

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.

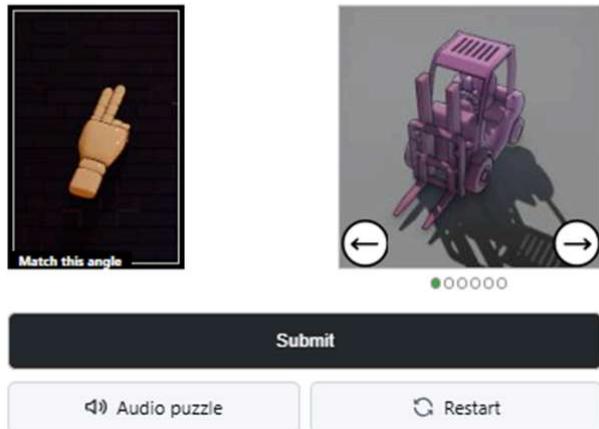
Visual puzzle

⟳ 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)



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 your 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

X

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.

The screenshot shows the GitHub Home page. At the top, there's a search bar and several navigation icons. Below the search bar, there's a section titled "Ask Copilot" with links to "What are Python decorators?", "Python Panda data analysis", and "Rails authentication endpoint". The main "Home" section features several cards: "Introduction to GitHub", "GitHub Pages", "Code with Copilot", and "Hello GitHub Actions". To the right, there's a sidebar titled "Explore repositories" with cards for "bagisto / bagisto", "erebe / wstunnel", and "Homebrew / homebrew-cask". At the bottom left, there's a "Start writing code" section with a "Create" button.

Exercise 2: Adding a project and repo

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

The screenshot shows the GitHub repository creation menu. A red box highlights the "+" icon in the top toolbar. A dropdown menu is open, showing options: "New repository" (which is selected and highlighted with a red box), "Import repository", "New codespace", "New gist", "New organization", and "New project". Below the menu, there are sections for "Explore repositories" featuring "bagisto / bagisto" and "erebe / wstunnel", and a "See more" link.

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 *

 patelpooja11 /

TestRepo is available.

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

Description (optional)

Testing repository

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

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.](#)

① You are creating a public repository in your personal account.

Create repository

Click Create repository.

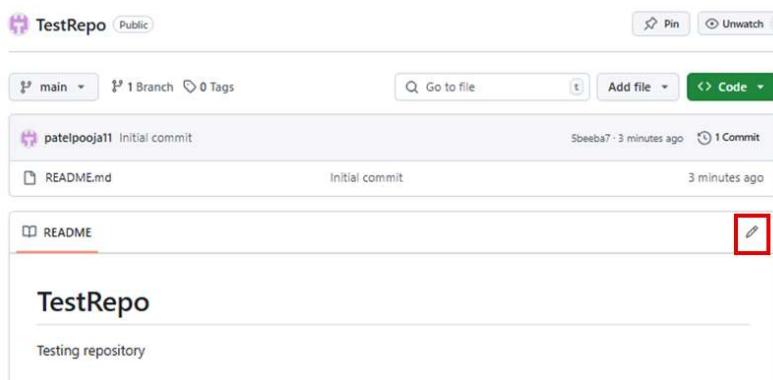
Now, you will be redirected to the repository you have created.

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.



The screenshot shows a GitHub repository named "TestRepo". The repository is public. It has 1 branch and 0 tags. The main page displays the contents of the "README.md" file, which contains the text "Testing repository". To the right of the file name, there is a small edit icon (pencil symbol) enclosed in a red box, indicating that the file can be edited.

Step 2: Add some text to the file.

The screenshot shows a GitHub repository named 'TestRepo'. The 'Code' tab is selected. A modal window is open over the repository page, showing the content of the 'README.md' file. The file contains three lines of text:

```

1 # TestRepo
2 Testing repository
3 This is the first markdown file.

```

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

The dialog box has the following fields:

- Commit message:** Update README.md
- Extended description:** Add an optional extended description...
- Branch selection:** Commit directly to the main branch (radio button selected)
- Other options:** Create a new branch for this commit and start a pull request (radio button unselected)
- Buttons:** Cancel (grayed out) and Commit changes (highlighted in red)

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.

The screenshot shows the 'TestRepo' repository page. The 'Code' tab is selected. The 'README.md' file content is displayed:

```

TestRepo
Testing repository This is the first markdown file.

```

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

The screenshot shows the 'TestRepo' repository page. The 'Code' tab is selected. A dropdown menu is open at the top right, with the 'Create new file' option highlighted with a red box. The repository page shows the following details:

- Branch: main
- Commits: patelpooja11 Update README.md (c8f183c, 1 minute ago)
- File: README

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

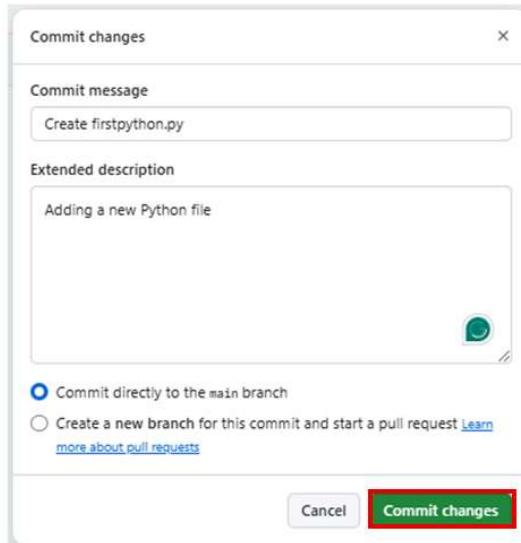
The screenshot shows the GitHub Copilot interface. At the top, it says "patelpooja11 / TestRepo". Below that is a navigation bar with "Code", "Issues", "Pull requests", "Actions", "Projects", and "Wiki". A search bar shows "TestRepo / firstpython.py in main". Below the search bar are two buttons: "Edit" and "Preview". A message says "Code 55% faster with GitHub Copilot". The code editor contains the following Python code:

```

1 # Print the output
2 print("New Python file")

```

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.

The screenshot shows the GitHub repository listing for "TestRepo". It lists files: README.md, firstpython.py, and README. The "Add file" button is highlighted with a red box. A dropdown menu shows options: "+ Create new file" and "Upload files", with "Upload files" also highlighted with a red box. The repository details show "1 Branch" and "0 Tags". The README file content is shown below.

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 Go to file 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