

Lab: GitHub Sign Up and Create Repo

Estimated time: 30 minutes

In this lab, you will get started with GitHub by creating a GitHub account and creating a new repository. You will then add a file to the repository using the GitHub web interface.

Objectives

After completing this lab, you will be able to:

1. Describe GitHub
2. Create a GitHub account
3. Add a project/repository
4. Create and edit a file
5. Upload and commit a file

GitHub overview

Before we learn to use GitHub, let's first review Git. Git is an open-source command-line-based version-control system for software development. While Git is a command-line tool and you also need to host and maintain a server on which you can use Git for versioning.

GitHub is a repository hosting service that uses Git. GitHub provides a web-based hosting service with a graphical user interface(GUI) and git command line interface (CLI). It also provides access control and several collaboration features, such as wikis and basic task management tools, for every project. GitHub provides cloud storage for source code, supports all popular programming languages, and streamlines the iteration process. GitHub includes a free plan for individual developers and for hosting open-source projects.

1. Creating a GitHub account

To create an account in GitHub, complete the following steps:

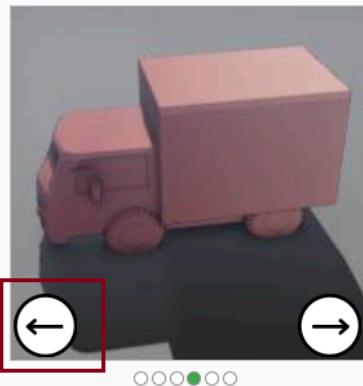
1. Go to the [Join GitHub](#) page and create an account. **Note:** If you already have a GitHub account, log in now.
2. Provide the necessary details to create an account as shown below:

The screenshot shows the GitHub sign-up page. At the top, there is a navigation bar with links for Product, Solutions, Open Source, and Pricing. On the right side of the navigation bar is a search bar labeled "Search or jump to..." and a "Sign in" button. Below the navigation bar, the main heading reads "First, let's create your user account". The form contains three input fields: "Username" (filled with a redacted value), "Email address" (filled with a redacted value), and "Password" (filled with a redacted value). Below the password field, a note says "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter." There is a link "Learn more.". At the bottom of the form, there is a section for "Email preferences" with a checked checkbox "Send me occasional product updates, announcements, and offers.".

3. Verify the account, and then click **Submit**.

Verify your account

Use the arrows to rotate the object to face in the direction of the hand. (1 of 2)



Submit



4. When you have finished, click **Create account**.

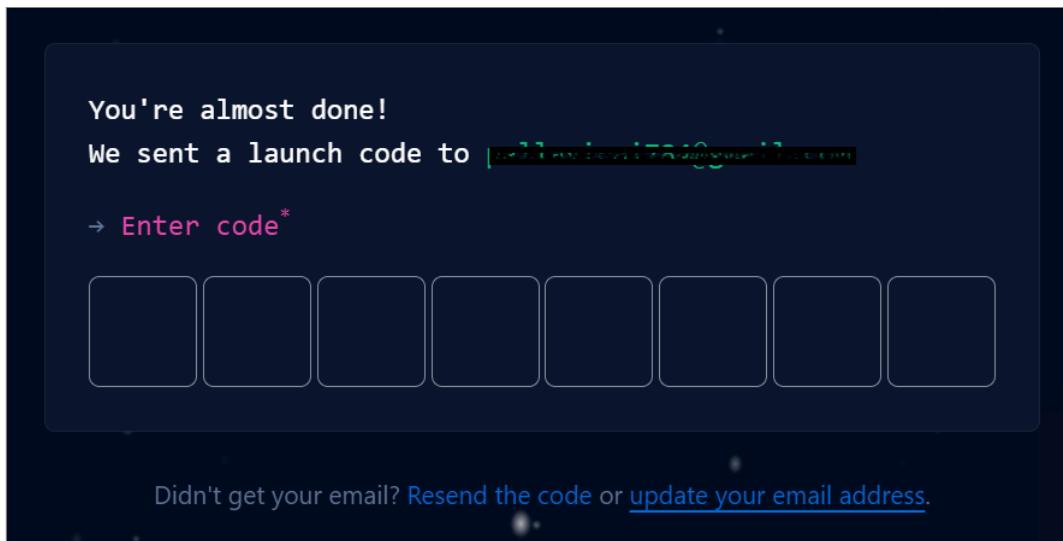
Verify your account



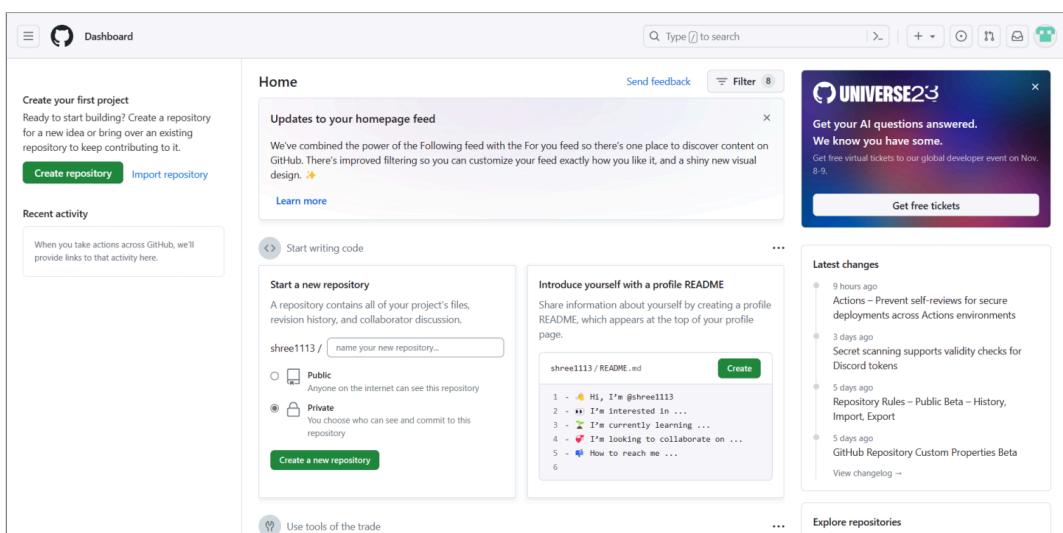
Create account

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.

5. You will receive a verification email from GitHub. Click the enclosed link to verify your email or enter the provided verification code. **Note:** If you do not receive a verification email, click **Resend the code**.



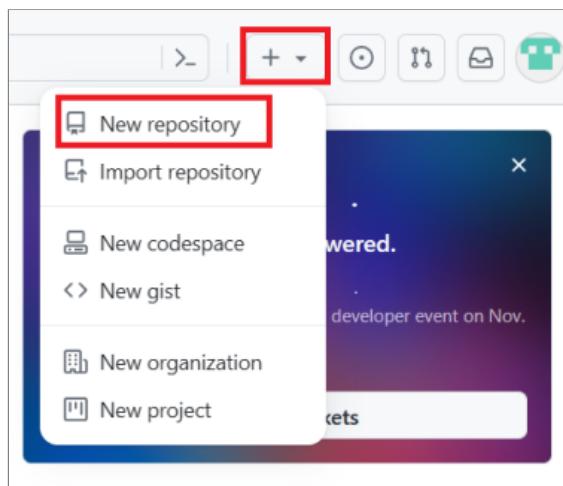
6. After entering the provided verification code, you will see a screen that looks like this. This indicates that you are logged into your account and you are ready to start using it.



2: Add a project/repository

To add a new repository, complete the following steps:

- At the top right of the GitHub home page, click on the "+" icon and select **New repository**.



- Enter a repository name and select the **Add a README file** check box under the **Initialize this repository with:** option.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *



Repository name *

testrepo

✓ testrepo is available.

Great repository names are short and memorable. Need inspiration? How about [silver-octo-computing-machine](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

3. Click **Create repository**.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

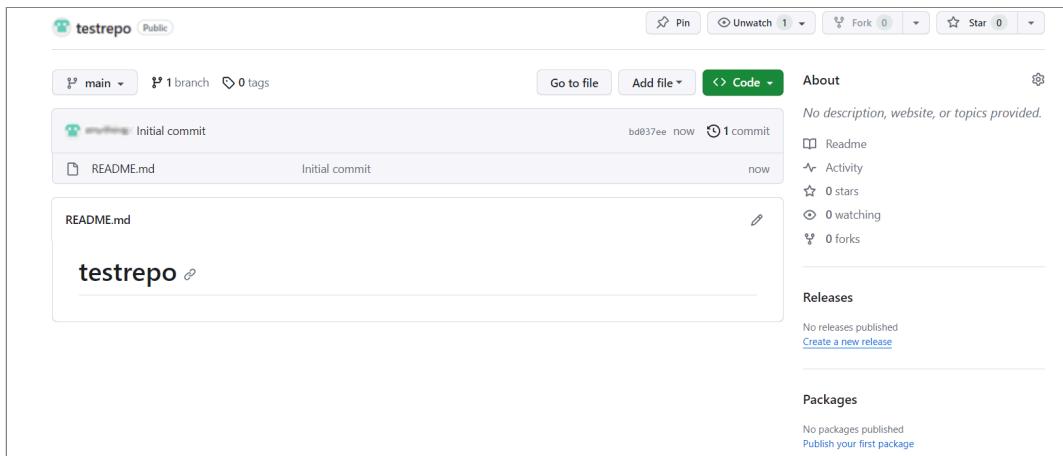
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set main as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

4. The repository is created and its home page is displayed.



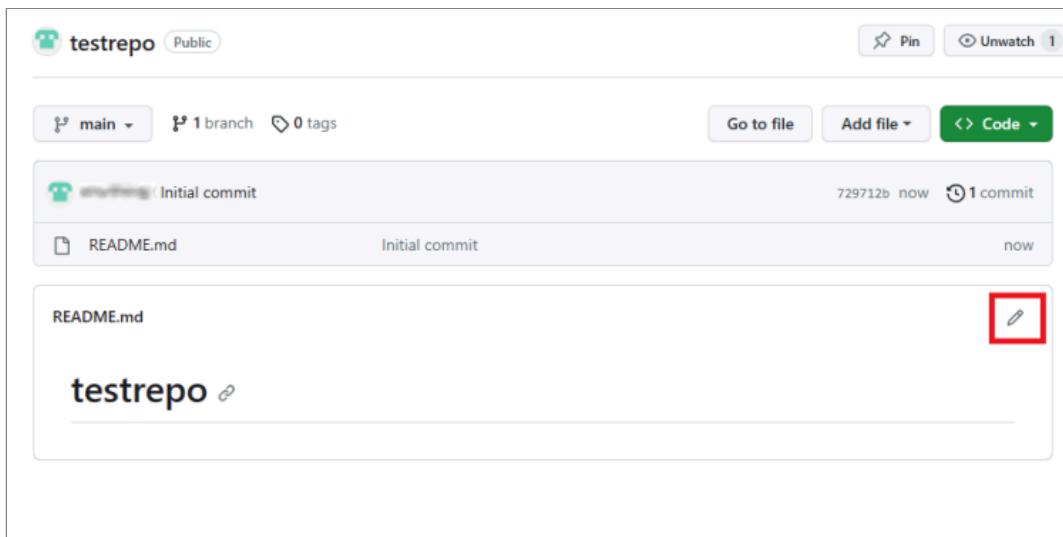
Next, you'll start editing the repository.

3: Create and edit a file

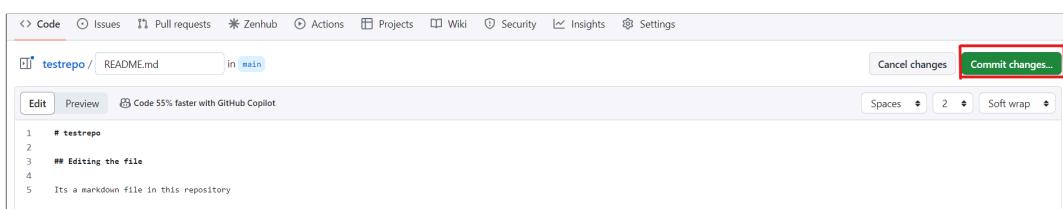
3.1 Edit a file

Although you will normally create a file before you edit it, in this case, the `README.md` file has already been created for you. To edit that file, complete the following steps:

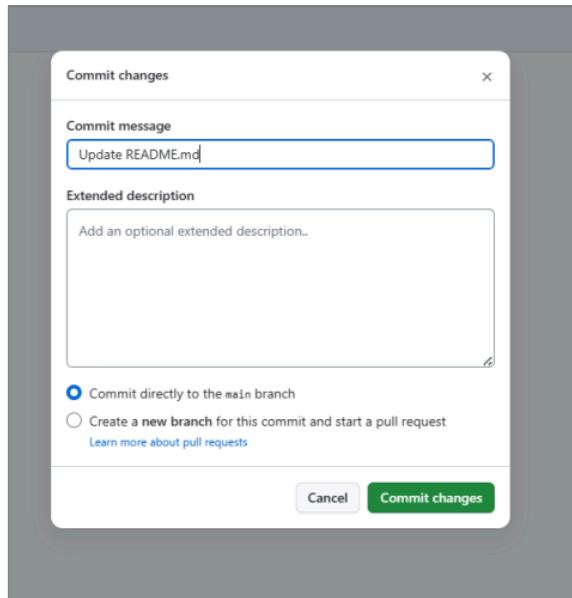
1. Your repository root folder contains just one file: `README.md`. Click the pencil icon at the right to edit the file.



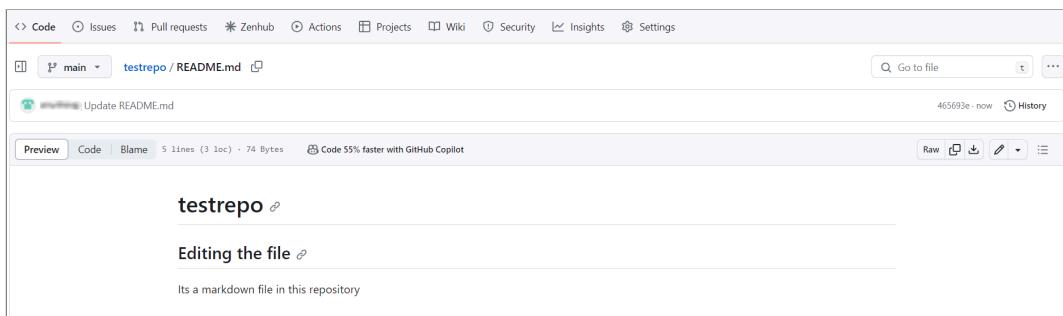
2. Add "It's a markdown file in this repository" text to the file and go to top right and click Commit changes



3. Add some additional description and click **Commit changes**.

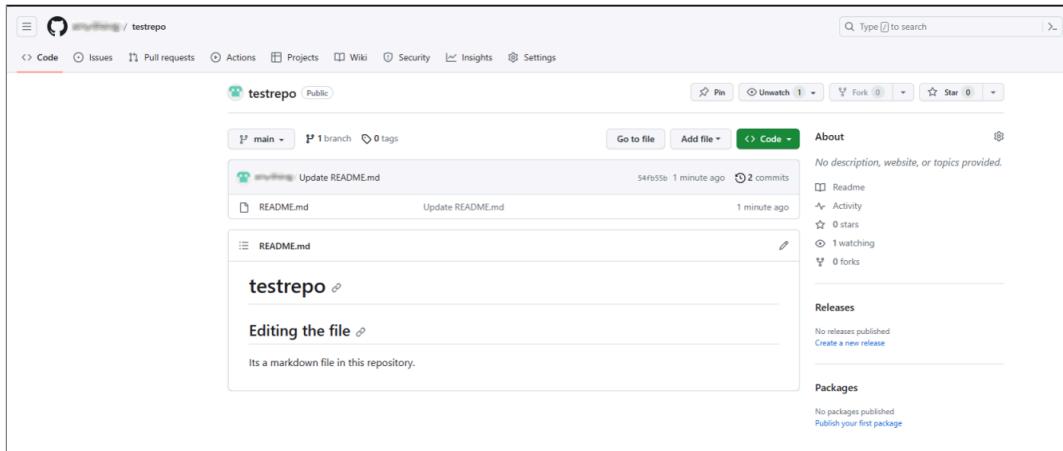


4. Confirm that the text you added to the file has been saved.

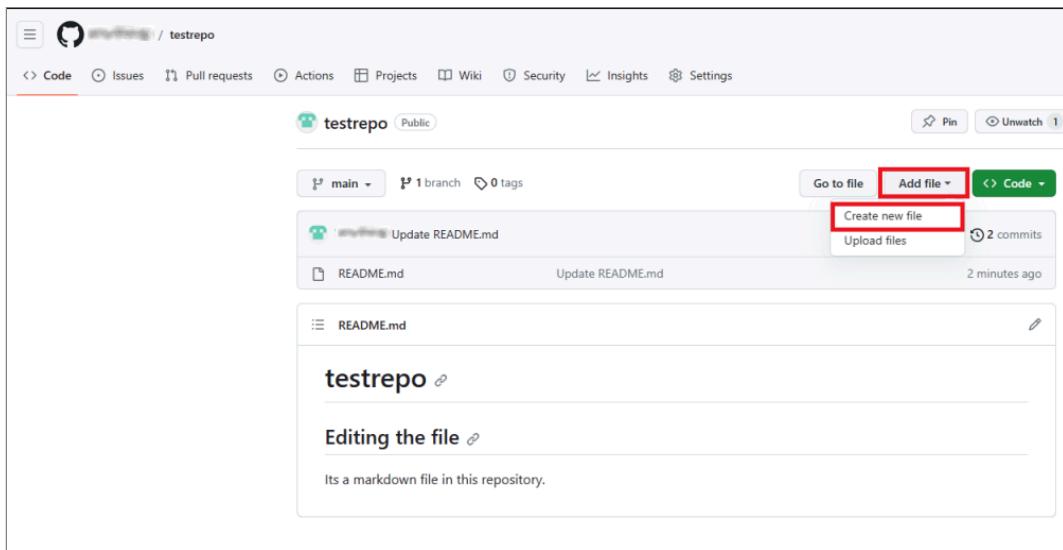


3.2 Create a new file

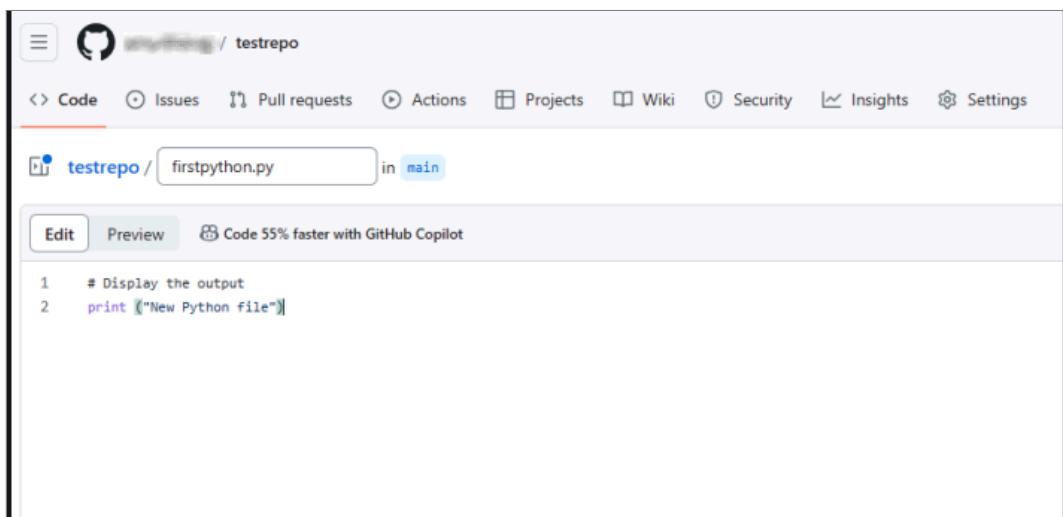
1. Click on the repository name **testrepo** to go back to the main branch.



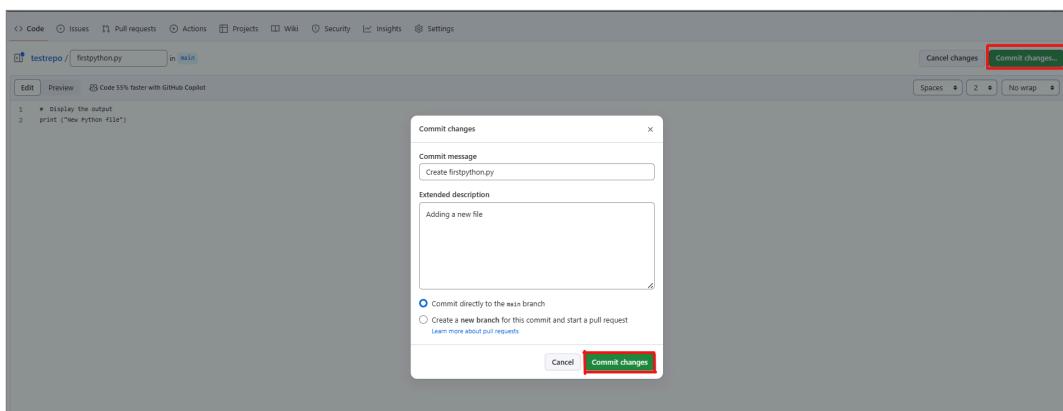
2. Click **Add file** and select **Create new file** to create a new file in the repository.



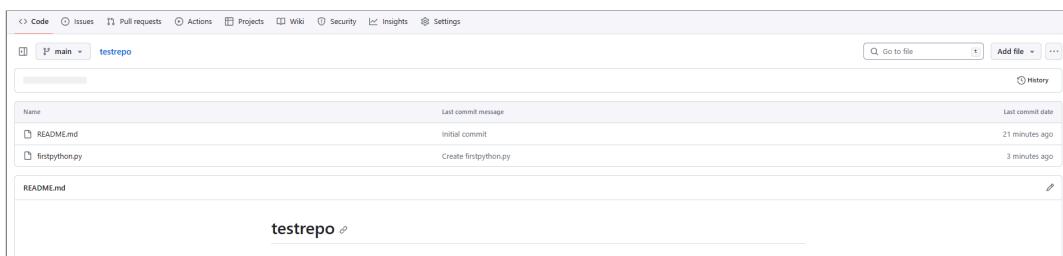
3. Enter a file name and extension; for example, `firstpython.py`. Add the lines of code displayed in the following image to your file:



4. Scroll to the top right of the page and click Commit changes. You can optionally add a description of your update (for example, "Adding a new file"). Click **Commit changes** to create your new file.



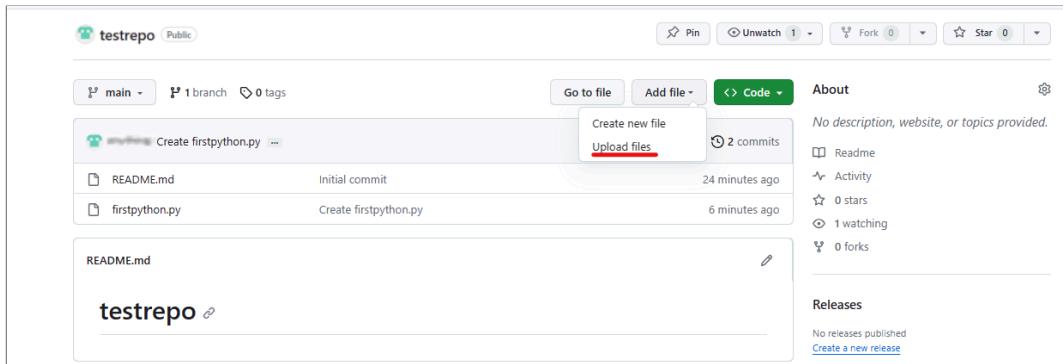
5. Your file is now added to your repository and the repository listing shows when the file was added or last revised.



4. Upload and commit a file

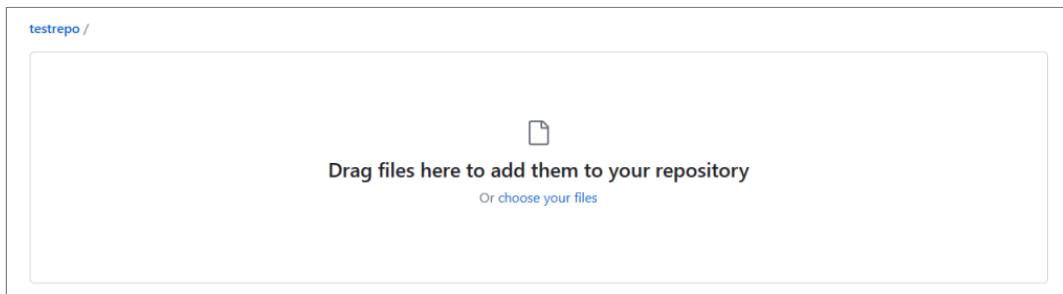
To upload a local file and commit it to your repository, complete the following steps:

1. On your repository page, click **Add file** and then select **Upload files** to upload a file.



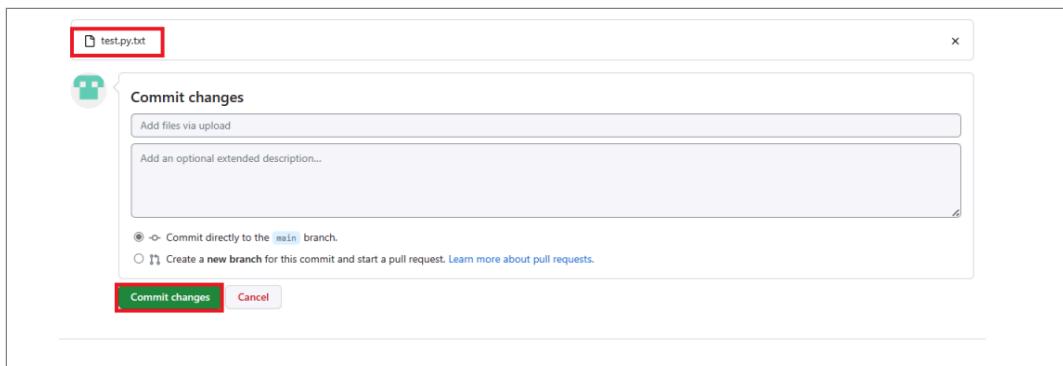
The screenshot shows a GitHub repository named 'testrepo'. The 'Code' tab is active. In the center, there's a code editor with the file 'README.md' containing the text 'testrepo'. At the top right of the code editor, there are buttons for 'Create new file' and 'Upload files'. The 'Upload files' button is highlighted with a red box. To its right, there's a button for '2 commits'. On the right side of the page, there's an 'About' section with basic repository statistics: 1 branch, 0 tags, 2 commits, 0 forks, 0 stars, and 1 watching.

2. Click **choose your files** and choose a file from your computer. You can upload any file (for example, a .txt, .ipynb, or .png file) to the repository.



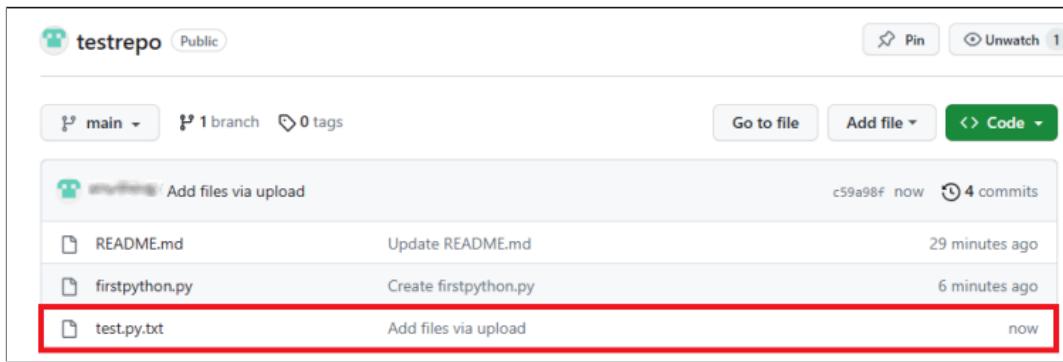
The screenshot shows a GitHub repository page with a large central area for uploading files. It features a paper icon and the text 'Drag files here to add them to your repository' in bold. Below this, there's a smaller link 'Or choose your files'.

3. When the file finishes uploading, click **Commit changes**.



The screenshot shows a 'Commit changes' dialog box. At the top, there's a file selection area with 'test.py.txt' selected. The main area contains fields for 'Add files via upload' and 'Add an optional extended description...'. Below these is a radio button group for committing: 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request.' At the bottom are 'Commit changes' and 'Cancel' buttons, with 'Commit changes' highlighted with a red box.

4. Your file is uploaded to the repository.



The screenshot shows the GitHub repository page again. The commit history now includes a new entry for 'test.py.txt': 'Add files via upload' at 'now'. This commit is highlighted with a red box. The other commits are 'README.md' at '29 minutes ago' and 'firstpython.py' at '6 minutes ago'.

Summary

Congratulations! In this lab, you have learned how to create a new repository, add a new file, edit a file, upload a file, and commit your changes. We encourage you to continue to update your repository to become familiar with the processes that you have learned.

Author details

Authors:

- Romeo Kienzler
- Malika Singla

Other contributors:

- Rav Ahuja
- Upkar Liddar



Skills Network