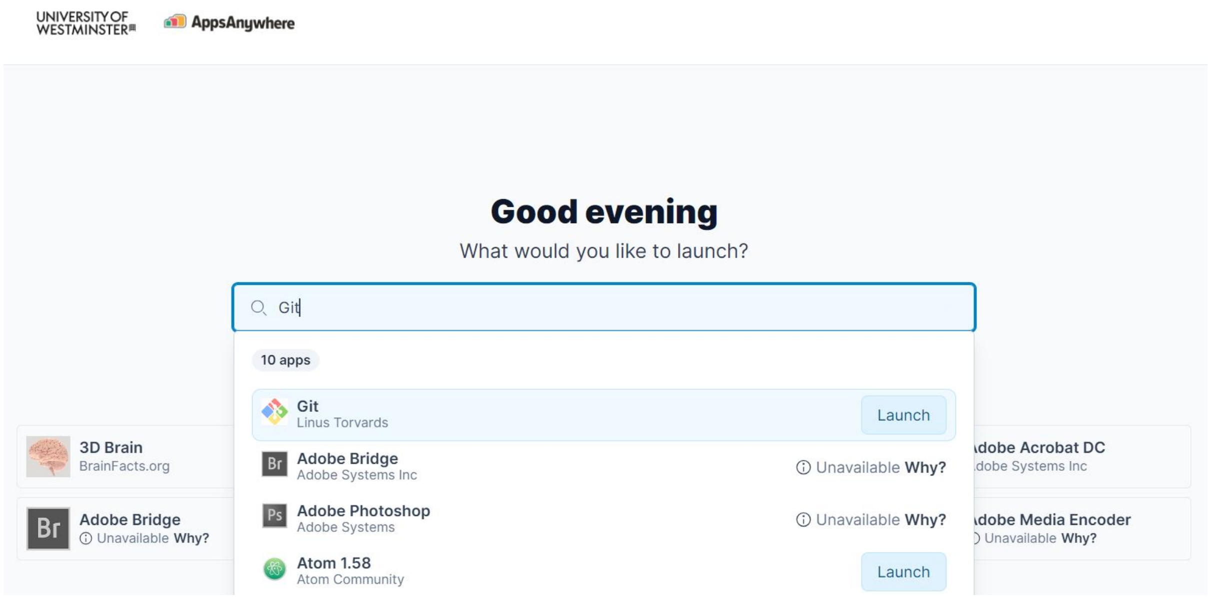
Week 8 Computer Seminar

1. Setting up

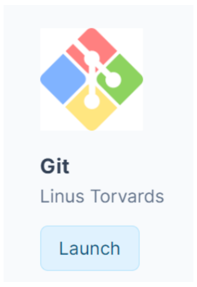
1.1. Installing Git

You may install Git from https://git-scm.com/ into your personal computer.

Git can also be installed from AppsAnywhere (https://appsanywhere.westminster.ac.uk/). Type "Git" in the search bar.



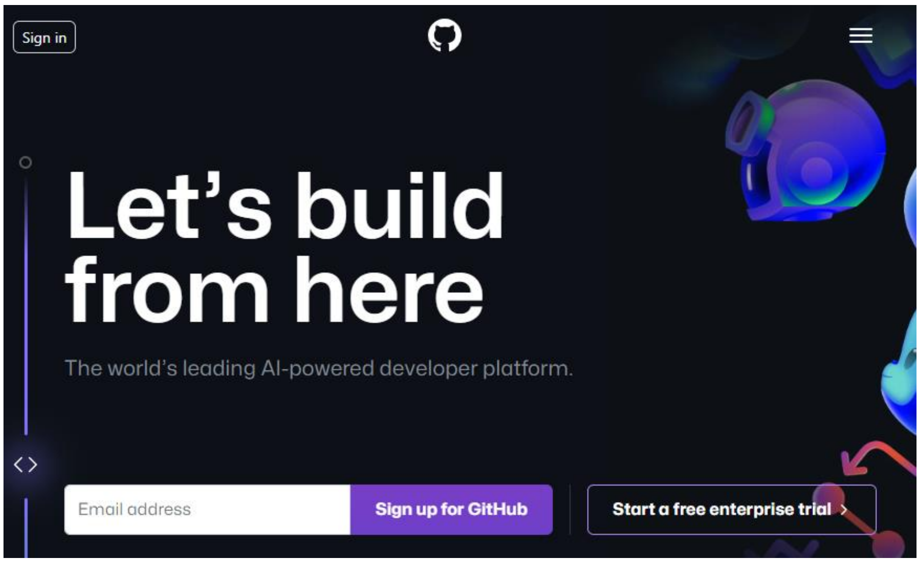
Click the "Launch" button.



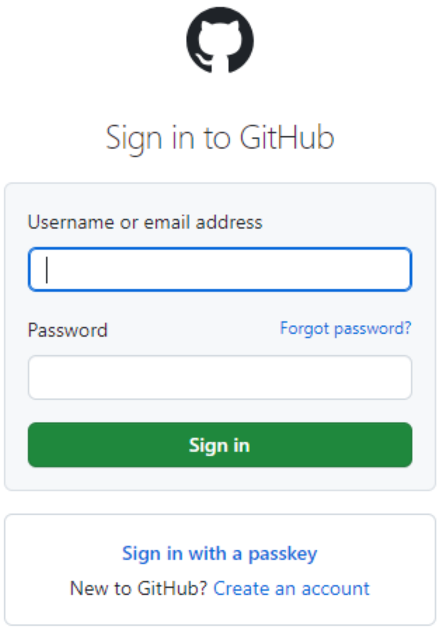
You can then have access to Git Bash for the command line and Git GUI.

1.2. Signing up for a GitHub account

You will need access to GitHub. You must sign up for a GitHub account (https://github.com/). Please complete the process for creating a GitHub account by first entering your email address and clicking on "*sign up for GitHub*". Follow the instructions.

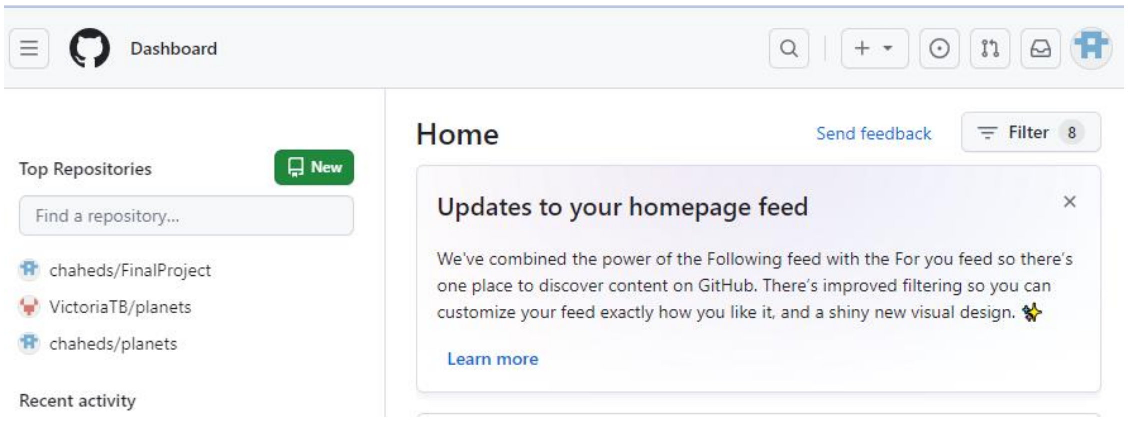

Sign in
Let's build from here
The world's leading AI-powered developer platform.
Email address
Sign up for GitHub
Start a free enterprise trial

If you already have a GitHub account, then simply sign in (https://github.com/login).



2. Exploring GitHub

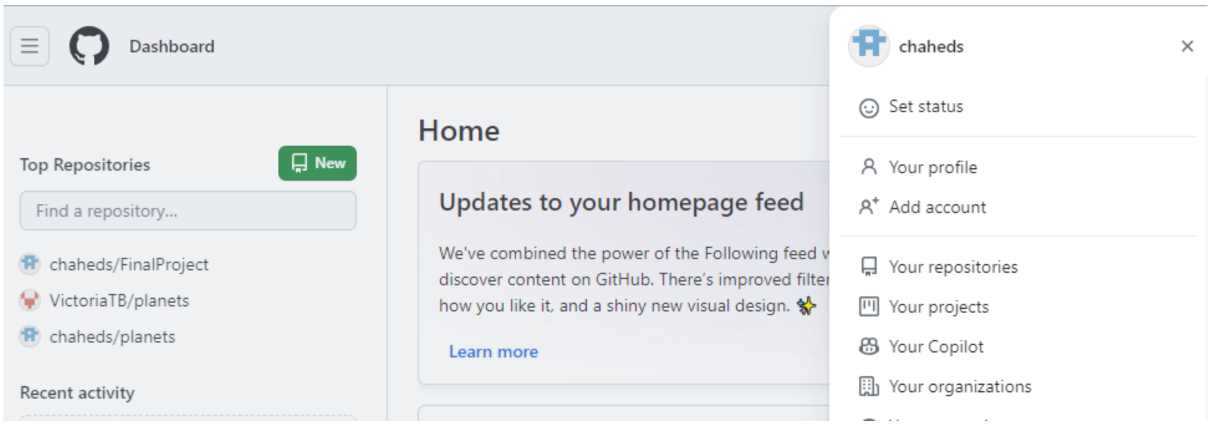
Once you create a GitHub account, you are on your GitHub website.



Updates to your homepage feed

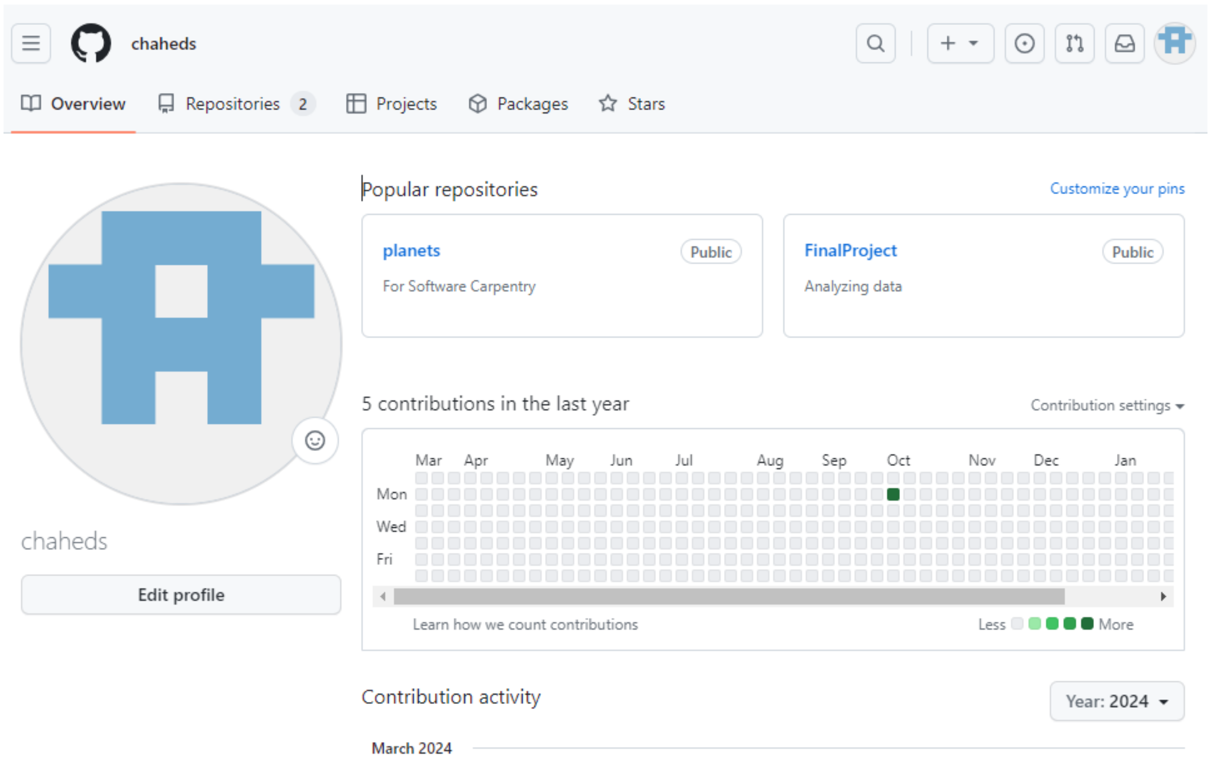
We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design.

You can access the main menu by clicking on the icon on the top right corner. You get a drop-down list of all key features including your profile, repositories, projects, settings associated with your account, etc.



Go to your profile. It contains your popular repositories, contribution graph and activities.

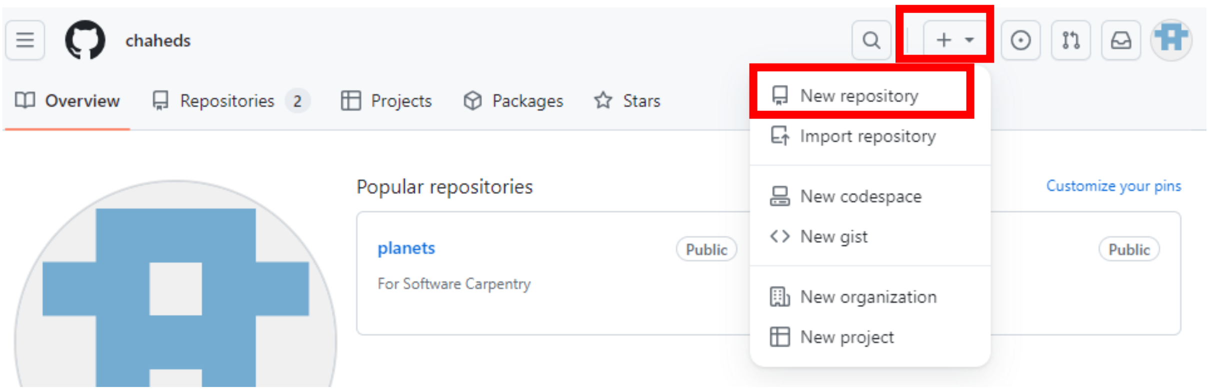
Your profile contributions graph is a record of contributions you've made to repositories on GitHub.com. Your first square is for joining GitHub, and you will earn more as you make additional contributions.



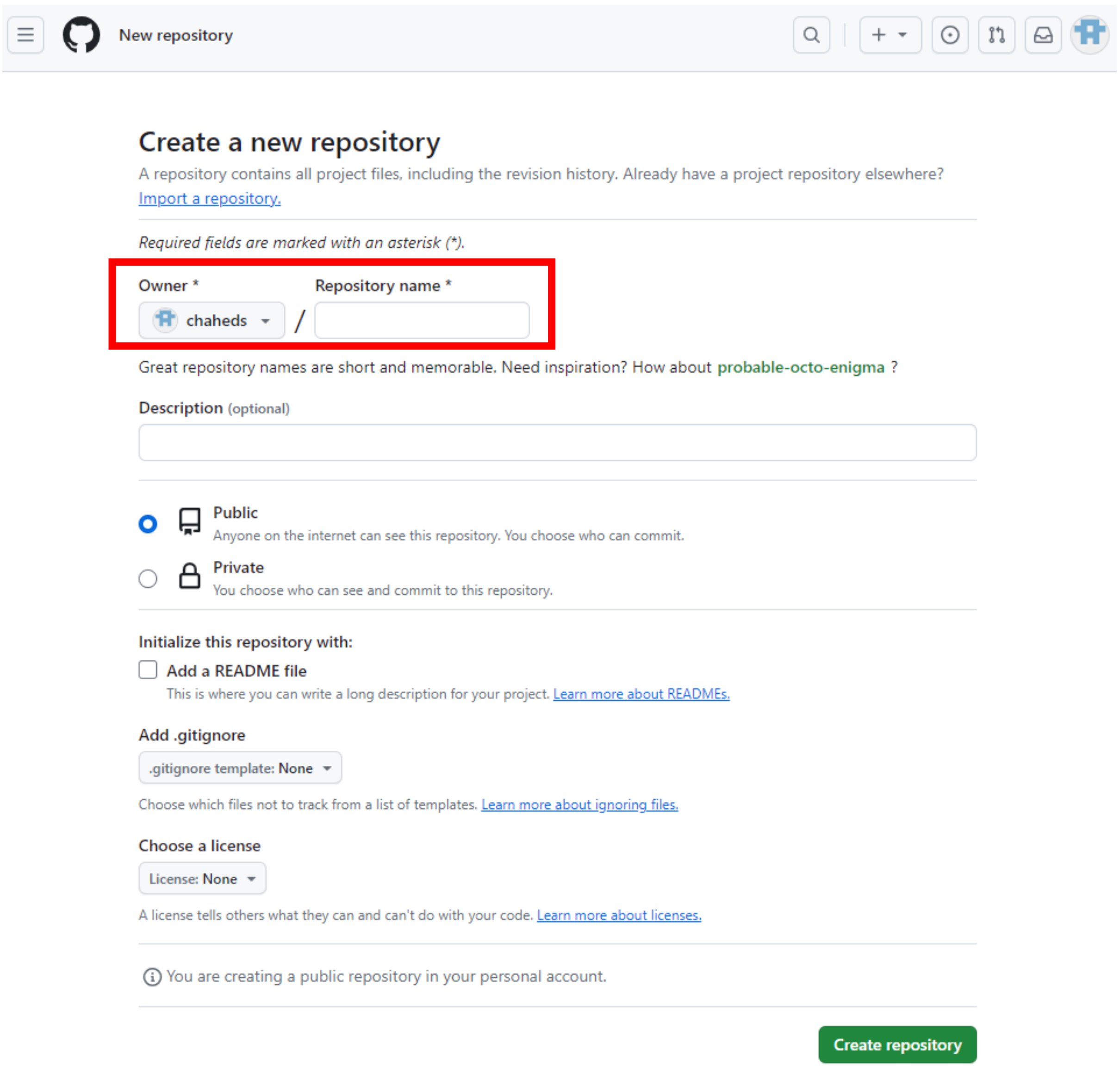
2.1. Creating a GitHub repository

Let’s create a repository. That is where you will be placing your project.

At the top right, click on the “**+**” to create a repository, and then from the drop down menu in the top right, select “**New repository**”.



Let’s fill in the requested information about the new repository starting by naming the repo.



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 * chaheds / Repository name *

Great repository names are short and memorable. Need inspiration? How about probable-octo-enigma ?

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

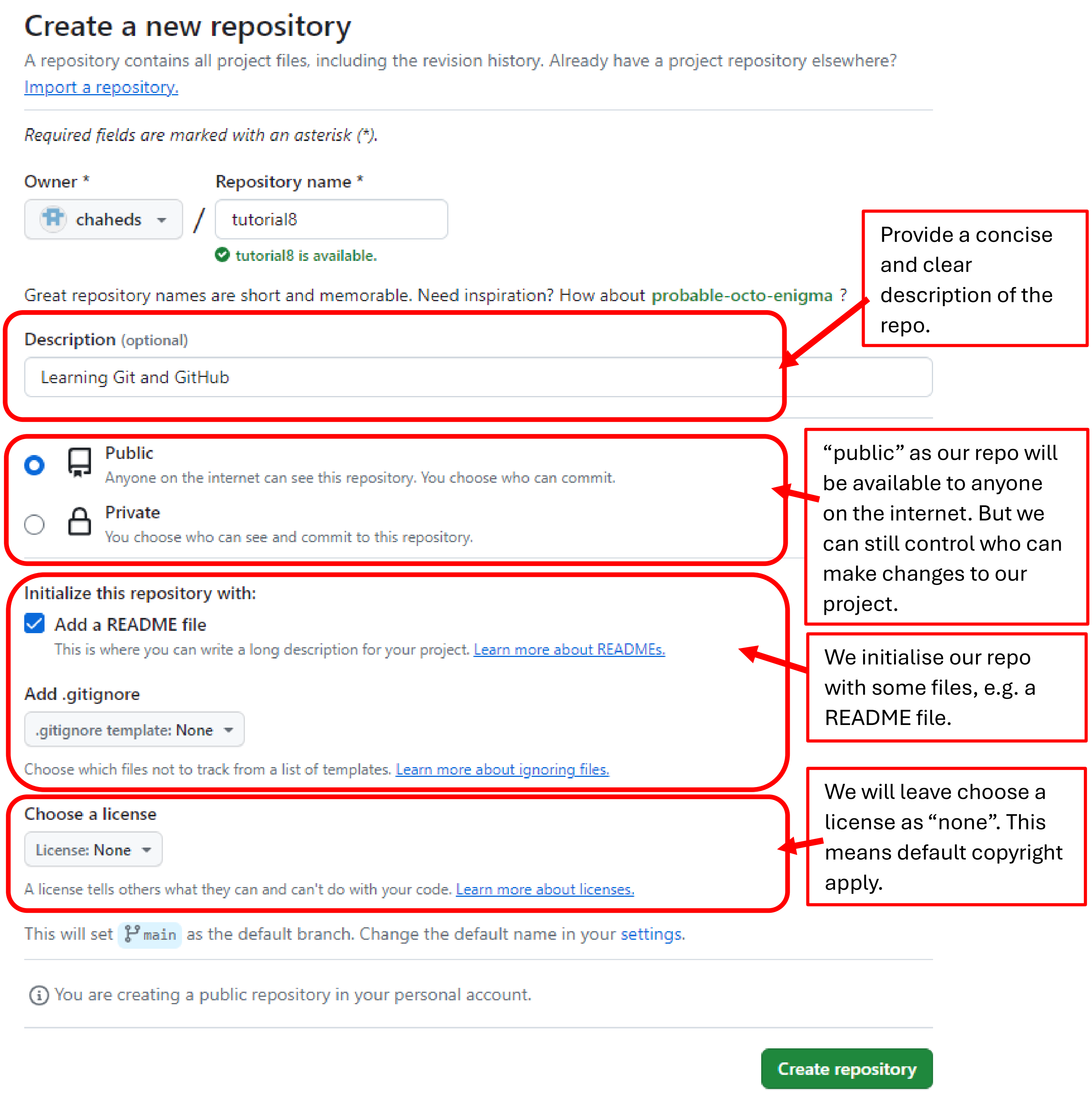
Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.



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 * chaheds / Repository name * tutorial8

tutorial8 is available.

Great repository names are short and memorable. Need inspiration? How about probable-octo-enigma ?

Description (optional)

Learning Git and GitHub

Provide a concise and clear description of the repo.

● 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.

"public" as our repo will be available to anyone on the internet. But we can still control who can make changes to our project.

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.

We initialise our repo with some files, e.g. a README file.

Choose a license

License: None

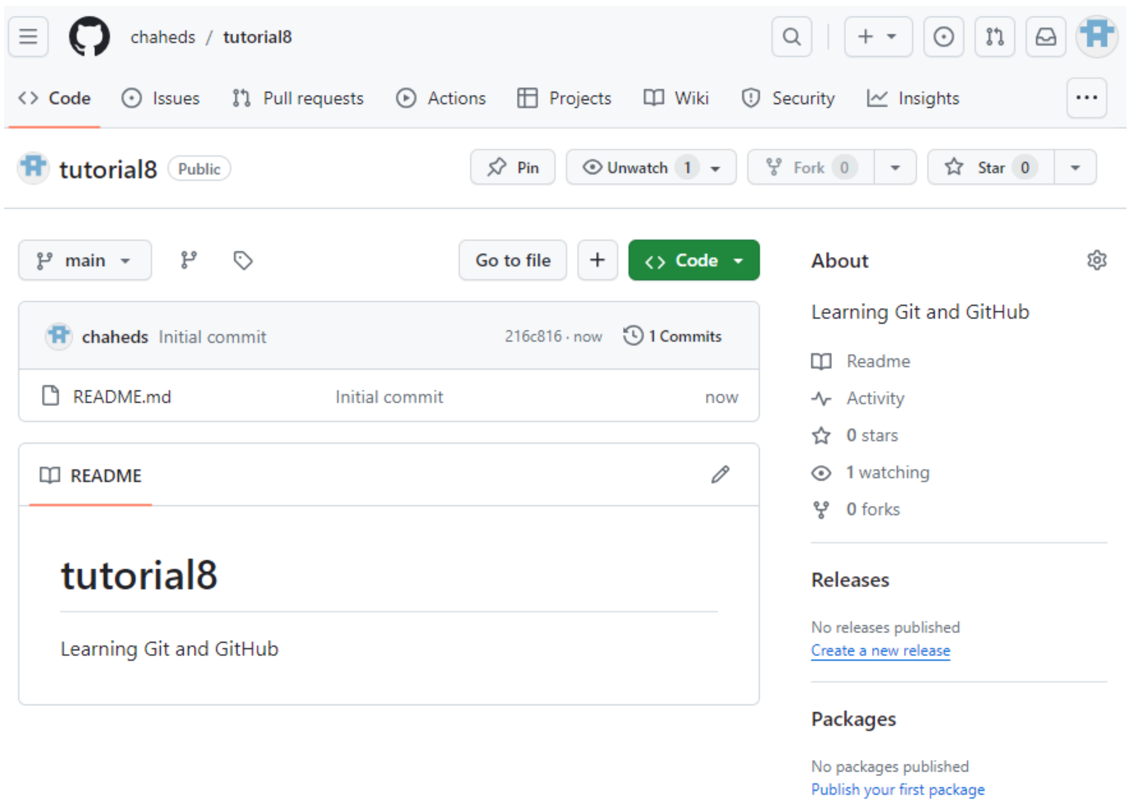
A license tells others what they can and can't do with your code. Learn more about licenses.

We will leave choose a license as "none". This means default copyright apply.

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.

Once all fields have been completed, press the green button to create your repo.



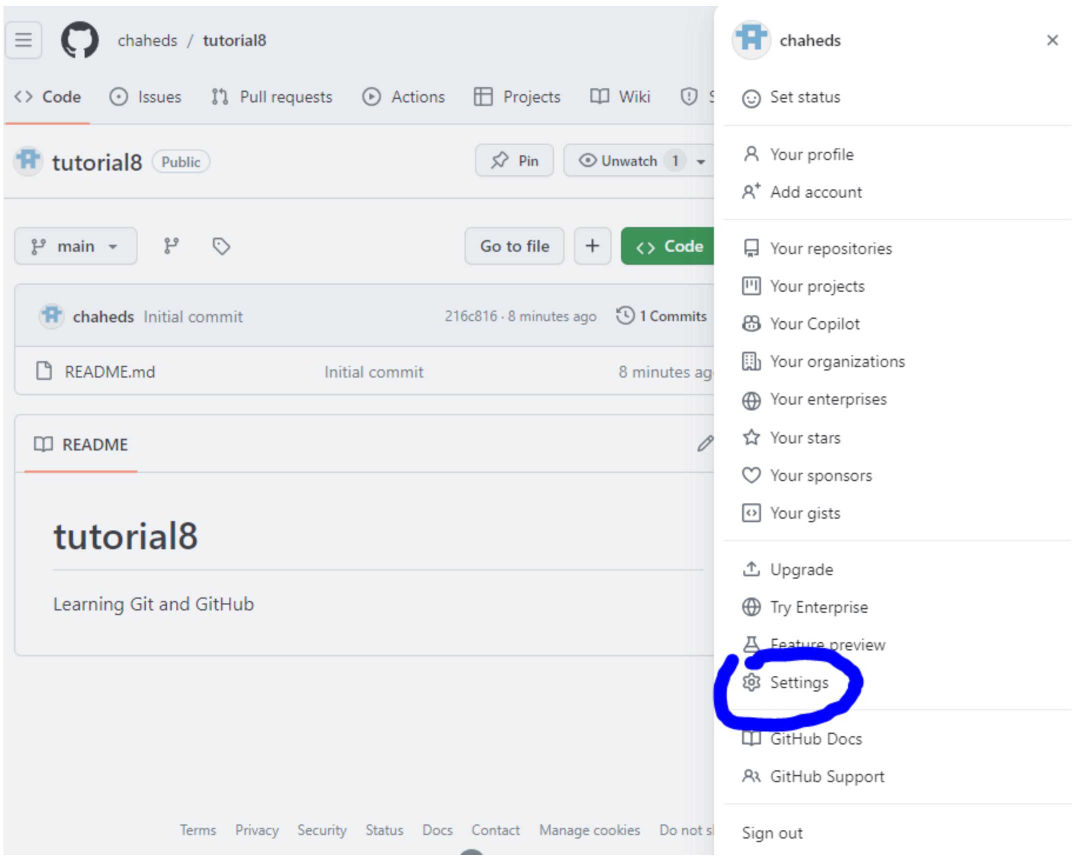
Congratulations you have created your first repo on GitHub!

You should now see the repository home page. Mine can be found at the URL: https://github.com/chaheds/tutorial8 [Yours should contain your username and your repository's name.]

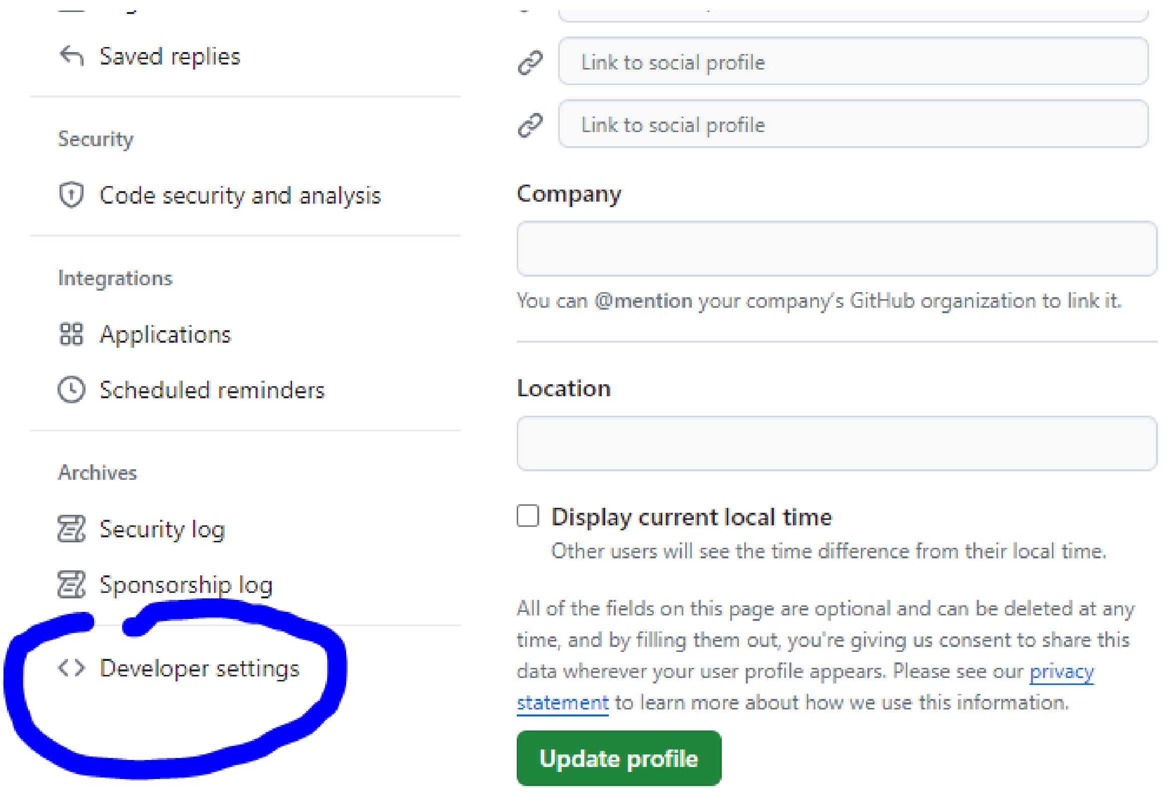
2.2. Generating a Personal Access Token (PAT)

We will now generate a Personal Access Token (PAT) which is an alternative to using passwords for authentication in the terminal. This is required since August 2021 instead of using passwords for interacting with a remote repo via the terminal. The generated PAT is valid for a limited period of time.

Go to "**Settings**".



Scroll down the Settings page. At the bottom of the menu on the left-hand side, click on "**Developer settings**".

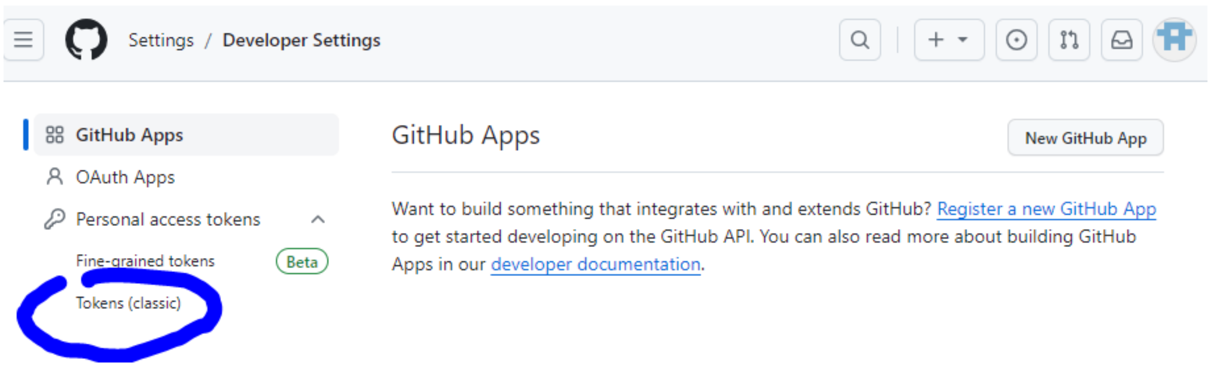


You can @mention your company's GitHub organization to link it.

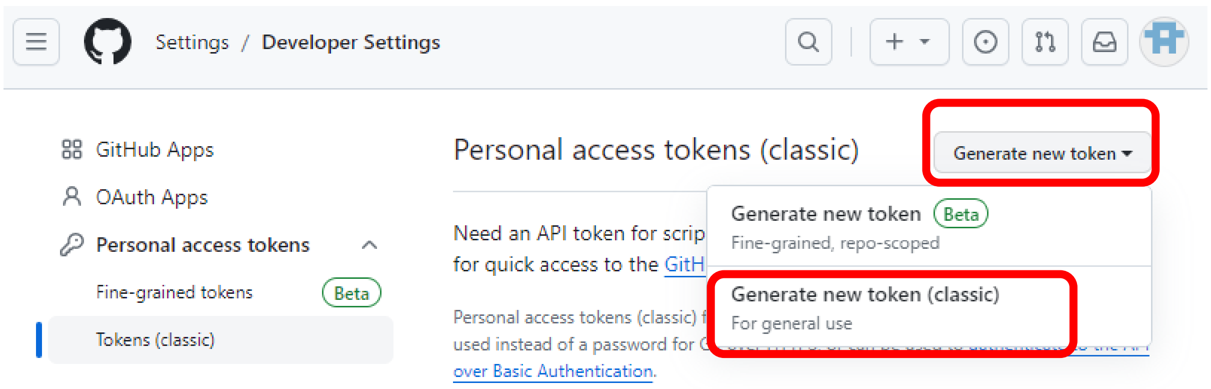
Other users will see the time difference from their local time.

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.

There is an option to generate new token. Under “**Personal access tokens**”, click “**Tokens (classic)**”, and “**Generate new token**” -> “**Generate new token (classic)**”.



Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API. You can also read more about building GitHub Apps in our developer documentation.

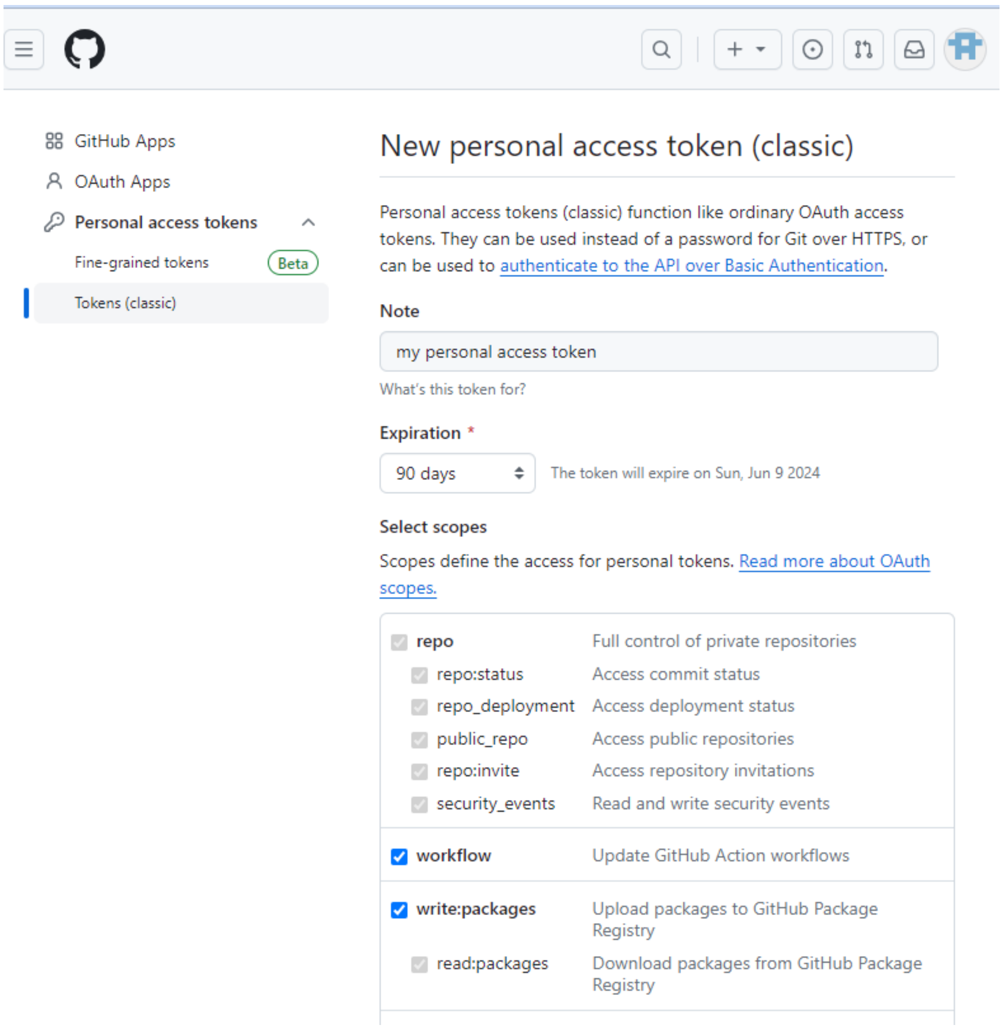


Need an API token for scrip
for quick access to the GitH

Personal access tokens (classic) f
used instead of a password for G
over Basic Authentication.

When you create a token, you need to give it a name and specify an expiration. Make sure your selected expiration duration covers your project.

Under “**Select scopes**”, tick all the boxes- to get all the permissions (e.g. writing, editing). Finally click on “**Generate token**” at the bottom of the page.



New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

my personal access token

What's this token for?

Expiration *

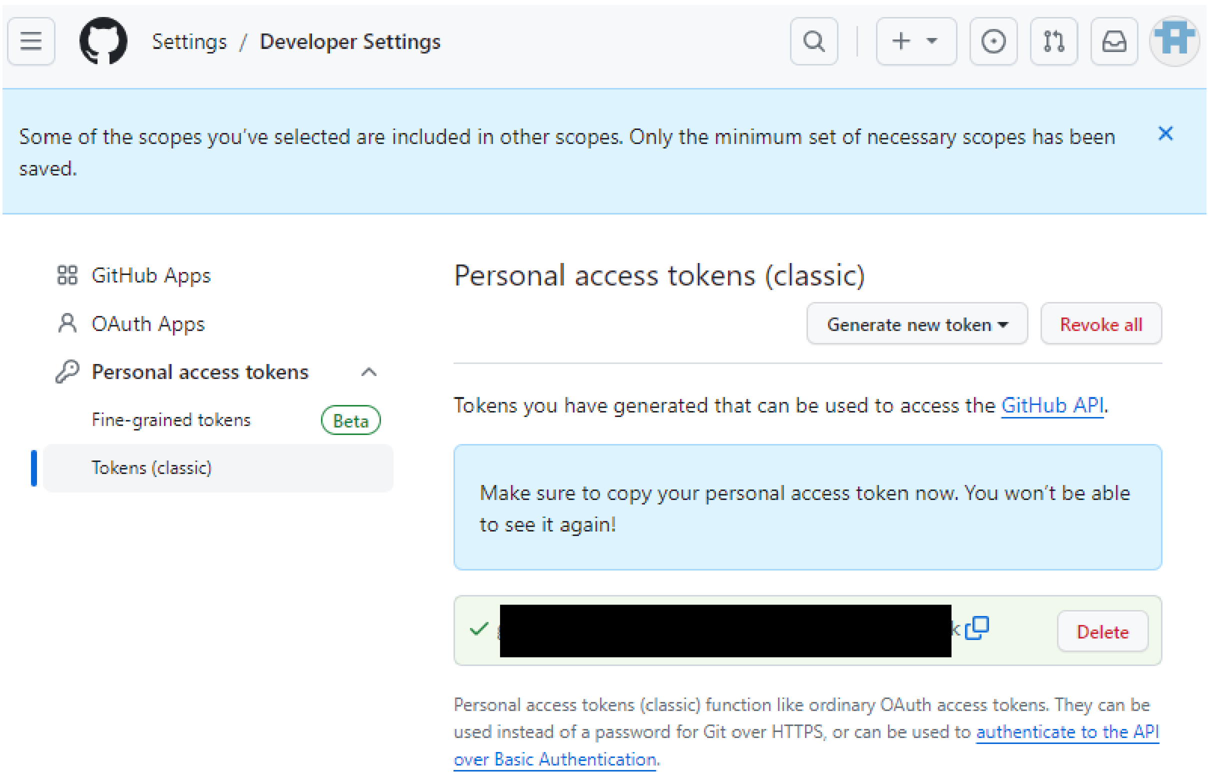
90 days — The token will expire on Sun, Jun 9 2024

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

✓ repo	Full control of private repositories
✓ repo:status	Access commit status
✓ repo_deployment	Access deployment status
✓ public_repo	Access public repositories
✓ repo:invite	Access repository invitations
✓ security_events	Read and write security events
✓ workflow	Update GitHub Action workflows
✓ write:packages	Upload packages to GitHub Package Registry
✓ read:packages	Download packages from GitHub Package Registry

Scroll down and click “**Generate token**”. This will reload the page with a prompt to make sure we copy our PAT as we won’t be able to see it again. **Don’t share your PAT!**



Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic)

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

[redacted]

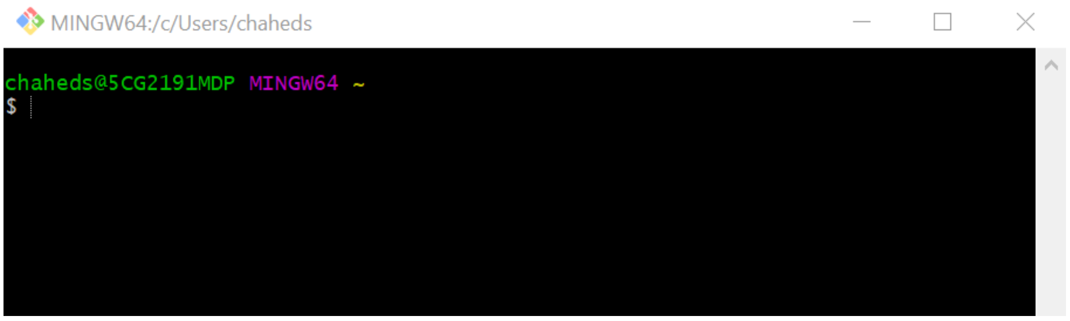
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

3. Using Git

3.1. Getting set up

Let's go back to the local machine, and we will be using Git.

On your computer go to the start menu and search for the "**Git Bash**" app. Click it to open.

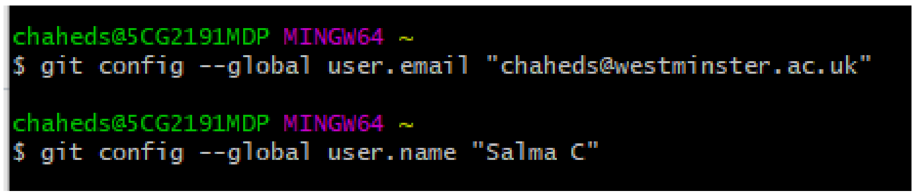


```
chaheds@5CG2191MDP MINGW64 ~
$
```

The command prompt window should be kept open while using GitHub.

Git keeps track of who performs version control actions, so you must configure Git with your own name and email. You are configuring the Git software that is installed on your machine.

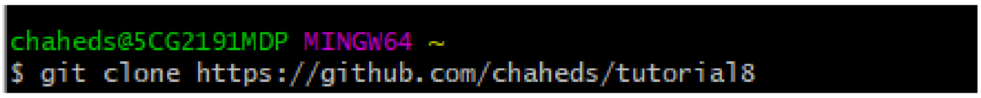
We set up the email address and the name that we would like to attach to any commits that we make (when you change and save a file). Use the **git config** command as shown in the screenshot below [but you should obviously type in your name and your email address!].



```
chaheds@5CG2191MDP MINGW64 ~
$ git config --global user.email "chaheds@westminster.ac.uk"

chaheds@5CG2191MDP MINGW64 ~
$ git config --global user.name "Salma C"
```

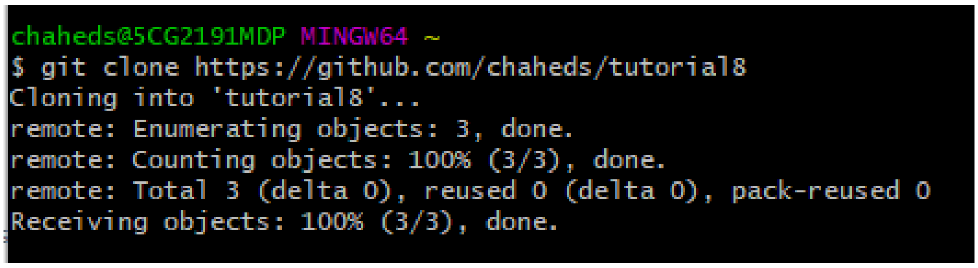
3.2. Cloning a repository to your local machine

Use **Git clone** command to download a copy of the repository you created earlier to your local machine. Specify your repository URL link [not mine!].



```
chaheds@5CG2191MDP MINGW64 ~
$ git clone https://github.com/chaheds/tutorial8
```

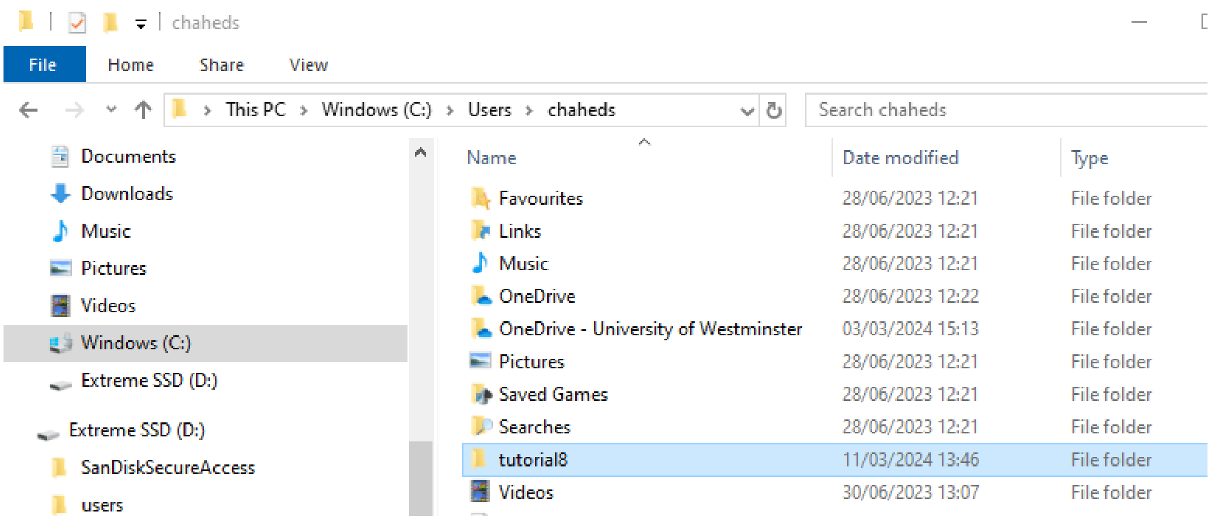
The output will be as shown as in the screenshot below, indicating that it unpacked 100% of the objects.



```
chaheds@5CG2191MDP MINGW64 ~
$ git clone https://github.com/chaheds/tutorial8
Cloning into 'tutorial8'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Once you cloned your GitHub repository to your local machine, go to the location on your computer associated with your account, you should find a folder named after your repository (e.g. tutorial8).

Any files or folders that you want to be tracked by git should be placed into this folder.

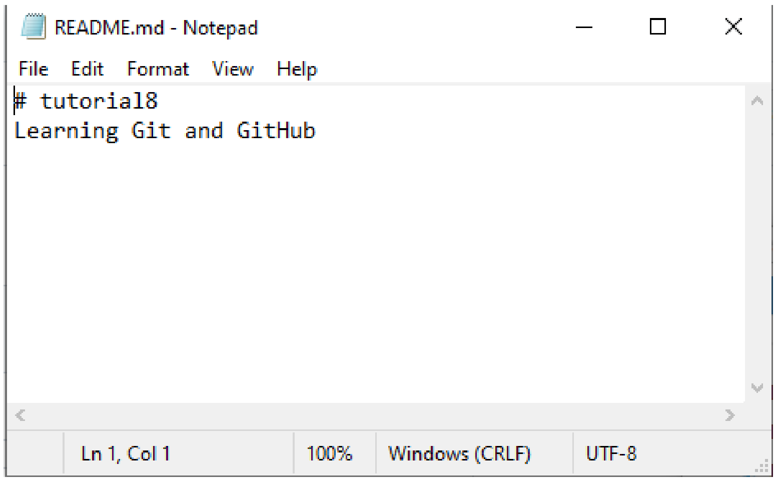


Name	Date modified	Type
Favourites	28/06/2023 12:21	File folder
Links	28/06/2023 12:21	File folder
Music	28/06/2023 12:21	File folder
OneDrive	28/06/2023 12:22	File folder
OneDrive - University of Westminster	03/03/2024 15:13	File folder
Pictures	28/06/2023 12:21	File folder
Saved Games	28/06/2023 12:21	File folder
Searches	28/06/2023 12:21	File folder
tutorial8	11/03/2024 13:46	File folder
Videos	30/06/2023 13:07	File folder

Inside that folder, there is a **README.md** file. This file was created to initialise the newly created GitHub repository.

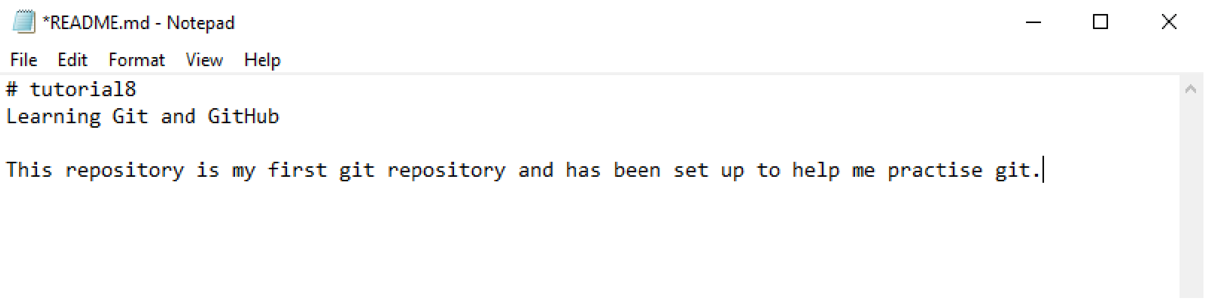
3.3. Comparing files

Initially the README.md file contains the name of the repository and the description.



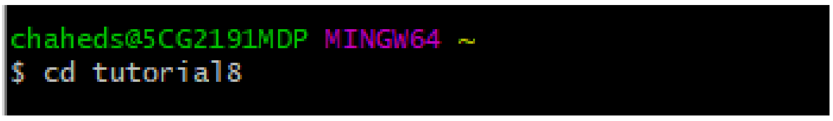
```
# tutorial8
Learning Git and GitHub
```

Let's add a new line and save the file.



```
# tutorial8
Learning Git and GitHub

This repository is my first git repository and has been set up to help me practise git.
```

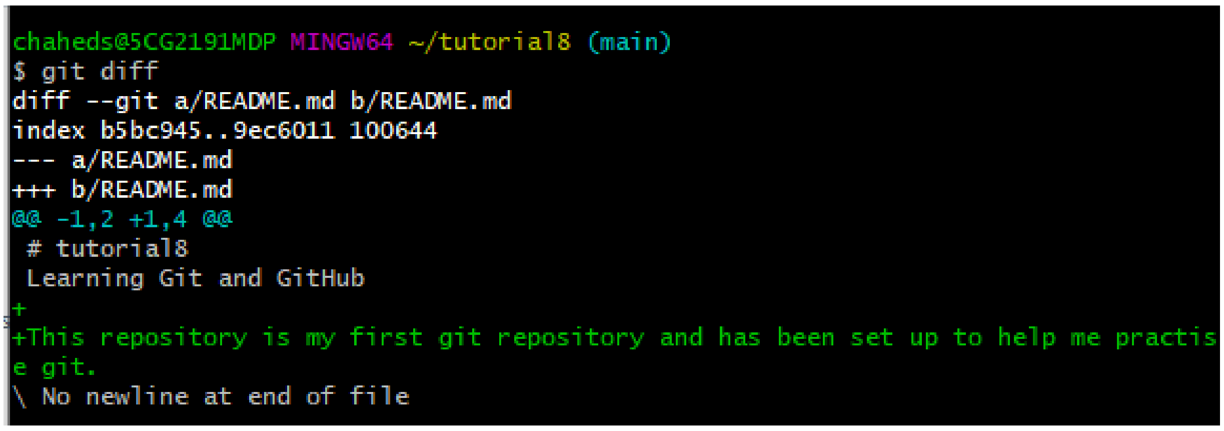
Back to command line, use **cd** command to change into the tutorial8 folder (i.e. my GitHub repo).



```
chaheds@5CG2191MDP MINGW64 ~
$ cd tutorial8
```

Run **git diff** command to see a list of changes made to the file in my repo, i.e. compare the current file to the file in the history.

In the output, the new line added to the README file is indicated by a "**+**" and is displayed in green.



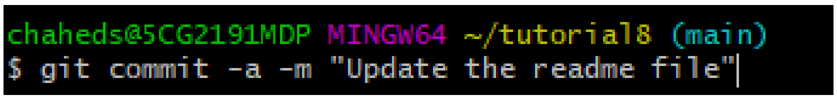
```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git diff
diff --git a/README.md b/README.md
index b5bc945..9ec6011 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,4 @@
 # tutorial8
 Learning Git and GitHub
+
+This repository is my first git repository and has been set up to help me practis
e git.
\ No newline at end of file
```

3.4. Recording changes to the repository

To make this change permanent and part of the history of this file, it needs to be committed. We will commit this change to the repository by using the **git commit** command.

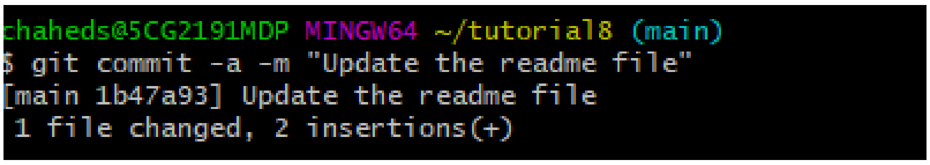
-a implies commit all changes I have done so far

-m is the commit message (this is to remind us why we made that change)



```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git commit -a -m "Update the readme file"
```

The output:



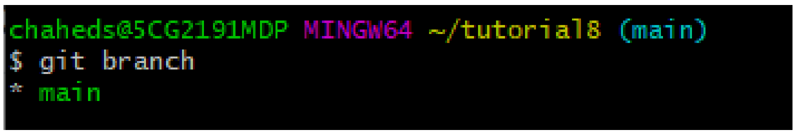
```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git commit -a -m "Update the readme file"
[main 1b47a93] Update the readme file
 1 file changed, 2 insertions(+)
```

As expected, one file changed and 2 insertions (i.e. one blank line and one line that was displayed in green in the previous screenshot).

This confirms that the local copy of the repository has been updated and the change has been committed.

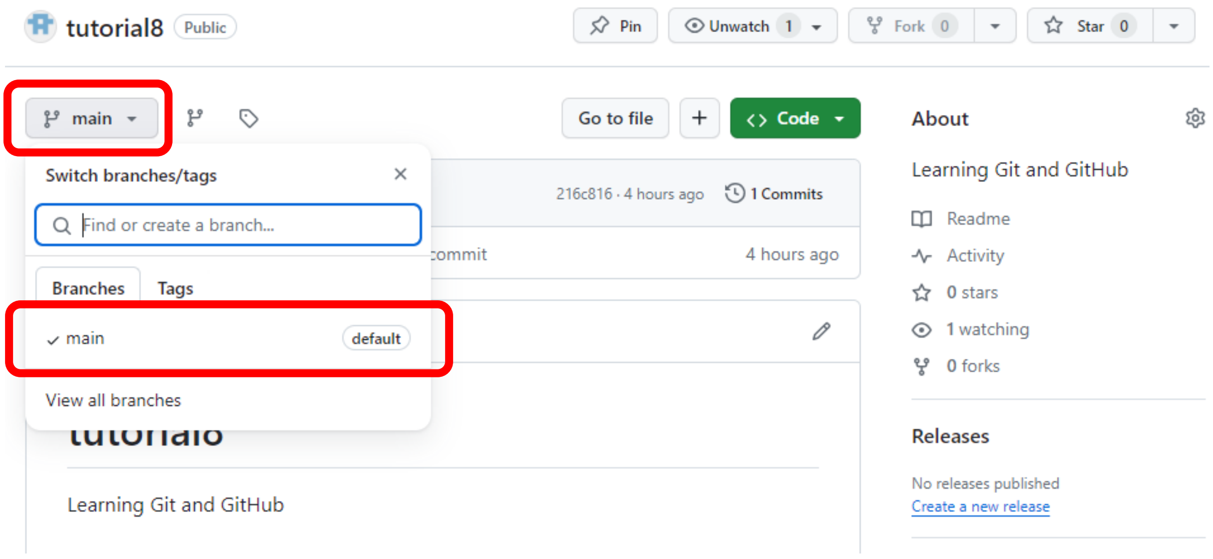
4. Branches

The **git branch** command shows what branches of our repository are available. The output illustrated in the screenshot below indicates that there is only one branch and that is the one we are using. The current branch (i.e. being used) is indicated with an asterisk.



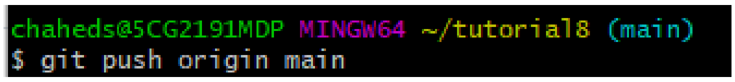
```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git branch
* main
```

We can also see this information on GitHub. Let's go back to GitHub. Looking at the repository view, click on "main" at the top left.



5. Pushing your changes

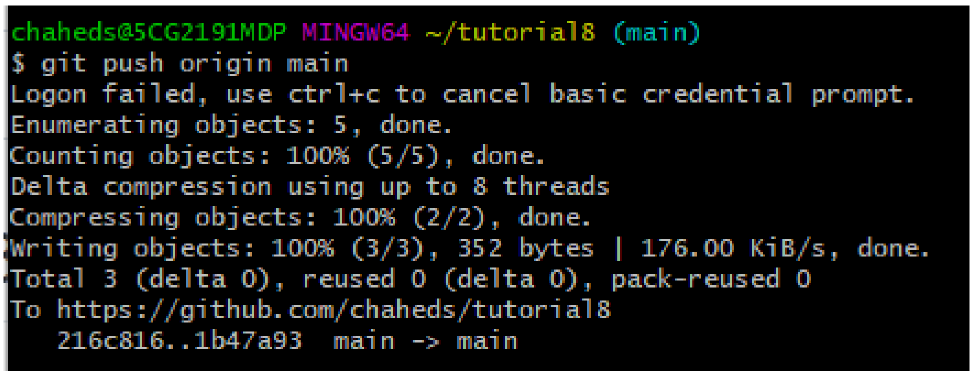
We had made changes locally to the README.md file, and now we want to save those changes to the remote repository on GitHub. For this purpose, we use **git push** command which send the local copy of the repository back to the remote repository on GitHub. It synchronises the local main branch with the remote main branch.



```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git push origin main
```

Origin refers to the remote GitHub repository (remote main branch). **Main** is the name of the branch we want to send (local main branch).

You will be prompted to enter your email as the username and your personal authentical token (PAT) as the password.



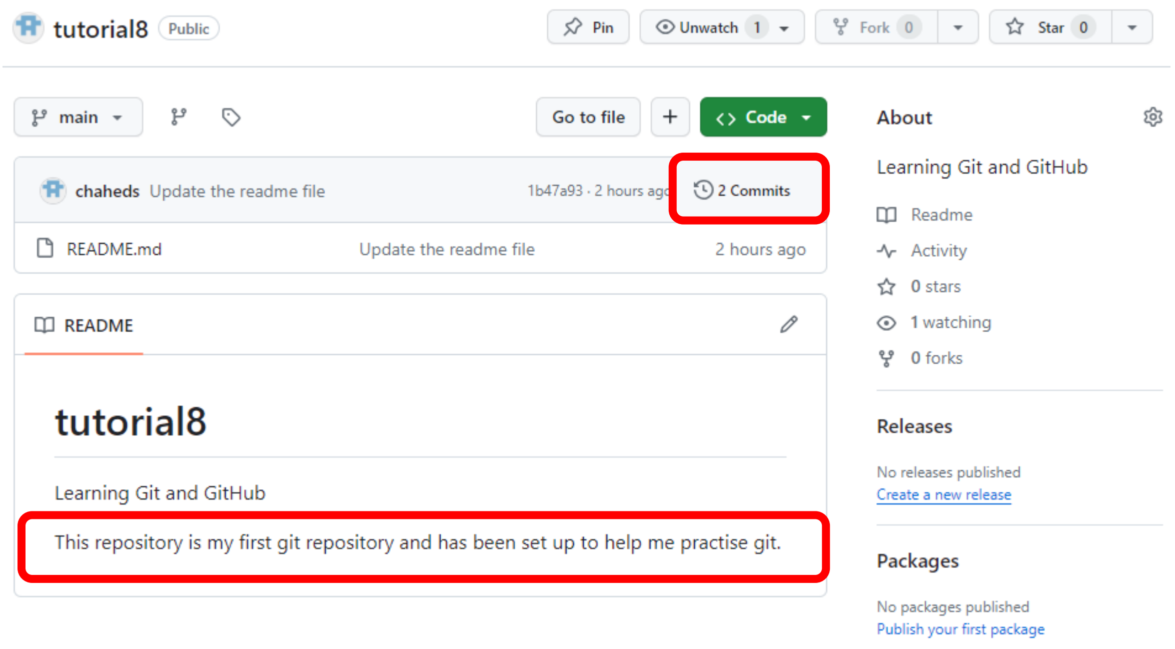
```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git push origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 352 bytes | 176.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/chaheds/tutorial8
   216c816..1b47a93  main -> main
```

Let's check our GitHub to check whether our changes are reflected in our remote repository.

There are two commits:

i. The first commit is related to the creation of the repository.
ii. The second commit is related to the changes added to the README.md file.

The README file has been updated as it is illustrated under README box in the repository view.



chaheds Update the readme file 1b47a93 · 2 hours ago 2 Commits

README.md	Update the readme file	2 hours ago

README

tutorial8

Learning Git and GitHub

This repository is my first git repository and has been set up to help me practise git.

About

Learning Git and GitHub

Readme
Activity
0 stars
1 watching
0 forks

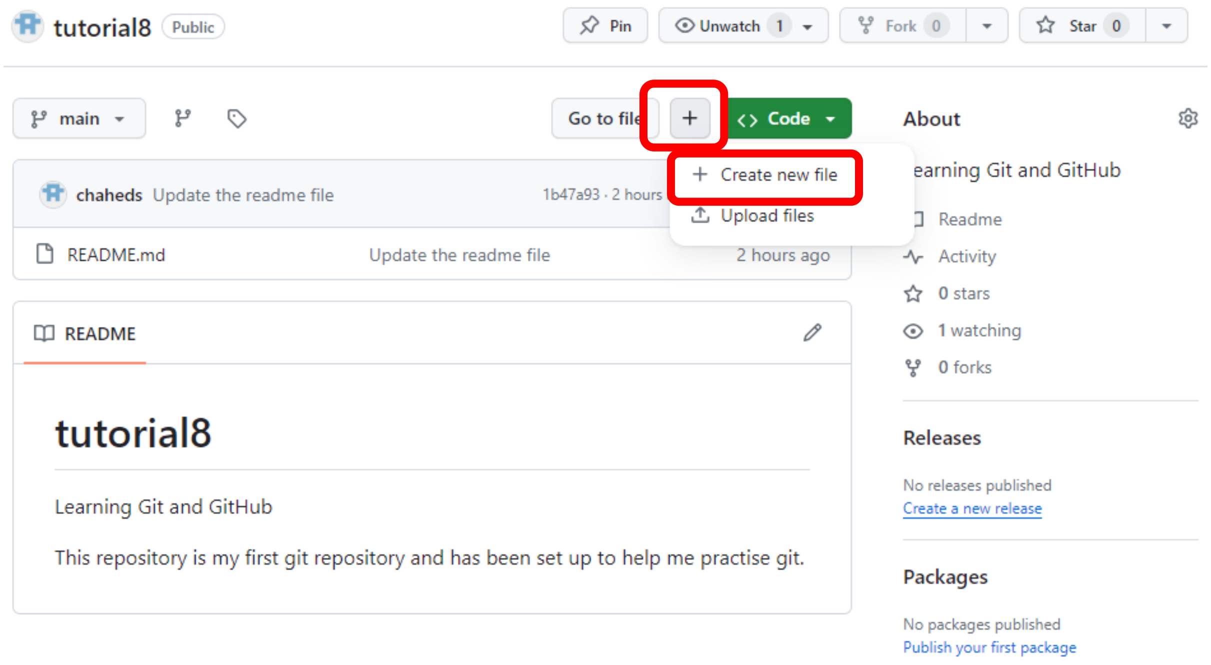
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

We can add a file directly from GitHub by clicking on “**+**” at top right of the repository view.



chaheds Update the readme file 1b47a93 · 2 hours

README.md Update the readme file 2 hours ago

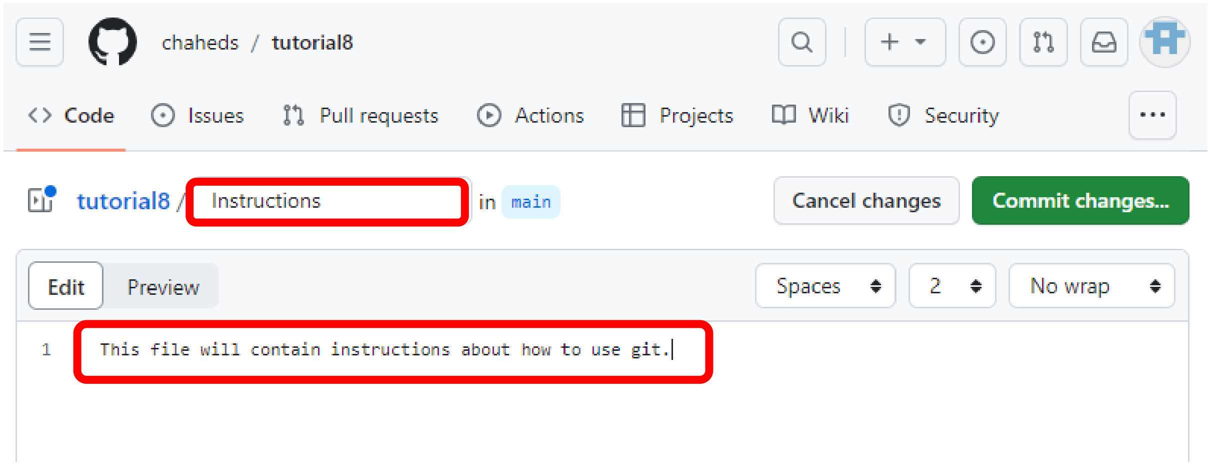
README

tutorial8

Learning Git and GitHub

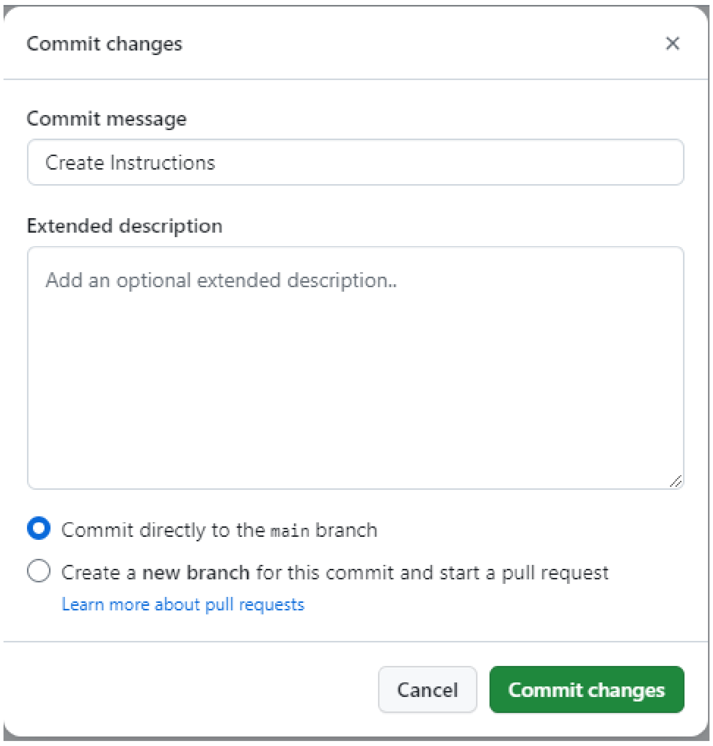
This repository is my first git repository and has been set up to help me practise git.

Add a file name and a message on the file content. Then click on “**Commit changes…**” (green button) on the right side of the window. See screenshot below.

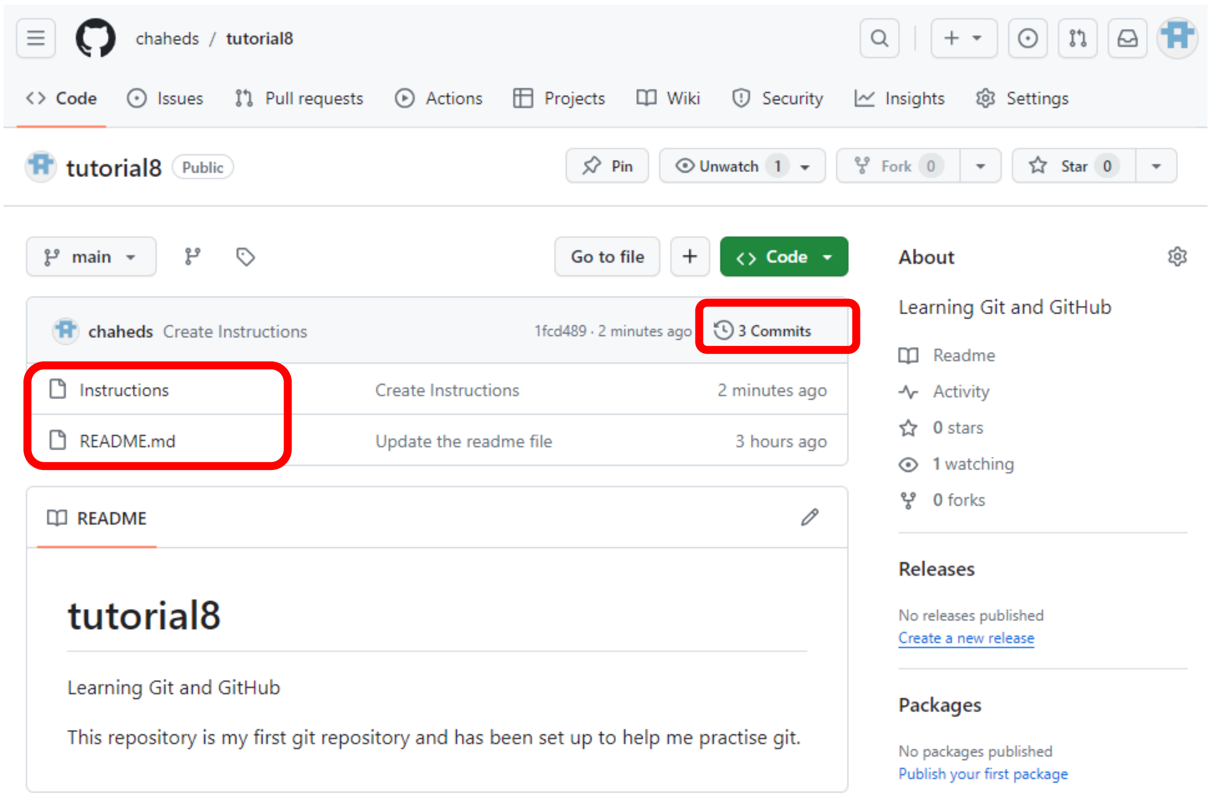


```
This file will contain instructions about how to use git.
```

A window will pop up. Check the commit message. “**Commit directly to the main branch**” is selected by default. Click “**Commit changes**”.


Commit changes
×
Commit message
Create Instructions
Extended description
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
Cancel
Commit changes

We have now three commits and two files in our GitHub repository.



Instructions	Create Instructions	2 minutes ago
README.md	Update the readme file	3 hours ago

tutorial8

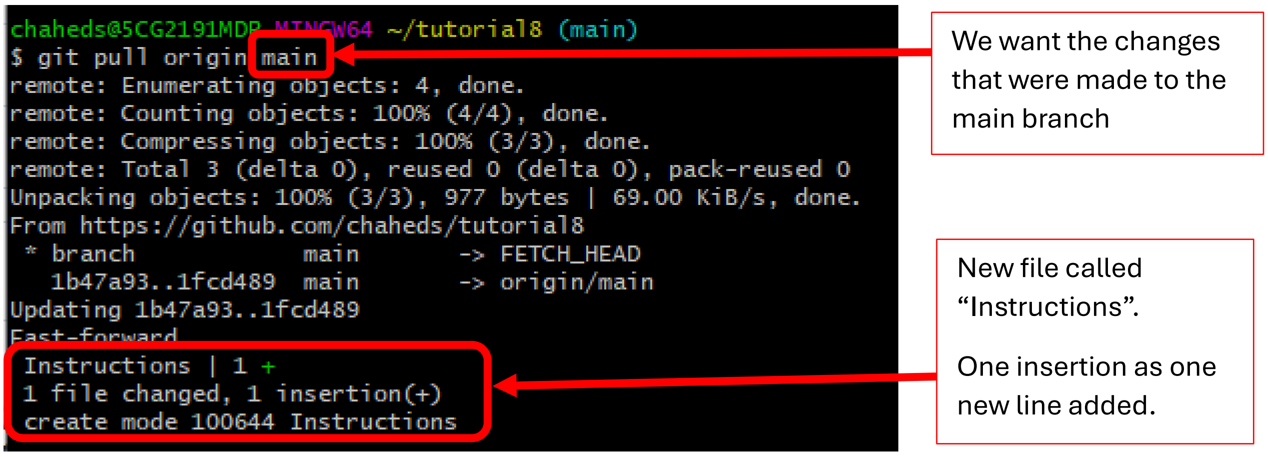
Learning Git and GitHub

This repository is my first git repository and has been set up to help me practise git.

Every time we make a change, the repository is logged recording who did it, when they did it and what they did.

6. Keeping in sync

Our local repository has become out of sync with the remote GitHub repository. We had added a new file to the remote repository on GitHub but that installed file is not automatically added to our local repository. We have to update our local repository using **git pull** command. We pull in all the changes that have been made to the remote repository to our local copy.



```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 977 bytes | 69.00 KiB/s, done.
From https://github.com/chaheds/tutorial8
 * branch            main       -> FETCH_HEAD
   1b47a93..1fcd489  main       -> origin/main
Updating 1b47a93..1fcd489
Fast-forward
 Instructions | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Instructions
```

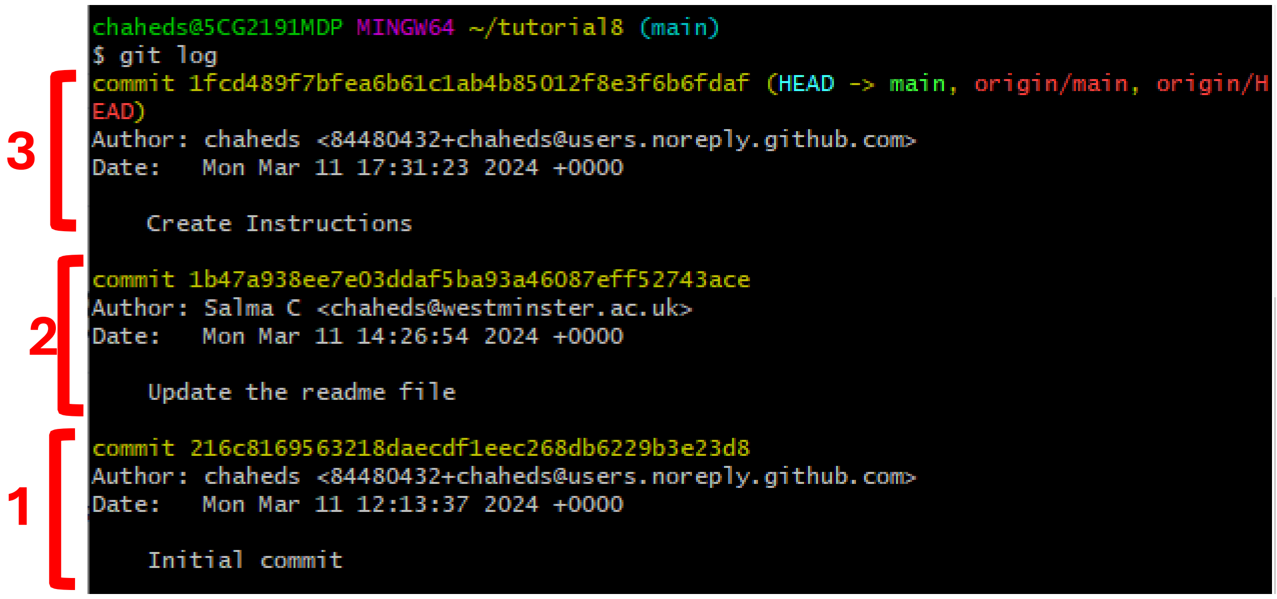
We want the changes that were made to the main branch

New file called "Instructions".

One insertion as one new line added.

7. Viewing history of repository

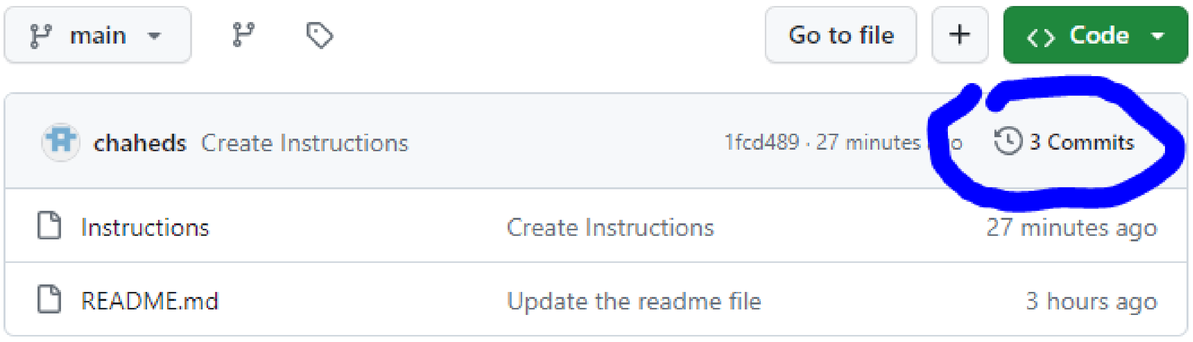
All changes in our repository are tracked. We can view this using **git log** command. This command will enable us to see all the changes that have taken place to our repository (i.e. view history of repository).



```
chaheds@5CG2191MDP MINGW64 ~/tutorial8 (main)
$ git log
commit 1fcd489f7bfea6b61c1ab4b85012f8e3f6b6fdaf (HEAD -> main, origin/main, origin/H
EAD)
Author: chaheds <84480432+chaheds@users.noreply.github.com>
Date:   Mon Mar 11 17:31:23 2024 +0000

    Create Instructions

commit 1b47a938ee7e03ddaf5ba93a46087eff52743ace
Author: Salma C <chaheds@westminster.ac.uk>
Date:   Mon Mar 11 14:26:54 2024 +0000

    Update the readme file

commit 216c8169563218daecdf1eec268db6229b3e23d8
Author: chaheds <84480432+chaheds@users.noreply.github.com>
Date:   Mon Mar 11 12:13:37 2024 +0000

    Initial commit
```

Every commit has a unique identifier (in yellow). Commits are displayed in chronological order with the most recent commit appears at the top of the log. We can also view the author and the time of the change.

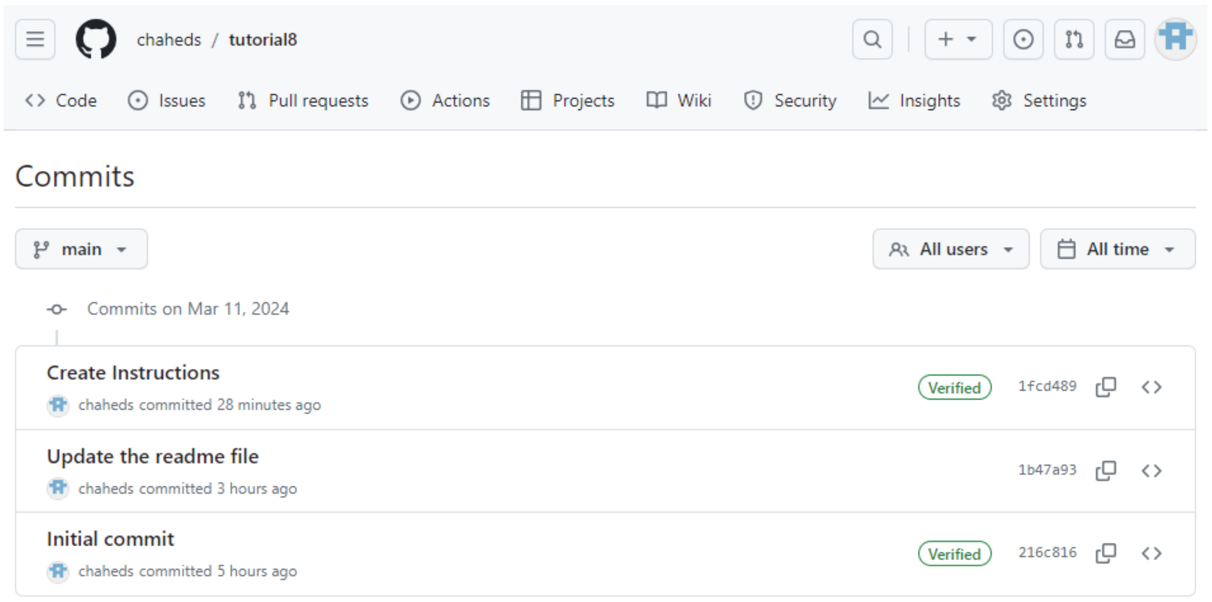
We can also view the history of repository on GitHub by clicking on the number of commits in the repository view.



chaheds Create Instructions 1fcd489 · 27 minutes ago

Instructions	Create Instructions	27 minutes ago
README.md	Update the readme file	3 hours ago

Here is the history of the repository on GitHub:



Commits

Commits on Mar 11, 2024

Create Instructions
chaheds committed 28 minutes ago
Verified 1fcd489

Update the readme file
chaheds committed 3 hours ago
1b47a93

Initial commit
chaheds committed 5 hours ago
Verified 216c816

Disclaimer: The computer seminar material was extracted and adapted from 6DATA007W Data Science Final Project Lecture (2022) delivered by Philip Worrall.