

`GIT` Immersion



Welcome to the new era ;)

Will you become one of the professional programmers?

Speaker: Tanay

Introduction



Github

Git

Git Pages

Let's get to know each other a little bit:

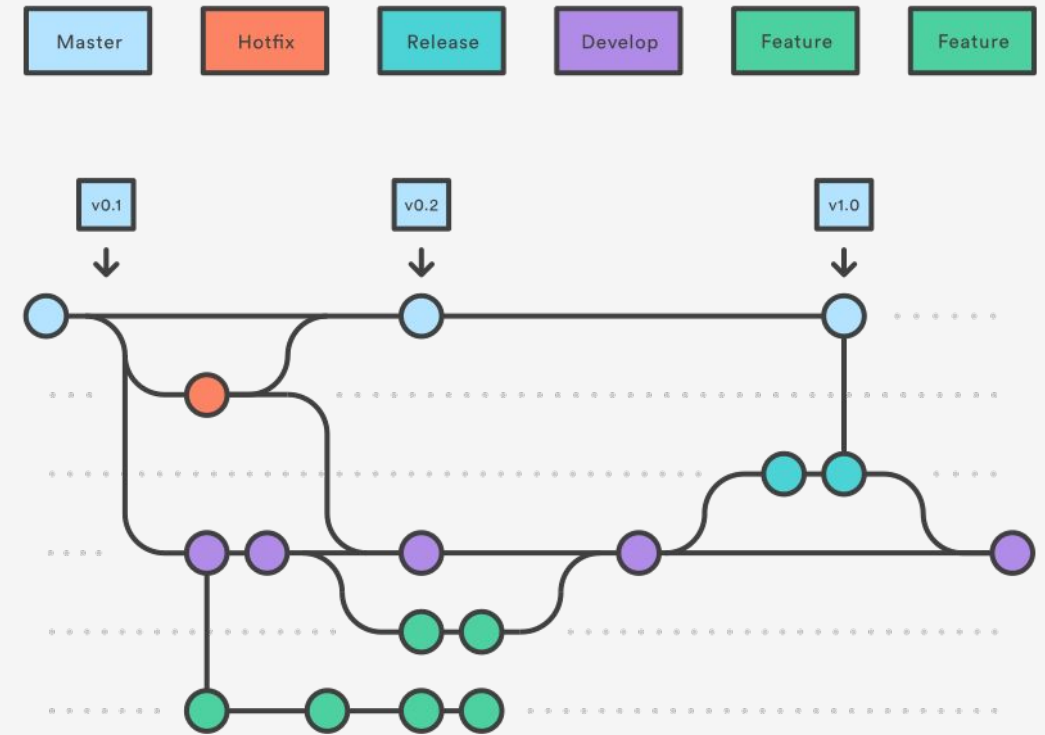
- Name
 - Where do you study/work?
 - One interesting fact about you.
-

What is Git?

Distributed version control system
It is free and open source

Primary use:

- Code management
- Data integrity is assured
- Team collaboration is allowed
- Keep track of changes in files



Github

Git

Git Pages

Steps to follow

Github Account

Create Repository

Create a branch

Github

Git

Git Pages

Step 1: Create a Github account



Github

Git

Git Pages

Step 2: Create a repository



Github

Git

Git Pages

Step 3: Create a new branch



Github

Git

Git Pages

Steps to follow

Github Account

Download
Git

Open
terminal

Step 1: Create a Github account

Step2: Download Git

Step3: Open terminal (macOS)
Download GitBash (windows)



Github

Git

Git Pages

Project: Hello World

Purpose : Simple exercise to get used to GIT

Things you will learn after today :

1. Create and use a repository
2. Start and manage a new branch
3. Make changes to file => push them to GIT as commits
4. Open a pull request
5. Merge

Lets start !!!

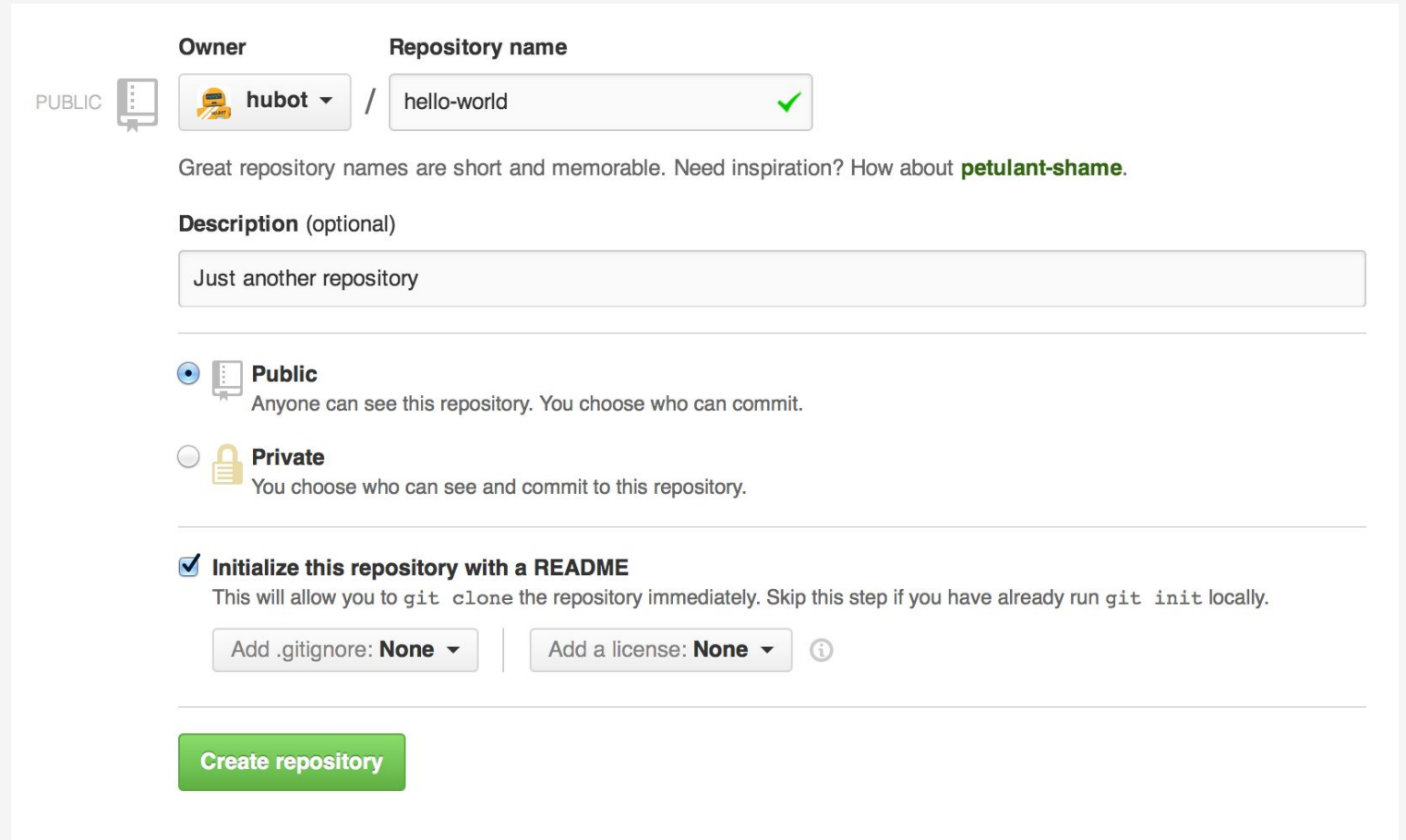
Github

Git

Git Pages


1. Create and use a repository

1. In the upper right corner, next to your avatar or identicon, click and then select **New repository**.
2. Name your repository hello-world.
3. Write a short description.
4. Select **Initialize this repository with a README**.
5. Create repository



The screenshot shows the GitHub 'Create new repository' form. At the top, there's a 'PUBLIC' label with a monitor icon. The 'Owner' dropdown is set to 'hubot' with a small avatar icon. The 'Repository name' field contains 'hello-world' and has a green checkmark. Below this, a hint says 'Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.' The 'Description (optional)' field contains 'Just another repository'. Under the 'Visibility' section, 'Public' is selected with a radio button and a monitor icon, with the text 'Anyone can see this repository. You choose who can commit.' Below it, 'Private' is unselected with a radio button and a lock icon, with the text 'You choose who can see and commit to this repository.' The 'Initialize this repository with a README' option is checked with a blue checkmark. Below this, a note says 'This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. A large green 'Create repository' button is at the very bottom.

PUBLIC

Owner:  hubot

Repository name: hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional): Just another repository

☒ **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 allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: None | Add a license: None ⓘ

Create repository

2. Start and manage a new branch

Why do we need branching ? We can work on more than 1 repository at 1 time

1. Go to your new repository **hello-world**.
2. Click the drop down at the top of the file list that says **branch: master**.
3. Type a branch name, readme-edits, into the new branch text box.
4. Select the blue **Create branch** box or hit “Enter” on your keyboard.

3. Make changes to file then push them to GIT as commits

+ Save changes are called commits

Steps:

1. Click the README.md file.
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself.
4. Write a commit message that describes your changes.
5. Click **Commit changes** button.

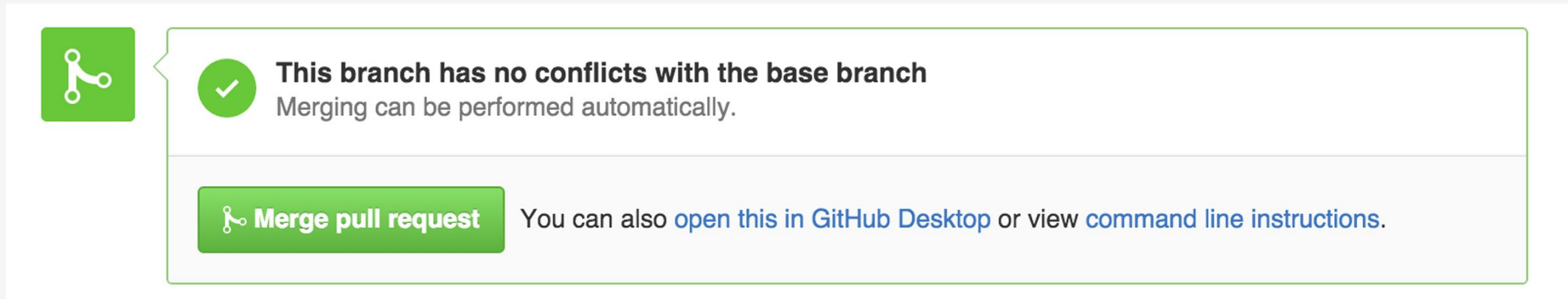
The screenshot shows the GitHub web interface for the repository 'hubot / hello-world'. At the top, there are buttons for 'Unwatch', 'Star', and 'Fork'. Below this, there are tabs for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Code' tab is selected, and the 'hello-world / README.md' file is open. The file content is visible in the editor, showing a README template. Below the editor, there is a 'Commit changes' section with a text input field containing 'Finish README' and a larger text area containing 'And mention moon tacos'. There are two radio buttons for the commit destination: 'Commit directly to the readme-edits branch' (selected) and 'Create a new branch for this commit and start a pull request'. At the bottom, there are two buttons: 'Commit changes' (green) and 'Cancel' (red).

4. Open and merge a pull request

1. Click the **Pull Request** tab, then from the Pull Request page, click the green **New pull request** button.
2. In the **Example Comparisons** box, select the branch you made, readme-edits, to compare with master (the original).
3. Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.
4. When you're satisfied that these are the changes you want to submit, click the big green **Create Pull Request** button.
5. Give your pull request a title and write a brief description of your changes.
6. **Create pull request** when you are done.

Merge a pull request

- ❖ Merging your readme-edits branch into the master branch.
- ❖ Steps:
 - 1) Click the green **Merge pull request** button to merge the changes into master.
 - 2) Click **Confirm merge**.
 - 3) Go ahead and delete the branch, since its changes have been incorporated, with the **Delete branch** button in the purple box.



5. GitHub Exploration

Collections:

- Machine learning
- Government apps
- Made in Africa
- How to choose your first open source project
- Open journalism
- Design essentials



Github

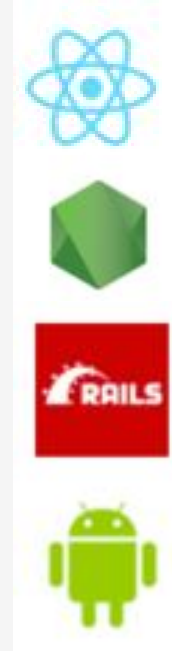
Git

Git Pages

5. GitHub Exploration

Topics:

- Node.js
- Rails
- Android
- React



Github

Git

Git Pages

5. GitHub Exploration

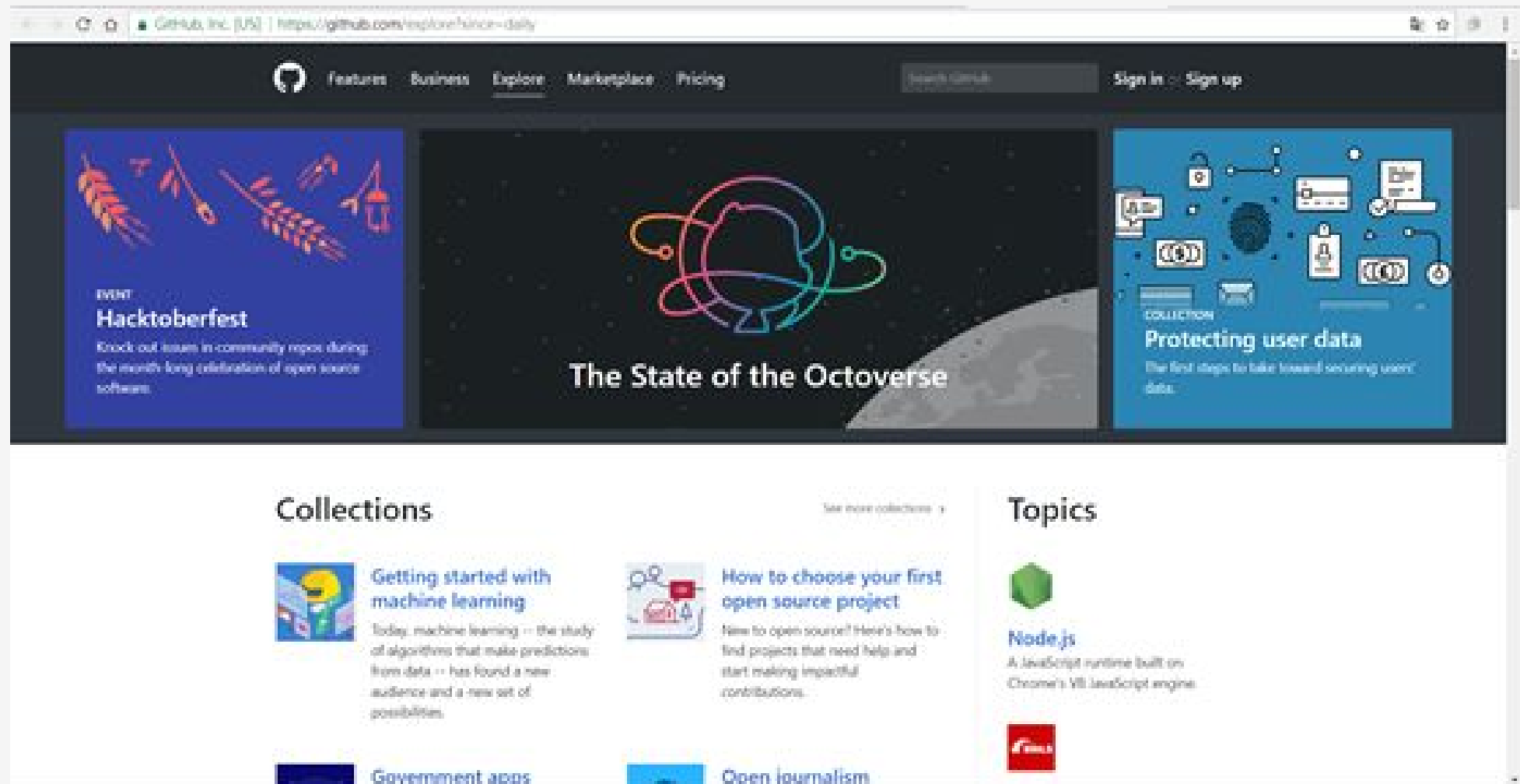
Events:

- GitHub Universe
- GitHub Constellation
- Git Merge
- GitHub Satellite

Github

Git

Git Pages



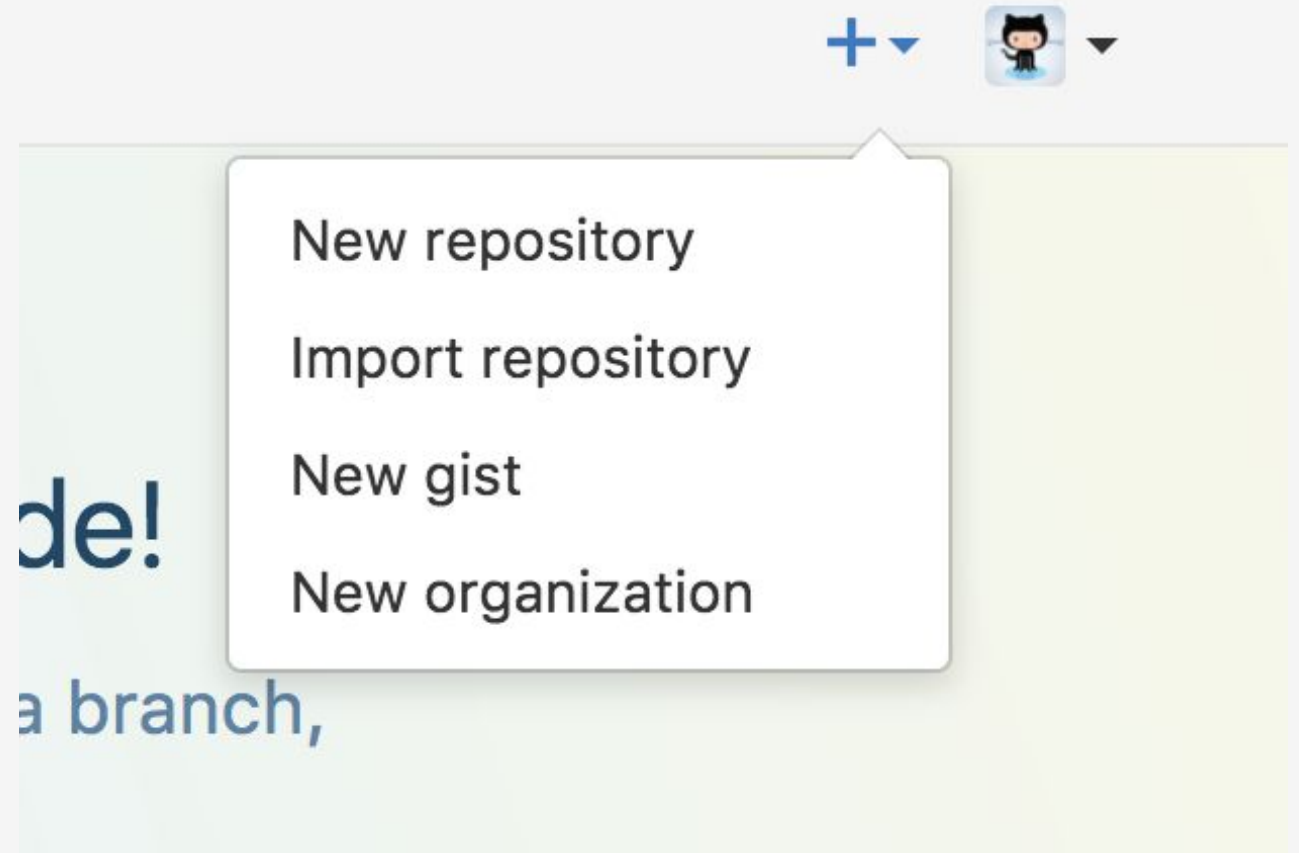
Github

Git

Git Pages

6. Git Pages

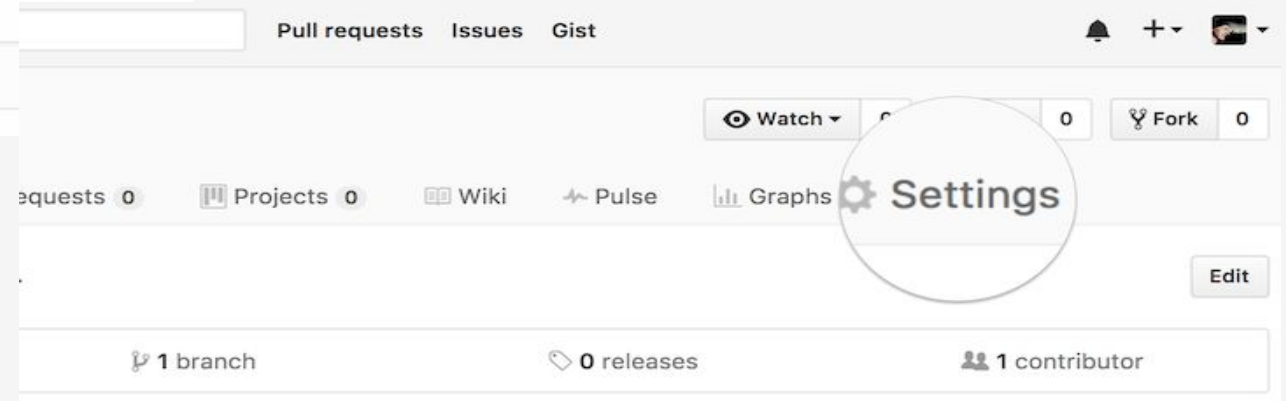
You'll create a new repository to get started.



Github

Git

Git Pages



On the new repository screen, you need to give this repository a special name to generate your website. Your website's files will live in a repository named `username.github.io` (where "username" is your actual GitHub user name). To begin setting up your site, you have to open the Settings tab

Github

Git

Git Pages

If you scroll down on the settings page, you'll see the **GitHub Pages** section near the bottom. Click the **Choose a theme** button to start the process of creating your site.

GitHub Pages

Your site is ready to be published at <https://octocat.github.io/>.

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

Source
Your GitHub Pages site is currently being built from the `master` branch. [Learn more.](#)

master branch ▾

Save

User pages must be built from the `master` branch.

Theme chooser
Select a theme to build your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain
Custom domains allow you to serve your site from a domain other than `octotester-sj.github.io`. [Learn more.](#)

Save

☒ **Enforce HTTPS** — Required for your site because you are using the default domain (`octotester-sj.github.io`)
HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.
When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Github

Git

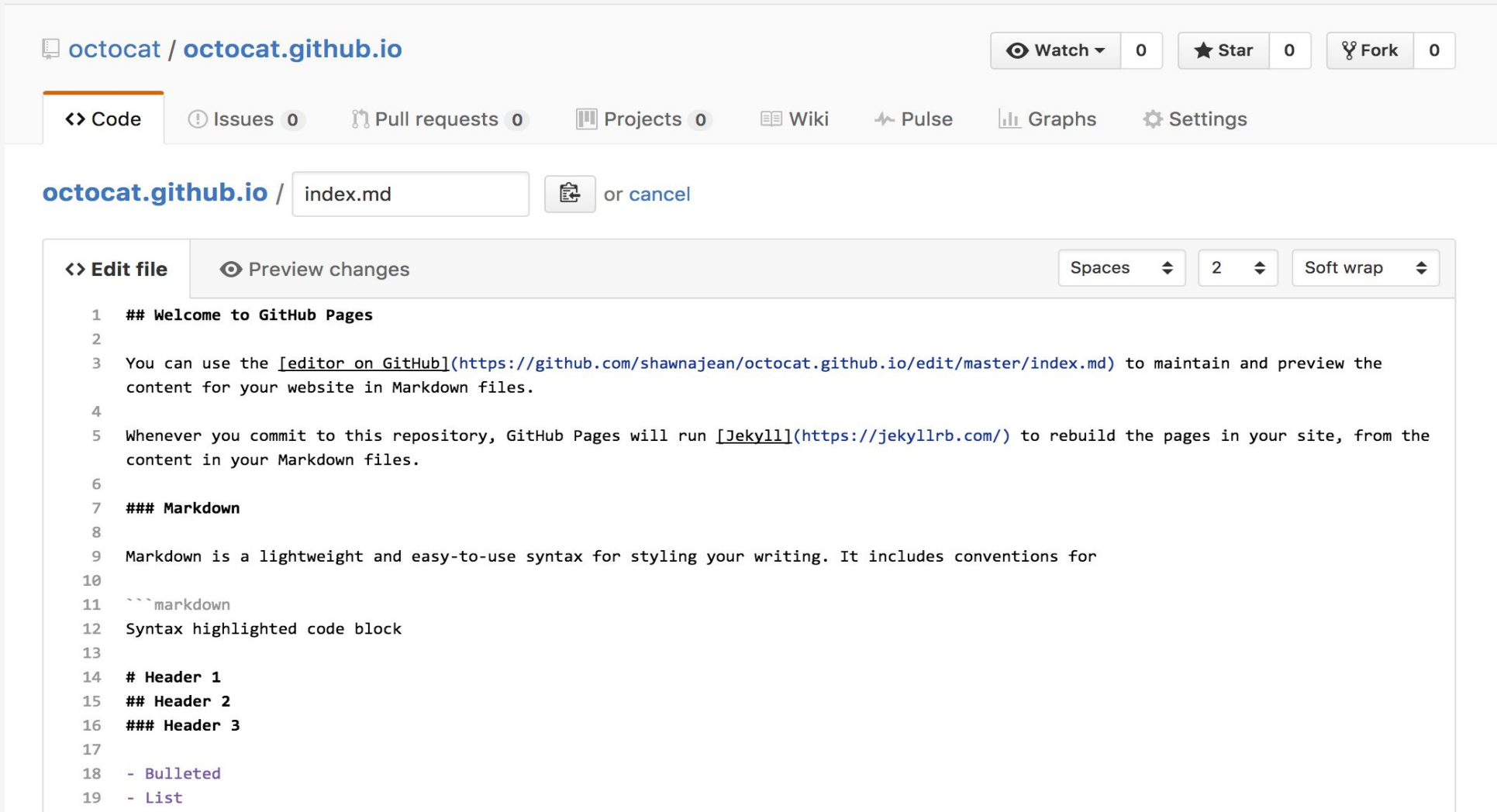
Git Pages

Once you've clicked the button, you'll be directed to the Theme Chooser. You'll see several theme options in a carousel across the top of the page. Click on the images to preview the themes. Once you've selected one, click **Select theme** on the right to move on. It's easy to change your theme later, so if you're not sure, just choose one for now.

Github

Git

Git Pages



The screenshot shows the GitHub interface for the repository `octocat / octocat.github.io`. At the top, there are buttons for `Watch`, `Star`, and `Fork`, each with a count of 0. Below these are navigation tabs for `Code`, `Issues`, `Pull requests`, `Projects`, `Wiki`, `Pulse`, `Graphs`, and `Settings`. The `Code` tab is selected, and the file `index.md` is open in the editor. The editor has tabs for `Edit file` and `Preview changes`. The code in `index.md` is as follows:

```
1  ## Welcome to GitHub Pages
2
3  You can use the [editor on GitHub](https://github.com/shawnajean/octocat.github.io/edit/master/index.md) to maintain and preview the
4  content for your website in Markdown files.
5
6  Whenever you commit to this repository, GitHub Pages will run [Jekyll](https://jekyllrb.com/) to rebuild the pages in your site, from the
7  content in your Markdown files.
8
9  ### Markdown
10
11  Markdown is a lightweight and easy-to-use syntax for styling your writing. It includes conventions for
12
13  ```markdown
14  Syntax highlighted code block
15
16  # Header 1
17  ## Header 2
18  ### Header 3
19
20  - Bulleted
21  - List
```

Now, you'll write your content. This part requires a bit more thought, but you can keep the default content for now, if you'd like.

Github

Git


Git Pages

file.

34

35 **### Support or Contact**

36



Commit changes

Add content to new Pages site

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a **new branch** for this commit and start a pull request. [Le](#)

Commit changes

Cancel

Once you're finished editing, scroll down to the bottom of the page and click **Commit changes**.

Github

Git

Git Pages


```
cout<<"Thank you for attending my workshop\n";
```

Github

Git

Git Pages