

Lesson

Tuesday

# Introduction to Programming

## (/introduction-to-programming)

### / Git, HTML and CSS (/introduction-to-programming/git-html-and-css)

### / GitHub Pages

Text

Cheat sheet

It's more fun to make web pages when we can show them off to our friends. There are a lot of ways to put our pages online. One of the easiest is to utilize GitHub Pages to automatically deploy our projects online for free.

First, make sure to have a GitHub account set up. The awesome thing about GitHub pages is that you get one main site per account and as many project pages as you would like. That means that when you are ready to create a portfolio of all of your coding work, you can have a main page all about you and a page for each of your coding projects that you do here at Epicodus or at home!

## Project Structure

A quick note about structuring projects before we begin. When we start a new project, it is a good idea to create a folder to hold the HTML file and the `css` and `img` folders. So far, we have been using a descriptive name for our HTML file, such as `favorite-things.html` or `my-first-webpage.html`. Often, though, you'll want to give the descriptive name for your folder, and name the web page's main HTML file `index.html`. As we start to add pages to our web site, the `index.html` file will serve as the home page of our site.

For this lesson, we will be putting our `favorite-things.html` file on GitHub Pages. Let's go in the Terminal to our `my-first-webpage` project folder. **Since we want `favorite-things.html` to be the home page of our site, we need to rename it `index.html`. GitHub Pages won't recognize it as the main page if we don't.**

```
$ mv favorite-things.html index.html
```

The `mv` command in the Terminal moves the content of the `favorite-things.html` file into a new file called `index.html`.

Now that we've made this change, let's commit this new change and push it to GitHub.

```
$ git status
$ git add .
$ git commit -m "Change name of favorite-things.html file to index.html."
$ git push origin main
```

## Adding a Branch for `gh-pages`

In order for GitHub Pages to work, we need to create a branch called `gh-pages`.

First we will create the new branch and switch into it by running the commands:

```
$ git branch gh-pages
$ git checkout gh-pages
```

# Pushing Branches to Github

Now that we are in the `gh-pages` branch, let's push this new branch to GitHub:

```
$ git push origin gh-pages
```

## Accessing our Web Page

Now our project should be available for viewing at *my-github-username.github.io/repository-name*. So, for instance, if our GitHub username was *sample-epicodus-student*, and our repository was named *my-first-webpage* we would navigate to *sample-epicodus-student.github.io/my-first-webpage*. Check out the URL that corresponds with your GitHub username and repository name (**Note:** Repository names are case-sensitive, use the same capitalization pattern as you did when naming the repo!)

You can also set up the GitHub Pages webpage for your project in the browser by going to your project's repository, clicking the branch drop down on the left, typing `gh-pages` and selecting `Create branch: gh-pages`. Done!

## Updating GitHub Pages

Note that if you make further changes to your project, you should do so on your `main` branch rather than on your `gh-pages` branch. To get back to your main branch, `git checkout main`. We'll learn in the future about merging branches, but for now the easiest way to update your `gh-pages` branch is to delete it on GitHub and then recreate it based on the updated main branch.

You can also check out GitHub's documentation (<https://pages.github.com/>) on deploying to GitHub pages. You'll see it's very simple.

Alternatively, there is an option to deploy to GitHub Pages through a repository's settings (<https://docs.github.com/en/free-pro-team@latest/github/working-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site>). We recommend sticking with the approach covered in this lesson, which also gives you additional practice working with the terminal and pushing and committing code.

[Previous \(/introduction-to-programming/git-html-and-css/practice-markdown-by-writing-readmes\)](#)

[Next \(/introduction-to-programming/git-html-and-css/practice-github-pages\)](#)

Lesson 33 of 64

Last updated March 25, 2023

[disable dark mode](#)



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.