



Using GitHub

UPLOADING YOUR WEBSITE TO GITHUB

Most of the coding work you produce this term will be uploaded to GitHub. GitHub enables you to share your website with your clients, friends, family and anyone else that needs to review your work. In order to complete this step, you need a GitHub account and the free GitHub Desktop software. If you do not have both of these, please stop now and get them.

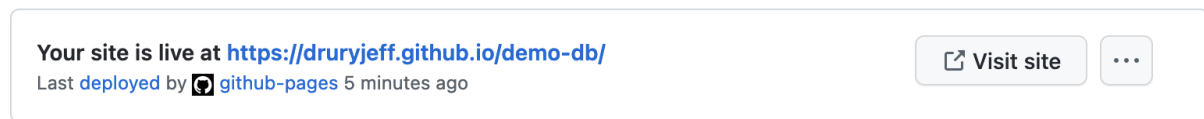
1. Go to <https://github.com/> and log in using your username and password.
2. Next, open the GitHub Desktop application on your computer.
3. If you have not opened this application before, GitHub Desktop will ask you to enter your GitHub username and password. Follow the instructions for adding this information.
4. You can ignore the pop-up window that opens initially and just select File > New Repository from the top menu to set up a new GitHub repository.
5. In the window that opens after selecting File > New Repository in the previous step, enter a name for your repository. Use your project name (for example, "bike-pvd").
6. You can skip the Description field or add a description of your website.
7. In the Local Path field, click the "Choose" button to locate your project files. When you locate your project folder, double-click the folder to open it. (You want to be looking at the contents of your folder – the view where you can see your HTML files, "css" folder, etc..) Click "Open".
8. You can ignore the next few fields in the setup window ("Initialize", "Git Ignore" and "License").
9. Click the "Create Repository" button.
10. Next, go out to your Desktop or Documents and navigate to your project folder. You will see a new folder (with the name added in step 5) alongside your existing files. This is your new GitHub repository. Select all of your files – not including the new folder – and drag the files into the new folder.
11. Next, go back to the GitHub Desktop application. On the left side of the interface, you'll see that your repository now has your files in it. This is the result of what you did in the previous step.
12. Add the phrase "Initial Commit" to the lower left side of the interface (replacing the word "Summary"). Click the "Commit to Main" button. Wait a second while the command processes.
13. Next, click the "Publish repository" button at the top right of the interface. In the window that opens, uncheck "Keep this code private" and then click the "Publish Repository" button. Wait a minute while the changes take effect. You'll see a spinning icon while you wait.
14. Go back to the GitHub website (<https://github.com/>). Go to "Your Repositories" (from the user menu at the top right of the page).
15. Click on the name of the repository you just made. You should now see all of your files listed.

16. Click on the dropdown menu labeled “main” (located just above your files). In the field that opens, type “gh-pages” (no quotes, no spaces, all lowercase) and then click the “Create branch: gh-pages” button. Wait a minute while the page refreshes.

Note – You just created a new “branch” with the same files you initially committed. This new branch will be used to display your files as a functioning web site.

17. Click on the Settings button at the top right of the page. On the left side of the page, click the “Pages” link. On the “GitHub Pages” page, you should see a rounded-corner, gray-bordered box with a blue link containing the address of your website. Click the “View site” button to see your live website posted on GitHub Pages. This is the location of your web site.

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.



Important – If you do not see this gray-bordered box, refresh the page a few times. Sometimes it takes a few minutes, or in rare cases a few hours, to publish.

EDITING YOUR GITHUB PAGES WEBSITE

So far, you have uploaded your work to GitHub. You have created a new GitHub repository and “pushed” your work to it. What happens when it comes time to edit these files?

1. Important – Before you open Sublime Text to edit your files, open the GitHub Desktop application.
2. Select the “gh-pages” branch from the “Current Branch” tab at the top of the interface.

Note – If you do not see a “gh-pages” branch, and you are sure you have created one on the GitHub website, click the “Fetch Origin” button at the top of the interface. This will refresh the application and show the “gh-pages” branch.

3. Once you’ve selected the “gh-pages” branch from the GitHub Desktop application, you can go back to Sublime Text and make your changes, edits and revisions. Edit the site in Sublime Text as you normally would do.
4. When you are done editing your HTML and CSS in Sublime Text, come back to the GitHub Desktop application. You will notice a number next to the “Changes” tab under the “Current Repository” tab. This indicates the number of changes you made to your files.
5. Next, add a note summarizing your edits – something simple, yet descriptive like “Changed the background image for the header” or “Added a new services page” or “Changed the main typeface” – and click the “Commit to gh-pages” button.
6. Last, click the “Push origin” button at the top of the page. This will upload your changes to the main GitHub website. In the process, it will update your web page/website online.

Important – If you go out to the browser and do not see your site edits right away, come back in five minutes or so and check it again. Sometimes it takes a few minutes, or, in some cases, a few hours, to update.