**Creating Data Visualizations using ggplot**

A logo with purple text

Description automatically generated with medium confidence

**Objective for Exercise**

We will create different data visualizations using the ggplot package using the inbuilt dataset in R called mtcars

1. Click on the + symbol on the top left and choose R Script from the menu to open a new R edit window in RStudio:

A screenshot of a computer

Description automatically generated

1. Read and view the first 5 rows of the Data using the following:
2. 1
3. 2
4. 3
5. 4
6. 5
7. 6
8. 7
9. 8
10. 9
11. library(datasets)
12. #Load Data
13. data(mtcars)
14. #View first 5 rows
15. head(mtcars, 5)

Copied!

1. Type this ?mtcars to get information about the variables. This will print the information at the bottom right panel, on the Help tab
2. Copy and paste the following code to load the ggplot package and create a scatterplot of disp and mpg.
3. 1
4. 2
5. 3
6. 4
7. 5
8. 6
9. #load ggplot package
10. library(ggplot2)
11. #create a scatterplot of displacement (disp) and miles per gallon (mpg)
12. ggplot(aes(x=disp,y=mpg,),data=mtcars)+geom\_point()

Copied!

1. Use the following code to add a title.
2. 1
3. 2
4. 3
5. 4
6. #Add a title
7. ggplot(aes(x=disp,y=mpg,),data=mtcars)+geom\_point()+ggtitle("displacement vs miles per gallon")

Copied!

1. Use the following code to change the name of the x-axis and y-axis
2. 1
3. 2
4. 3
5. 4
6. #change axis name
7. ggplot(aes(x=disp,y=mpg,),data=mtcars)+geom\_point()+ggtitle("displacement vs miles per gallon") + labs(x = "Displacement", y = "Miles per Gallon")

Copied!

1. Use the following to create a boxplot of the the distribution of mpg for the individual Engine types vs Engine (0 = V-shaped, 1 = straight)  
   To do this you have to make vs a string or factor.
2. 1
3. 2
4. 3
5. 4
6. 5
7. 6
8. #make vs a factor
9. mtcars$vs <- as.factor(mtcars$vs)
10. #create boxplot of the distribution for v-shaped and straight Engine
11. ggplot(aes(x=vs, y=mpg), data = mtcars) + geom\_boxplot()

Copied!

1. Add color to the boxplots to help differentiate:
2. 1
3. 2
4. 3
5. ggplot(aes(x=vs, y=mpg, fill = vs), data = mtcars) +
6. geom\_boxplot(alpha=0.3) +
7. theme(legend.position="none")

Copied!

1. Finally, let us create the histogram of weight wt.
2. 1
3. ggplot(aes(x=wt),data=mtcars) + geom\_histogram(binwidth=0.5)

Copied!

This concludes this lab, we hope that you had fun!

**Author(s)**

[Aije Egwaikhide](https://www.linkedin.com/in/aije-egwaikhide/)

**Change log**

| **Date** | **Version** | **Changed by** | **Change Description** |
| --- | --- | --- | --- |
| 2023-05-04 | 1.1 | Benny | Added page numbers and republished |
| 2020-12-14 | 1.0 | Aije | Created initial version of the lab |

+-

# Hands-on Lab: Getting Started with GitHub

A logo with purple text

Description automatically generated with medium confidence

**Effort:** 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 us 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.

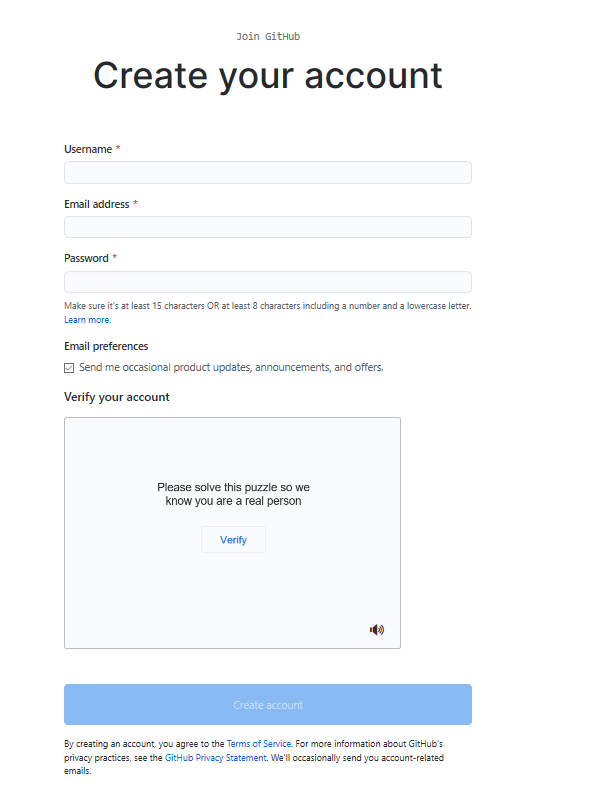
## Exercise 1: Creating a GitHub Account

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

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

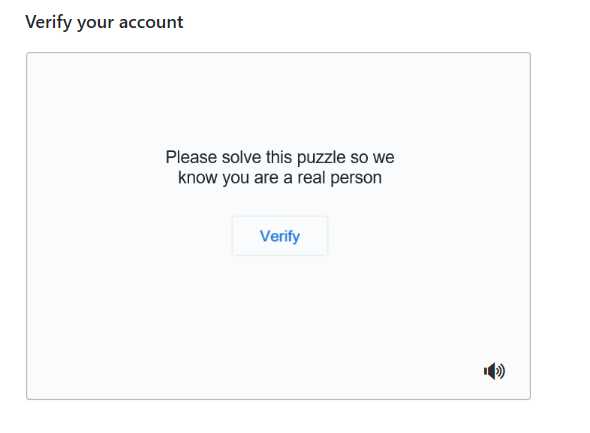
**NOTE:** If you already have a GitHub account, you can skip this step and simply log in to your account.

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

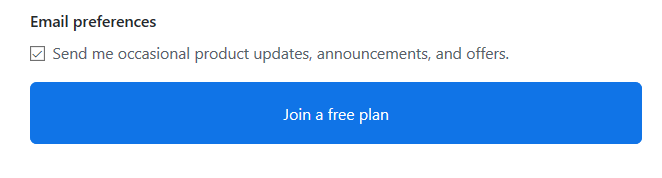


Click Create account.

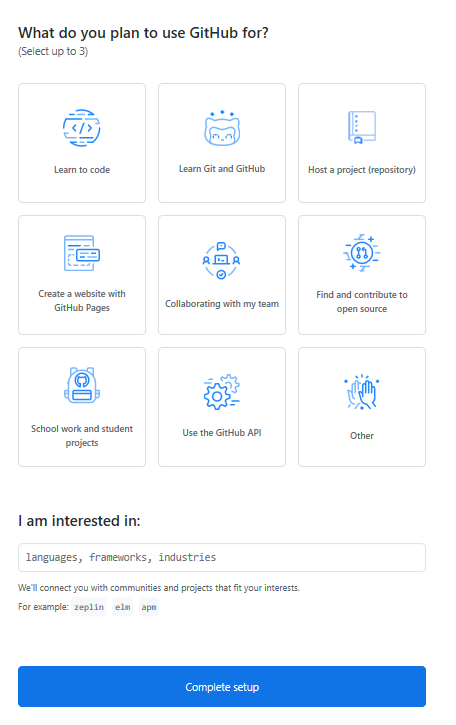
Step 3: Click Verify to verify the account and click Done.



Step 4: After verification, click Join a Free Plan.

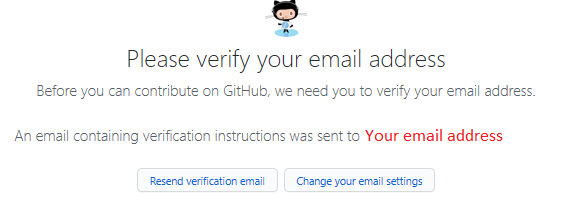


Step 5: Select the details as shown below and click Complete setup.

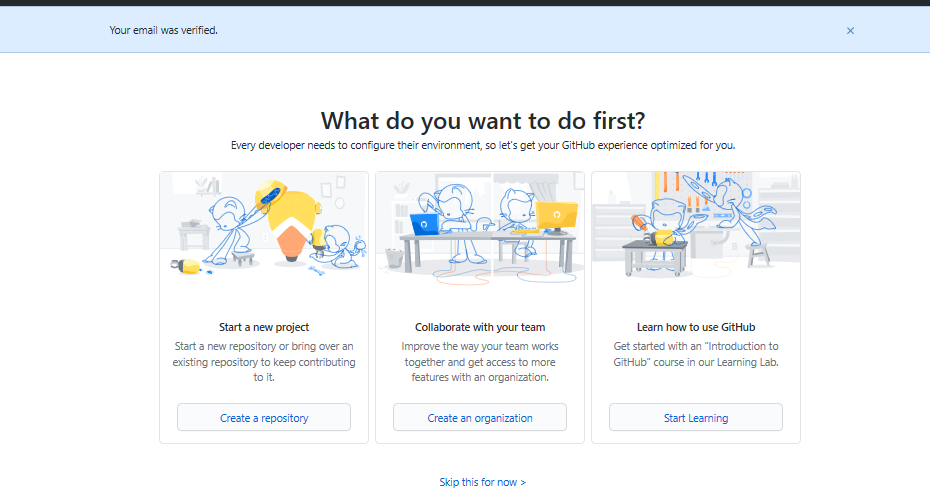


Step 6: Go to your email, find the verification email from GitHub, and click the Verify your email button or link in that email to verify.

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

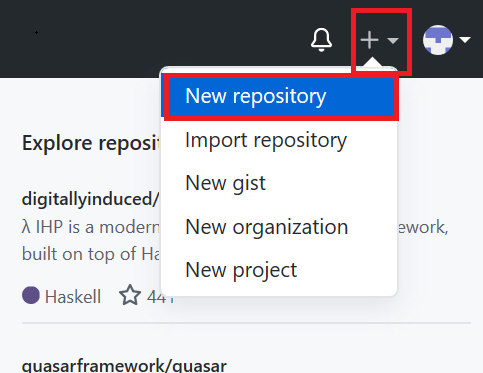


Email is verified.

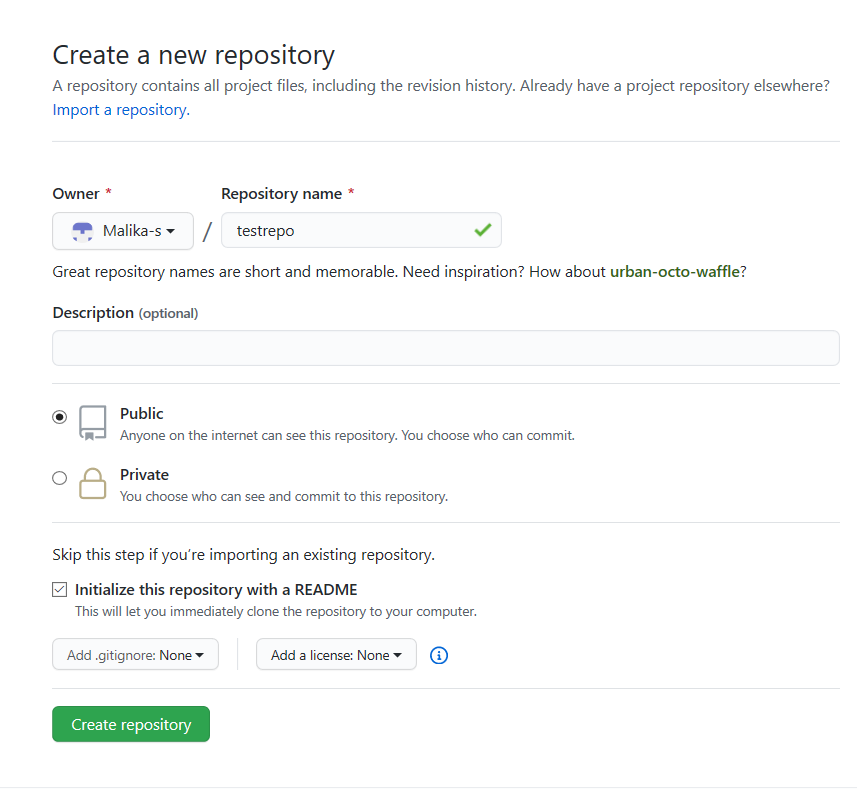


## Exercise 2: Adding a project and repo

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.



Click Create repository.

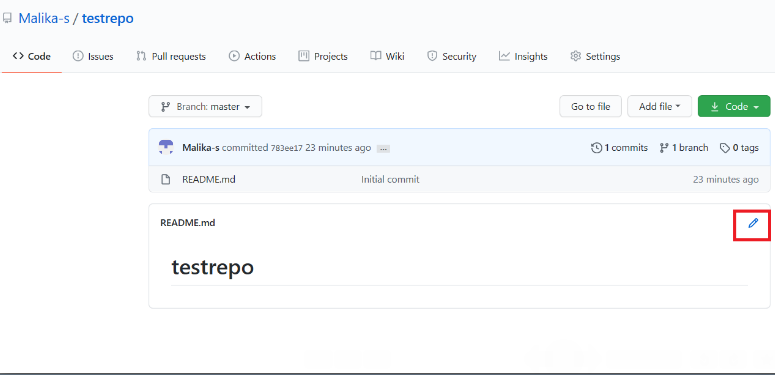
Now, you will be redirected to the repository you have created.

Let’s start editing the repository.

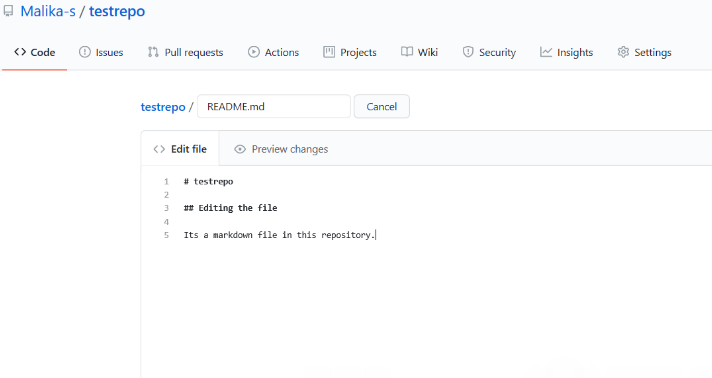
## Exercise 3: Create and edit a file

### Exercise 3a: Edit a file

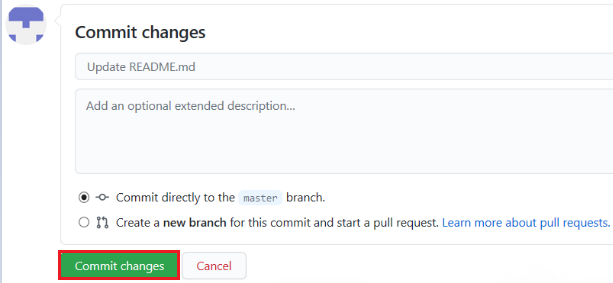
Step 1: Once the repository is created, the root folder of your repository is listed by default, and has just one file, ReadMe.md. Click the pencil icon to edit the file.



Step 2: Add some text to the file.



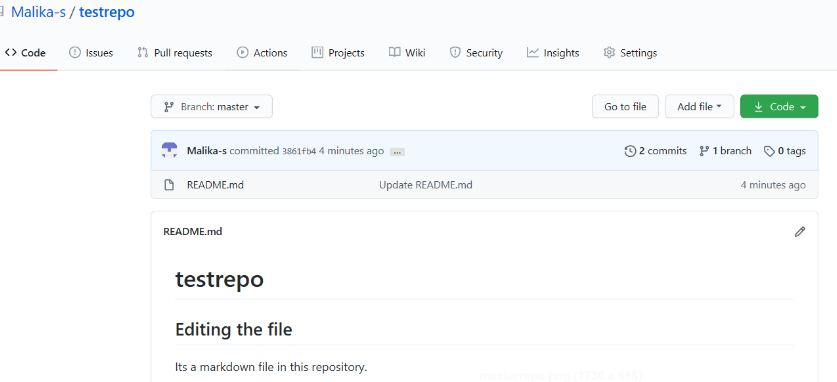
Step 3: Scroll down the page after adding the text and click Commit Changes.



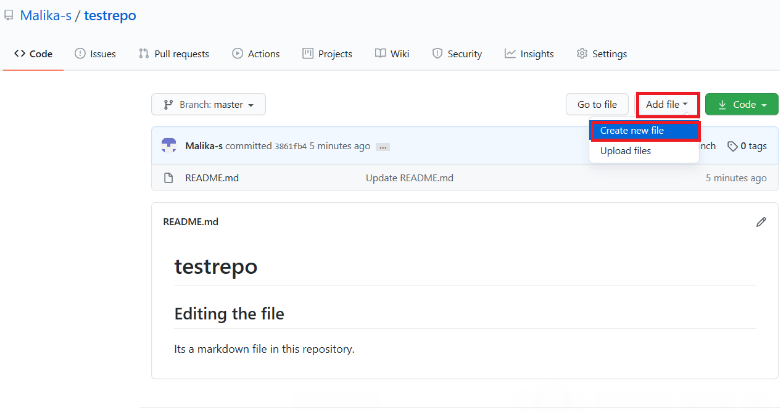
Now, check that your file is edited with the new text.

### Exercise 3b: Create a new file

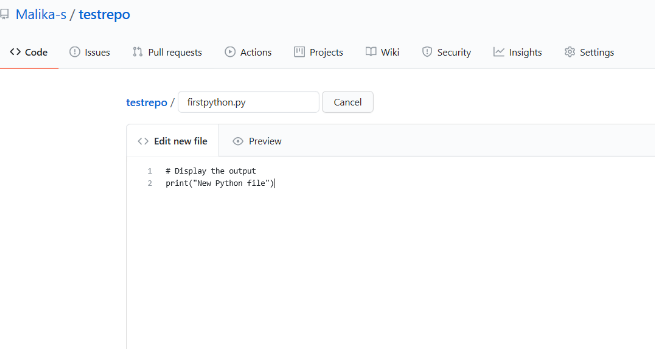
Step 1: Click the repository name to return to the master branch, like in this testrepo.



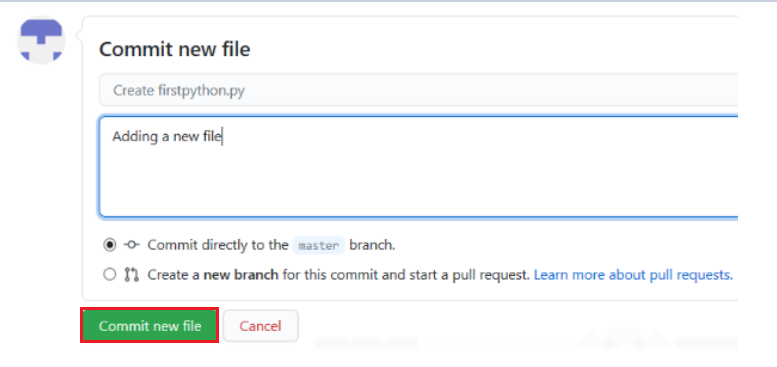
Step 2: Click Add file and select Create New file to create a file in the repository.



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



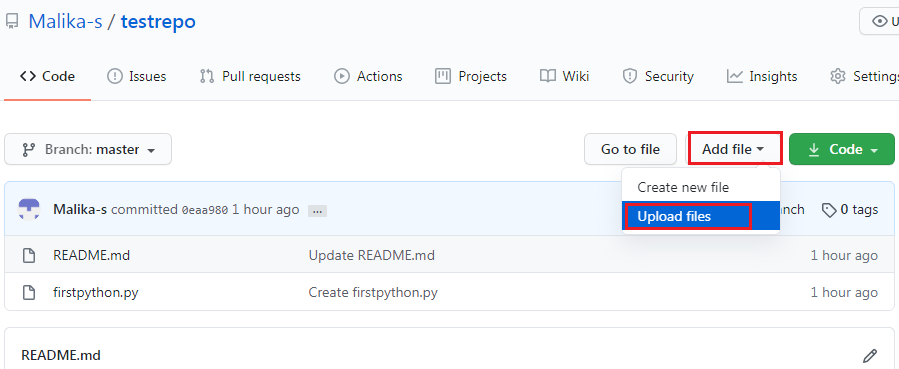
Step 4: Scroll down the page after adding the text. Add description of the file (optional) and click Commit new file.



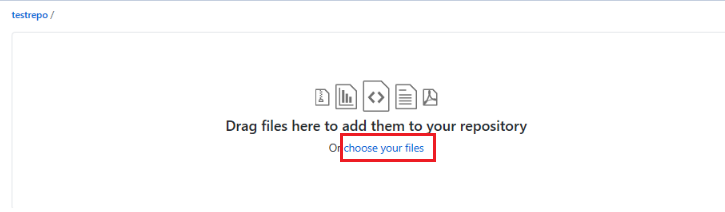
Step 5: Your file is now added to your repository, and the repository listing shows when the file was added and changed.

## Exercise 4: Upload a file & Commit

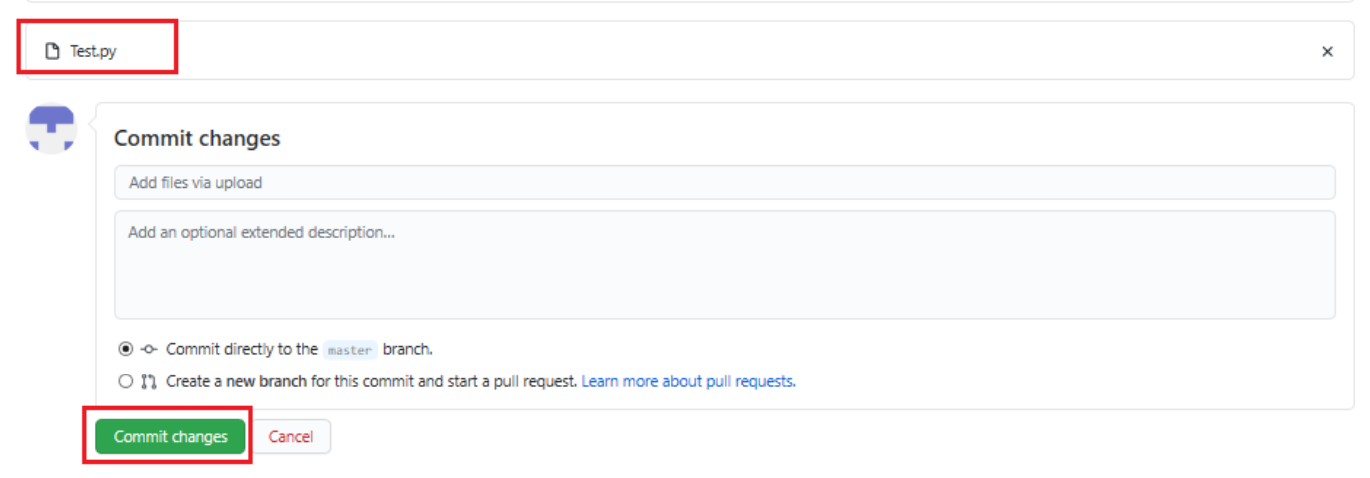
Step 1: Click Add file and select Upload files to upload a file (any .txt, .ipynb, .png file) in the repository from the local computer.



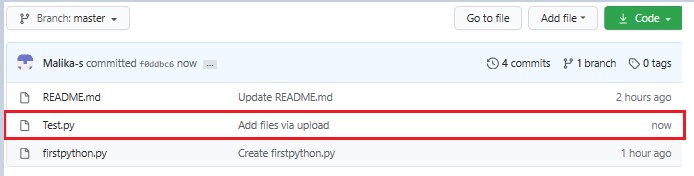
Step 2: Click choose your files and select any files from your computer.



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



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



## 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

## Changelog

| **Date** | **Version** | **Changed by** | **Change Description** |
| --- | --- | --- | --- |
| 2023-01-17 | 0.6 | Steve Hord | QA pass with edits |
| 2020-07-16 | 0.5 | Malika Singla | Spell check, step number added |
| 2020-07-14 | 0.4 | Rav Ahuja | Changed logo, updated effort, title, intro, objectives, added Authors and Changelog |
| 2020-07-13 | 0.3 | Malika Singla | Added to GitLab and made some formatting changes, added objectives, etc. |
| 2020-07-03 | 0.2 | Malika Singla | Ported to markdown and added GitHub account signup, new screenshots, etc. |
| 2020-06-30 | 0.1 | Romeo Kienzler | Drafted initial version |

### © IBM Corporation 2020. All rights reserved.

Tikur azmud, rosmery, qurfd, qokonet, qaster oil=oil

Telba, rosmey, coconet, key shnkurt=conditioner

Butter,blue majic, telba, kurfd=conditionner