatch 1 ★ Star 0

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) [https://github.com/gasamplepage/gasamplepage.github.io.git](#)

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# gasamplepage.github.io" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/gasamplepage/gasamplepage.github.io.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/gasamplepage/gasamplepage.github.io.git
git push -u origin master
```

...or import code from another repository

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

[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

[Code](#)
[Issues](#) 0
[Pull Requests](#) 0
[Wiki](#)
[Pulse](#)
[Graphs](#)
[Settings](#)

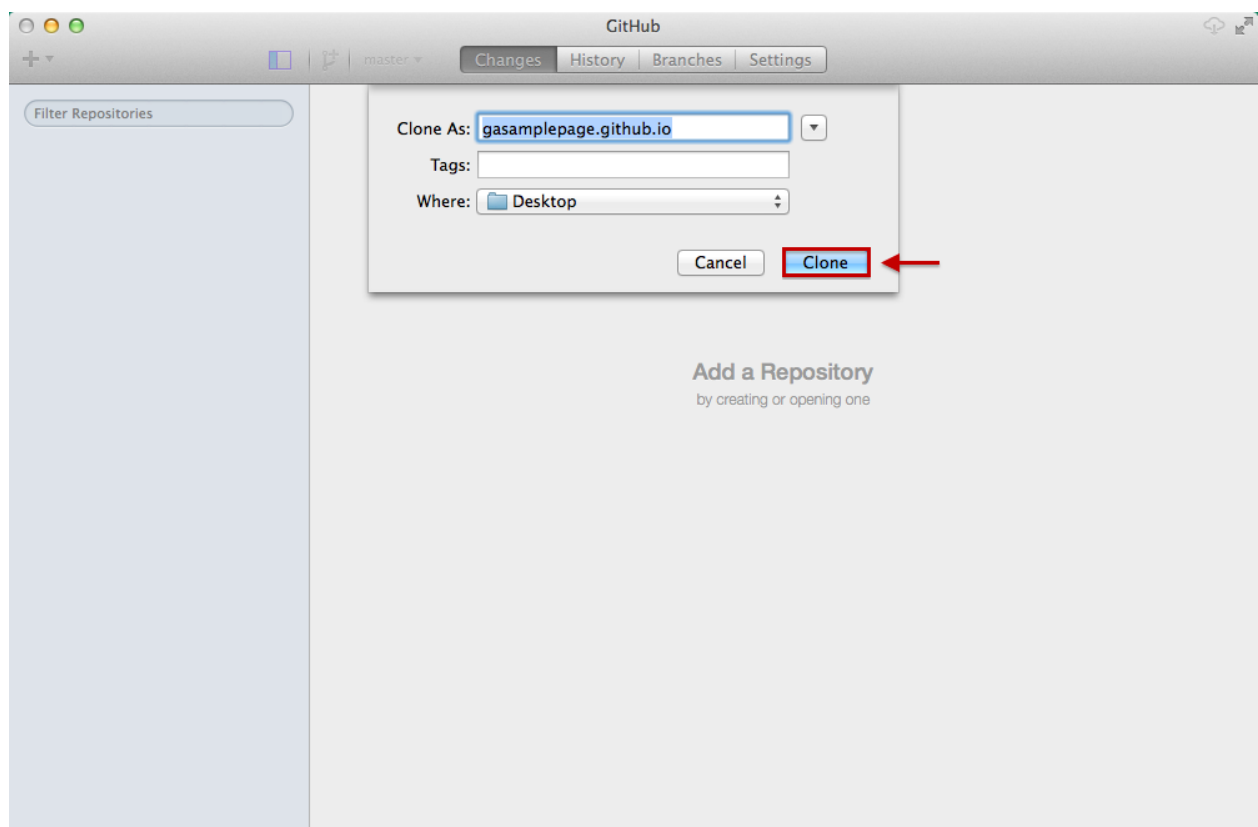
Step 7: “Cloning”our Repository

When you re-open the app, a message should pop-up that says “Clone As” with the name of our new repository pre-entered. Basically, “cloning” means that we’re copying our GitHub repository from the cloud and saving it as a local folder on our computer.

Doing this will allow you to add and edit files locally, then sync up those changes to your GitHub repository when you’re done.

Save this new repository folder somewhere that will be easy to find, like your Documents folder or your Desktop. Then, click “Clone.”

This will produce a local folder with the name written in the “Clone As” field, and it will be located wherever you decided in the “Where” field.

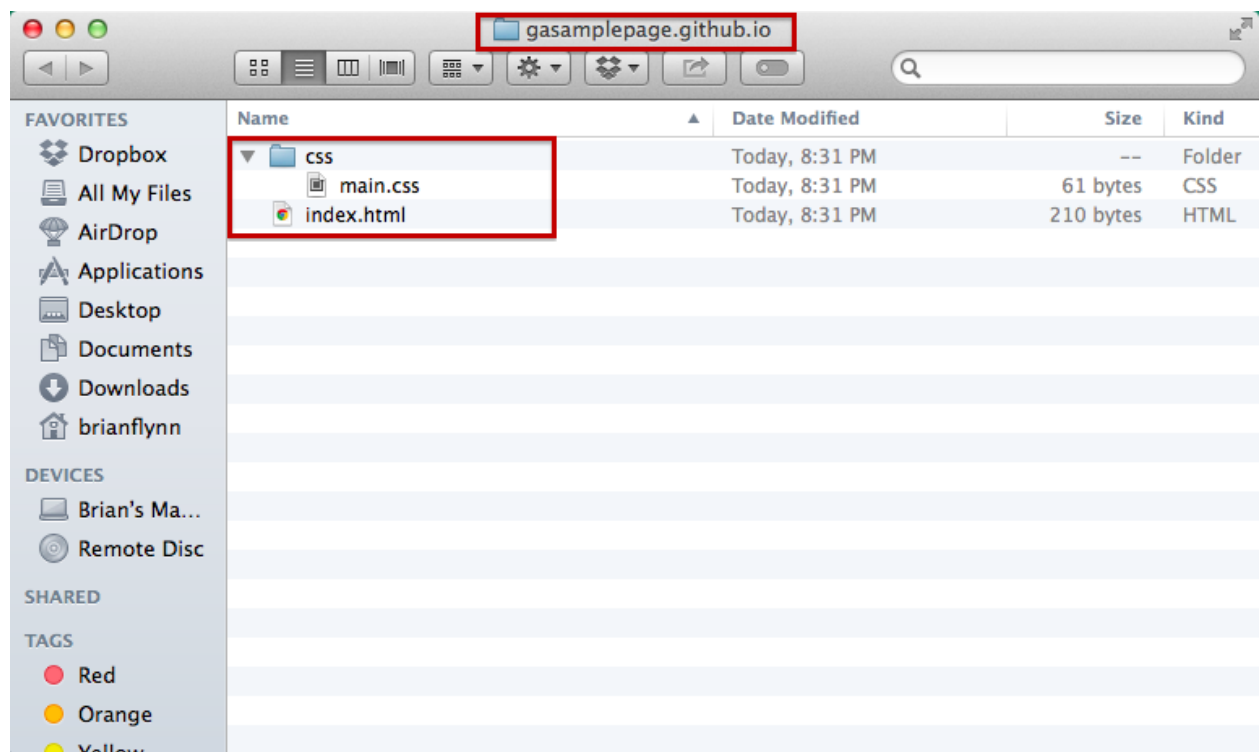


Step 8: Adding Files to our Repository

Shrink the GitHub app and find the cloned local folder you just created. If you saved it to your desktop, then go to your desktop. If you saved it to your documents folder, go there. etc.

Once you find your new cloned folder, put the HTML and CSS files making up your web page inside of it, preserving the directory structure.

Once you do that, go back to the GitHub App.



Step 9: Syncing our Repository — Version A

Even though we've added files to our local repository folder, we still need to sync this change with the corresponding GitHub repository in the cloud in order to put our site up on the web. So, back in the app, you'll need to do the following:

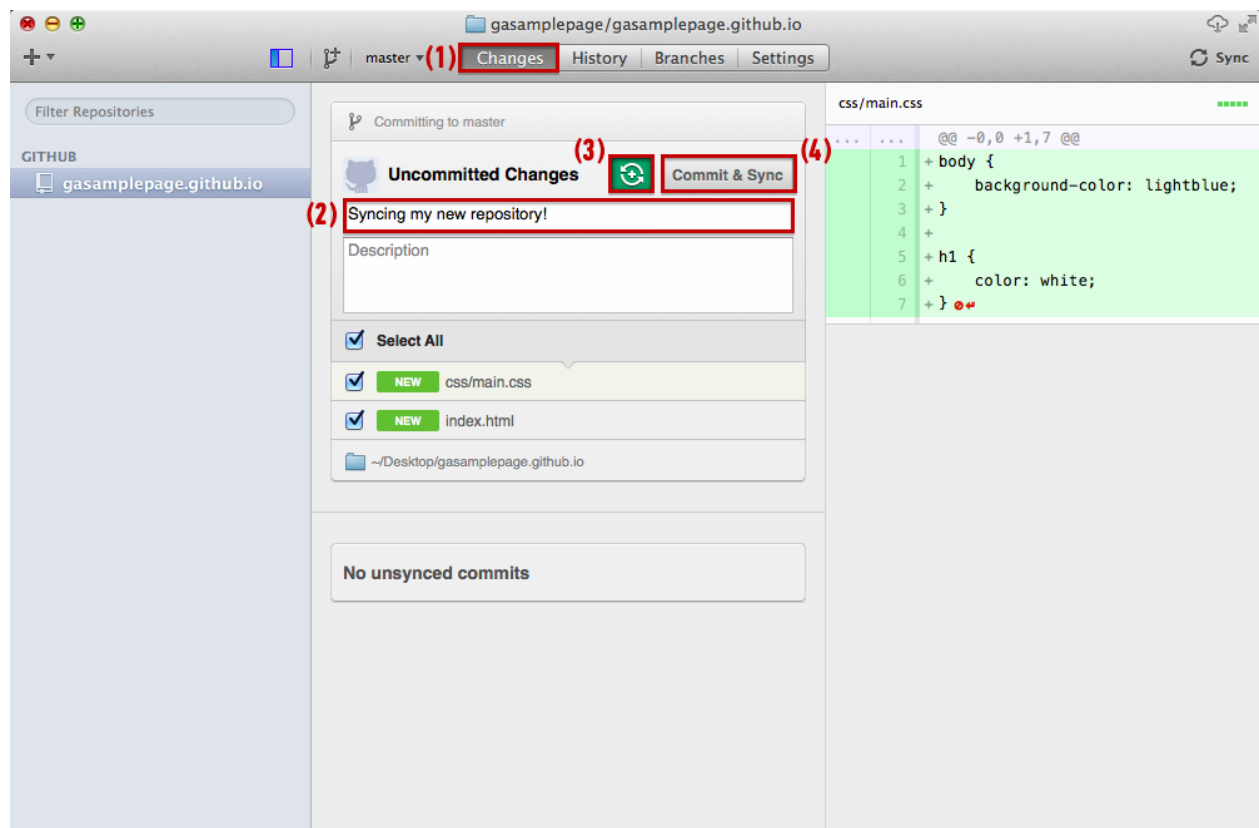
*** (Depending on your computer's software, the layout of your Github app may look one of two different ways. Don't worry, neither way is particularly better than the other. If the image below doesn't look like what's on your screen, check Step 9, Version B.) ***

(1) Make sure you're in the section labelled "Changes."

(2) In the summary field, type a message describing this update. We'll write "syncing my new repository!" Later, when we're making updates, our summaries might look like "changed the background color and logo size per client's request."

(3) Then, if it's not already pre-selected, click on the button that looks like a plus sign surrounded by two arrows. It should be dark green when selected.

(4) Click on the "Commit and Sync" button.



Step 9: Syncing our Repository — Version B

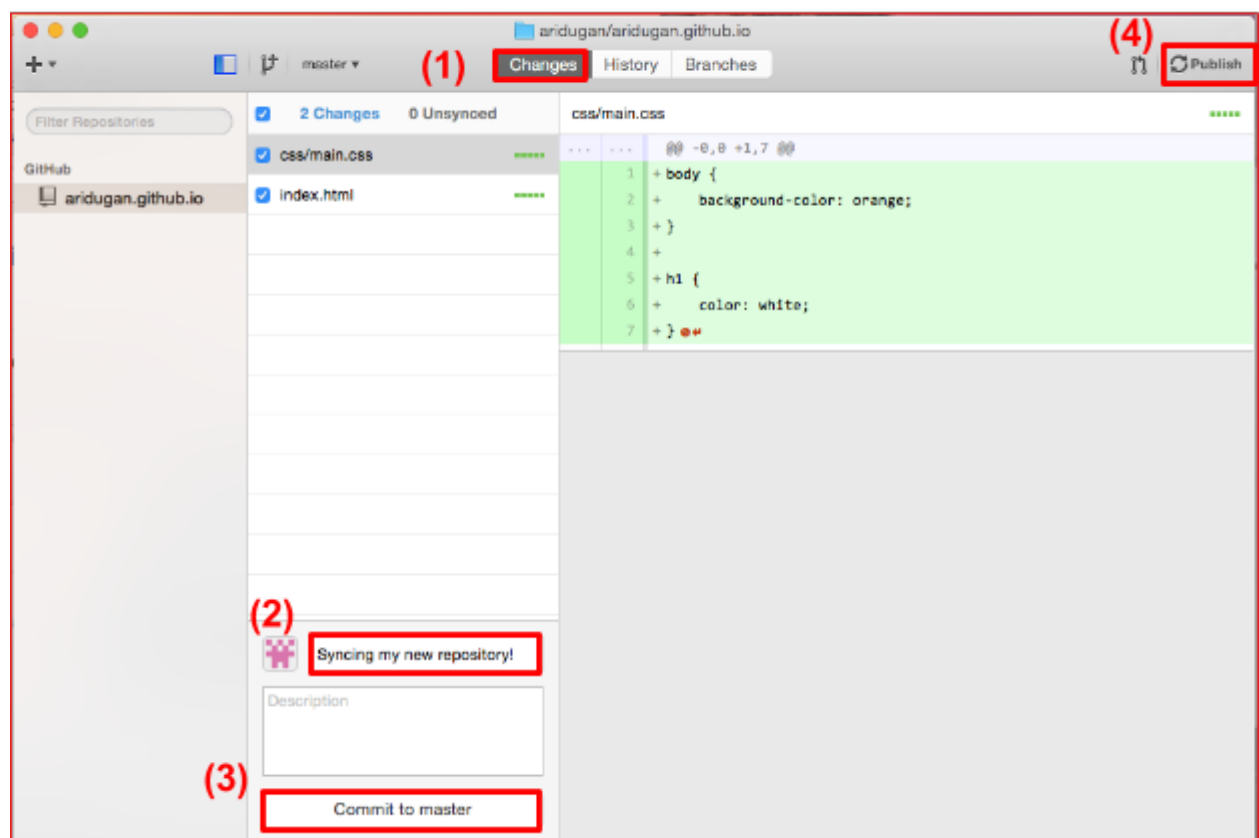
Depending on your software, the layout of your GitHub app may look like this image instead of the one shown in Step 9, Version A. Don't worry, for our needs, the difference is purely cosmetic. Just take the following steps, and you'll have your page online in no time!

(1) Make sure you're in the section labelled "Changes."

(2) In the summary field, type a message describing this update. We'll write "syncing my new repository!" Later, when we're making updates, our summaries might look like "changed the background color and logo size per client's request."

(3) Press the "Commit to Master" button.

(4) Click on the "Publish" button.



Step 10: Loading Your Page (aka OMG We're Finally Done!)

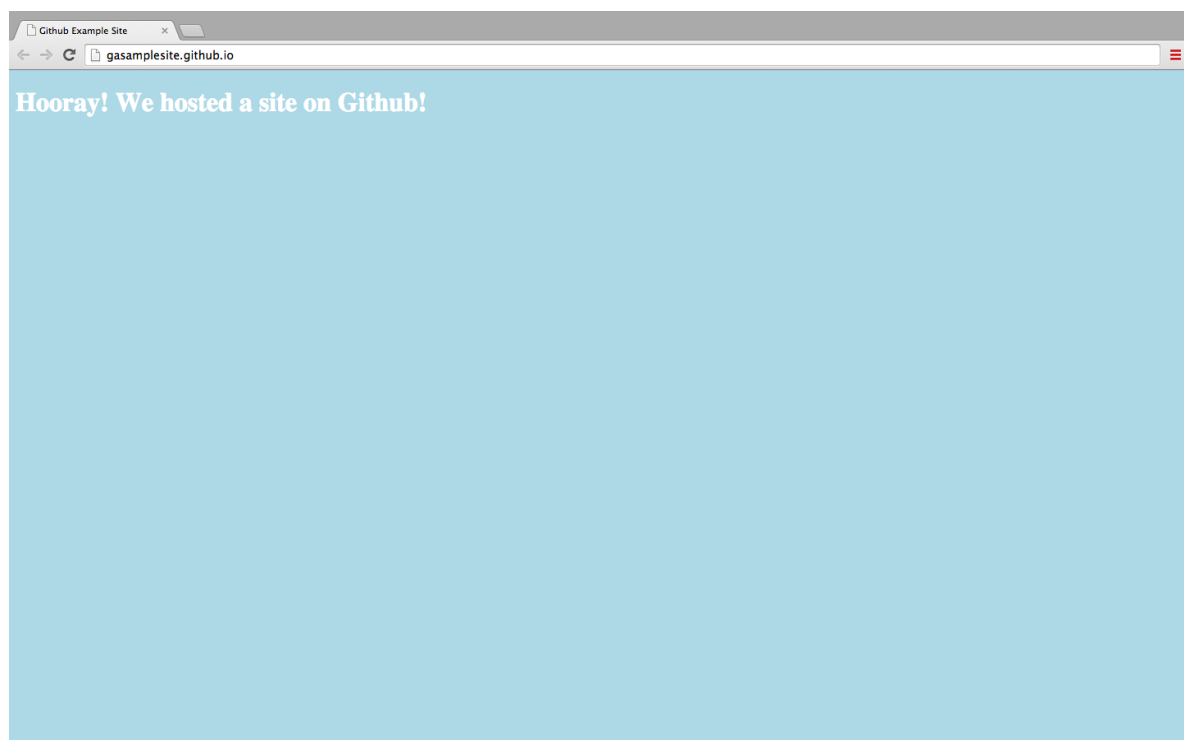
Our next step is to go back to our web browser and visit our newly hosted page.

Unfortunately, the very first time you commit and sync a new repository, it can take anywhere between 0 and 30 minutes for GitHub to prepare your files. Every other time you sync, it'll be pretty much instantaneous.

If you try to visit your site before it's ready, you'll get an error message telling you it's not ready yet. If that happens, try waiting a little longer. If you've waited more than 30 minutes, and it's still not working, double check the spelling in your username, repository, and URL.

If you've entered everything correctly and given GitHub enough time to prepare your new repository, and it *still* doesn't work, please don't smash your computer. It's okay. It's very possible you did everything correctly and there was some weird error behind the scenes. If you got stuck here or anywhere in this process, reach out to your producer and mentor and let us know how you got tangled. We're here to help you out!

However, if everything worked correctly, all you have to do is type in the name of your repository. So, we'll type in gasamplepage.github.io and **BAM!!!** We have a website!



If you ever want to update or adjust your page in any way, just change the files in your cloned local repository folder and re-commit and sync them in the GitHub app.

The End