



Checkpoint 8

GitHub

Remember when you started working with Repl.it? That was one of the first tools you learned about in this program! But as you know by now, there are many, many tools and technologies in the web developer's toolkit. In this checkpoint, you'll learn about GitHub.

Outline

When applying for jobs, web and software developers need to demonstrate to employers that they understand how to work with *version control systems*, which are systems that help tech professionals easily manage changes to files, software, websites, and other computer or web-based programs. And when it comes to version control, GitHub is an industry standard. GitHub also provides a free and easy method of deploying websites online so you can conveniently share your web projects with the world.

In this checkpoint, you'll start with the basics of GitHub. You'll learn the fundamentals and set up your free GitHub account. Then, you'll learn an easy way to post your web files onto GitHub. Finally, you'll learn how to update your project's settings so that everyone on the internet  view your hard work as a live website.

By the end of this checkpoint, you will be able to do the following:

- Showcase a public web project on GitHub.com
- Host a web project on GitHub.io

Introduction to GitHub

Today, nearly all web developers use GitHub, or a similar service, to help them accomplish three important tasks with their web applications and projects.



Outline

1. Version control

Version control allows many people to make and save changes for multiple versions of a project or program. This means you and other developers can experiment, make mistakes, and revert back to an

2. One safe place

These services securely store your projects in one safe place on the web. You can work locally and on more than one computer, but you always have an online backup. This one location makes storage easy and convenient.

3. Collaboration

More often than not, coding projects require multiple people to work nearly simultaneously. Version control systems allow many different contributors to edit and change the same set of files at the same time.

earlier version if
necessary.

Note

It's worth noting that for this checkpoint, you will be using GitHub simply as a way to store your code online and publish a web page. GitHub has many more features, too, but those will be your focus.

The GitHub language

As you get started with GitHub, you'll need to learn the lingo. Here are some new GitHub terms and concepts that will prove useful to you in your time at Thinkful and in your career.

- **Git:** This is the open-source system used for version control. It's used by many companies, but GitHub is the most popular.
- **GitHub.com:** The website where users can store and manage versions of their code projects online.
- **GitHub.io:** The website that displays websites that are stored on GitHub.com online, for all users to view.
- **GitHub Desktop:** A free app that is installed locally on your computer. It makes it easy to update code projects from your computer to GitHub.com.
- **Repository:** This is a project's folder that contains all of the files for the project, as well as each file's revision history. This is commonly

called a *repo*. There are local repositories, which are on your computer, and remote repositories, which are on GitHub.com. These will be explained in more depth below.

- **Public repo:** These are publicly accessible repositories. GitHub provides free access to any public repo.
- **Private repo:** This is a repo that only you (or others who you give access to) can see. GitHub also makes it free to store private repos; however, to access most of the functionality of a private repo, you'll need to pay a monthly fee.
- **Owner:** The person who creates a repo and therefore determines if it is public or private. They determine how code that is submitted to a project from others is managed.
- **Collaborator:** A person who has been invited by the repository owner to contribute to a project. They have been granted access to a repo, and they have the ability to edit the code in that repo.
- **Commit:** A fancy technical term that's used to refer to saving new changes to a version of a file.

Outline

Don't worry if these terms sound confusing or foreign right now. With a little practice, you'll quickly find yourself using these terms in conversations with other developers, and your classmates. And before too long, you'll start to sound like a true web developer.

GitHub overview

Now that you're a bit more familiar with the lingo, you can start getting a feel for what GitHub is all about. [GitHub.com](#) provides a free service of

storing the files for any web project online. Web developers, designers, and many other tech professionals make public profiles on GitHub that showcase projects they've worked on. They can even create a visual display highlighting the public contributions they've made to remote repos throughout the year.

This video will provide you with more information about GitHub and its capabilities.

Outline

The image shows a video player interface. At the top left, there is a white rectangular button with the text "HTML/CSS". The main title "GitHub: Overview" is displayed in large white font, with a play button icon integrated into the letter "o". Below the video area, there is a control bar with a play button, a volume icon, and a timestamp "0:00 / 3:47". To the right of the timestamp are icons for closed captions ("CC"), a 1x speed setting, and a share symbol.

It can be helpful to look at some user profiles to better understand GitHub's features. Here is a link to the profile of [Sarah Dayan](#), a French web developer. As you can see, she has had a lot of activity over the last year. All of her public projects are pinned at the top of her profile, so

they are completely accessible for you and others to view. Feel free to click around.

The screenshot shows a GitHub profile for a user named Sarah Dayan. At the top, there's a large profile picture of a woman with dark hair and glasses. Below the picture, her name 'Sarah Dayan' and handle 'sarahdayan' are displayed. A 'Follow' button is present. To the right, there are sections for 'Overview', 'Repositories 42', 'Projects 0', 'Stars 40', 'Followers 593', and 'Following 7'. The 'Overview' tab is selected. Under 'Pinned', there are four repository cards: 'dinero.js', 'snippets.js', 'browserstore.js', and 'frontstuff'. Each card includes a thumbnail, the repository name, a brief description, and stats like programming language, stars, and forks. Below these is a heatmap titled '1,988 contributions in the last year' showing activity by month and day. A legend at the bottom of the heatmap indicates contribution levels from 'Less' to 'More'.

Outline

At the beginning of your career, your GitHub profile will look a bit empty, much like this fictional account by [Jenny Thinkful](#). This makes sense—you're just getting started!

Click the above link to Jenny's profile to explore a bit more.

The screenshot shows Jenny Thinkful's GitHub profile page. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation is a section titled "Popular repositories" which lists "pizza" and "CSS". A chart titled "2 contributions in the last year" shows activity across months from April to February. Below this is a "Contribution activity" section for February 2020, showing she created 1 commit in 1 repository named "jennythinkful/pizza". The year selector at the top right of this section is set to 2020.

Outline

Once you're on Jenny's GitHub profile page, click the project *pizza* under *Popular repositories* to view all the project files.

Outline

jennythinkful / pizza

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

No description, website, or topics provided.

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

jennythinkful Initial Upload Latest commit c7c18dd 10 days ago

File	Description	Time
images	Initial Upload	10 days ago
README.md	Initial Upload	10 days ago
index.html	Initial Upload	10 days ago
pizza.jpg	Initial Upload	10 days ago
style.css	Initial Upload	10 days ago

README.md

Pizza

Single Web Page: Favorite Pizza Toppings

This is a single web page listing my favorite pizza toppings.

<https://github.com/jennythinkful/pizza/actions>

There, you should see the four files listed below:

- `index.html`
- `style.css`
- `images` (This folder contains a single image: `pizza.jpg`.)
- `README.md`

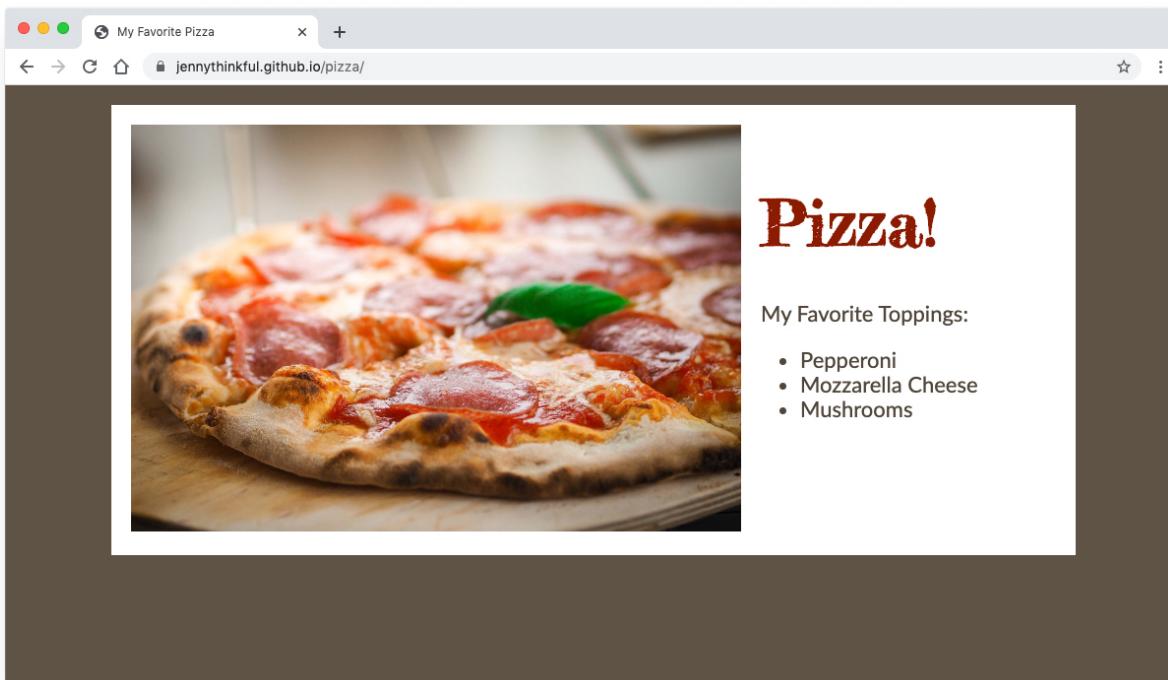
Some of these filenames should seem familiar to you! But one, `README.md`, may be new. Generally, *README files* are simple documents that accompany a web project to describe, explain, or

provide instructions for that web project. Although README files can come in various plain text formats, the file type `.md` here means this README file is in *Markdown* format. Markdown files offer you a shorthand technique to write simple HTML files with different elements, such as headings and paragraphs, without actually having to write the basic HTML code. However, it's important to note that CSS cannot be applied to Markdown files. All projects on GitHub are *strongly* encouraged to have a single `README.md` file.

Among its many functionalities, GitHub also allows web projects that use HTML, CSS, and JavaScript to be shown as web pages. Here is the URL of Jenny's pizza web project, which is hosted on GitHub.io:

<https://jennythinkful.github.io/pizza/>.

Outline



GitHub hosting

It's free to access all public repos on GitHub and to share those websites online using GitHub.io. GitHub also provides free storage of private repos, but as noted above, much of the functionality for private repos is still hidden behind a monthly fee. As a newcomer to the web development world, you need as much visibility as you can get! Early in your career, it's important for you to be brave and share everything on your profile. Make your projects accessible to the public (which includes potential employers).

It's fairly straightforward to set up a website for yourself that's hosted on GitHub.io. You can also purchase a domain name to point to that website, which you'll learn more about below.

Understanding the GitHub basics

Once you get the hang of it, working with GitHub isn't too difficult. But it certainly poses some unique challenges if you've never used anything like it before. The steps below will help you get oriented before you dive in and try out the process yourself.

Step 1: Start a project

It certainly helps to have a project already started before jumping into GitHub, even if that's just a project folder on your computer with the following core files:

- `index.html`, containing the basic framework HTML code
- `style.css`, which will be empty until you add to it
- `script.js`, which will be empty until you add to it

Step 2: Connect the project folder to GitHub

There are three ways of getting your local project files onto GitHub to be saved as a remote repository. You can review each option below.

1. **Drag and drop a folder into GitHub.com.** For your first project in these checkpoints, you'll see how easy it is to drag and drop your project folder into GitHub. This is a great, simple approach for smaller files. But as you work on larger projects, you'll find that the two other methods listed here are more useful.
2. **Use GitHub Desktop to upload project files onto GitHub.com.** As discussed above, GitHub Desktop is a free app that monitors any changes made to the files of a local project. By clicking a single button, you can then update any web files that have been edited locally. This means that you apply any changes you've made on your computer to the files that exist on the remote, online repo on GitHub.com. Working with GitHub Desktop is an easy way to monitor larger projects with multiple files. But as you get more comfortable with this process, you'll soon learn to use the command line to upload your files to GitHub.
3. **Use the command line to upload project files.** Command-line interfaces, like the Terminal application in macOS systems, are already on your computer and will give you far greater control in your work with GitHub. At their most basic level, these systems are text-based applications that allow you to communicate with, view, and organize various files and directories. This is especially important as you work with multi-person teams on the same project files and collaborate on various *forks*, or copies, of a

project. By the end of this program at Thinkful, you'll feel extremely comfortable with all three of these methods. And believe it or not, you'll feel especially comfortable with the advanced techniques using the command line.

Step 3: Commit changes to a project onto GitHub

As noted above, when using GitHub for a project, you'll be working with two repos: a local repository and a remote repository. For any given project, these two repos would be the same. The difference is where you're making changes to the files.

- The *local repository*, or local repo, is the project folder that is stored on your computer. It's the local (in other words, not web-based) folder that contains all the files for the project. Edits can be made to any of these files, even if you're offline. When you're online, you can upload those changes to the files that are on GitHub.com.
- The *remote repository*, or remote repo, is the GitHub project folder that is stored on the GitHub.com website. If the remote repo is a public repo, anyone can view the code and the organization of those files. If it's a private repo, only people who are invited can view the code and files. These repos can be downloaded onto any computer and synced with changes made from any team members.

As you write HTML, CSS, and JavaScript code and add additional files and images to a local repo, you will be able to easily upload your

changes to the remote repo on GitHub. You'll be able to do this without much management or having to upload the files manually. In fact, when you use GitHub Desktop or Terminal, your projects will continuously be checked for any changes made to the project folder. With just a few clicks, you'll be able to seamlessly upload all of your code changes and additions to GitHub.com.

Step 4: Create a live website on GitHub.io

Notably, the web projects on GitHub.com only showcase code. They do not showcase the *actual* web pages. But that's where GitHub.io comes in. In the project's settings, you can easily make it so that GitHub.com will host the web project on the GitHub.io website. And you can do this for free!

Outline

Set yourself up with GitHub

You're ready to get started with GitHub. Follow the steps below to learn how.

Start with a project folder

At this point, you may not have a web project ready to add to GitHub.com. Fortunately, you can practice with the pizza project that Jenny Thinkful has on her profile.

You can find the [pizza repo here](#). You'll also want to view [Jenny's Pizza Repl.it](#) and [Jenny's deployed web page](#) as you follow along.

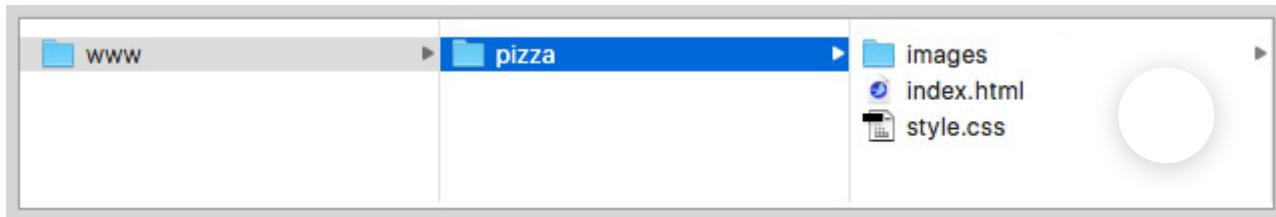
Depending on the interface that you're working with, you'll either select the **Clone or download** option or click the **Code** button. From there,

select the **Download ZIP** button to download the *pizza* repo onto your computer.

The screenshot shows a Mac OS X desktop environment. On the left, there's a file browser window with a sidebar containing a folder named 'www'. In the main pane, there's a folder named 'pizza' which contains three files: 'images', 'index.html', and 'style.css'. On the right side of the screen, a GitHub repository page for 'jennythinkful/pizza' is open. The repository summary shows 1 commit, 1 branch, 0 packages, 0 releases, 1 environment, and 1 contributor. Below this, there's a list of files: 'images' (Initial Upload), 'README.md' (Initial Upload), 'index.html' (Initial Upload), 'pizza.jpg' (Initial Upload), and 'style.css' (Initial Upload). At the bottom of the GitHub page, there's a 'Clone or download' button with options for 'Clone with HTTPS' and 'Download ZIP'. A hand cursor is hovering over the 'Download ZIP' button. The GitHub URL 'https://github.com/jennythinkful/pizza/archive/master.zip' is also visible at the bottom of the page.

Outline

Now that you have a local *pizza* repo, it's important to develop some organizational principles with your project folders. Keep your web files well organized, in a specific order and folder hierarchy. Consider moving the *pizza* folder into a specific location on your computer. For instance, you could create a folder on your desktop and appropriately name it *WWW*, *GitHub*, *Web Projects*, or something else that is relevant and logical. Then, you could place every web project within that folder.



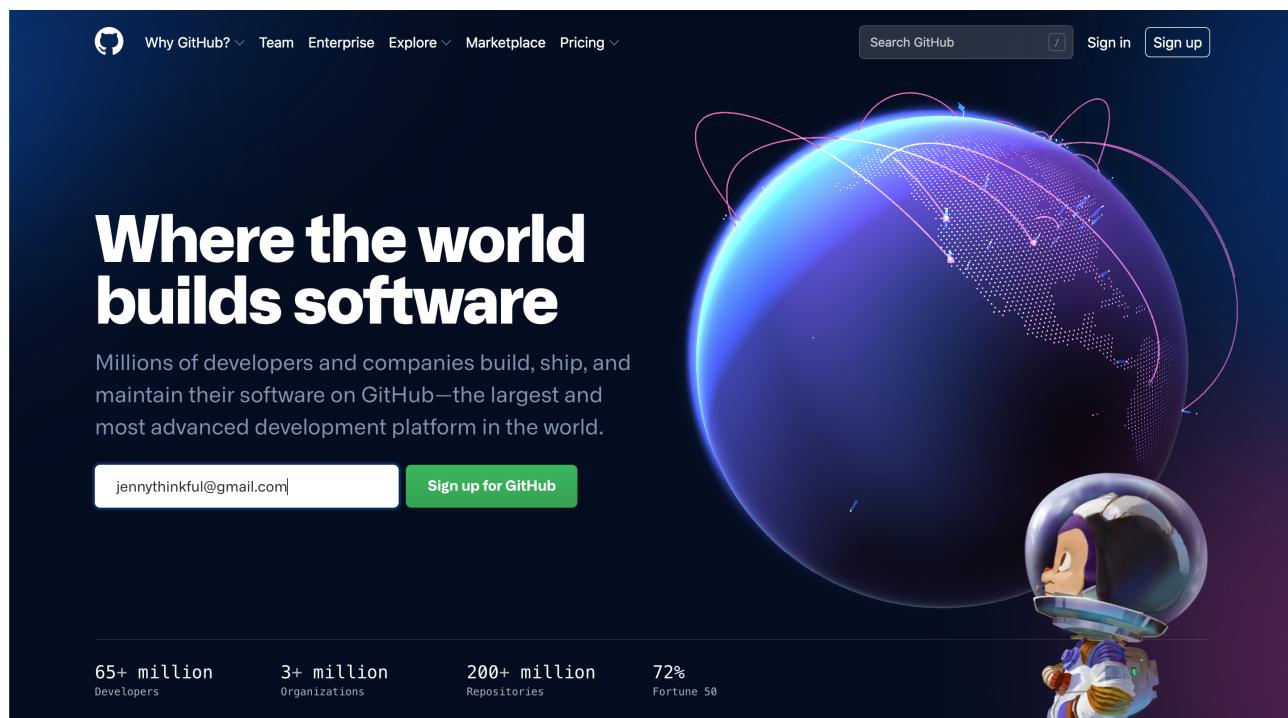
If necessary, you can move your projects to another location on your computer after they are connected to GitHub Desktop. But staying organized from the start will be very beneficial.

Open a new account

Now that you have a project to work with, you're ready to jump in. If you haven't already, you'll need to create a free account on [GitHub.com](https://github.com).

The first thing that you'll have to do is to sign up with your email address:

Outline



Next, you'll be asked to create your account with your username, email address, and password. On this page, you can also set your email preferences, and you may be asked to solve a simple puzzle to verify that you're a real person using the website.



Join GitHub

Create your account

Username *

JennyThinkfulCodes

Email address *

jennythinkful@gmail.com

Password *

.....

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

 Send me occasional product updates, announcements, and offers.

Verify your account

[Create account](#)

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

© 2021 GitHub, Inc.

[Terms](#)[Privacy](#)[Security](#)[Status](#)[Docs](#)[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)

Create a username with your name, or make a unique username that's available. Consider using variations of your name, your initials, and the work you do. Remember, potential employers, fellow developers, and anyone else who searches for your work will see this username, so aim to create one that is memorable and appropriate. This username will help others find your projects, and it will also be part of any public URLs associated with your projects. Here are two examples from Jenny's usernames:

- **jennythinkful** : <https://github.com/jennythinkful>

- [jennythedeveloper](https://github.com/jennythedeveloper): <https://github.com/jennythedeveloper>

When you're ready, press the **Create account** button.

Next, you'll be asked to fill out a survey to help GitHub customize your account for you.

Outline

The screenshot shows the GitHub account creation survey. At the top, it says "Welcome to GitHub" and "Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there." Below this, there are two sections of questions with multiple-choice options. The first section asks "What kind of work do you do, mainly?" with options like Software Engineer (I write code), Student (I go to school), Product Manager (I write specs), UX & Design (I draw interfaces), Data & Analytics (I write queries), Marketing & Sales (I look at charts), Teacher (I educate people), and Other (I do my own thing). The second section asks "How much programming experience do you have?" with options like None (I don't program at all), A little (I'm new to programming), A moderate amount (I'm somewhat experienced), and A lot (I'm very experienced). At the bottom, it asks "What do you plan to use GitHub for? (Select up to 3)" with three icons: a laptop, a GitHub logo, and a notepad.

When you're done completing the survey, press the **Complete setup** button at the bottom of the page.

Verify your email address

Once you're done setting up your account, you'll need to verify your email address. Check your email for a message from GitHub and follow the directions. Select the option to verify your email.

Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to john123@aol.com.

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

GitHub

Product	Platform	Support	Company
Features	Developer API	Help	About
Security	Partners	Community Forum	Blog
Enterprise	Atom	Professional Services	Careers
Customer stories	Electron	Learning Lab	Press
Pricing	GitHub Desktop	Status	Social Impact
Resources		Contact GitHub	Shop

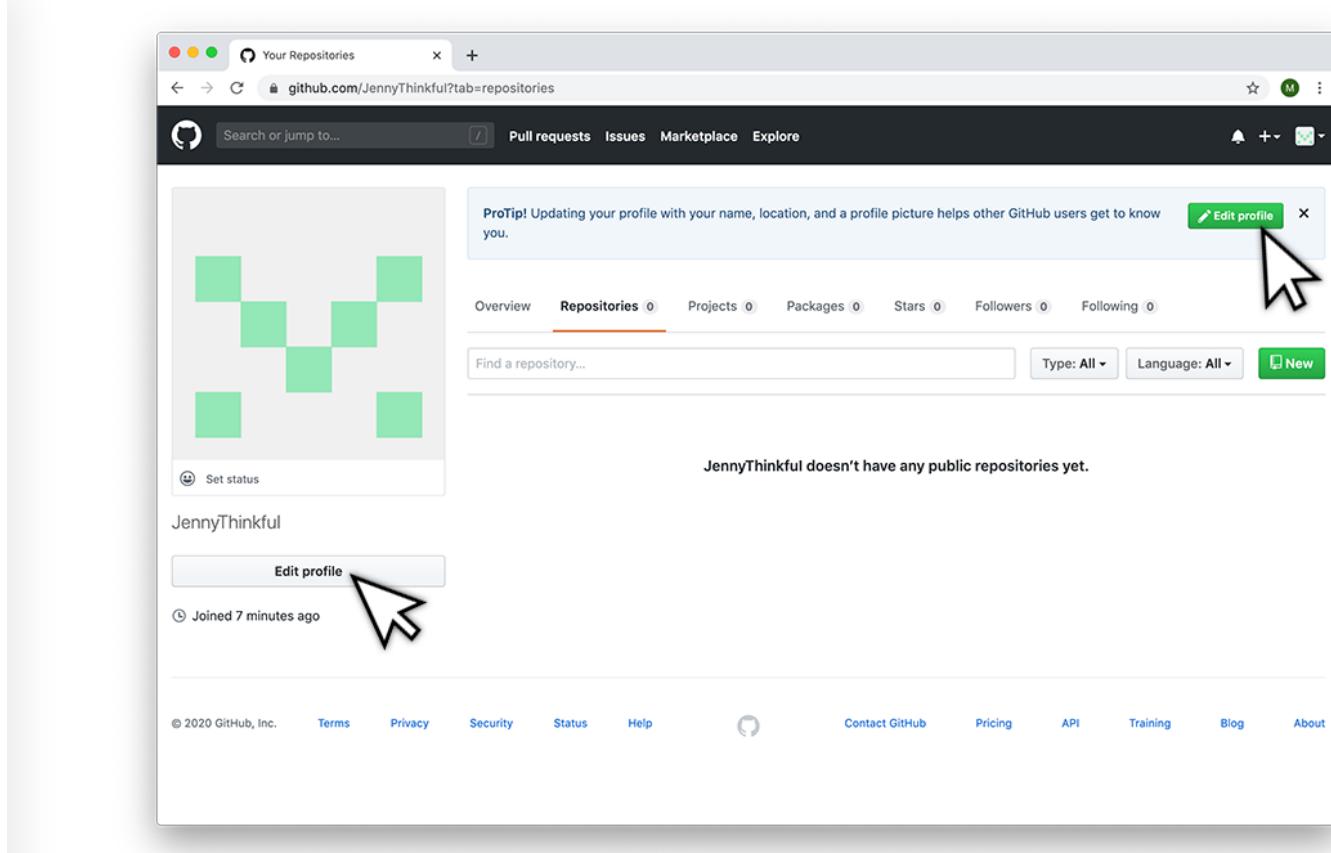
© 2020 GitHub, Inc. [Terms](#) [Privacy](#)

[Twitter](#) [Facebook](#) [YouTube](#) [LinkedIn](#) [GitHub](#)

Outline

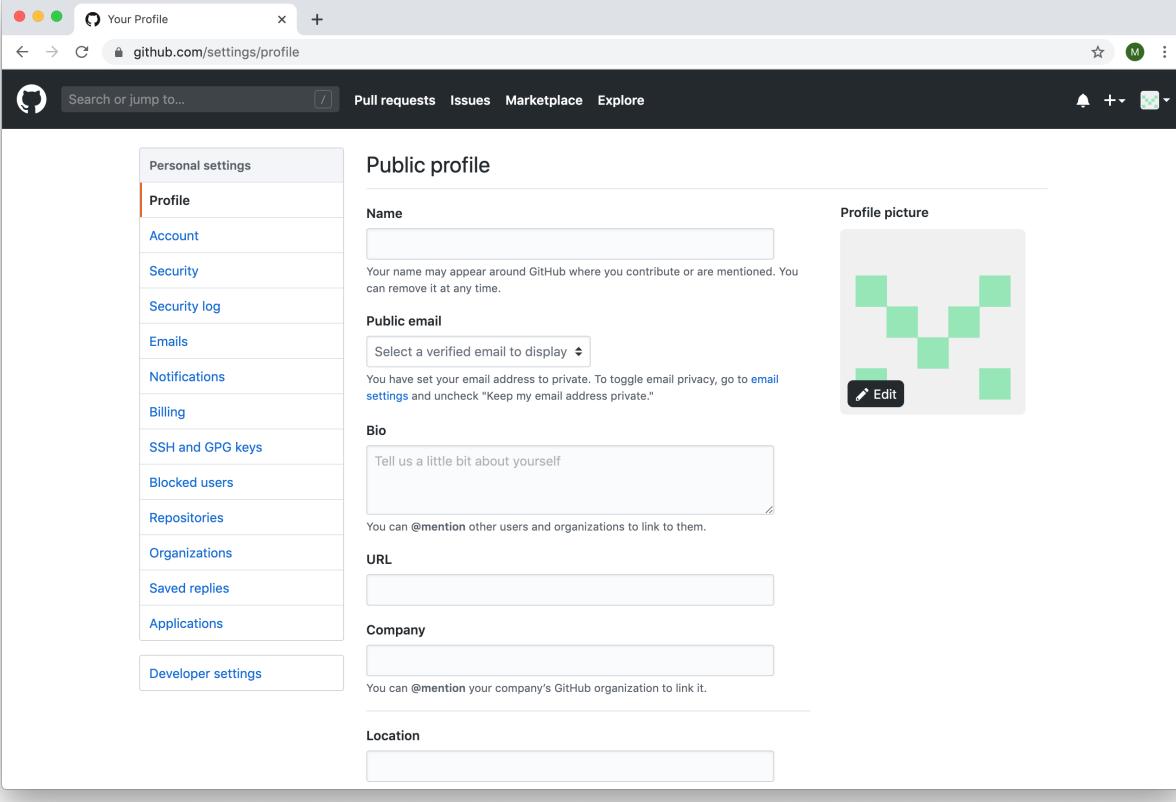
Edit and update your profile

At this point, you can (and should) take a minute to update and personalize your GitHub account. Click your profile icon in the top-right corner, which should reveal a drop-down menu. From there, select **Your profile** and then click the **Edit profile** button.



Outline

Your classmates and any future employers will see the content on your profile. You'll want to add a professional profile photo, and you can update the rest of the content with any relevant information you feel comfortable sharing. And you can always revisit your profile page as needed to make updates.

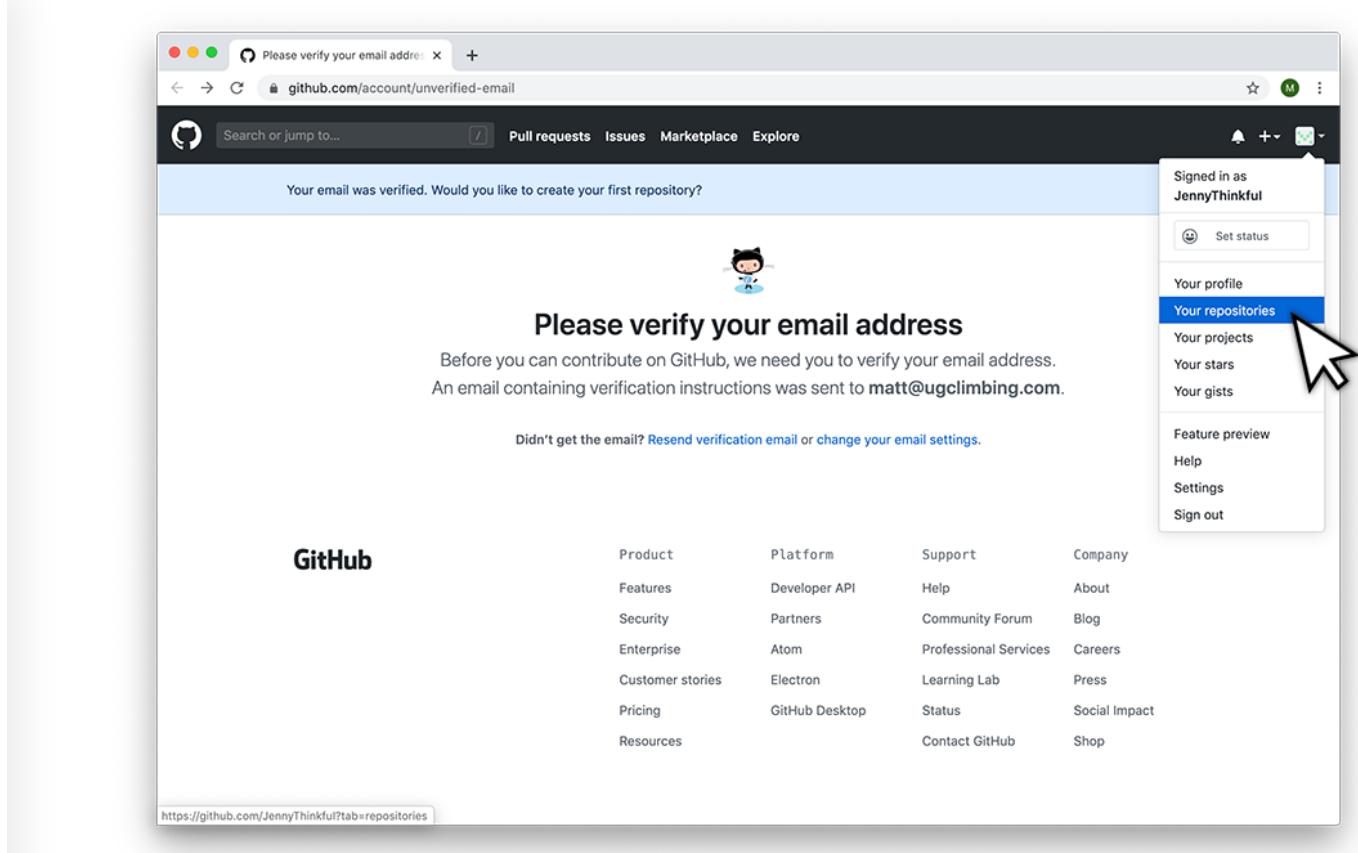


The screenshot shows the GitHub profile settings interface. On the left, a sidebar lists various sections: Personal settings, Profile (which is selected and highlighted in orange), Account, Security, Security log, Emails, Notifications, Billing, SSH and GPG keys, Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings. The main area is titled 'Public profile' and contains fields for Name, Public email, Bio, URL, Company, and Location. A placeholder image for the profile picture is shown, along with an 'Edit' button.

Outline

Go to your repo page

Once you've verified your email address and personalized your profile page, you can get to work! Now that you're in GitHub, you can click your profile icon (or your photo, when you've uploaded one) in the top-right corner. From that drop-down menu, select **Your repositories** to begin building and exploring.



Your email was verified. Would you like to create your first repository?

Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.
An email containing verification instructions was sent to matt@ugclimbing.com.

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

Signed in as
JennyThinkful

- [Set status](#)
- [Your profile](#)
- [**Your repositories**](#)
- [Your projects](#)
- [Your stars](#)
- [Your gists](#)

Feature preview
Help
Settings
Sign out

GitHub

Product	Platform	Support	Company
Features	Developer API	Help	About
Security	Partners	Community Forum	Blog
Enterprise	Atom	Professional Services	Careers
Customer stories	Electron	Learning Lab	Press
Pricing	GitHub Desktop	Status	Social Impact
Resources		Contact GitHub	Shop

<https://github.com/JennyThinkful?tab=repositories>

Outline

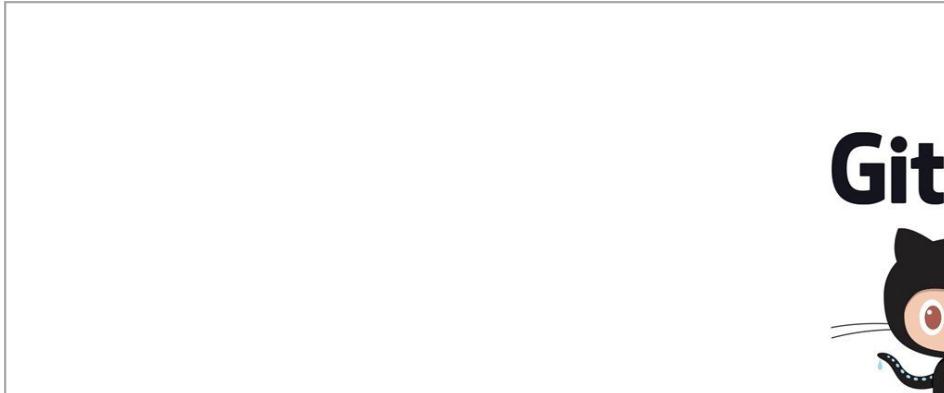
Create a new repo

You're ready to create a new repo! Watch the video below to get started.

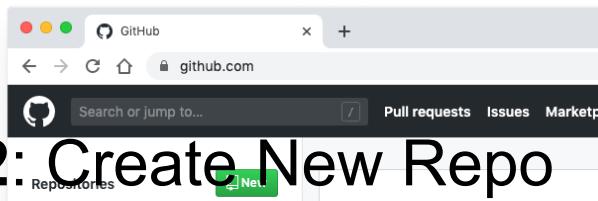
The video player interface displays a video titled "GitHub: First Commit". The thumbnail image shows a white box with the text "HTML/CSS". Below the video title is a play button icon. At the bottom of the player, there is a control bar with a play button, a volume icon, and a timestamp indicating the video is at 0:00 of 2:07. There are also icons for closed captions, 1x speed, and a share or embed option.

Outline

These slides highlight the key steps that are outlined in the video.



Step 1: Navigate to your repo



Step 2: Create New Repo

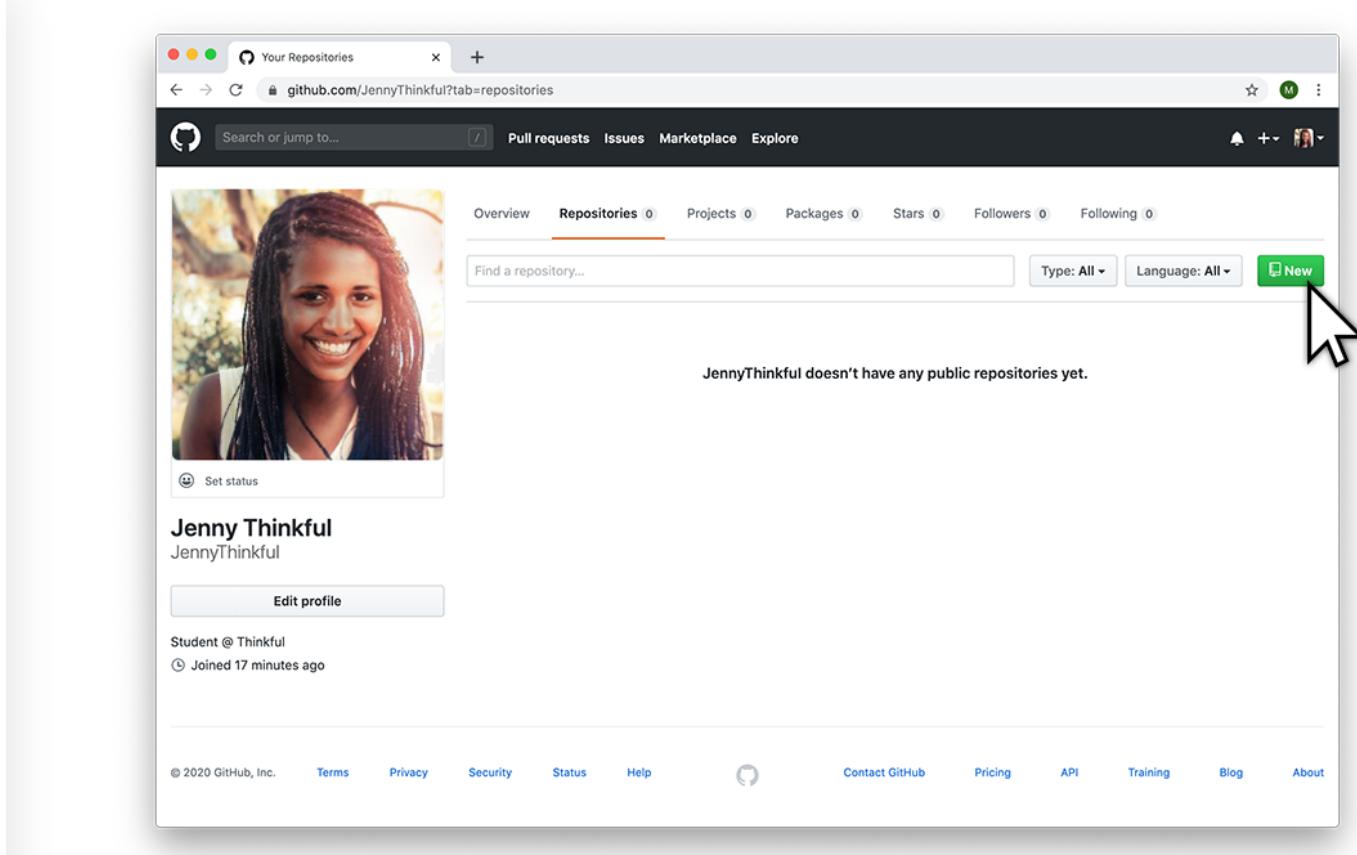
Step 3: Fill out repo info

By [Thinkful](#)



New repo documentation

You're ready to create your first repo. Depending on the interface you're looking at, you may click the **New** button, or you'll select the **+** icon next to your profile picture. In the drop-down menu, you'll click **New repository**.



Outline

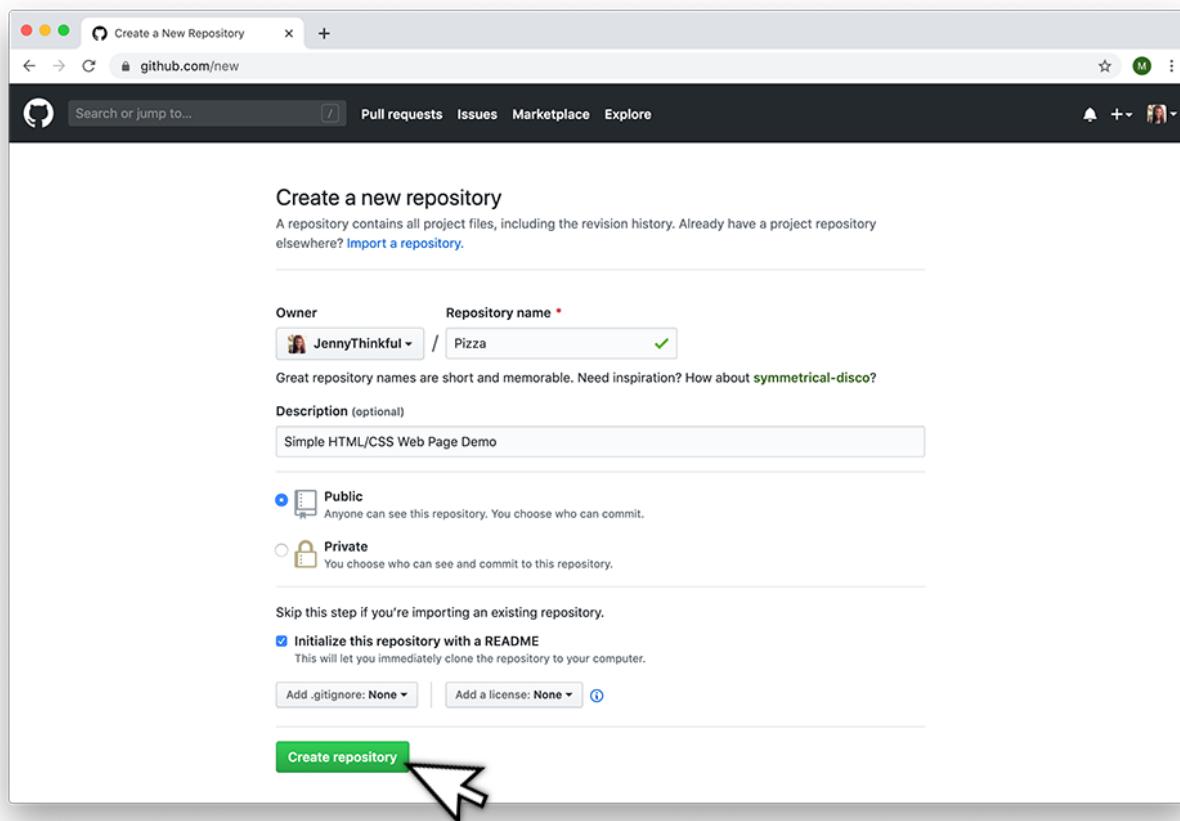
This will open the *Create a new repository* web page. Now, you'll fill in a bunch of important information about your new repo.

- **Repository name:** This is where you name your repo. Typically, you'll want to give the remote repo the same name as the local project folder; this will help you stay organized. However, the names don't have to be exactly the same.
- **Description:** You can provide an optional description that briefly explains the project. This can be really useful later on, when you or other developers need to quickly identify the project.
- **Public or private:** Down the line, you'll likely use these options to determine whether to make your repos public or private. However, for now, keep this project (and others you make in this project) public to get the full functionality and to share your work.

- **Initialize this repository with a README file:** Because you are creating a new repo, you'll probably want to create a README file to initialize this repository. As noted above, a `README.md` file is a simple Markdown document that provides a description and additional notes explaining what this web project is all about.

When you're ready, select the **Create repository** button.

Outline

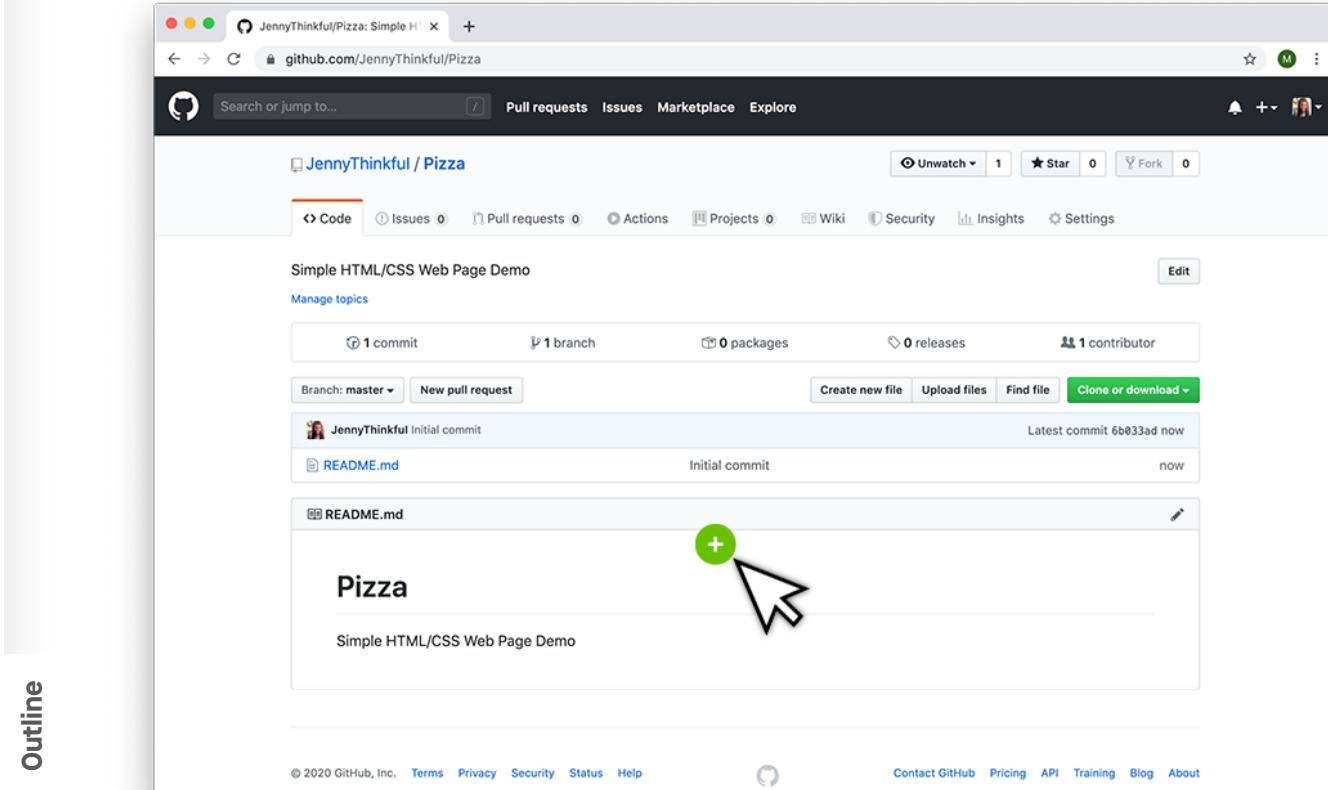


Upload your project files

Now, you are seeing your first remote repo! It's pretty empty, but it's there!

At this point, let's practice adding a project folder. Find your *piz* project folder on your computer. You should be able to find it in *...*

folder you created, such as *Desktop > WWW > Pizza*. Now, you can drag and drop that folder right onto the GitHub page to upload it.



Outline

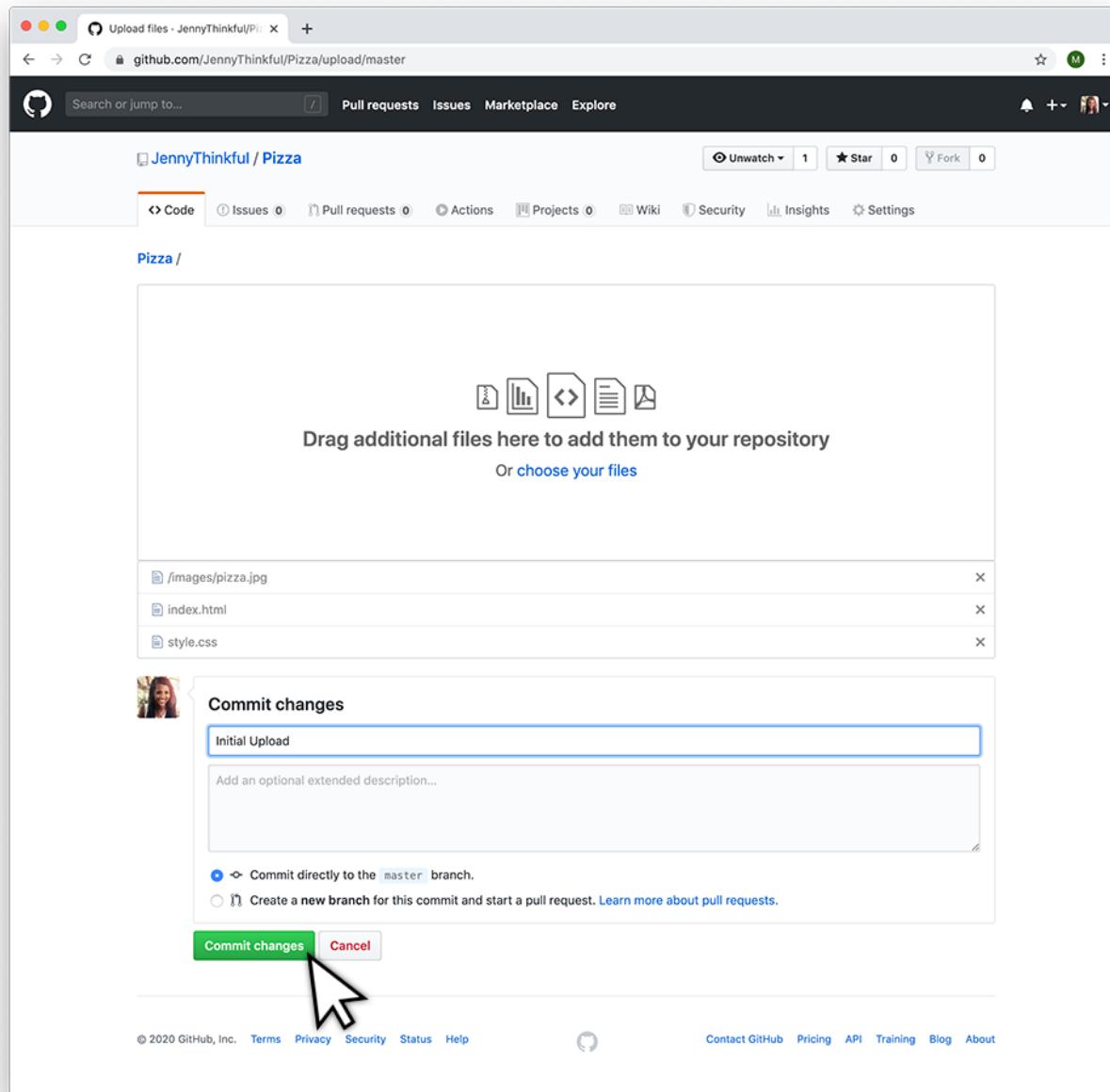
Committing project files

Now that you've uploaded the *pizza* project folder, GitHub recognizes your `index.html` page, your `style.css` page, and the `pizza.jpg` file within the images folder.

Adding new files to a repo involves making a commit. As you learned above, a commit is essentially like saving a change to a version of a project. As you commit new files to your repo, you'll need to briefly identify what the commit is doing. In this case, you might write a commit message like *Initial Upload* to tell yourself, and others, that this commit is the first time you're uploading files to the repo. And as you

commit changes to larger, more complex projects, this title or commit message can help identify what changes you've made. This will help you keep track of edits to the project and the different versions you and your team are working with.

When you are ready, click the **Commit changes** button.



Setting files on GitHub.io

At this point, your local repo is duplicated online as a remote repo. The link to this repo page can be shared with classmates and potential employers to showcase your hard work. Now that the files are online, you can get GitHub.com to present your files as a live web project using GitHub.io. Watch the video to learn more.

Outline



Now that you're oriented, you can put this into action. Start by clicking the **Settings** option.

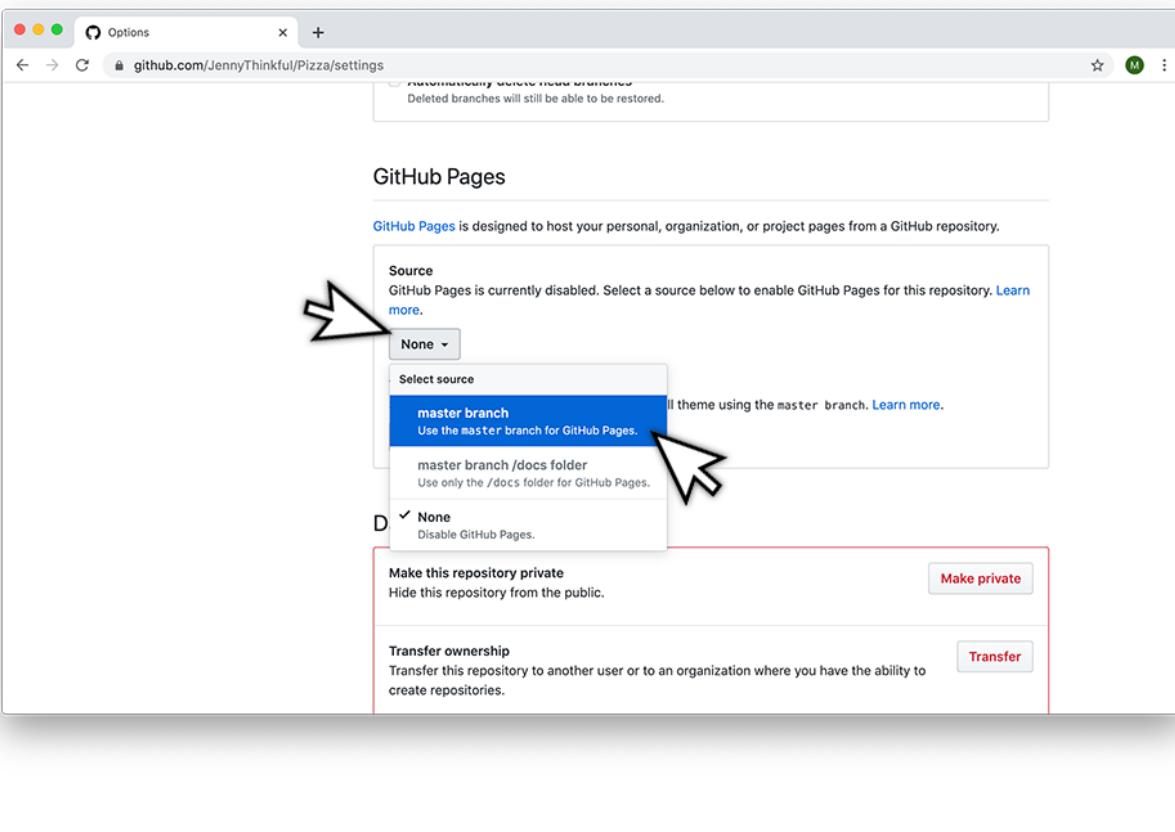
The screenshot shows a GitHub repository page for 'JennyThinkful/Pizza'. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation is a search bar and a header for the repository. The repository name is 'JennyThinkful / Pizza'. To the right of the repository name are buttons for Unwatch (with 1), Star (with 0), Fork (with 0), and Edit. A cursor arrow points towards the 'Edit' button. Below the header, there's a summary section with metrics: 2 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. A dropdown menu shows 'Branch: master' and a 'New pull request' button. To the right of these are buttons for Create new file, Upload files, Find file, and Clone or download. Below this summary is a table of files and their upload history:

File	Description	Time
images	Initial Upload	now
README.md	Initial commit	3 minutes ago
index.html	Initial Upload	now
style.css	Initial Upload	now

Below the file table is a preview of the 'index.html' file content, which contains the word 'Pizza'.

Outline

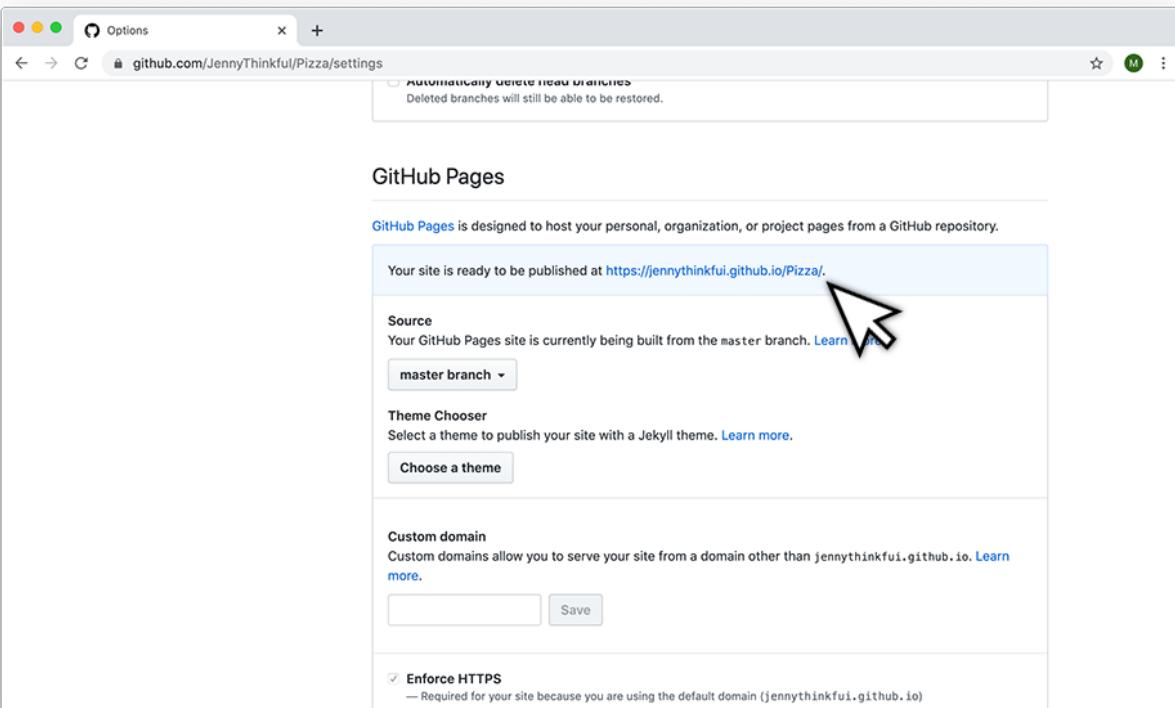
On the **Settings** page, scroll down to reach the **GitHub Pages** option. Here, select the drop-down option currently displaying as **None** and change it to **main branch**, as seen below.



Outline

Note: This is showing the `master` branch, but GitHub has now changed to using `main` as the default.

Your web pages are now live! But to find them, you still need the URL. While you're still under the **GitHub Pages** option on the *Settings* page, locate the GitHub.io URL where you can find your website. You can share this URL with others so that they can see your live website files!



Outline

You have just created your first remote GitHub repository and published it as a live website. Congrats! And now, if you make any changes to the code within the files or you drag and drop new files into your remote repo, your changes will then be seen on both GitHub.com and your personal website on GitHub.io.

More on README files

Now that you've got GitHub set up, you'll take a moment to revisit README files. The video below will give you a bit more information on how these files work.



Outline

As you've likely noticed, the README filename is in capital letters, which defies some of the web conventions that you've learned about previously. But a `README.md` file isn't a web file. It is used to provide a description of the project on GitHub. In fact, one reason that README files are in capital letters is to help them stand out among the other files in a folder. This makes sense because it is probably the first thing that a new user would want to look at in order to learn a little more about the project.

A good README file will highlight your public project on GitHub and help others understand what your project is about. Here are some common and helpful pieces of information to add to a README file:

- A single paragraph describing the project
- Information about the author or creator
- Any acknowledgments, such as expressing thanks to anyone whose code you used and mentioning anyone whose work inspired you

You can also check out this [curated list](#) of strong README files to get a sense of what to include. But don't be intimidated—many of the projects listed there are quite sophisticated, and you're not yet an expert developer. That list simply showcases good examples of comprehensive README files.

More on Markdown

The power of Markdown files is that they provide a shorthand method for writing HTML. The trouble is this shorthand method works only in `.md` files and will not work with `.html` files. So it's important not to get these shortcuts confused with HTML and CSS code. If you wish to read more about Markdown, check out this [resource](#).

More on custom domain names

As noted briefly above, you can buy a custom domain name (URL) that can easily showcase your projects by pointing to your GitHub.io website. In fact, a domain name can be used to point to your GitHub account or any of the projects that are hosted on GitHub.io. When you're ready to purchase a custom domain name, follow the directions below. These are the two primary steps:

1. Purchase your domain name and point the domain to GitHub.

2. Update GitHub to point to the domain, too.

A domain name usually costs around \$10 for yearly access. (So, if Jenny were to create her web portfolio and host it publicly on GitHub, then she could run <http://www.jennythinkful.com> for around \$10 per year!) Three popular domain sites are listed below:

- [GoDaddy](#): The cost of a domain name is about \$12 per year.
Check out this [tutorial](#) to learn more.
- [Name.com](#): The cost of a domain name is about \$11 per year.
Check out this [tutorial](#) to learn more.
- [Namecheap](#): The cost of a domain name is about \$10 per year.
Check out this [tutorial](#) to learn more.

If you have any questions, reach out for assistance.

Assignment

For this assignment, you'll showcase a public web project on GitHub.com. Complete the following steps using the instructions provided in this checkpoint.

1. Create a GitHub account.
2. Download the *pizza* project from Jenny Thinkful. (Remember, you can find her repo [here](#).)
3. Create a new repo.
4. Upload your project files.
5. Commit your changes.

When you're done, you should have a remote repo hosted online, and you should be able to share that link with others so that they can see it, too.

Checkpoint

Submit your ideas or a link to your work here and use it as a conversation starter during your next mentor session.

This checkpoint will not be graded, but is still required.

Your work

Outline

03.11.21



Share your ideas here...

****bold** _italic_ `code` > quote - bullet list**

Preview

Completed

Next checkpoint



How would you rate this content?

[Report a typo or other issue](#)

[Go to Overview](#)

Outline