# AEC- Project Managment with GIT (1:0:0)-AEC59

By,
Uzma Sulthana

# Syllabus

| Continuous Internal Evaluation (CIE) : 50 Marks | | |
|---|---|---|
| **Assessment Tools** | Marks | Course Outcomes addressed |
| **Internal Test-I (CIE-I)** | 30 | CO1,CO2 & CO3 |
| **Internal Test-II CIE-II)** | 30 | CO4 & CO5 |
| **Average of the two CIE will be taken for 30 marks** | | |
| **Other Components** | | |
| **Skill to troubleshoot and solve Git- related issues** | 20 | CO1, CO2, CO3, CO4 & CO5 |
| **The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Quiz-I & II +Marks scored in Assignment –I** | | |
| **Semester End Examination (SEE)** | | |
| **Course End Examination (Answer One full question from each Unit-Internal Choice)** | 100 | CO1, CO2, CO3 CO4 and CO5 |

## Syllabus

- **Unit 1:** Getting Started with GitHub.com, Starting your first solo project.

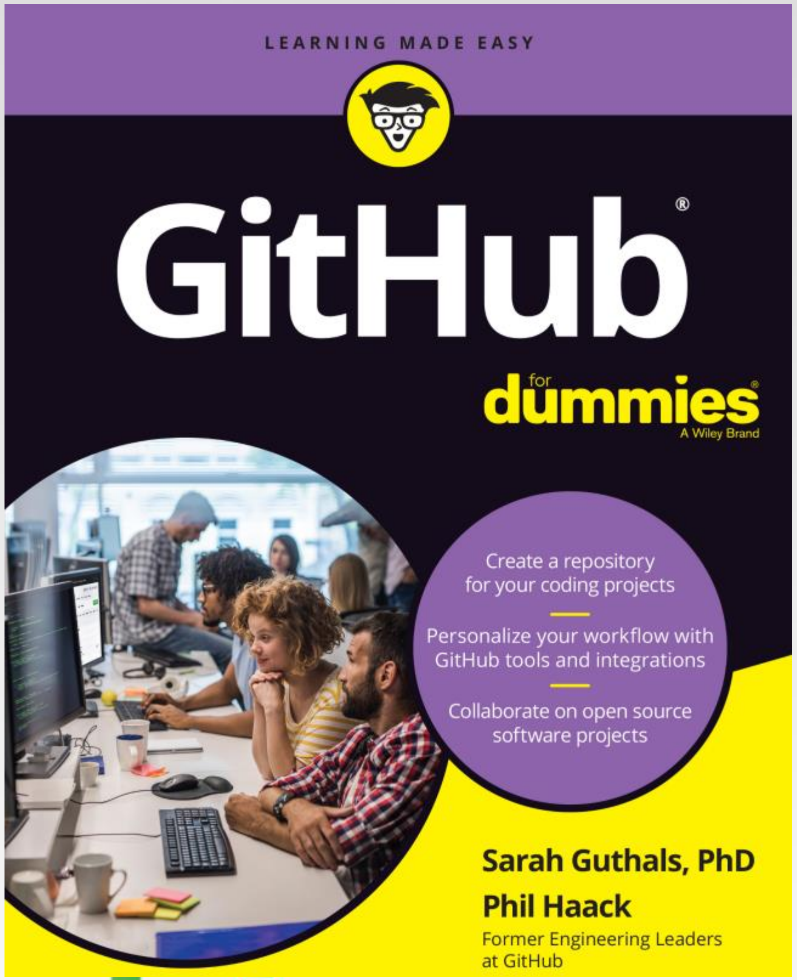
- **Unit 2:** Continuation of first solo project, Contributing to Your first Project

# Syllabus

- **Unit 3**: Continuation of Unit 2 , Manage and Contribute to Large Projects.

- **Unit 4:** Continuation of Unit 3, Make GitHub Work For You.

- **Unit 5:** The GitHub Ecosystem, The parts of Tens



LEARNING MADE EASY

GitHub®

for dummies®
A Wiley Brand

Create a repository for your coding projects

Personalize your workflow with GitHub tools and integrations

Collaborate on open source software projects

Sarah Guthals, PhD
Phil Haack
Former Engineering Leaders at GitHub

## UNIT-1: GITHUB.COM

# Contents

## Understanding the Git in GitHub

### -Introducting GitHub

- **GitHub creates an environment**
  - allows to store code on a remote server
  - Provided the ability to share code with other people
  - Easy for more than one person to add, modify, or delete code to the same file and project, while keeping one source of truth for that file .

  - Example: **Google Docs** — a place online where you can write code with other people and not have to email different versions back and forth.

- What makes **GitHub work behind the scenes is Git**

# Understanding the Git in GitHub

## -Understanding Version control

**Definition:**

- Version control systems (also known as **source control management, or SCM**) are software that keep track of each version of each file in a coding project, a timestamp for when that version was created, and the author of those changes.

# Understanding the Git in GitHub

## -Understanding Version control

**The basic workflow of coding with version control system support is as follows:**

1. Create a project, typically in a folder on your computer.

2. Tell your version control system of choice to track the changes of your project/folder.

3. Each time your project is in a working state, or you’re going to walk away from it, tell your version control system of choice to save it as the next version.

4. If you ever need to go back to a previous version, you can ask your version control system to revert to whichever previous version you need.

# Understanding the Git in GitHub

## -Git Version control

**GitHub, as the same would suggest, is built on Git.**

- **Git** is a type of version control system, and it is free and open source, which means that anyone can use it, build on top of it, and even add to it.
- Try simple git on the terminal:
  - Mac or Linux, a terminal is already installed on your computer.
  - Windows computer, install **Git for Windows**
    - Just go to https://gitforwindows.org and click Download to gain access to **Git Bash**, an emulator that allows you to interact with Git just like you would on a Linux or Mac terminal.
    - You also get Git GUI, which gives you a user interface for almost all Git commands you might type into Git Bash, and shell integration so that you can quickly open Git Bash or Git GUI from any folder.
    - Many developers on Windows prefer to use PowerShell as their terminal environment.

# Understanding the Git in GitHub

## -Git Version control

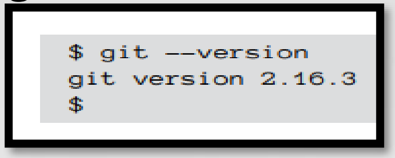
**GitHub, as the same would suggest, is built on Git.**

- **First, find the Terminal application:**
  - » On Mac, you can click the magnifying glass at the top right of your desktop, type Terminal, select the terminal from the list of applications, and press Enter or click it.
  - » On Linux, press Ctrl-Alt-T all at the same time, and the terminal window opens.
  - » On Windows, click the Windows menu in the bottom right of your desktop, search Git Bash, select the Git Bash application from the list of search results, and press Enter or click it.
- **When the application opens**,
  - » type **git --version** in the terminal.
  - » If you have Git installed, you should see a version number, as shown in the following code (the $ should already be on the line, so you do not need to type it).
  - » Otherwise, you can follow the instructions on https://git-scm.com/book/en/v2/GettingStarted-Installing-Git.

# Understanding the Git in GitHub

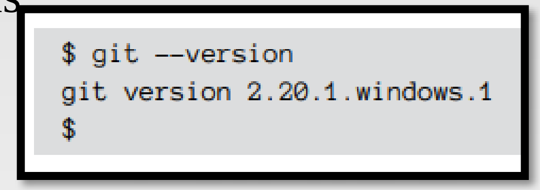
## -Git Version control

- For Mac or Linux, you should see something like this:



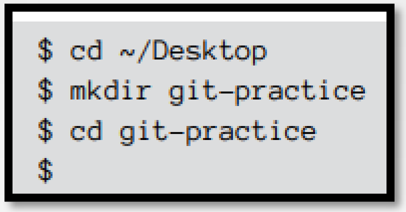
```
$ git --version
git version 2.16.3
$
```

- For Windows, you should see something like this:



```
$ git --version
git version 2.20.1.windows.1
$
```

- Next, using the terminal, go to your desktop and create a new folder called git practice. To do this, you should type the following commands:

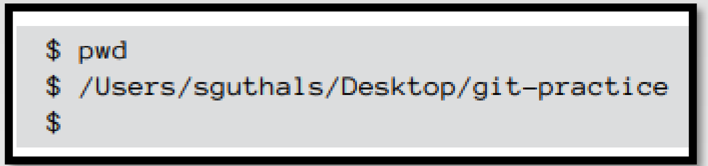


```
$ cd ~/Desktop
$ mkdir git-practice
$ cd git-practice
$
```

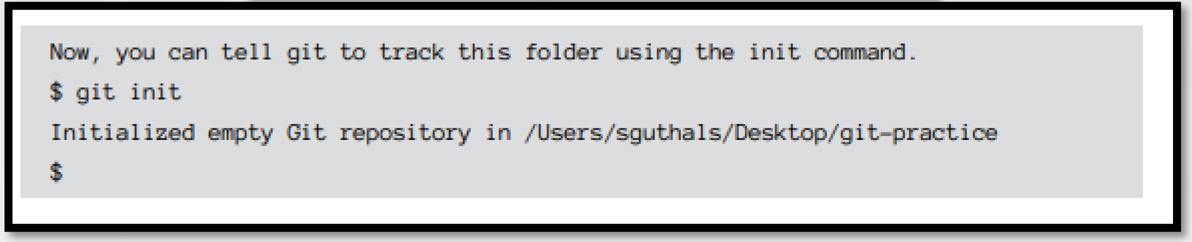
# Understanding the Git in GitHub

## -Git Version control

- If you type pwd, you should see that you are now in the folder git-practice, which is on your desktop. It might look something like this:

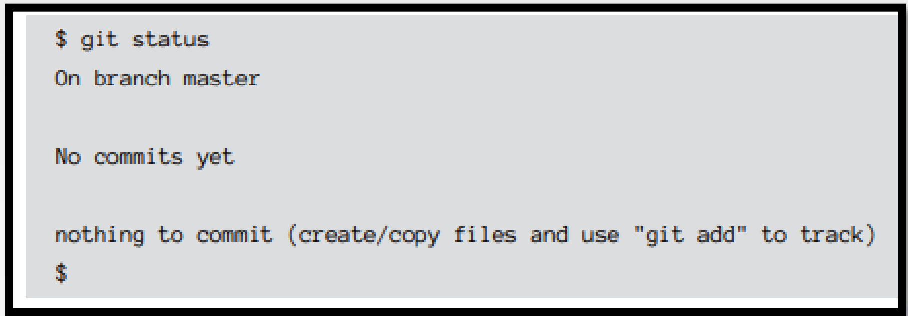


```
$ pwd
$ /Users/sguthals/Desktop/git-practice
$
```



```
Now, you can tell git to track this folder using the init command.
$ git init
Initialized empty Git repository in /Users/sguthals/Desktop/git-practice
$
```

- Then make sure that you have a clean folder. You can check with the status command:

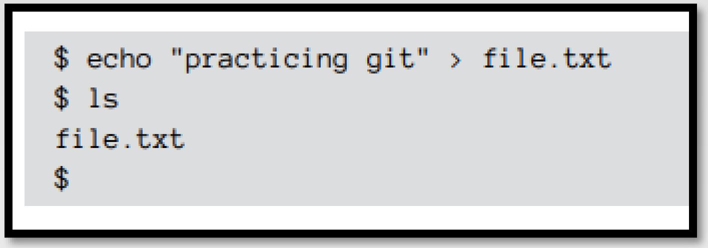


```
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
$
```

# Understanding the Git in GitHub

## -Git Version control

- Then, you can create a file to have Git start tracking and confirm the file is in the folder:



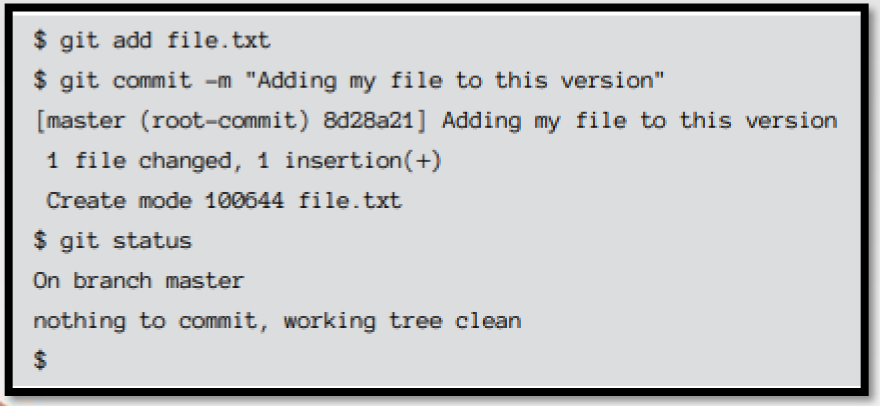
```
$ echo "practicing git" > file.txt
$ ls
file.txt
$
```

- After the folder is open, double-click the file called file.txt, and the file opens with TextEdit on Mac, gedit on Linux, and Notepad on Windows. You can see that the words “practicing git” are actually there.

## Understanding the Git in GitHub

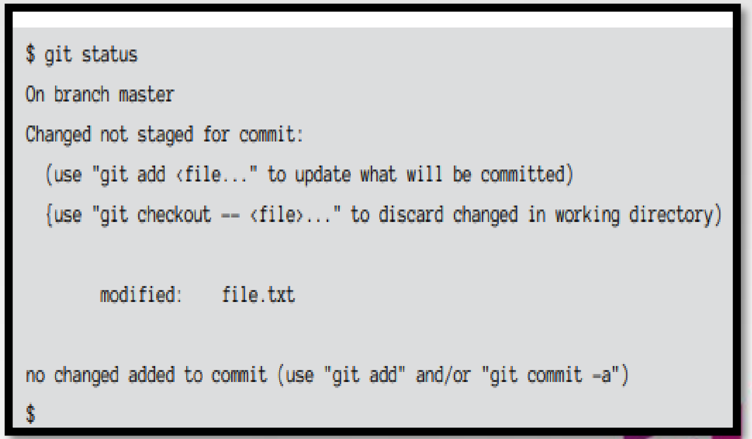
### -Git Version control

- Close the file. Now, you can tell Git that you want to save this as a particular version. Back in the terminal:



```
$ git add file.txt
$ git commit -m "Adding my file to this version"
[master (root-commit) 8d28a21] Adding my file to this version
 1 file changed, 1 insertion(+)
 Create mode 100644 file.txt
$ git status
On branch master
nothing to commit, working tree clean
$
```

- You can make a change to your file in the text file. Open the file again, change the text to say “Hi! I’m practicing git today!” and then click File⇨ Save and close the text application. When you go back to the Terminal to check the status of your project again:

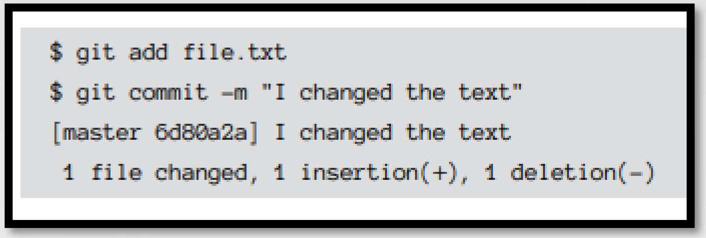


```
$ git status
On branch master
Changed not staged for commit:
  (use "git add <file..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changed in working directory)

        modified:    file.txt

no changed added to commit (use "git add" and/or "git commit -a")
$
```

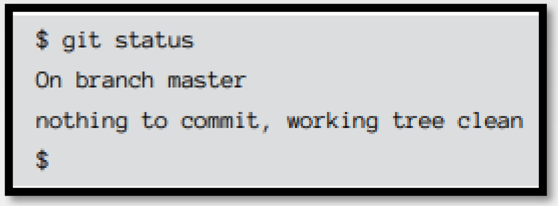
# Understanding the Git in GitHub

## -Git Version control

- Commit this version of your file again and notice that Git recognizes that everything has been saved to a new version:



```
$ git add file.txt
$ git commit -m "I changed the text"
[master 6d80a2a] I changed the text
 1 file changed, 1 insertion(+), 1 deletion(-)
```

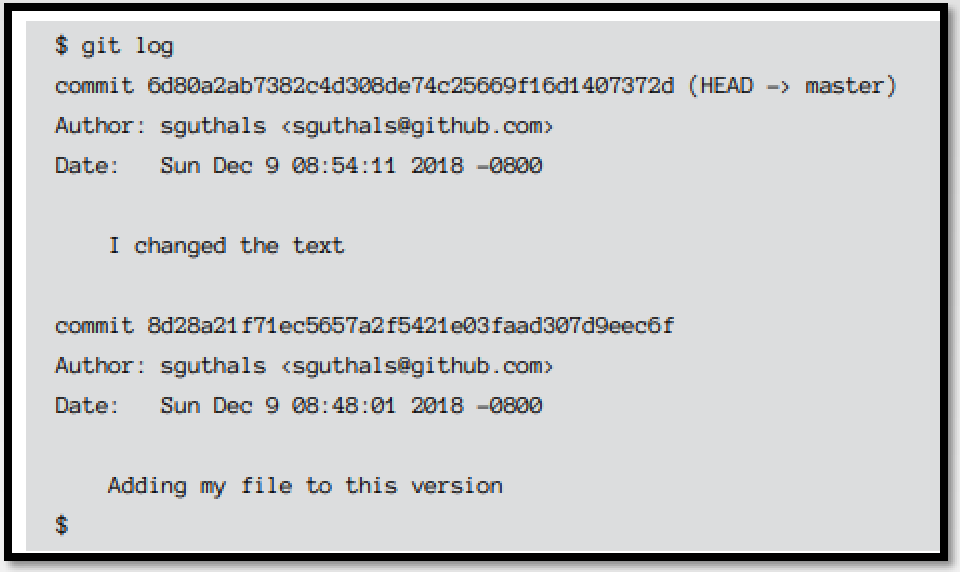


```
$ git status
On branch master
nothing to commit, working tree clean
$
```

# Understanding the Git in GitHub

## -Git Version control

- Say that you actually want to go see the original change; when you added “practicing git”. First, get the log of all the commits you have made:

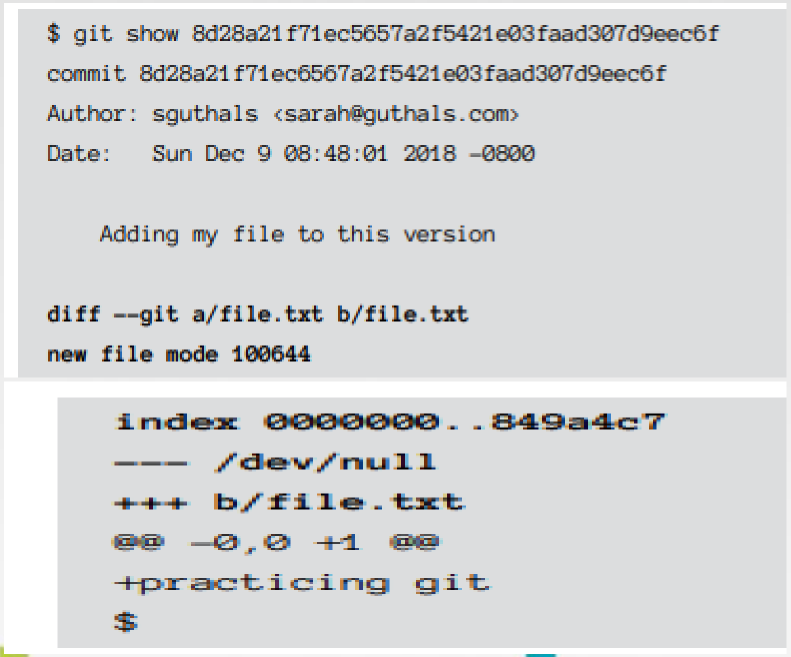


```
$ git log
commit 6d80a2ab7382c4d308de74c25669f16d1407372d (HEAD -> master)
Author: sguthals <sguthals@github.com>
Date:   Sun Dec 9 08:54:11 2018 -0800

    I changed the text

commit 8d28a21f71ec5657a2f5421e03faad307d9eec6f
Author: sguthals <sguthals@github.com>
Date:   Sun Dec 9 08:48:01 2018 -0800

    Adding my file to this version
$
```

# Understanding the Git in GitHub

## -Git Version control

- Then ask Git to show you the first commit you made (the bottom most one). Make sure that you're typing your unique commit hash. In this book, the hash starts with 8d28a2. Make sure you type the entire hash that appears in your Git log:



```
$ git show 8d28a21f71ec5657a2f5421e03faad307d9eec6f
commit 8d28a21f71ec6567a2f5421e03faad307d9eec6f
Author: sguthals <sarah@guthals.com>
Date:   Sun Dec 9 08:48:01 2018 -0800

    Adding my file to this version

diff --git a/file.txt b/file.txt
new file mode 100644
index 0000000..849a4c7
--- /dev/null
+++ b/file.txt
@@ -0,0 +1 @@
+practicing git
$
```

# Understanding the Git in GitHub

## -Git Version control

### Git branching by collaborator:

- Git is different from other version control systems because it has fast branching, shown in Figure 1-1.
- **Branching** is a Git function that essentially copies code (each branch is a copy of the code), allows you to make changes on a specific copy, and then merges your changes back into the main (master) branch.
- When you're writing code, you will add files and commit changes to your master branch.

# Understanding the Git in GitHub

## -Git Version control

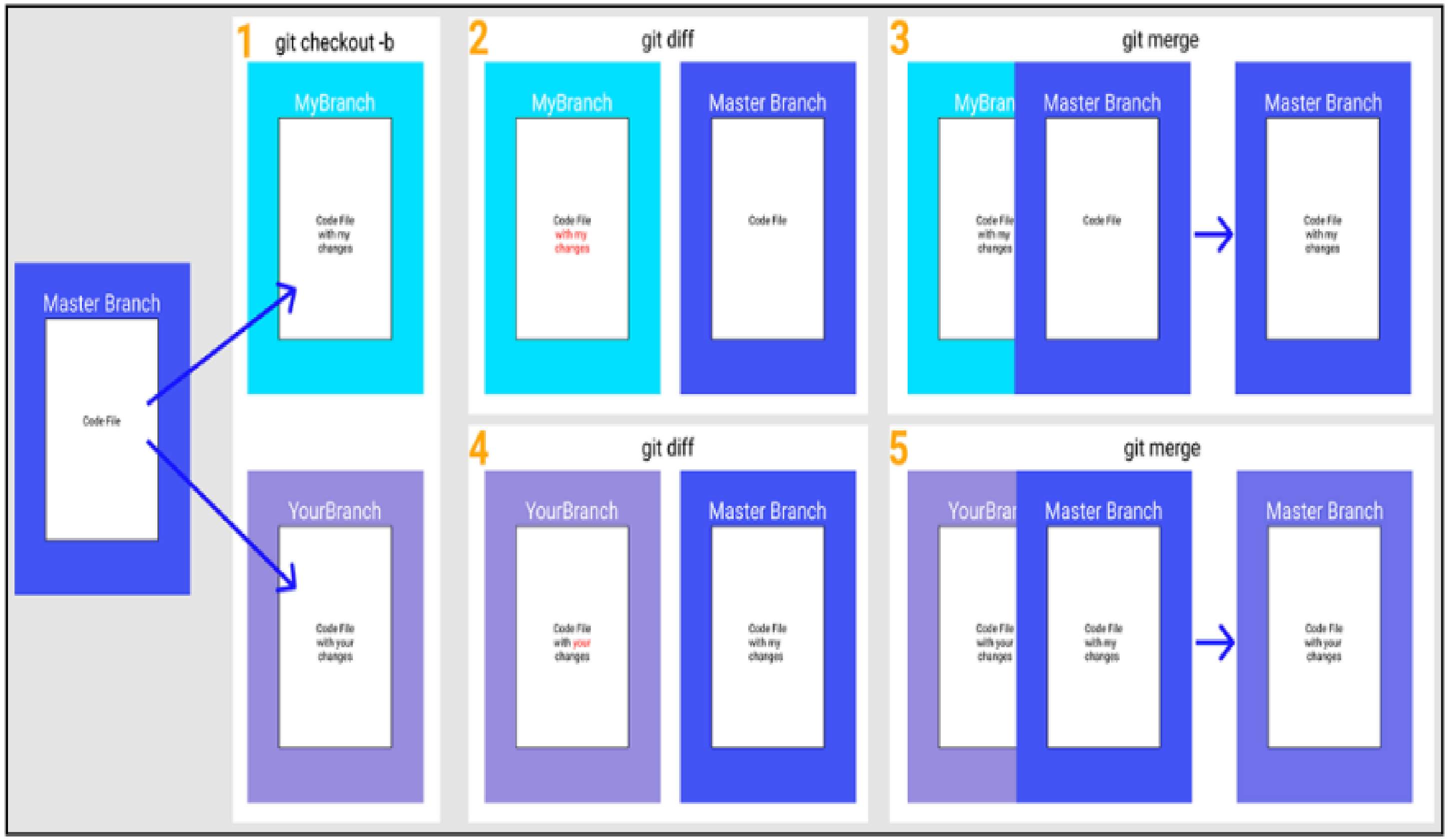


**FIGURE 1-1:** Example workflow for Git branches.

# Understanding the Git in GitHub

## -Git Version control

### Git branching by feature:

- Another common way to use branching is to have each feature that you develop be on a different branch, regardless of the collaborator who is building the feature.
- You can extend the idea of branching by feature to also have one branch that is your production branch. This branch is what your users will see. Then you can have a development branch, which is one that you can merge features into without changing what your users see
- This type of branching allows you to build a lot of different features, merge them each into the development branch, make sure they all work the way you want, and then merge the development branch into the production branch when you know it's ready for your users.

# Understanding the Git in GitHub

## -Git Version control

### Git branching for experimentation:

- You can also create branches to test to see whether something works and then completely throw the branch away.
- Example: This type of branching can be useful if you want to try a completely new layout of a website

# Understanding the Git in GitHub

## -Git Version control

**Git features that GitHub supports:**

» **Repository:** Each repository contains all the files and folders related to your project and gives you control of permissions and collaborators' interaction with your code.

» **Clone:** When you want to make changes to your code, you will often want to create a copy, or clone, of the project on your local computer. The cloned project is still tightly connected with the version on GitHub.com; it's just your local copy.

» **Fork:** Forking a project is when you create your own copy of the entire project. When you fork a project, GitHub.com creates a new repository with your copy of all the files. You can still suggest changes back to the original copy, but you can also take your version and go in a new direction.

# Understanding the Git in GitHub

## -Git Version control

### Git features that GitHub supports:

» **Branches:** GitHub.com supports branching and even provides a useful tool — pull requests — to compare the diff between branches and merge branches.

» **Commits:** GitHub.com tracks all the commits that you push to its servers and gives you an easy interface for browsing the code at different branches and different commits

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**The following steps guide you through signing up for a free, individual GitHub.com account:**

**1. Go to GitHub.com and fill out the Sign Up form.**

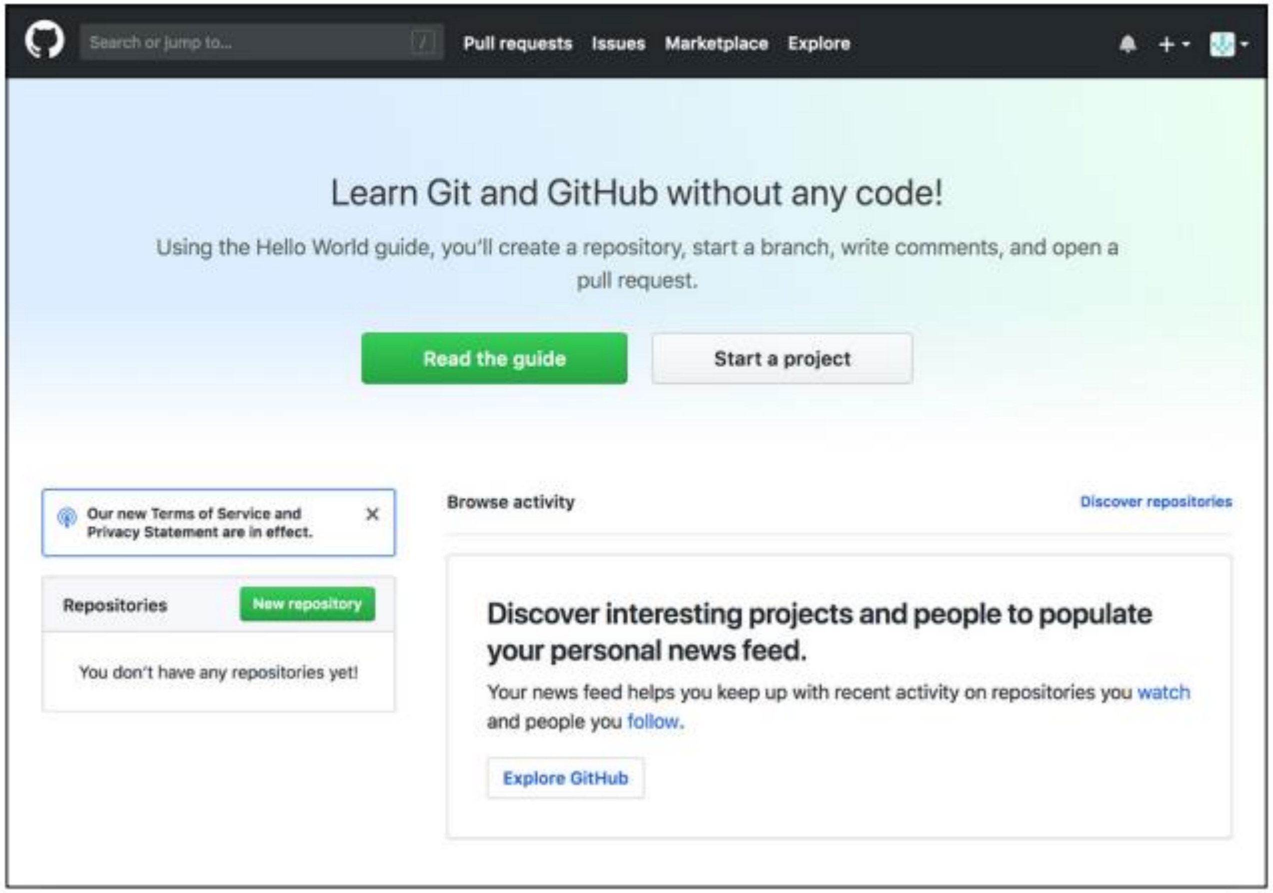
**2. Choose the plan you want.**

For the purpose of this book, you can use the Free plan. You can always upgrade to a paid plan later if you decide you want to have more than three collaborators for your private repository and other pro GitHub features.

**3. Complete the brief survey.**

This survey helps GitHub understand who is using the software and helps them support workflows specific to their users.

You’re now at the home page, shown in Figure 1-2.



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Our new Terms of Service and Privacy Statement are in effect.

You don't have any repositories yet!

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch and people you follow.

**FIGURE 1-2:** The GitHub.com home page when you're logged in.

# Understanding the Git in GitHub

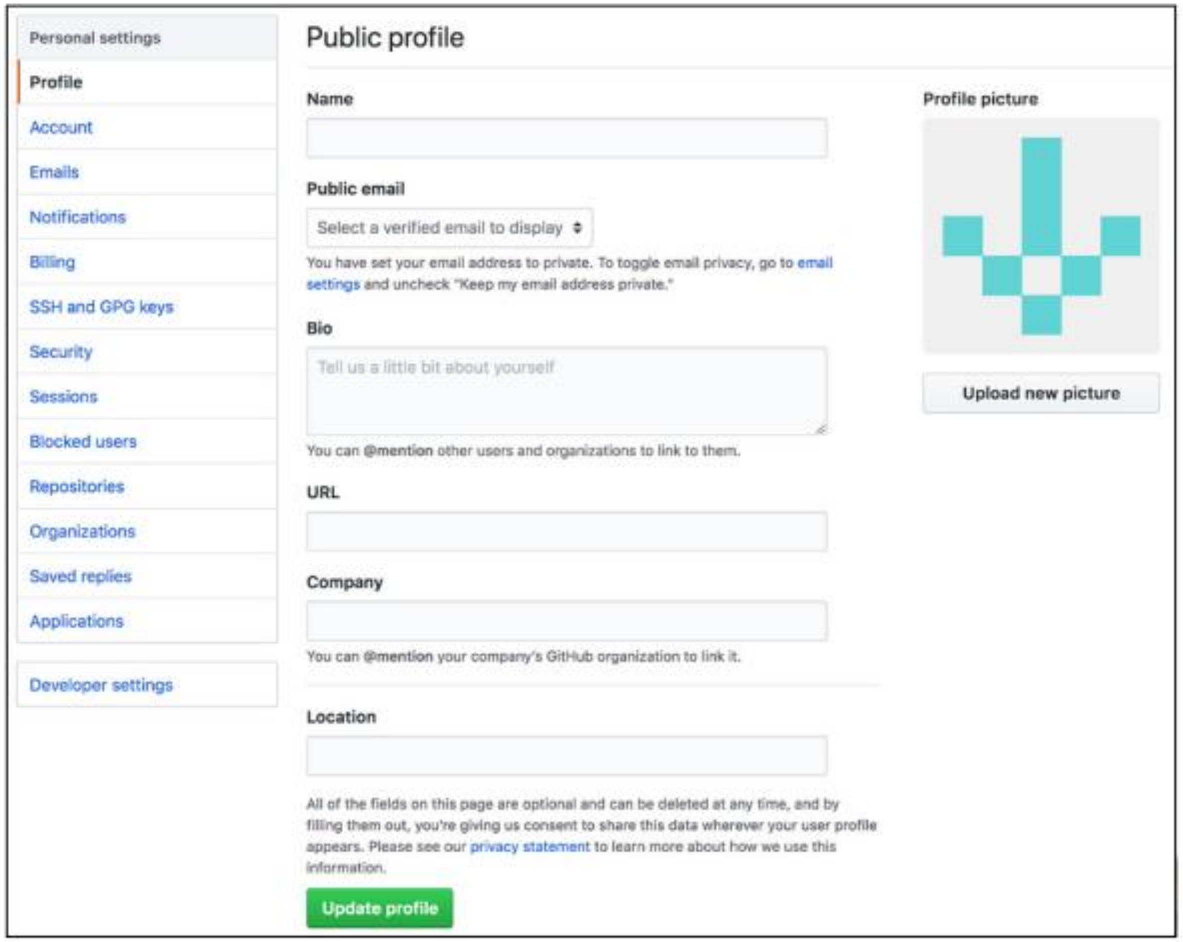
## -Signing up for GitHub.com

**Personalizing Your GitHub.com Account:**

To complete your GitHub.com profile:

1. Click the avatar icon in the top right corner and choose Your Profile.
2. Click Edit Profile on the page that appears.
3. Fill out the form on the Personal Settings page, shown in Figure 1-3.
4. Click Update profile when you’re finished.

On the Personal Settings page, you can also adjust a number of different settings to continue personalizing your account.



Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- Sessions
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

Developer settings

Public profile

Name

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Company

You can @mention your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our privacy statement to learn more about how we use this information.

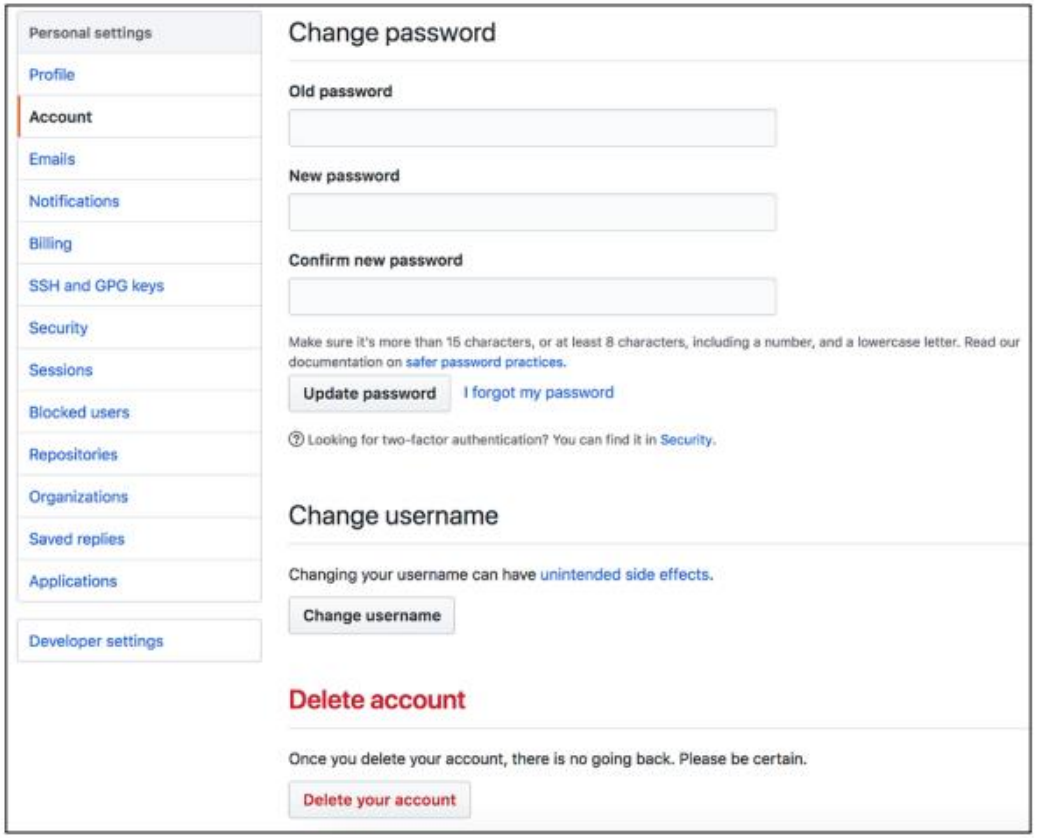
**FIGURE 1-3:** The Personal Settings page.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Account:**

In Account settings, you can change your password, change your username, or delete your account.



Change password

Old password

New password

Confirm new password

Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter. Read our documentation on safer password practices.

Looking for two-factor authentication? You can find it in Security.

Change username

Changing your username can have unintended side effects.

Delete account

Once you delete your account, there is no going back. Please be certain.

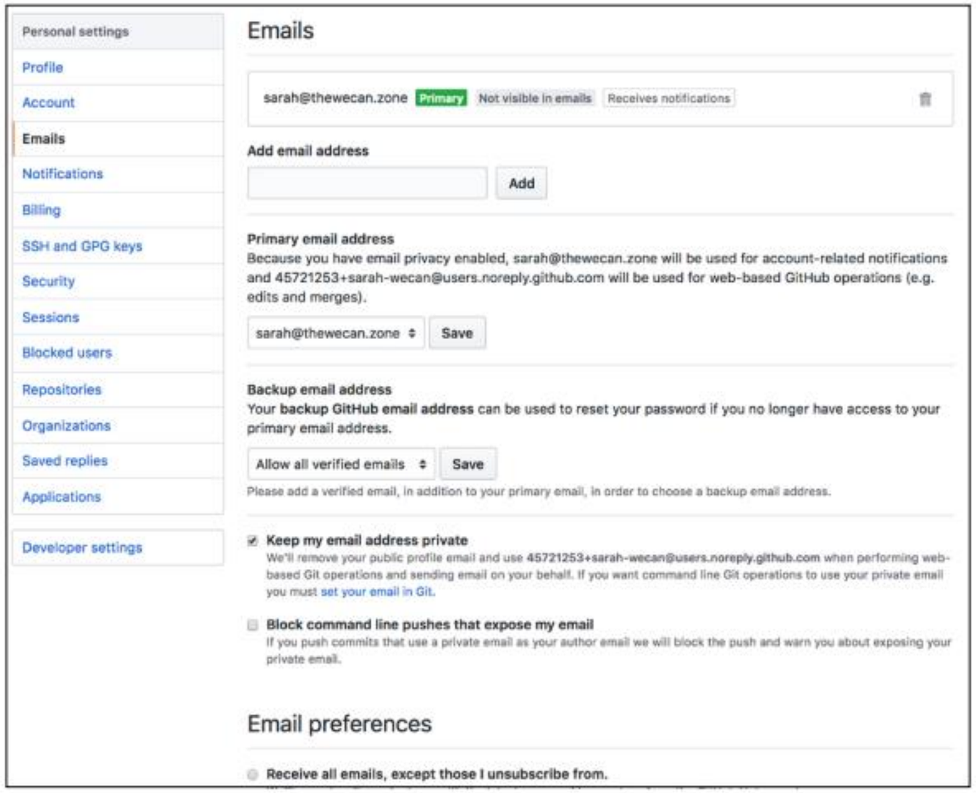
**FIGURE 1-4:** Account settings.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Emails:**

GitHub allows you to link multiple email addresses to your account. Notice that you can add email addresses, keep your email address private, and even block Git commands that may expose your email address.



Emails

sarah@thewecan.zone Primary Not visible in emails Receives notifications

Add email address

[ ] Add

Primary email address
Because you have email privacy enabled, sarah@thewecan.zone will be used for account-related notifications and 45721253+sarah-wecan@users.noreply.github.com will be used for web-based GitHub operations (e.g. edits and merges).

sarah@thewecan.zone Save

Backup email address
Your **backup GitHub email address** can be used to reset your password if you no longer have access to your primary email address.

Allow all verified emails Save

Please add a verified email, in addition to your primary email, in order to choose a backup email address.

☑ **Keep my email address private**
We'll remove your public profile email and use **45721253+sarah-wecan@users.noreply.github.com** when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must set your email in Git.

☐ **Block command line pushes that expose my email**
If you push commits that use a private email as your author email we will block the push and warn you about exposing your private email.

Email preferences

○ Receive all emails, except those I unsubscribe from.

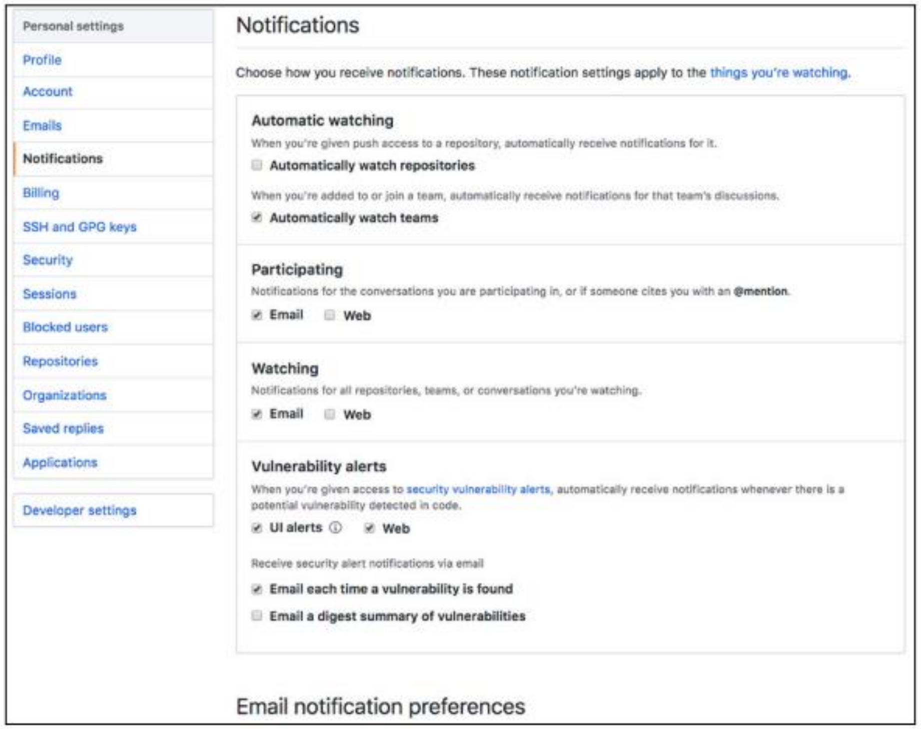
**FIGURE 1-5:** The Email settings.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Notifications:**

Notifications can get really overwhelming. Though you can choose your level of granularity for receiving notifications per repository, this page creates your default preferences for notifications.



Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys
Security
Sessions
Blocked users
Repositories
Organizations
Saved replies
Applications

Developer settings

Notifications

Choose how you receive notifications. These notification settings apply to the things you're watching.

Automatic watching

When you're given push access to a repository, automatically receive notifications for it.

☐ Automatically watch repositories

When you're added to or join a team, automatically receive notifications for that team's discussions.

☑ Automatically watch teams

Participating

Notifications for the conversations you are participating in, or if someone cites you with an **@mention**.

☑ Email ☐ Web

Watching

Notifications for all repositories, teams, or conversations you're watching.

☑ Email ☐ Web

Vulnerability alerts

When you're given access to security vulnerability alerts, automatically receive notifications whenever there is a potential vulnerability detected in code.

☑ UI alerts ☑ Web

Receive security alert notifications via email

☑ Email each time a vulnerability is found

☐ Email a digest summary of vulnerabilities

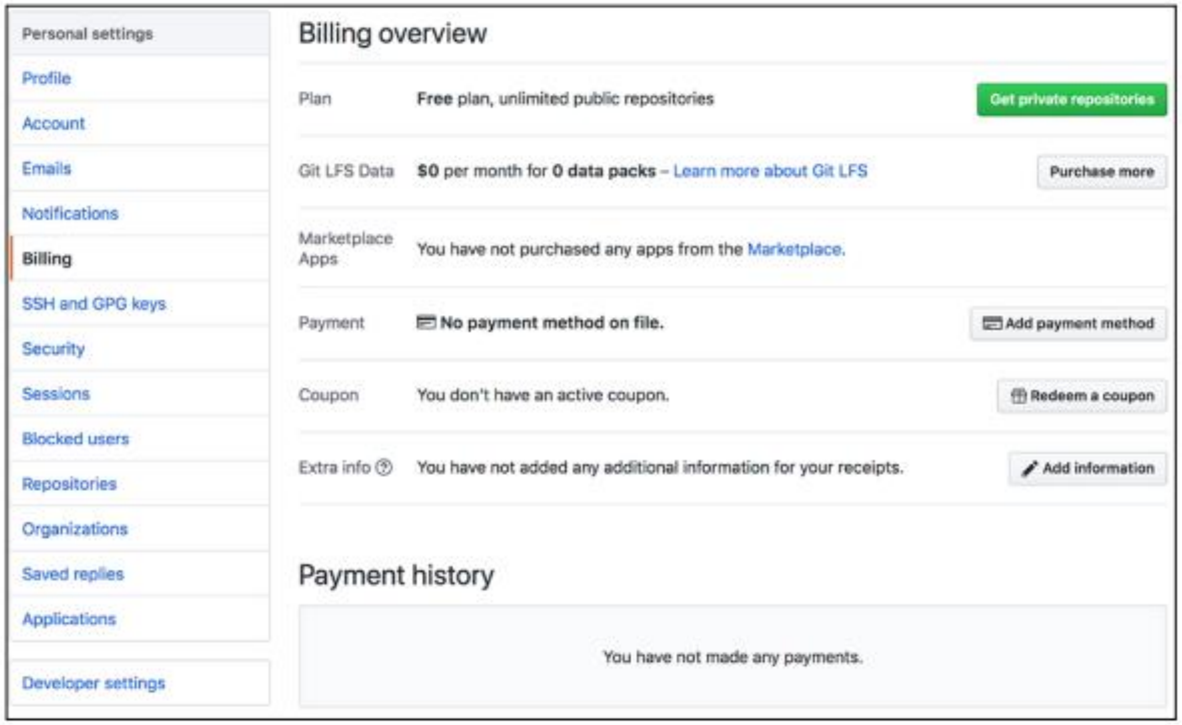
Email notification preferences

**FIGURE 1-6:** The Notifications settings.

## Understanding the Git in GitHub

### -Signing up for GitHub.com

**Billing:**



Billing overview

| | | |
|---|---|---|
| Plan | Free plan, unlimited public repositories | Get private repositories |
| Git LFS Data | $0 per month for 0 data packs – Learn more about Git LFS | Purchase more |
| Marketplace Apps | You have not purchased any apps from the Marketplace. | |
| Payment | No payment method on file. | Add payment method |
| Coupon | You don't have an active coupon. | Redeem a coupon |
| Extra info | You have not added any additional information for your receipts. | Add information |

Payment history

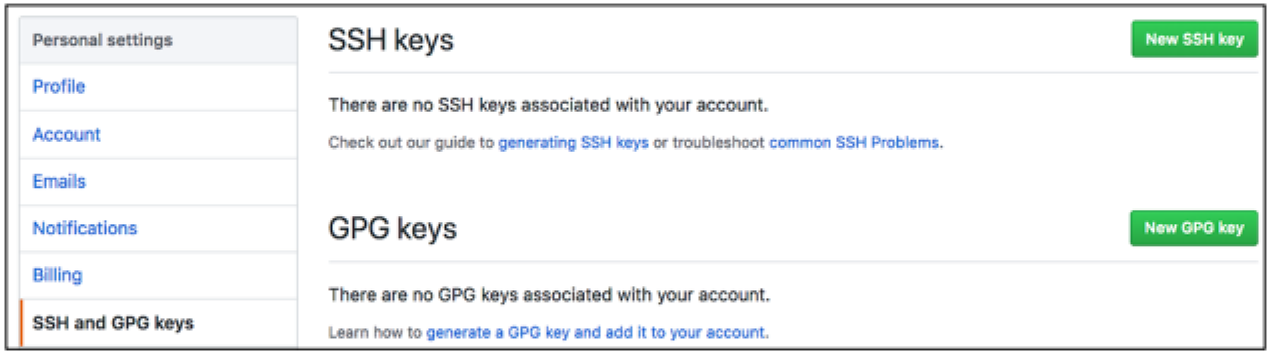
You have not made any payments.

FIGURE 1-7: The Billing settings.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**SSH and GPG keys:**



Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys

SSH keys
New SSH key

There are no SSH keys associated with your account.

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

GPG keys
New GPG key

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

**FIGURE 1-8:** The place to create SSH and GPG keys.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Security**

Not only should your password be complex, but you should also consider enabling two-factor authentication. Two-factor authentication means that when you type the correct password, you're asked to further verify that it is you who is attempting to log in through an app or SMS.

**Sessions**

Sessions allows you to see every computer address, city, and country where you're logged in or connecting to GitHub.com.

**Blocked users**

In the Blocked users settings, you can block users from all your repositories.

**Repositories**

The Repositories section lists all the repositories that you have created or been invited to as a collaborator. You also can leave repositories from this page.

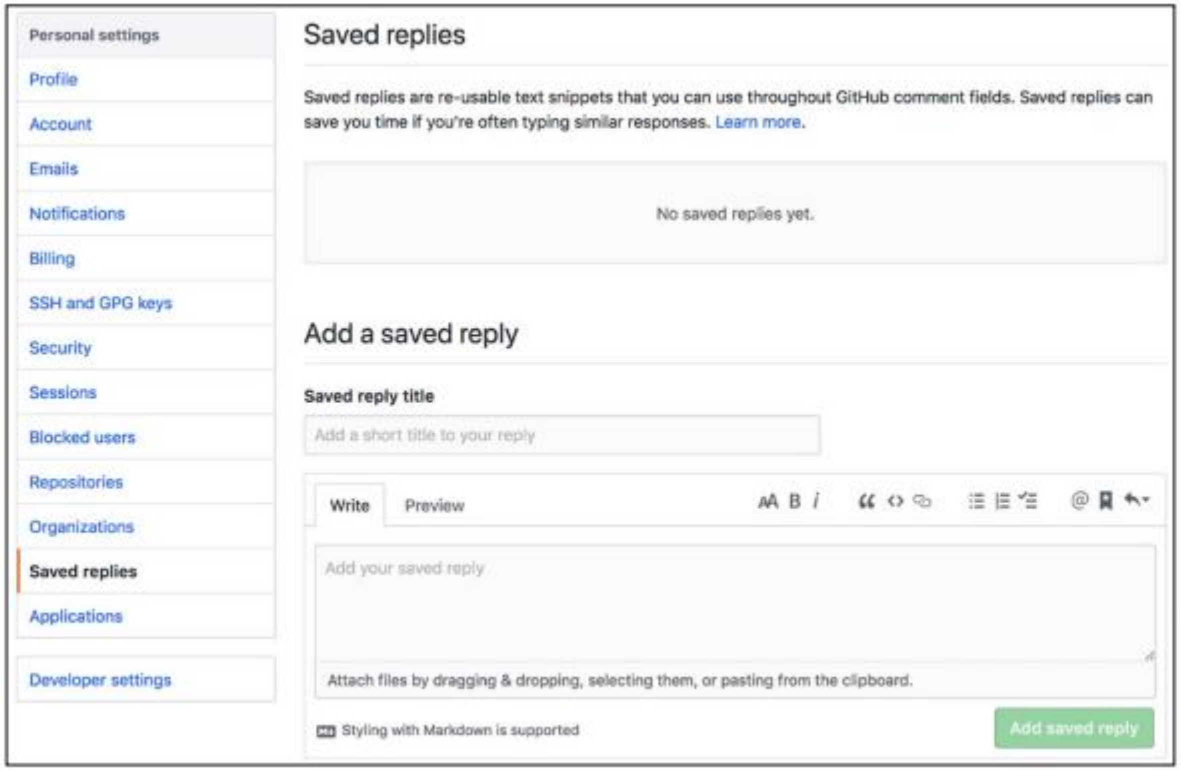
**Organizations**

Organizations enable you to put GitHub users and repositories under similar settings

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Saved replies:**


Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys
Security
Sessions
Blocked users
Repositories
Organizations
Saved replies
Applications
Developer settings

Saved replies

Saved replies are re-usable text snippets that you can use throughout GitHub comment fields. Saved replies can save you time if you're often typing similar responses. Learn more.

No saved replies yet.

Add a saved reply

Saved reply title

Add a short title to your reply

Write Preview

Add your saved reply

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Add saved reply

**FIGURE 1-9:** The Saved replies settings.

## Understanding the Git in GitHub

### -Signing up for GitHub.com

**Applications:**

You can connect three kinds of applications with your GitHub.com account:

» **Installed GitHub apps:** GitHub applications that you are using with your account. One example is GitHub Learning Labs.

» **Authorized GitHub apps:** Applications that you have authorized to access your account. One example is Slack.

» **Authorized OAuth apps:** Applications that you have authenticated with using GitHub credentials. One example is GitHub Desktop.

# Understanding the Git in GitHub

## -Signing up for GitHub.com

**Developer settings:**

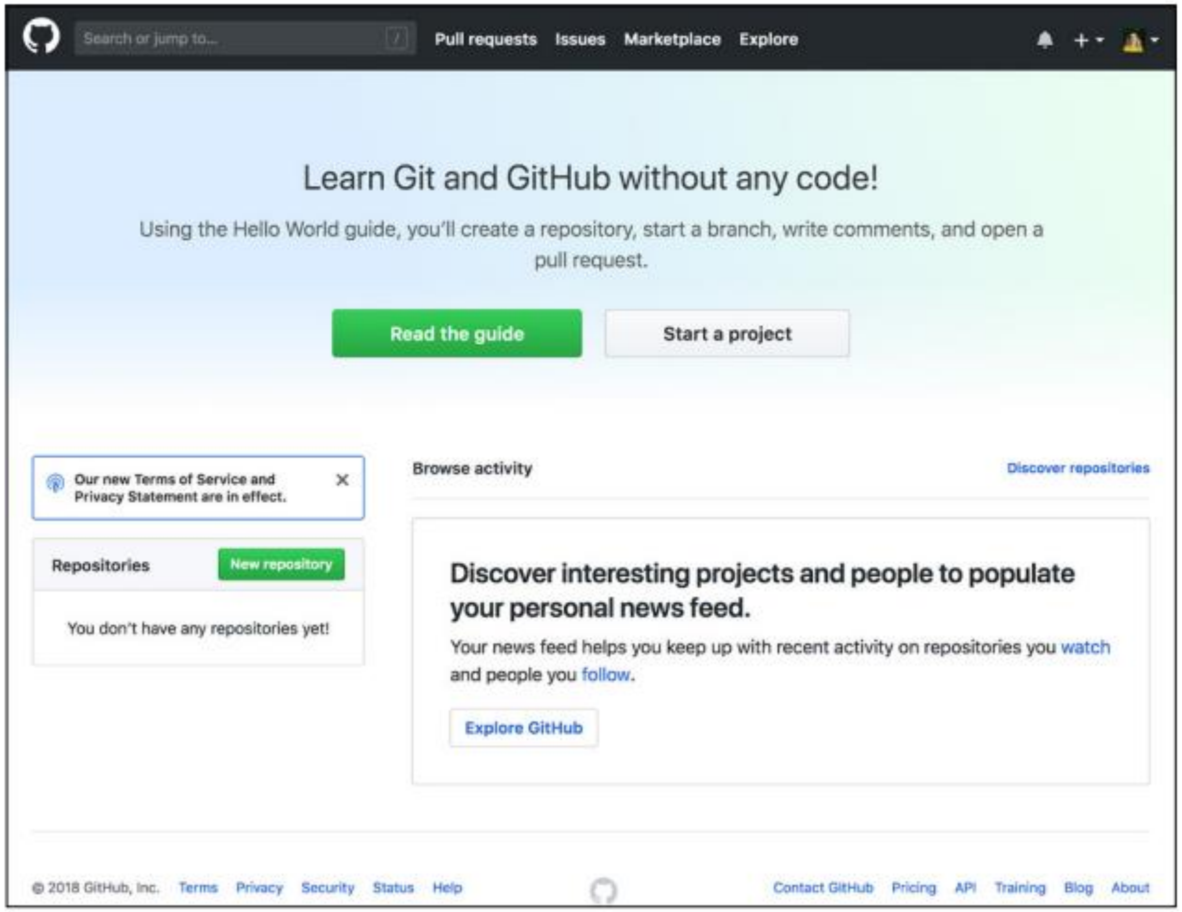
» **OAuth apps:** Applications you have registered to use the GitHub API.

» **GitHub apps:** Applications that integrate with and extend GitHub.

» **Personal access tokens:** Similar to SSH keys, tokens that allow you to access the GitHub API without requiring authentication.

# Chapter -2
# Setting Up Your Collaborative Coding Environment
# -Exploring GitHub.com

The home page of GitHub.com, shown in Figure 2-1, is a great starting point for many tasks, including starting your own project, learning about a topic, or exploring existing repositories.



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

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide
Start a project

Our new Terms of Service and Privacy Statement are in effect.

Repositories
New repository

You don't have any repositories yet!

Browse activity
Discover repositories

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch and people you follow.

Explore GitHub

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

**FIGURE 2-1:** The home page of GitHub.com.

# Chapter -2
# Setting Up Your Collaborative Coding Environment
# -Exploring GitHub.com

1. Exploring GitHub.com
2. Understanding Your Profile
3. Getting to Know GitHub Desktop
4. Setting up GitHub Desktop
5. Introducing Atom

# Chapter -2
# Setting Up Your Collaborative Coding Environment
# -Exploring GitHub.com

1. **Exploring GitHub.com**
   1. **GitHub home page:**
   2. **Search bar:**
   3. **Pull requests:**
   4. **Issues:**
   5. Marketplace:
   6. Explore:
   7. Notifications:
   8. Quick pick:
   9. Account menu:

# Chapter -2
# Setting Up Your Collaborative Coding Environment -Exploring GitHub.com

1. **Exploring GitHub.com**

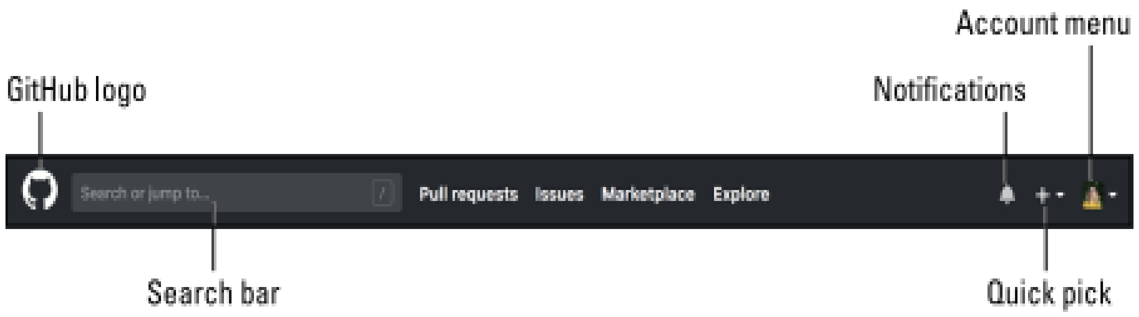
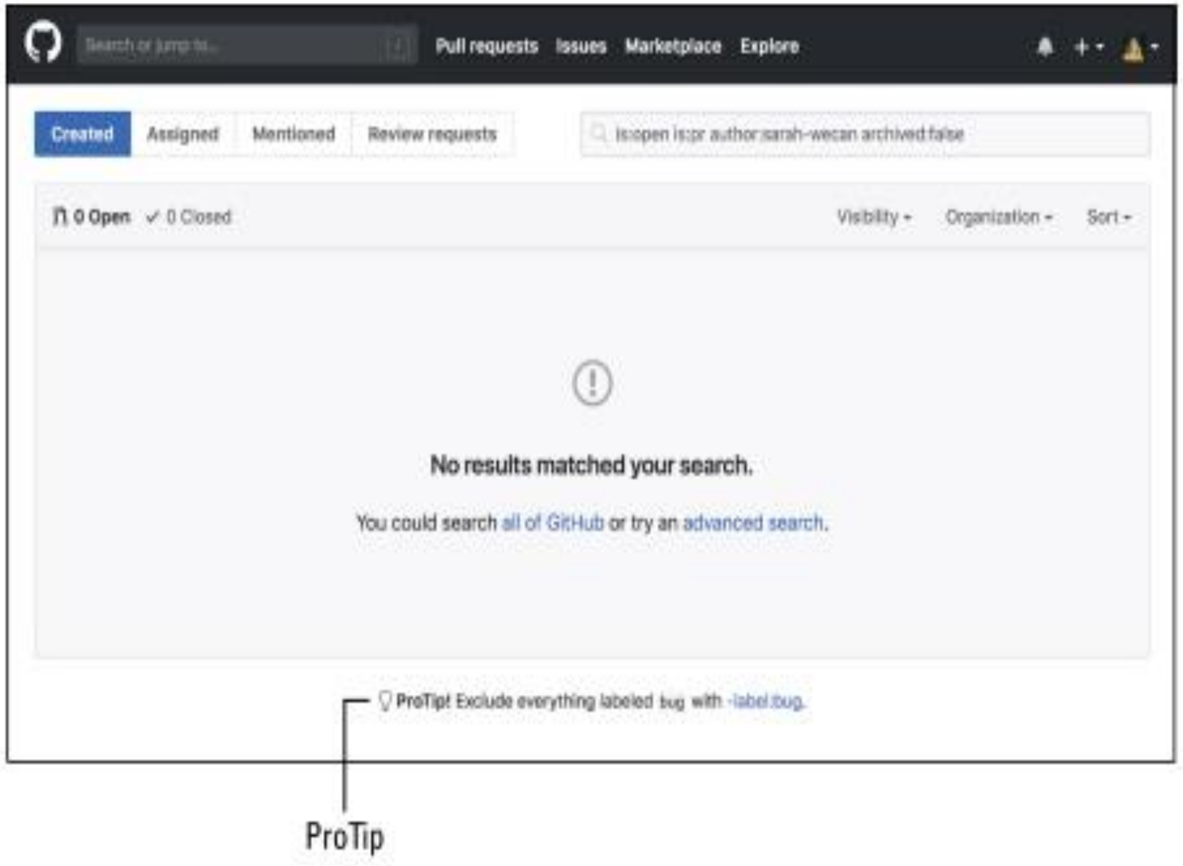


**FIGURE 2-2:** The top menu bar of GitHub.com.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

1. **Exploring GitHub.com**



No results matched your search.

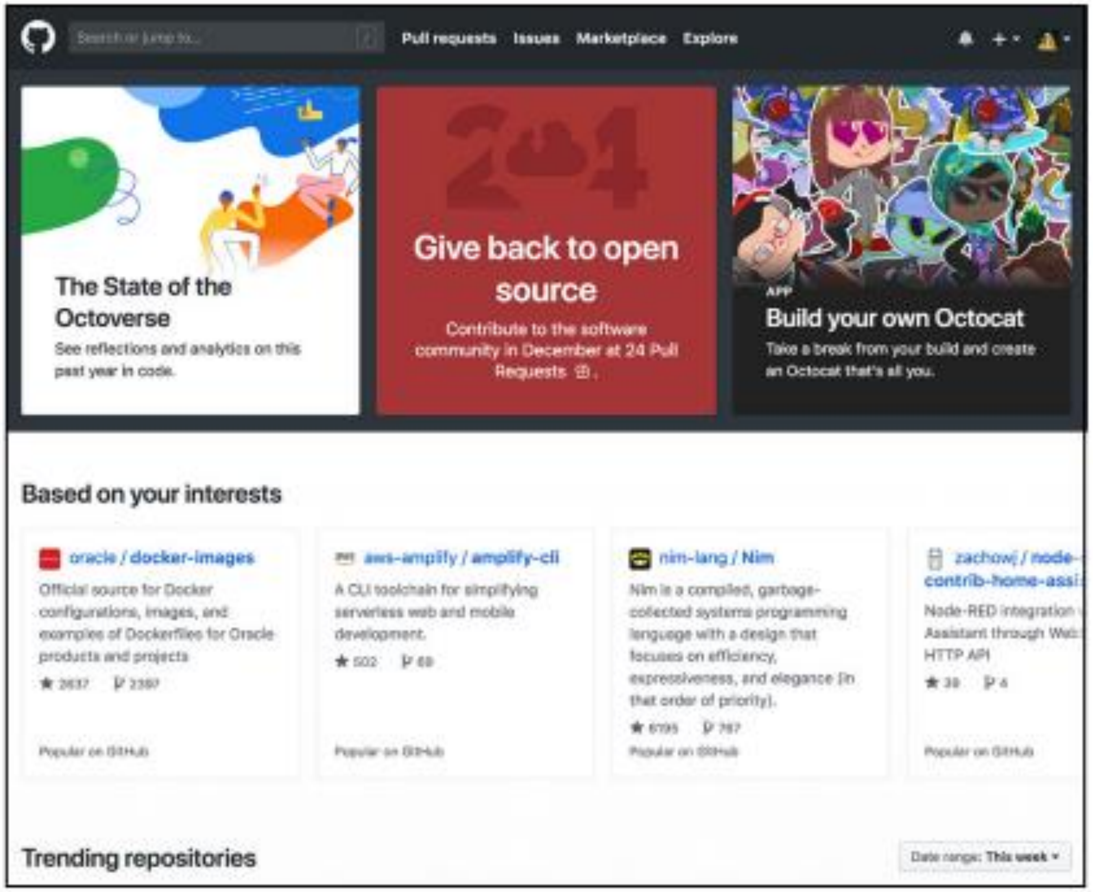
You could search all of GitHub or try an advanced search.

ProTip! Exclude everything labeled bug with -label:bug.

FIGURE 2-3: ProTip found on the Pull request page.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

1. **Exploring GitHub.com**



The State of the Octoverse

See reflections and analytics on this past year in code.

Give back to open source

Contribute to the software community in December at 24 Pull Requests.

APP

Build your own Octocat

Take a break from your build and create an Octocat that's all you.

Based on your interests

oracle / docker-images

Official source for Docker configurations, images, and examples of Dockerfiles for Oracle products and projects

★ 2637 2387

Popular on GitHub

aws-amplify / amplify-cli

A CLI toolchain for simplifying serverless web and mobile development.

★ 502 69

Popular on GitHub

nim-lang / Nim

Nim is a compiled, garbage-collected systems programming language with a design that focuses on efficiency, expressiveness, and elegance (in that order of priority).

★ 6795 787

Popular on GitHub

zachowj / node-contrib-home-assi

Node-RED integration Assistant through Web HTTP API

★ 38 4

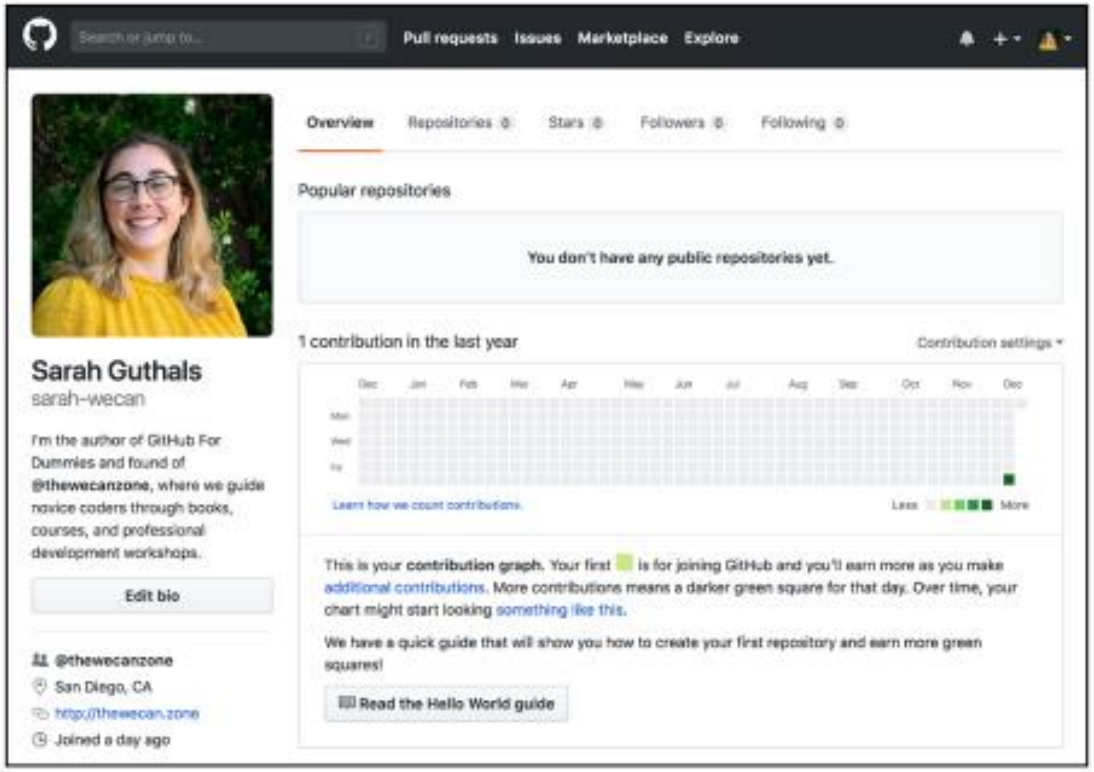
Popular on GitHub

Trending repositories

**FIGURE 2-4:** Curated list of repositories on GitHub.com.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

1. **Understanding Your Profile( Public View)**



Sarah Guthals
sarah-wecan

I'm the author of GitHub For Dummies and found of **@thewecanzone**, where we guide novice coders through books, courses, and professional development workshops.

@thewecanzone
San Diego, CA
http://thewecan.zone
Joined a day ago

Popular repositories

You don't have any public repositories yet.

1 contribution in the last year

Learn how we count contributions.

This is your **contribution graph.** Your first 🟩 is for joining GitHub and you'll earn more as you make additional contributions. More contributions means a darker green square for that day. Over time, your chart might start looking something like this.

We have a quick guide that will show you how to create your first repository and earn more green squares!

FIGURE 2-5: My profile.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 3. Getting to Know GitHub Desktop

1. GitHub Desktop is a free, open source application that makes it easier for Mac and Windows users alike to manage repositories and GitHub connections on their local computer.

**To install the GitHub Desktop on your computer**

**1. Go to https://desktop.github.com and click Download for the platform you're using. :** Google Chrome and a Mac.

Desktop works on a Windows PC as well because it's built using Electron, which allows it to work on both operating systems.

**2. After the download finishes, click the file that was downloaded.**

**The file automatically unzips.**

On Mac, the GitHub Desktop application appears in your Downloads folder, next to the zip file. On Windows, the application immediately opens after you unzip the file.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 3. Getting to Know GitHub Desktop

1. GitHub Desktop is a free, open source application that makes it easier for Mac and  Windows users alike to manage repositories and GitHub connections on their local computer.

**To install the GitHub Desktop on your computer**

**3. On Mac, drag the purple GitHub Desktop application into your  Applications folder.**

**4. On Mac, go to your Applications folder and double-click the GitHub**
**Desktop icon.**

**The application opens, shown in Fig**

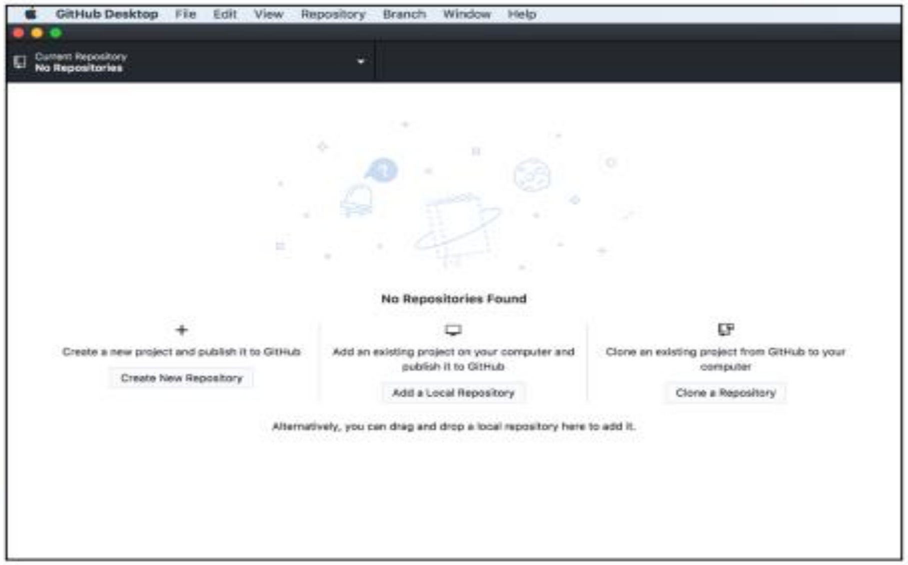

GitHub Desktop File Edit View Repository Branch Window Help
Current Repository
No Repositories
No Repositories Found
Create a new project and publish it to GitHub
Create New Repository
Add an existing project on your computer and publish it to GitHub
Add a Local Repository
Clone an existing project from GitHub to your computer
Clone a Repository
Alternatively, you can drag and drop a local repository here to add it.

FIGURE 2-6: The GitHub Desktop application default view.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 3. Setting Up GitHub Desktop

**1. Open the GitHub Desktop application.**

**2. Choose File ➪Preferences.**

**3. On the Accounts tab, click Sign In on the GitHub.com row.**

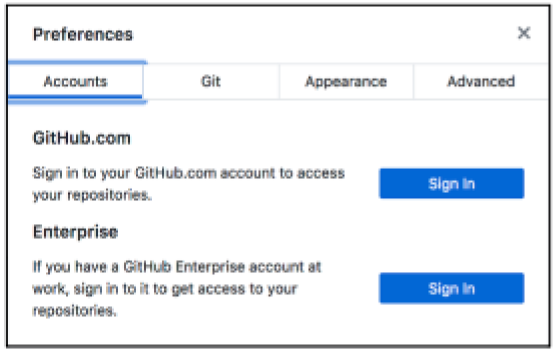
The Sign In dialog box, shown in Figure 2-7, appears.

**4. Type your username and password and click the Sign in button or click Sign in using your browser.**

When you click Sign in, all the dialog boxes close.

**5. Repeat Step 1 to re-open the preferences.**

Your account with your avatar, name, and GitHub username appears under the GitHub.com row, confirming that you have successfully logged in.



GitHub.com

Sign in to your GitHub.com account to access your repositories.

Enterprise

If you have a GitHub Enterprise account at work, sign in to it to get access to your repositories.

FIGURE 2-7: The Sign in dialog box for the GitHub Desktop application.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 3. Setting Up GitHub Desktop

**6. Click on the Git tab.**

The information has been auto-filled for you.

**7. On the Appearance tab, choose Light or Dark.**

Screenshots in this are in Light mode.

**8. Set other preferences, such as the Editor and usage data, on the Advanced tab.**

This book uses **Atom** and **Visual Studio Code** as example editors, but you can select whichever editor you prefer.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 4. Introducing Atom

Atom is a free, open-source editor. Just like GitHub Desktop, Atom is built on  Electron, making it work on Mac or Windows PC. Atom is extensible, meaning you can add your own features to it.

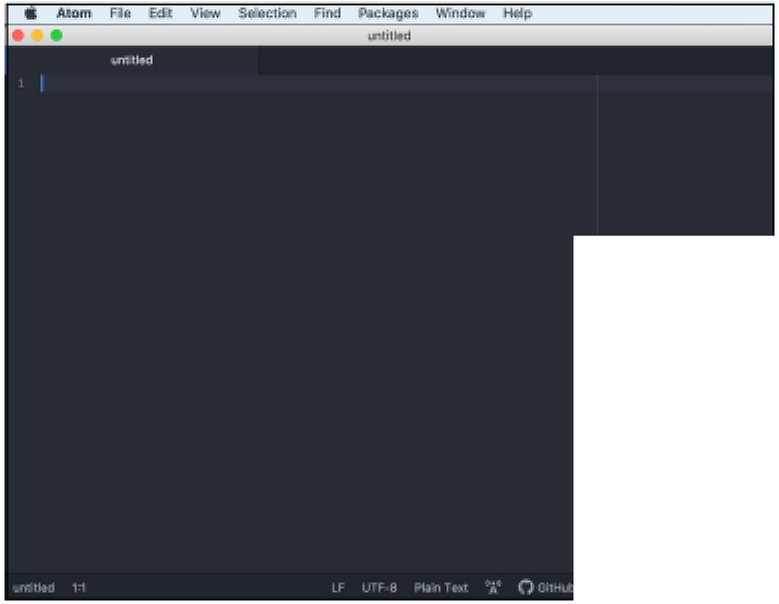
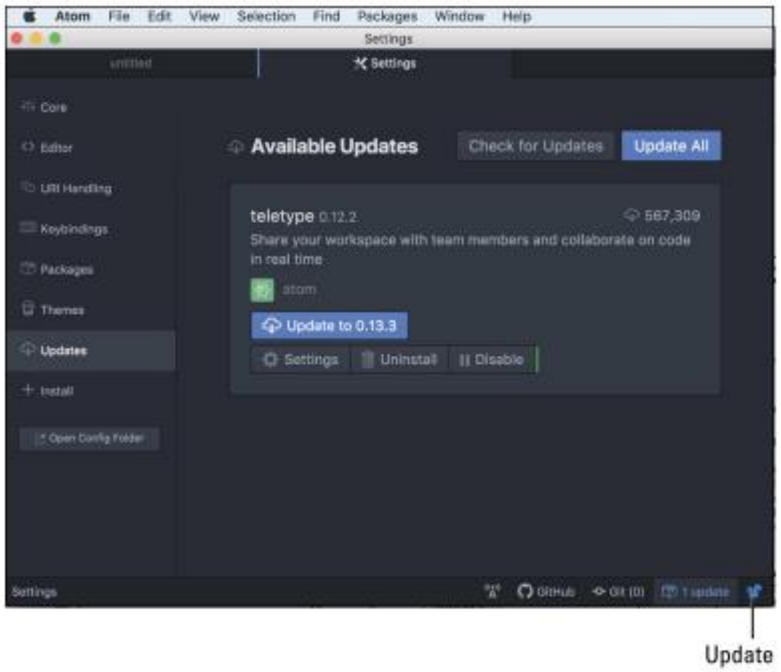

Atom File Edit View Selection Find Packages Window Help
untitled
untitled
untitled 1:1
LF UTF-8 Plain Text GitHub

FIGURE 2-8: The Atom application default view.



teletype 0.12.2 567,309

Share your workspace with team members and collaborate on code in real time

atom

FIGURE 2-9: the Squirrel icon takes you to the About page where you can update Atom.

# Chapter -2
# Setting Up Your Collaborative Coding Environment

## 4. Introducing Atom

Atom is a free, open-source editor. Just like GitHub Desktop, Atom is built on Electron, making it work on Mac or Windows PC. Atom is extensible, meaning you can add your own features to it.

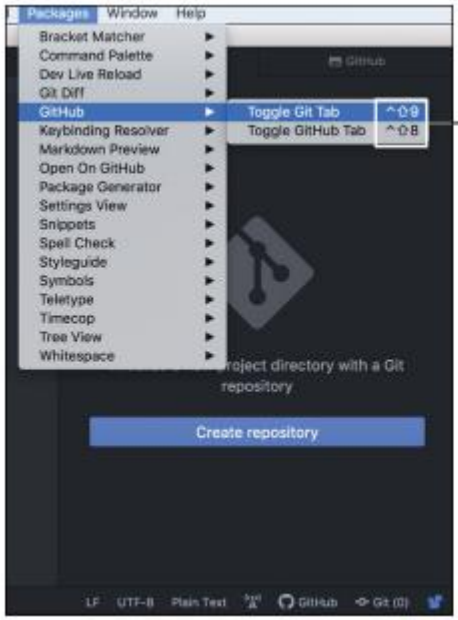
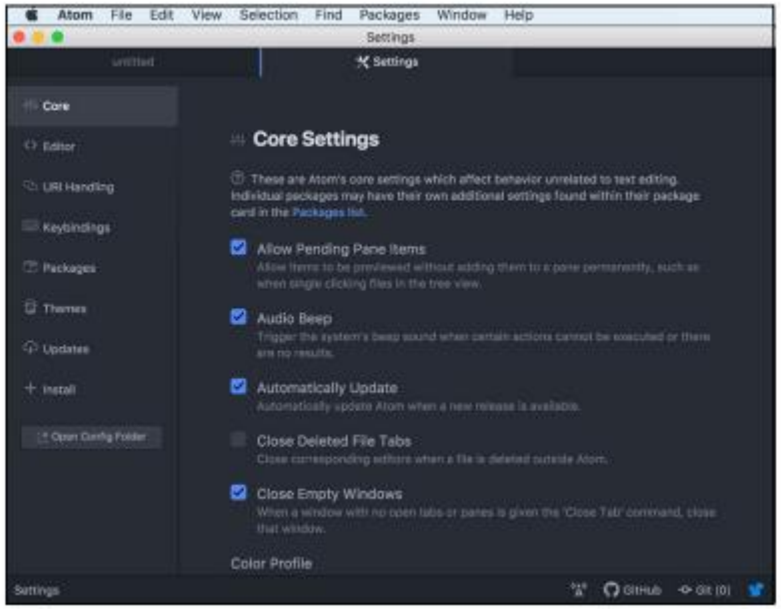


FIGURE 2-10: Keyboard bindings for toggling the Git and GitHub tabs.



Core Settings

These are Atom's core settings which affect behavior unrelated to text editing. Individual packages may have their own additional settings found within their package card in the Packages list.

[✓] Allow Pending Pane Items
Allow items to be previewed without adding them to a pane permanently, such as when single clicking files in the tree view.

[✓] Audio Beep
Trigger the system's beep sound when certain actions cannot be executed or there are no results.

[✓] Automatically Update
Automatically update Atom when a new release is available.

[ ] Close Deleted File Tabs
Close corresponding editors when a file is deleted outside Atom.

[✓] Close Empty Windows
When a window with no open tabs or panes is given the 'Close Tab' command, close that window.

Color Profile

FIGURE 2-11: The Atom Settings page.

# Chapter -3
# Starting your First Solo Project:

1. Introducing GitHub Repositories
2. Setting Up a GitHub Website Repo
3. Creating a website with GitHub Pages

## 1. Introducing GitHub Repositories

### 1. Setting up a Repositories:

A GitHub repository is a folder with all the files needed for your project, including the files that track all the versions of your project so that you can revert back if you make a mistake.

## create your first GitHub repo::

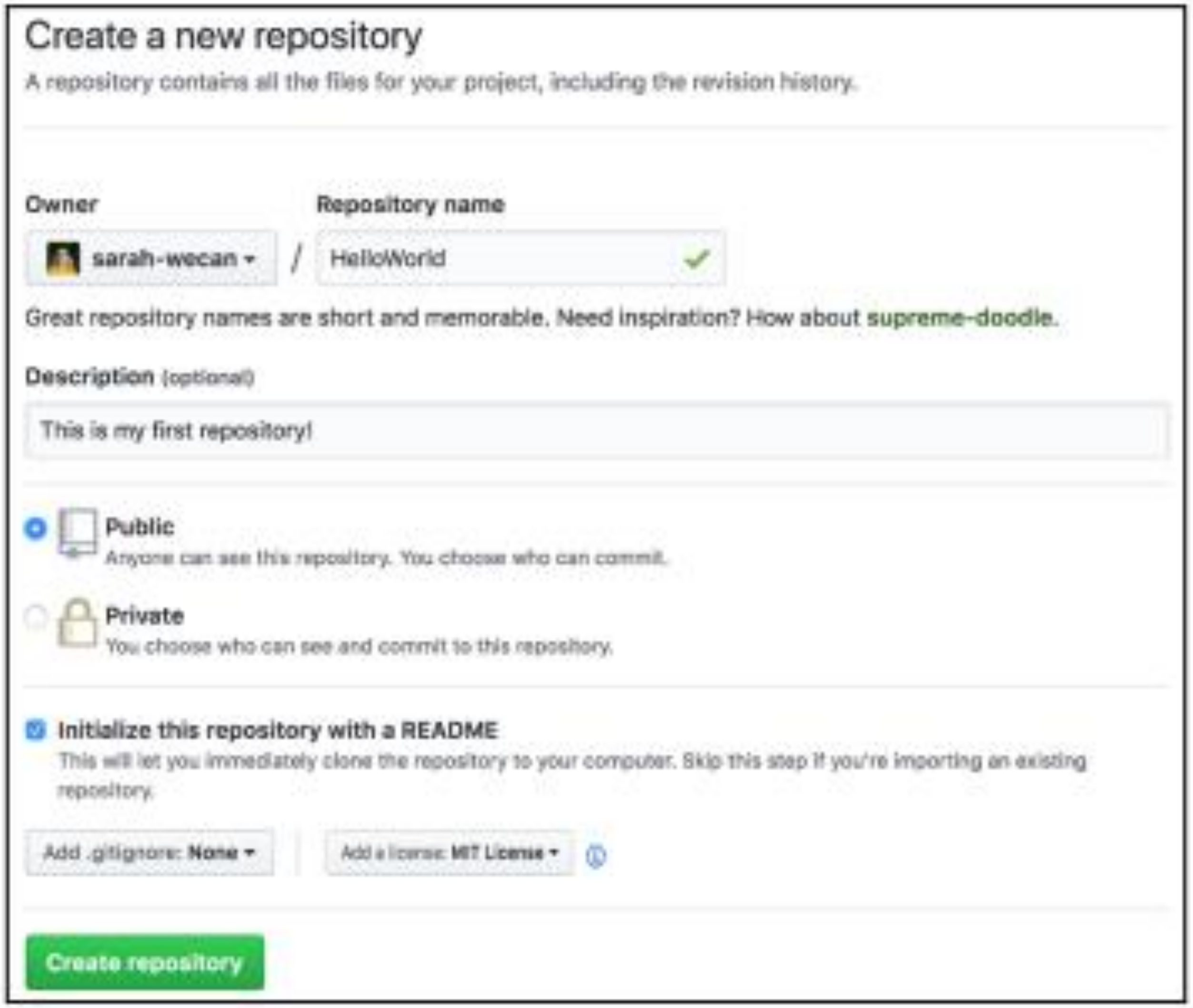
1. **Go to the home page of GitHub.com by clicking the Octocat.**

A list of your repositories appears on the bottom left side of the screen.

**2. Click the green New Repository button.**

The Create a New Repository dialog page, shown in Figure 3-1, opens

## 1. Introducing GitHub Repositories



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: sarah-wecan / Repository name: HelloWorld ✓

Great repository names are short and memorable. Need inspiration? How about supreme-doodle.

Description (optional): This is my first 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 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: MIT License ▾

FIGURE 3-1: The page to create a new repository.

## 1. Introducing GitHub Repositories

create your first GitHub repo::

**3. Type the name of your repository in the Repository name text box.**

I named my repository HelloWorld.

**4. Type a short description in the Description text box.**

**5. Select the Public radio button.**

**6. Click the Initialize this repository with a README check box.**

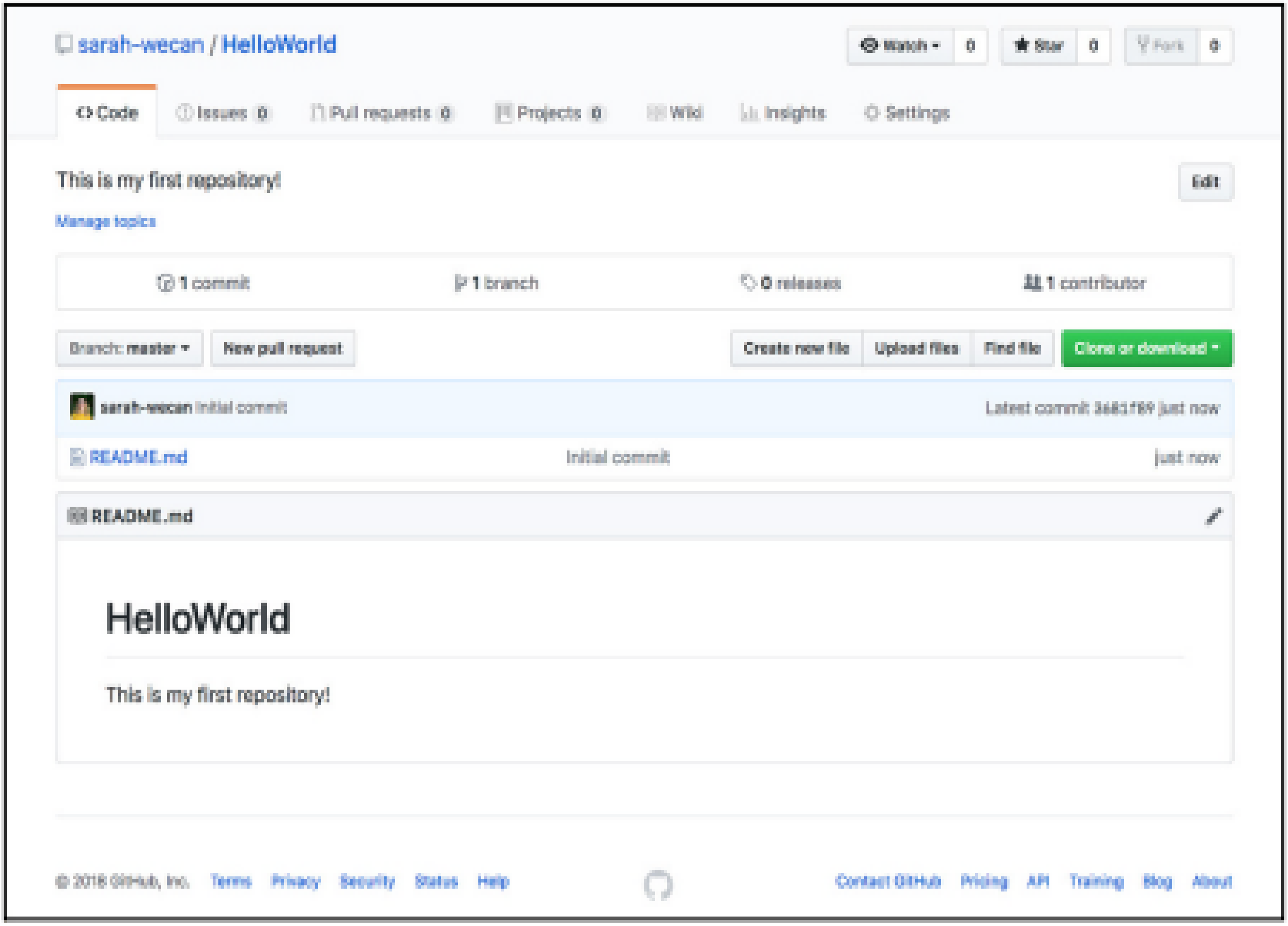
You do not need to add a .gitignore.

**7. Choose a license from the Add a license drop-down list.**

If you’re interested in finding out more information about licenses, see the nearby “Software licenses” sidebar.

**8. Click Create Repository.**

**The home page of your repository appears.**



This is my first repository!

sarah-wecan Initial commit — Latest commit 3681f89 just now

README.md — Initial commit — just now

HelloWorld

This is my first repository!

© 2018 GitHub, Inc.

FIGURE 3-2: The home page of my HelloWorld repository.

In Chapters 4 and 5, you can create a website for yourself. This website can link back to your repository.

## 1. Introducing GitHub Repositories

### 1. Exploring Your Repository:

**Top information**

At the top of the repository is the username of the author and title of the reposi tory. When you fork a repository, you see the original author underneath for a quick link. To fork a repository is to make a copy of it, where the changes you make to your copy can be suggested to the original author.

**To the right of your username are three buttons:**

» **Watch:** You can choose what kind of notifications you want to receive based on the type of activity happening on this repo.

» **Star:** Starring can help you quickly navigate to certain repositories, as well as give GitHub insight into things you're interested in so that recommendations will be more accurate for you. To access your starred repositories, just click your avatar on the top right of GitHub.com and choose Your Stars.

» **Fork:** If you are not the author of the repository, then you have the option to fork it.

## 1. Introducing GitHub Repositories

### 1. Exploring Your Repository:

**Tabs**

Seven tabs appear across the top of your repo. Each tab provides different features for the repo:

1. **Code:** The Code tab is where you can find all your code and browse folders and files.
2. **Issues:** Issues can help you track things you want to still make, problems you're having, or suggestions for other people.
3. **Pull requests:** Pull requests, also referred to as PRs, are similar to issues in that they have a title and a description, but they also have code changes that you're requesting to be pulled into the main branch. The safest way to contribute code is to create a new branch, make your code changes on that branch, and then request for that branch to be merged with the master branch.

## 1. Introducing GitHub Repositories

### 1. Exploring Your Repository:

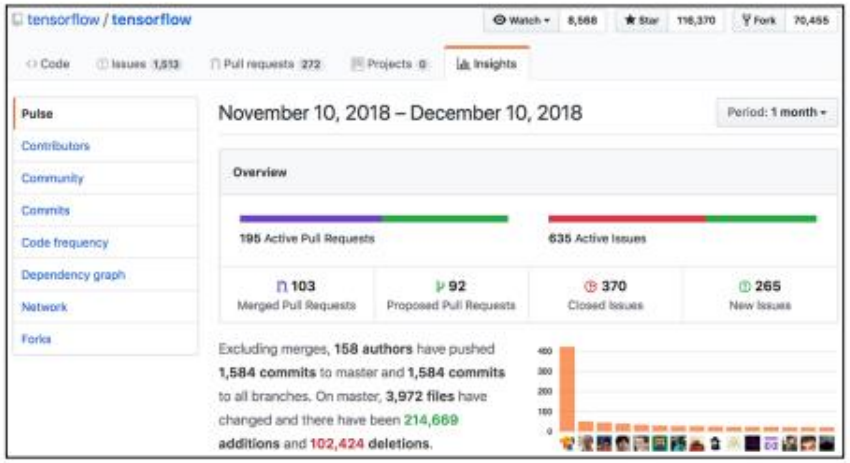
### Tabs

**4. Projects:** GitHub has project boards linked directly with your repo.

**5. Wiki:** Wikis are a great place to store documentation, project status, and roadmaps for your project.

**6. Insights:** The Insights tab, shown in Figure 3-3, gives you an overview of all collaborators and actions happening on the repo.

7. **Settings:** The Settings tab is visible only if you have the right permissions on the repository.



November 10, 2018 – December 10, 2018

195 Active Pull Requests

635 Active Issues

103 Merged Pull Requests

92 Proposed Pull Requests

370 Closed Issues

265 New Issues

Excluding merges, **158 authors** have pushed **1,584 commits** to master and **1,584 commits** to all branches. On master, **3,972 files** have changed and there have been 214,669 **additions** and 102,424 **deletions**.

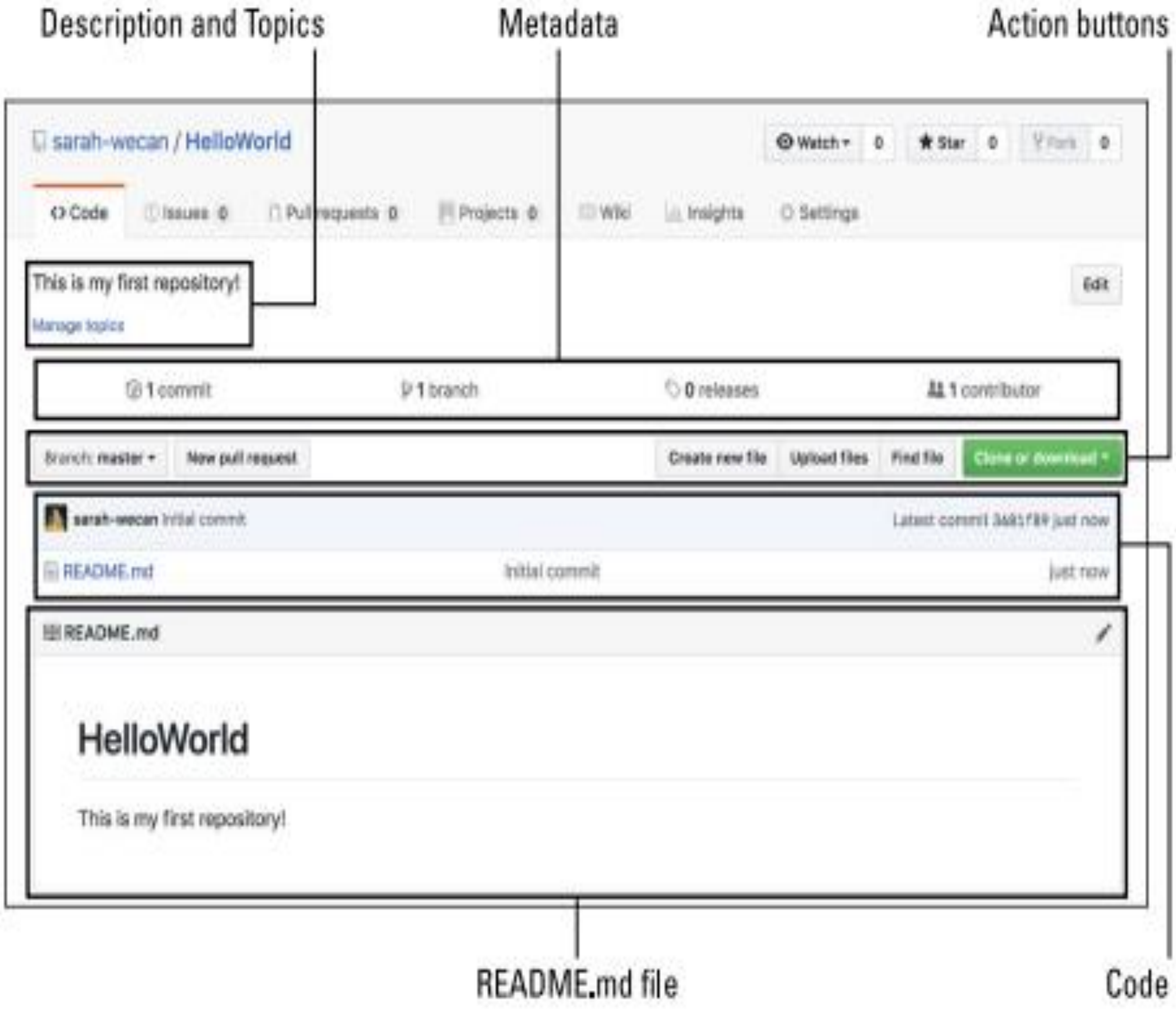
FIGURE 3-3: The Insights tab.

## 1. Introducing GitHub Repositories

### 1. Exploring Your Repository:

### Code Tab

The Code tab, shown in Figure 3-4, has a lot of additional important metadata about your repo that will come in useful in future development:



This is my first repository!

sarah-wecan Initial commit — Latest commit 3a81f89 just now

README.md — Initial commit — just now

README.md

HelloWorld

This is my first repository!

**FIGURE 3-4:** The Code tab.

## 1. Introducing GitHub Repositories

» **Description and topics:** At the top of the Code tab is a description and a place where you can put in topics to make your repository more discoverable.

Adding topics is particularly important if you want to attract other coders to help you build your software.

» **Metadata:** This bar contains information and links to the number of commits, branches, releases, and contributors for the repo.

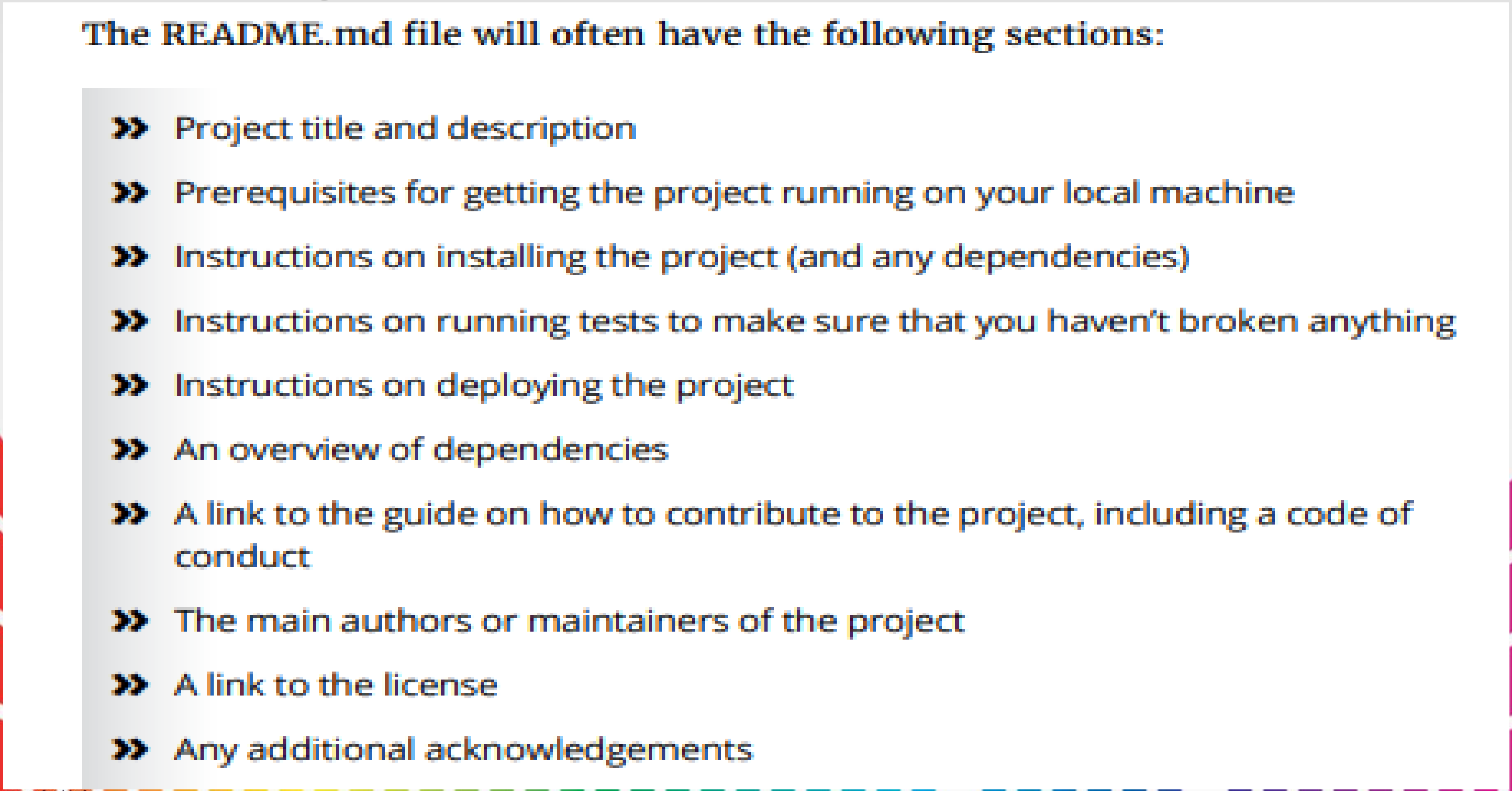
» **Action buttons:** On the left side of the repo is a drop-down menu where you can change which branch you're looking at or browse the files for a particular branch. The New pull request button allows you to quickly create a pull request.

## 1. Introducing GitHub Repositories

**>> Code:** At the bottom of the Code tab is a list of all the code in this repo. If a README.md file appears in this list, then the file shows up below the list.
For any file, you can click the filename to go to a page where you can see the file and edit it if you want

## 1. Introducing GitHub Repositories

## Modifying README.md



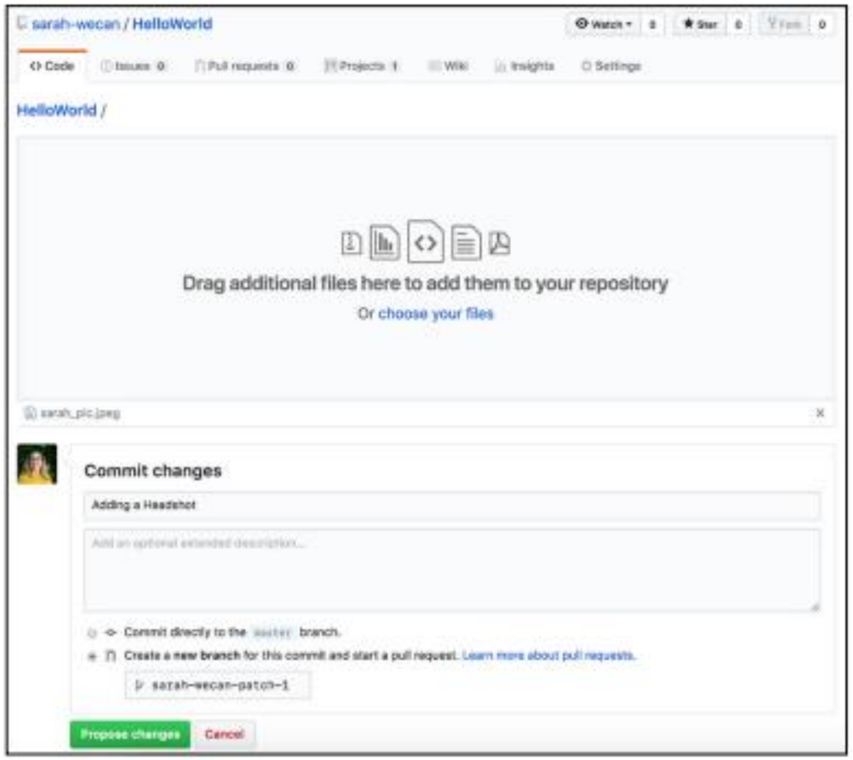
The README.md file will often have the following sections:

- » Project title and description
- » Prerequisites for getting the project running on your local machine
- » Instructions on installing the project (and any dependencies)
- » Instructions on running tests to make sure that you haven't broken anything
- » Instructions on deploying the project
- » An overview of dependencies
- » A link to the guide on how to contribute to the project, including a code of conduct
- » The main authors or maintainers of the project
- » A link to the license
- » Any additional acknowledgements

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md

**1.** Go to the Code tab of your repository and click the Upload files button. Steps 2 through 6 guide you through the page shown in Figure 3-5.



Adding a Headshot

Commit directly to the master branch.
[✓ Create a new branch for this commit and start a pull request.] Learn more about pull requests.

sarah-wecan-patch-1

**FIGURE 3-5:** The page to add a file to the HelloWorld repository.

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md

2. **Find a headshot of yourself (or any picture that you want to be on your README) and upload it using the drag box.**

Alternatively, you can click the Choose your files link and browse your files to find your photo.

3. **Type a title in the text box Commit changes.**

In this example, I added the title “Adding a Headshot.”

4. **Select Create a new branch for this commit and start a pull request.**

## Steps for Modifying README.md

**5. Type a name for the branch, or you can leave the default name for the branch.**

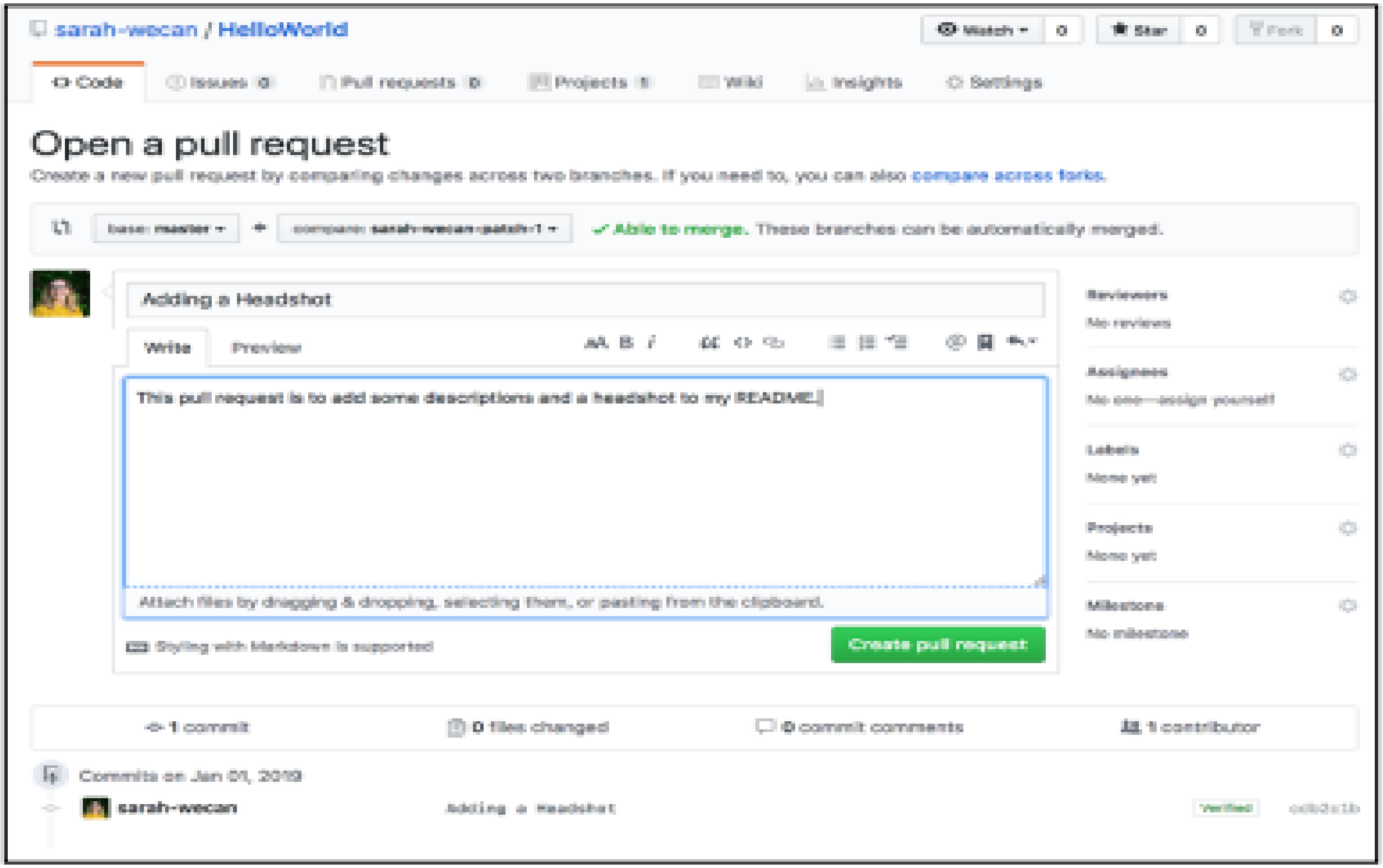
I left my default name, which is sarah-wecan-patch-1.

**6. Then click the Propose changes button.**

The Open a pull request page appears with your one commit to add a new headshot to your repo (see Figure 3-6

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

✓ Able to merge. These branches can be automatically merged.

Adding a Headshot

This pull request is to add some descriptions and a headshot to my README.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Commits on Jan 01, 2019

sarah-wecan Adding a Headshot

**FIGURE 3-6:** The Open a pull request page.

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md

**7. Add a short description to explain the change and then click the Create pull request button.**

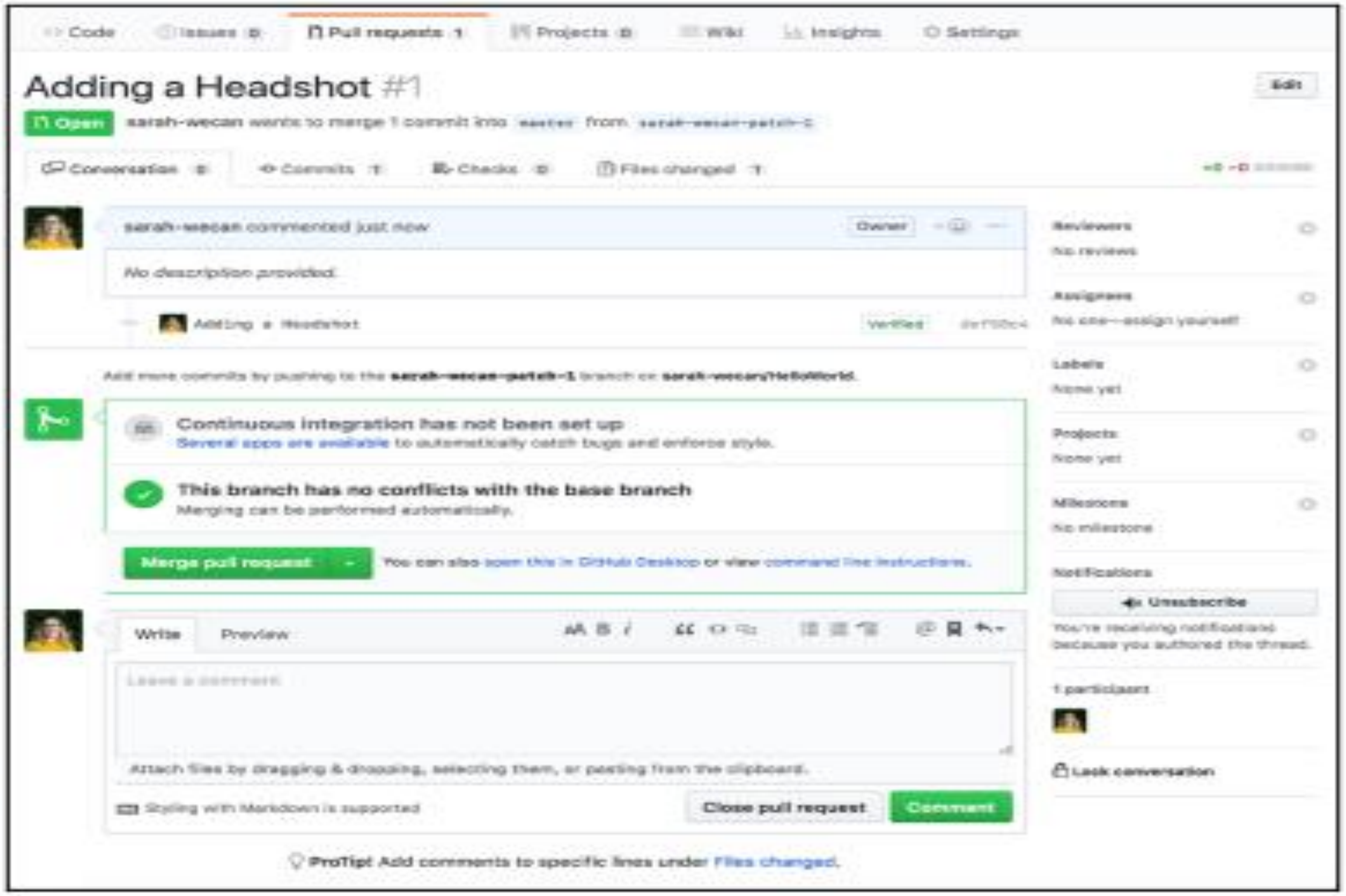
The Pull request tab displays your pull request with a title, number, status, description, list of commits, and a check to see whether it can be merged with master, shown in Figure 3-7.

**8. Click the Code tab to go back to the code in your repo.**

**9. Click the branch drop-down list and switch to the branch you created in Step 5**

1. **Introducing GitHub Repositories**

## Steps for Modifying README.md



Adding a Headshot #1

sarah-wecan wants to merge 1 commit into master from sarah-wecan-patch-1

sarah-wecan commented just now

*No description provided.*

Adding a Headshot

Add more commits by pushing to the **sarah-wecan-patch-1** branch on **sarah-wecan/HelloWorld**.

**Continuous integration has not been set up**
Several apps are available to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

**ProTip!** Add comments to specific lines under Files changed.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
You're receiving notifications because you authored the thread.

1 participant

**FIGURE 3-7:** The pull request description page for an open pull request.

## Steps for Modifying README.md

**10 Because your README file is still showing, click the little pencil so that you can change what the README says.**

If it exists, a README file always appears below the list of files.

**11. Using Markdown, write a little about yourself, including your career passions and some hobbies you enjoy.**

Figure 3-8 shows you an example of the type of information you may want to include.

**12. Click the Preview changes tab above the text.**

**A red line appears to the left of the first two lines to indicate they will be deleted (see Figure 3-9). Everything you wrote after that has a green line to the left of it to indicate that the text will be added to your description.**

**13. When you're satisfied with your description, scroll to the bottom of the file editor, add a title to the commit, and commit to the same branch you**

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md

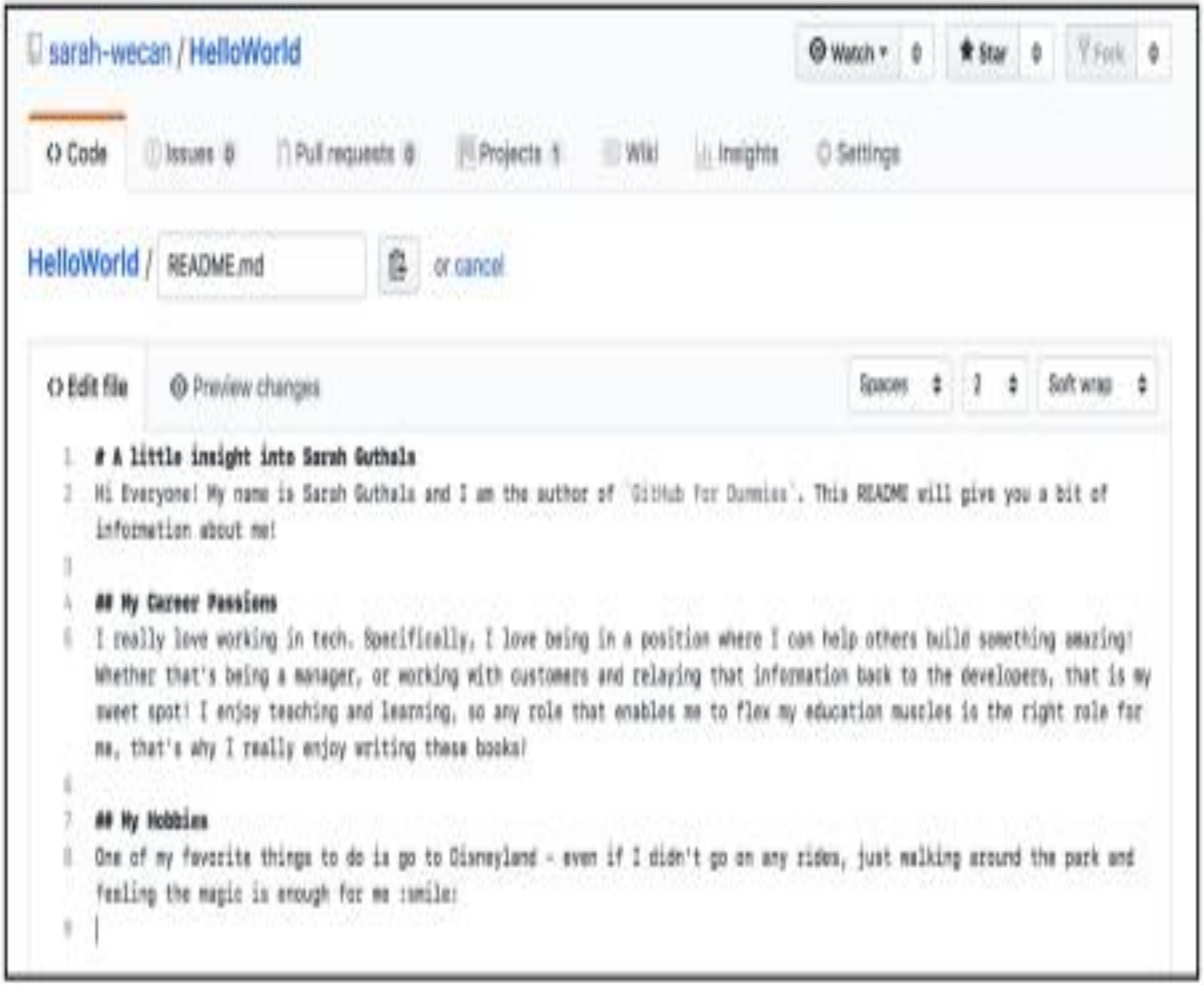
**12. Click the Preview changes tab above the text.**

A red line appears to the left of the first two lines to indicate they will be deleted (see Figure 3-9). Everything you wrote after that has a green line to the left of it to indicate that the text will be added to your description.

**13. When you're satisfied with your description, scroll to the bottom of the file editor, add a title to the commit, and commit to the same branch you just created.**

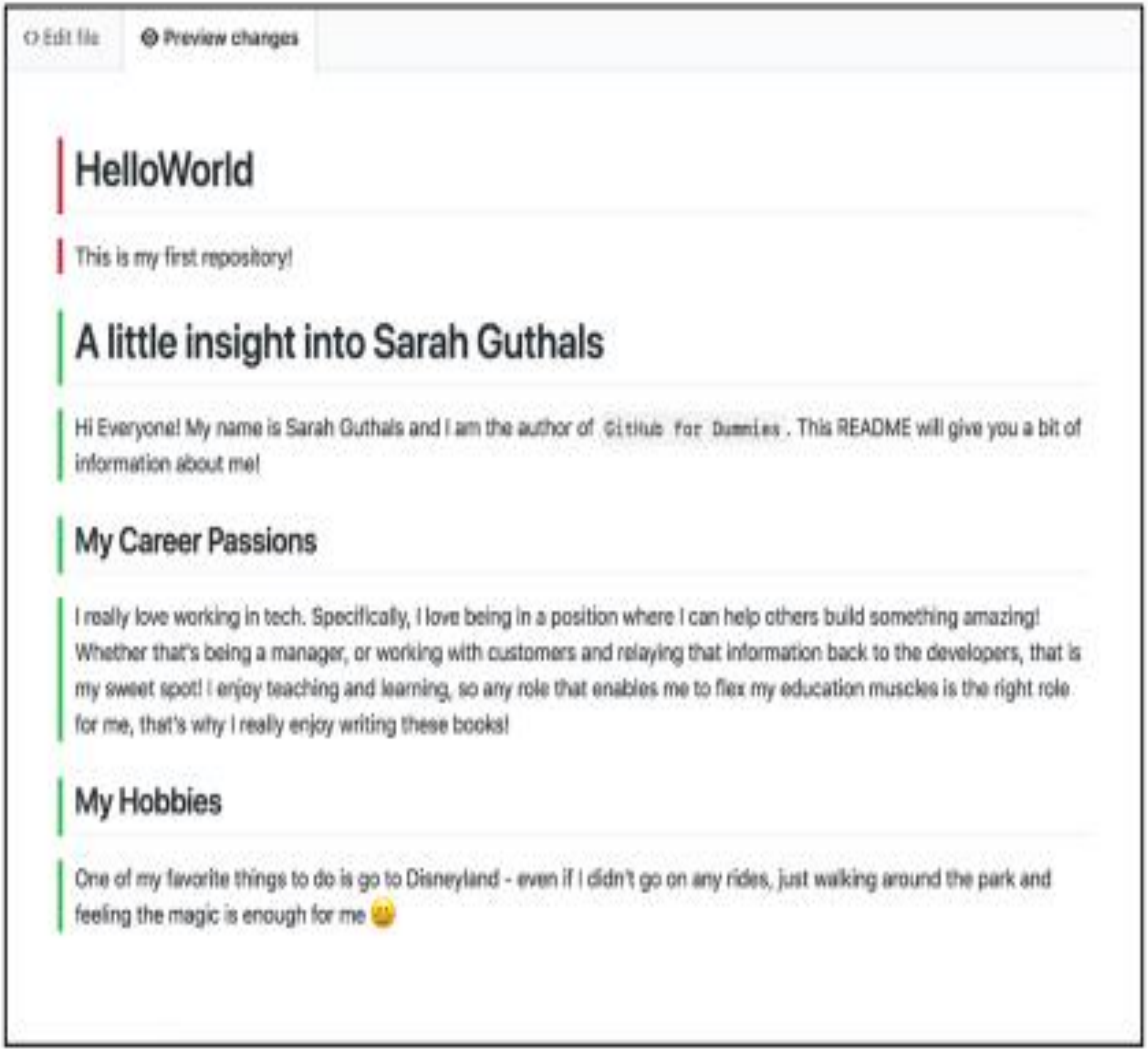
You see your README.md file in its final state.

**14. Click the HelloWorld link at the top where it says HelloWorld/README.md to return to your code home page.**



```
# A little insight into Sarah Guthals
Hi Everyone! My name is Sarah Guthals and I am the author of `GitHub for Dummies`. This README will give you a bit of information about me!

## My Career Passions
I really love working in tech. Specifically, I love being in a position where I can help others build something amazing! Whether that's being a manager, or working with customers and relaying that information back to the developers, that is my sweet spot! I enjoy teaching and learning, so any role that enables me to flex my education muscles is the right role for me, that's why I really enjoy writing these books!

## My Hobbies
One of my favorite things to do is go to Disneyland - even if I didn't go on any rides, just walking around the park and feeling the magic is enough for me :smile:
```

**FIGURE 3-8:** The README.md file in edit mode.



HelloWorld

This is my first repository!

A little insight into Sarah Guthals

Hi Everyone! My name is Sarah Guthals and I am the author of `GitHub for Dummies`. This README will give you a bit of information about me!

My Career Passions

I really love working in tech. Specifically, I love being in a position where I can help others build something amazing! Whether that's being a manager, or working with customers and relaying that information back to the developers, that is my sweet spot! I enjoy teaching and learning, so any role that enables me to flex my education muscles is the right role for me, that's why I really enjoy writing these books!

My Hobbies

One of my favorite things to do is go to Disneyland - even if I didn't go on any rides, just walking around the park and feeling the magic is enough for me 😀

**FIGURE 3-9:** The `README.md` file in `diff` mode.

## 1. Introducing GitHub Repositories

## Steps for Modifying README.md

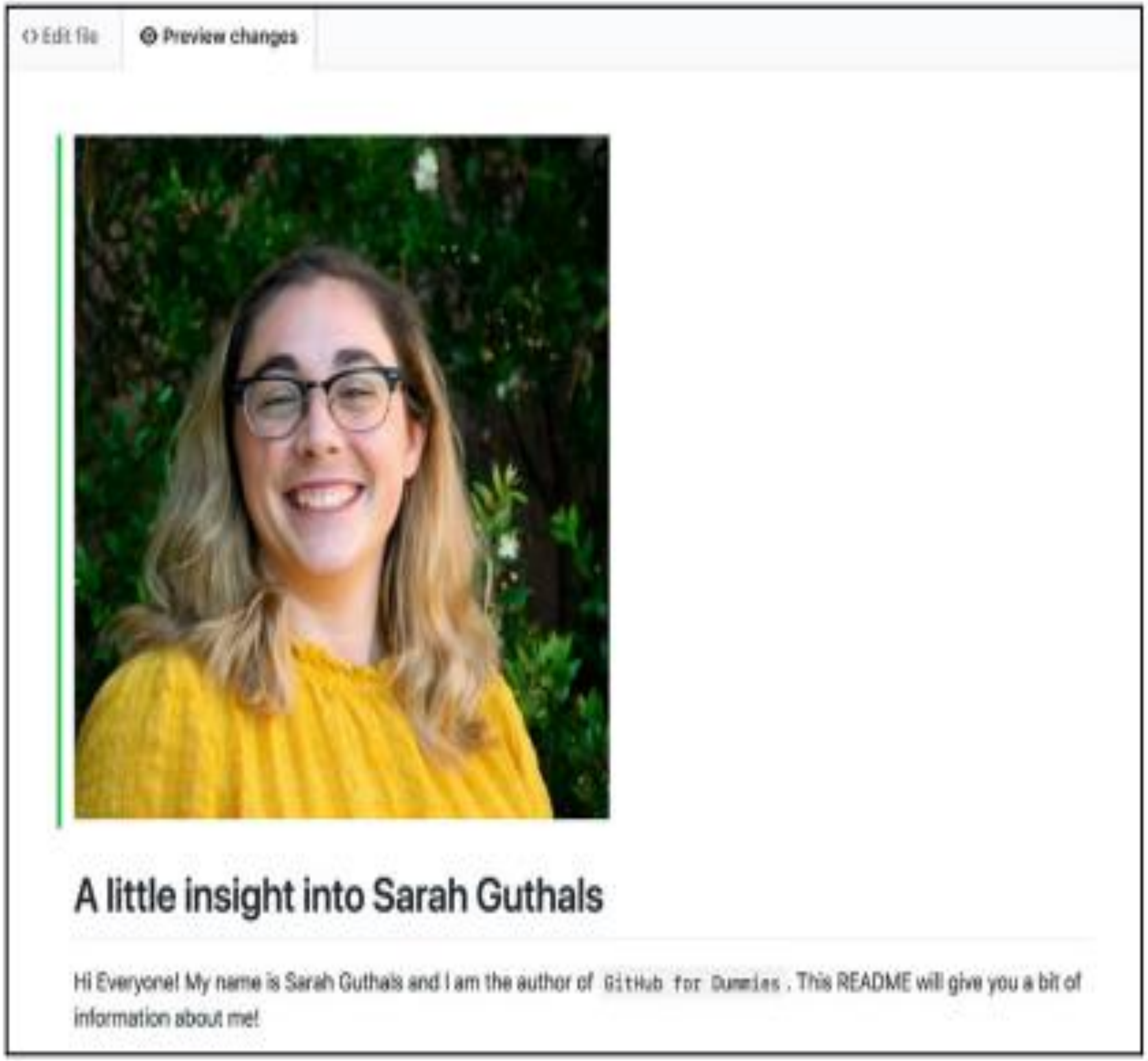
14. Click the HelloWorld link at the top where it says HelloWorld/README.md to return to your code home page.

15. Add your headshot by clicking on the pencil above the README.md file and adding the following line:

```
![headshot](sarah_pic.jpeg)
```

16. Preview the changes.

17. Scroll to the bottom and commit your changes.



## A little insight into Sarah Guthals

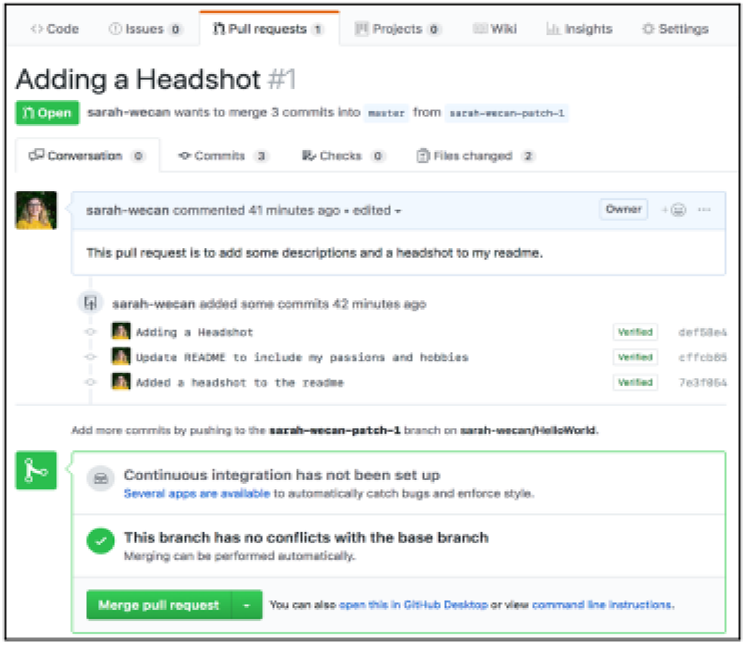
Hi Everyone! My name is Sarah Guthals and I am the author of `GitHub for Dummies`. This README will give you a bit of information about me!

**FIGURE 3-10:** The `README.md` file in `diff` mode with the headshot added.

## 1. Introducing GitHub Repositories

## Merging a Pull Request

1. On the main Code tab, click the View #1 button to get to your pull request.
2. Click the Files changed tab to see all the changes made to this repo



Adding a Headshot #1

sarah-wecan wants to merge 3 commits into `master` from `sarah-wecan-patch-1`

sarah-wecan commented 41 minutes ago • edited

This pull request is to add some descriptions and a headshot to my readme.

sarah-wecan added some commits 42 minutes ago

```
Adding a Headshot                                Verified   def58e4
Update README to include my passions and hobbies Verified   cffcb85
Added a headshot to the readme                   Verified   7e3f864
```

Add more commits by pushing to the **sarah-wecan-patch-1** branch on **sarah-wecan/HelloWorld**.

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

FIGURE 3-11: My pull request page.

## 1. Introducing GitHub Repositories

### Merging a Pull Request

3. (Optional) To change the way you see the diff, click the Diff Settings drop-down list and then click Split then Apply and reload

4. In the Conversation tab, scroll to the bottom of the pull request and click the big green Merge pull request button.

5. Click Confirm merge.

6. Click Delete branch.

7. Click the Code tab to go to your code.

## Using Issues and project Boards

### 1. Creating a project board and an issue

1. Click the Projects tab and then click the Create a project button.
2. Add a project name and description.
3. From the preloaded Template drop-down list, select a project template and click the Create project button.
4. Click the Manage button at the bottom of the To do column to open the dialog box to choose what automation should happen.

## Using Issues and project Boards

## 1. Creating a project board and an issue

5. Click the Manage button at the bottom of the In progress column.

6. Click the Manager button at the bottom of the Done column.

7. Delete the cards that were automatically added to the To do column by clicking the three dots on the top right of each column and choosing Delete note from the drop-down menu that appears.

## Using Issues and project Boards

## 1. Creating a project board and an issue

**8.** On the Issues tab, click New issue.

**9.** Add a title and description to the new issue and then, on the right side, assign yourself to it

**10.** Link this issue with a project by clicking Projects and choosing Tracking Changes to HelloWorld.

**11.** Click the Submit new issue button.

## Using Issues and project Boards

### 2. Closing an issue

1. From the Code tab on your repo, click the pencil icon for the README.md file to edit the file.
2. Add a section at the bottom of the README.md file with your favorite  books.
3. Scroll to the bottom of the page and add a title to the commit.
4. Choose Create a new branch for this commit and start a pull request.
5. Click Commit changes.
6. Add a description to the pull request.

## Using Issues and project Boards

### 2. Closing an issue

7. Link the project to this pull request.

8. Click Create pull request.

9. Return to your project board.

10 . Click the pull request card title to preview the pull request.

11. Click the Files changed tab and then click View file.

12. Click the Back button on your browser.

13. If you’re happy with the changes, click the Conversation tab, click Merge pull request button, click Confirm merge button, and then click the Delete branch button.

14. Click the Projects tab and choose the project that you created.

## END