

# Making a website for your HTML, JS, CSS or p5.js files

We are going to make a website using something called GitHub pages to make a website. Signing up for a GitHub account and subsequently creating a GitHub pages URL will allow you to upload your HTML, JS & CSS files online, so that you can access your website from anywhere.

## Bit of background (Feel free to skip)

GitHub is a website that stores what are called *repositories*. Repositories contain code, very much just like a folder on your computer. *Git* is what is called a *version control system* and to us what it means is that when you save files and *commit them* it will also save the previous version of your files too. This is very popular with a lot of people in the world of tech, because if something goes wrong, you can always go back to when it did work!

## Step 1: Signing up for GitHub

Go to <https://github.com> and sign up for an account, the form should be on the homepage. There will be a few extra forms when you signup, you can skip through these.

**Very important!!!!1!!1!**

The way that GitHub pages work is that in the end, your website will have the name of your username in it (.github.io). For instance, if my username was 'jonny' my GitHub pages URL will be <https://jonny.github.io>. So make sure you choose your username carefully!


## Step 2: Creating a repository

Now we have to create a repository. On the left hand side there should be a link that says **Create a repository**. Click this link and it should take you to this page:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner **Repository name \***

 cti-test /

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

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** ⓘ

### Very important!!!1!!!1!

This is where we need to set our repository name to equal exactly this structure: .github.io. So again if my username is 'jonny' in the **Repository name** field I would insert 'jonny.github.io'.

Keep the button marked as 'Public' and everything else as default and click **Create Repository**

## Step 3: Uploading your code

Now we need to upload our files to the Github repository. Hopefully after you created your repository you should see the below page. What we want to do is click the link that says **uploading an existing file**

Search or jump to... Pull requests Issues Marketplace Explore

ctl-test / ctl-test.github.io Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

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

Set up in Desktop or HTTPS SSH `https://github.com/ctl-test/ctl-test.github.io.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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

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

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

After that, if you drag your HTML, JS and CSS files (if you have them) onto the upload section, they should upload to your repository. **BUT!** You need to 'commit the changes' for them to be saved.

## Step 4: Committing your changes

This section is also relevant if you want to update your files too. Everytime you make changes to the files, you have to 'commit them'. To the Git repository, that basically means, storing these new files, committing them to memory.

Github makes this part quite easy. If you want to add a message you can do, but if not, all you have to do is click **Commit changes** and your files will be stored in the git repository.

**Commit changes**

This is me adding a message

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

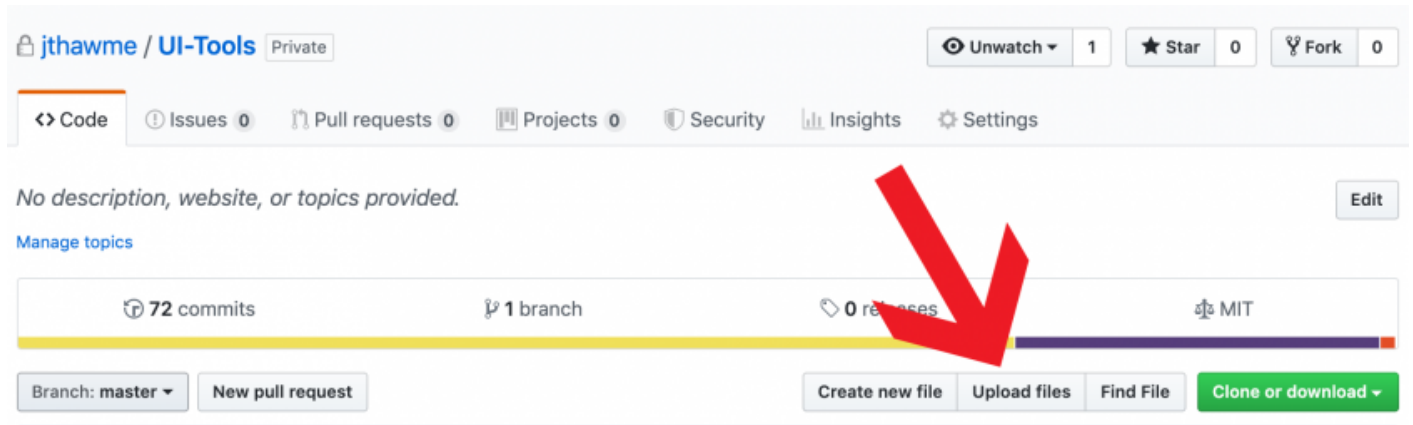
## Step 5: Viewing your website

*It may take a couple of minutes* but, after you have uploaded your code, you should be able to see your website running at `.github.io` (replacing  with your actual username).

Congratulations!

## Step 6: Updating files

To update files so that you make your newest files available on the internet, you need to follow the same as **Steps 3 & 4**. The only difference is you now click that says **Upload files** which you can find where, below



## Optional steps

Everything in this section isn't necessary but it might help you out, to work faster and better!

## Step 7: Using a Git application

We can download github's application to help speed up committing files. You can download the program here: <https://desktop.github.com/>. Once you've downloaded it, you need to log in and set up your name and e-mail address. Then you will see all of your repositories on your account.



# Let's get started!

Add a repository to GitHub Desktop to start collaborating



## Your Repositories

 [ctl-test/ctl-test.github.io](#)



Clone a Repository from the Internet...



Create a New Repository on your Hard Drive...



Add an Existing Repository from your Hard Drive...



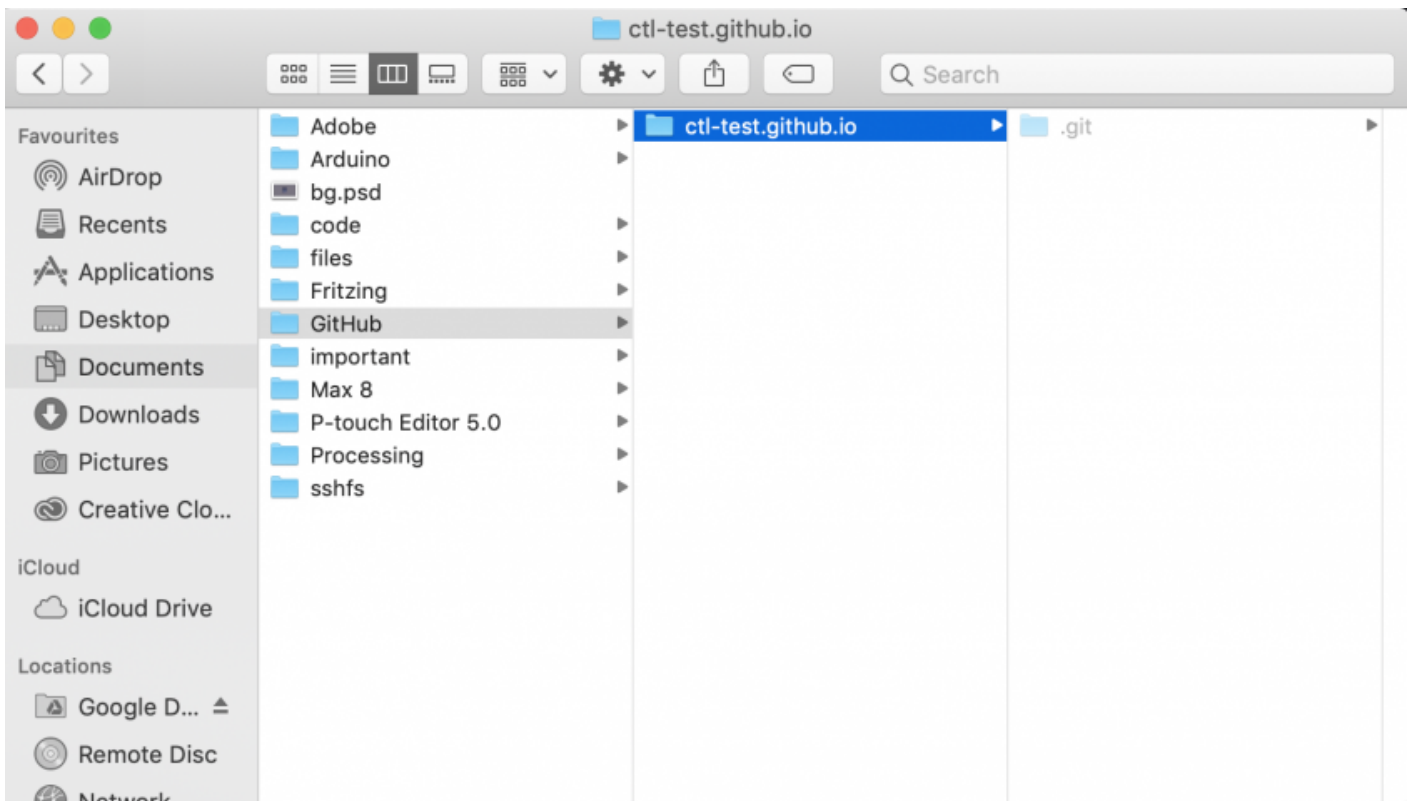
Clone [ctl-test/ctl-test.github.io](#)



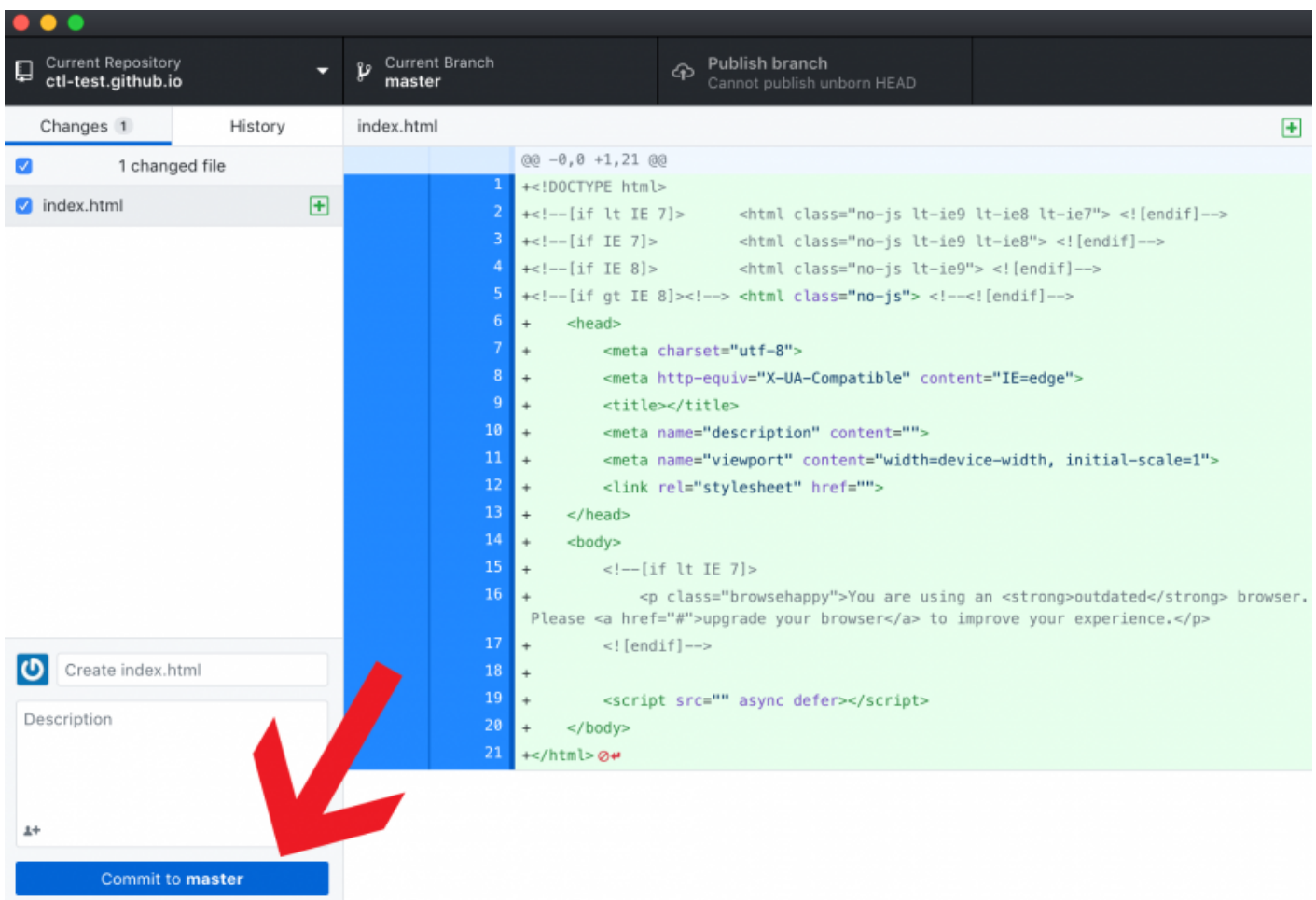
**ProTip!** You can drag & drop an existing repository folder here to add it to Desktop

If you click your .github.io link and then the blue 'clone' button beneath it, it will download the repository to your computer. But the more important part now is that from that folder it downloaded, any changes you make it will watch them so that you can upload them back to GitHub.

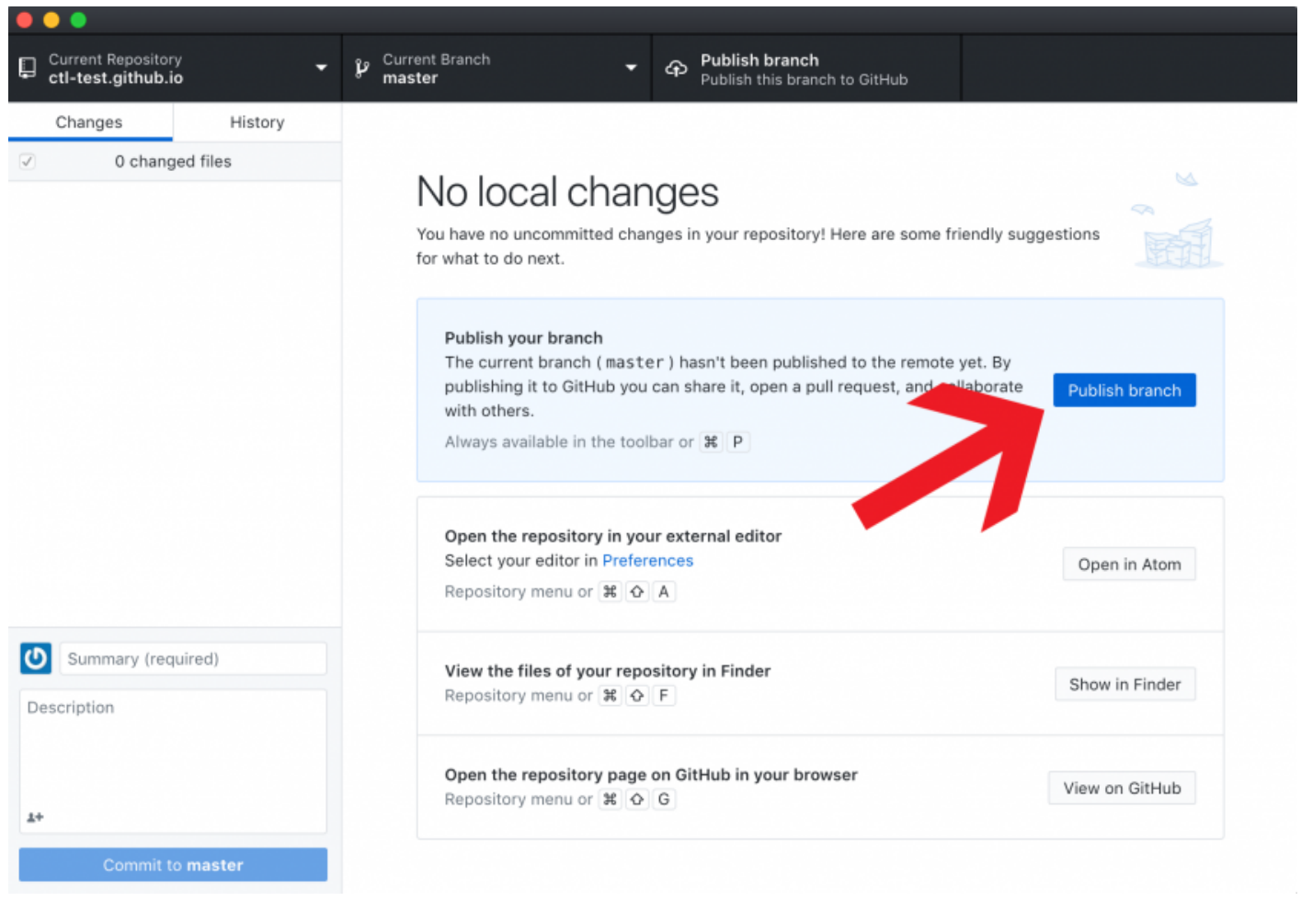
By default it will download the repository to this location: **Documents/Github/REPO\_NAME**



When you make changes to your files, if you check Github Desktop again, you should see that its detected changes in your files.



After you commit them in Github Desktop, in the bottom left of the screen (similar to Step 4) you can then publish the changes to your website



Revision #2

Created 6 June 2019 14:54:10

Updated 6 June 2019 15:07:22