

# Hands-on Lab: Getting Started with GitHub

Estimated time: **20 min**

In this lab, you will get started with GitHub by creating a GitHub account and project and adding a file to it using its Web interface.

## Objectives

After completing this lab, you will be able to:

1. Describe GitHub
2. Create a GitHub account
3. Add a project and repo
4. Edit and create a file
5. Upload a file and Commit

## GitHub Overview

First, let's introduce you to GitHub. GitHub is a collection of folders and files. It is a Git repository hosting service, but it adds many of its own features. Git is a command-line tool. It hosts and maintains a server via command line. GitHub provides this Git server and a Web-based graphical interface for you. It also provides access control and collaboration features, such as wikis and basic task management tools for every project. In addition, 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 hosting Open Source projects.

## Exercise 1: Creating a GitHub Account

Please use the following steps to create an account on GitHub:

Step 1: Create an account: <https://github.com/join>

### Sign up to GitHub

Email\*

Password\*

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username\*

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Continue >

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.

**NOTE:** If you already have a GitHub account, you can skip this step and simply log in to your account.

Step 2: Provide the necessary details to create an account as shown below:

Sign up to GitHub

Email\*

Password\*

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username\*

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Continue >

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.

Click Continue.

Step 3: Click Visual puzzle to verify the account.

Verify your account

Please solve a puzzle so we can safely create your account.


Visual puzzle

Audio puzzle

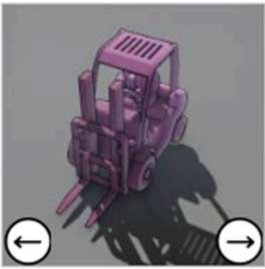
Step 4: Solve the puzzle and Submit.

Verify your account

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



Match this angle



Submit

Audio puzzle

Restart

Step 5: Open your email, find the GitHub verification email, copy the verification code, and return to the GitHub Signup page to enter it.

Step 6: Confirm youe email address using the verification code and Continue.

## Confirm your email address

We have sent a code to [redacted]@gmail.com

Enter code

Continue >

Didn't get your email? [Resend the code](#) or [update your email address](#).

**NOTE:** If you do not receive the verification email, click Resend the code.

Step 7: Sign in to GitHub, enter Id and Password and Sign in



## Sign in to GitHub

Your account was created successfully. Please sign in to continue



Username or email address

Password

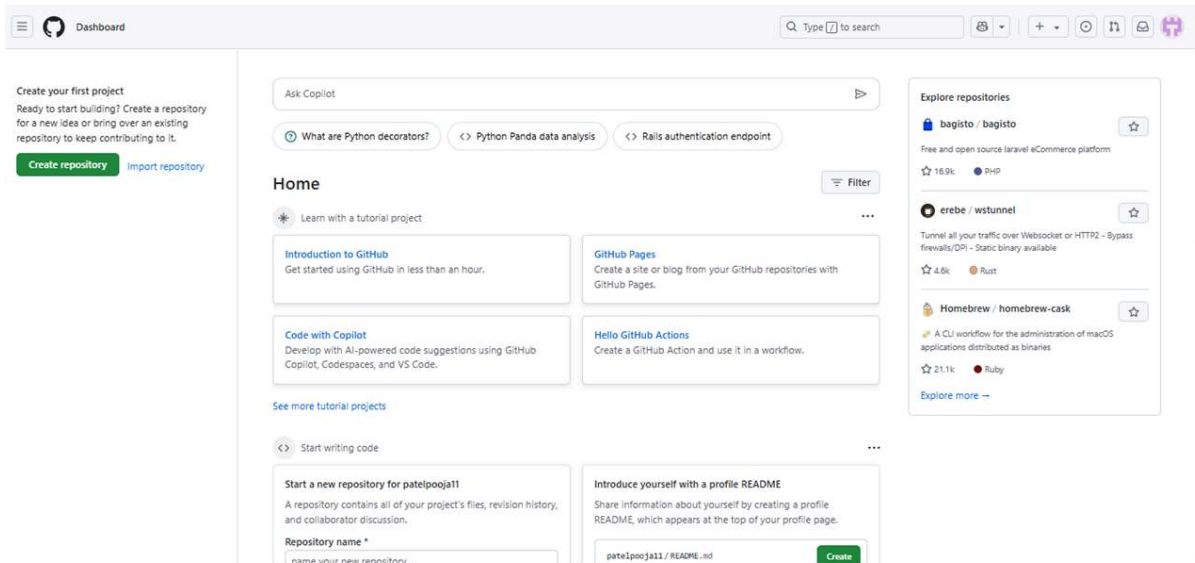
[Forgot password?](#)

Sign in

[Sign in with a passkey](#)

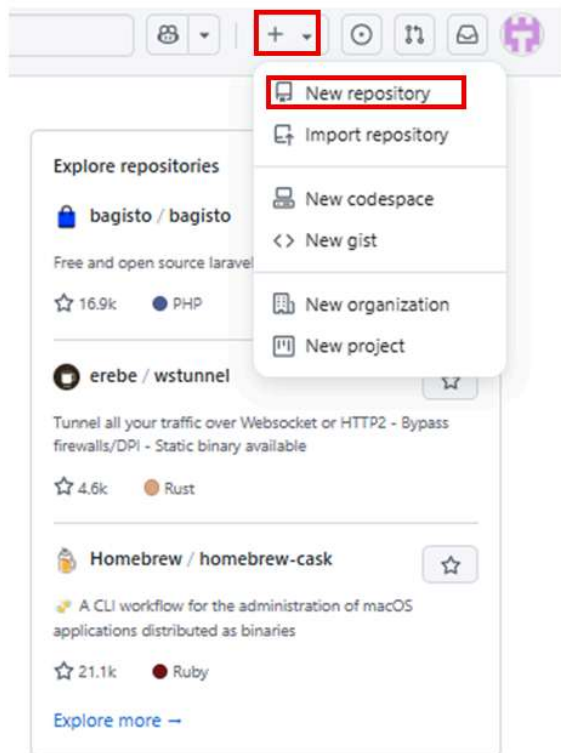
New to GitHub? [Create an account](#)

You will see the home page.



## Exercise 2: Adding a project and repo

Step 1: Click the + symbol and click New repository.



Step 2: Provide a name for the repository and initialize it with the empty README.md file.

## 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 (\*).

Owner \*  / Repository name \*

TestRepo is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-chainsaw](#) ?

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

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

Choose a license

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

You are creating a public repository in your personal account.

Create repository

Click Create repository.

Now, you will be redirected to the repository you have created.

Let us start editing the repository.

## Exercise 3: Create and edit a file

### Exercise 3a: Edit a file

Step 1: Once the repository is created, the root folder of your repository is listed by default, and has just one file, `ReadMe.md`. Click the pencil icon to edit the file.

TestRepo Public 1

main 1 Branch 0 Tags

patelpooja11 Initial commit 3 minutes ago 1 Commit

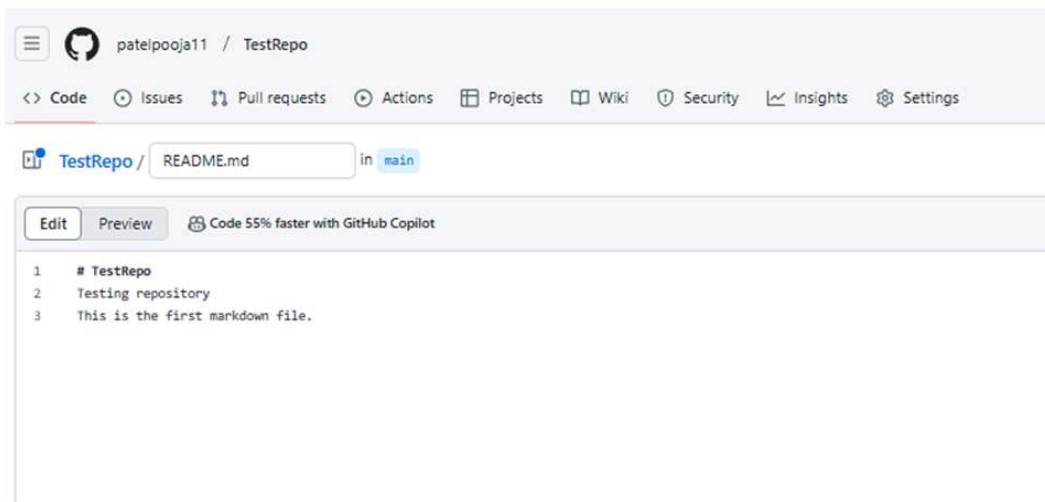
README.md Initial commit 3 minutes ago

README

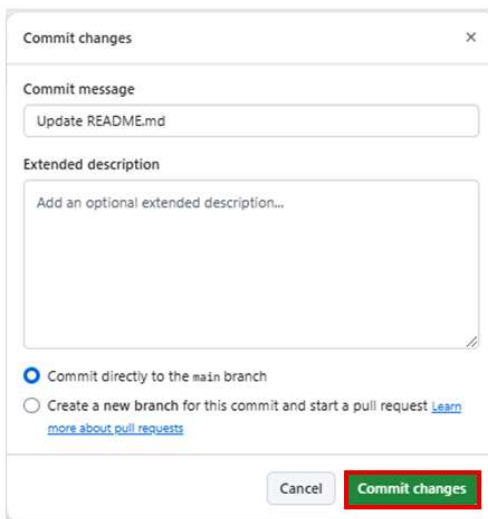
TestRepo

Testing repository

Step 2: Add some text to the file.



Step 3: After adding the text and click Commit Changes.



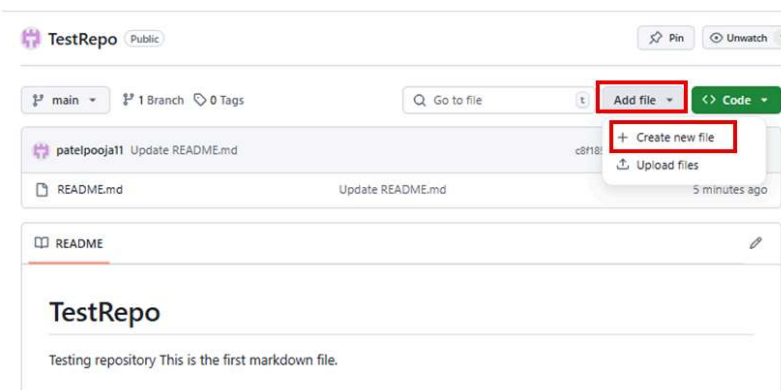
Now, check that your file is edited with the new text.

### Exercise 3b: Create a new file

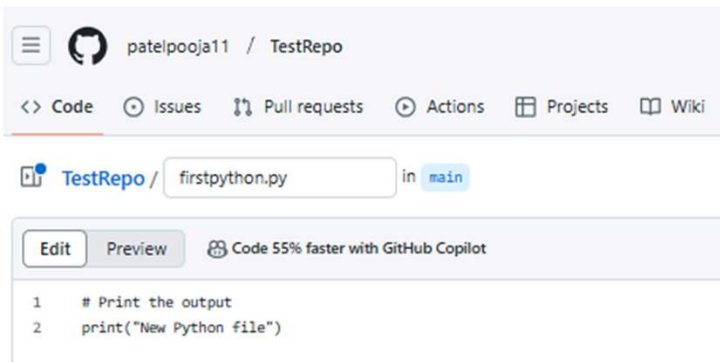
Step 1: Click the repository name to return to the master branch, like in this testrepo.



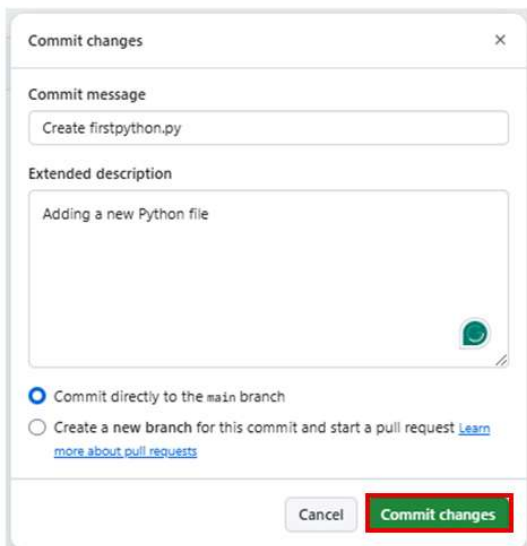
Step 2: Click Add file and select Create new file to create a file in the repository.



Step 3: Provide the file name and the extension of the file. For example, firstpython.py and add the lines.



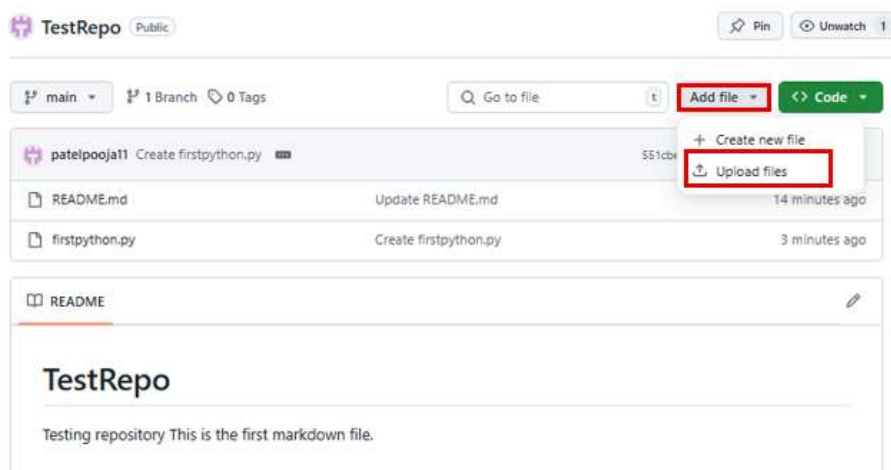
Step 4: Commit changes after adding the text. Add description of the file (optional) and click Commit changes.



Step 5: Your file is now added to your repository, and the repository listing shows when the file was added and changed.


## Exercise 4: Upload a file & Commit

Step 1: Click Add file and select Upload files to upload a file (any .txt, .ipynb, .png file) in the repository from the local computer.



Step 2: Click choose your files and select any files from your computer.

TestRepo /

  
Drag files here to add them to your repository  
[Or choose your files](#)

### Commit changes

Add files via upload

Add an optional extended description...

☒ Commit directly to the `main` branch.  
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

[Commit changes](#) [Cancel](#)

Step 3: Once the file finishes uploading, click **Commit changes**.

Test.ipynb

### Commit changes

Add files via upload

Add an optional extended description...

☒ Commit directly to the `main` branch.  
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

[Commit changes](#) [Cancel](#)

Step 4: Now, your file is uploaded in the repository.

TestRepo Public Pin Unwatch 1

main 1 Branch 0 Tags  Add file Code

patelpooja11 Add files via upload 35f86b6 · now 4 Commits

README.md	Update README.md	17 minutes ago
Test.ipynb	Add files via upload	now
firstpython.py	Create firstpython.py	7 minutes ago

## Summary

In this document, you have learned how to create a new repository, add a new file, edit a file, upload a file in a repository, and commit the changes.

## Author(s)

Romeo Kienzler

Malika Singla

## Other Contributor(s)

Rav Ahuja



# Skills Network