SIT225

Git and GitHub

Git is a distributed version control system that tracks versions of files. It is often used to control source code by programmers collaboratively developing software.

GitHub is a developer platform that allows developers to create, store, manage and share their code. It uses **Git** software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

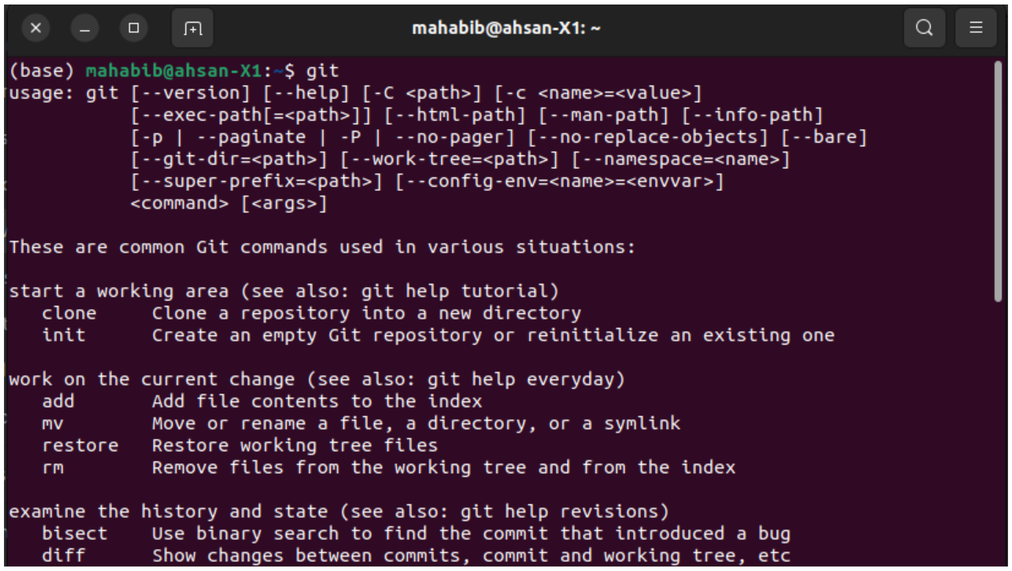
Why is GitHub for this unit?

SIT225 requires you to store your Arduino sketch and Python scripts to be stored in GitHub inside weekly folders under a root directory SIT225_2024T2. The content of week 1 should go in SIT225_2024T2/week_1, week 2 content should go in SIT225_2024T2/week_2 and so on.

As a future IT professional, you should have the technical skill to play with git so you can do collaborative software development with confidence.

Git installation manual

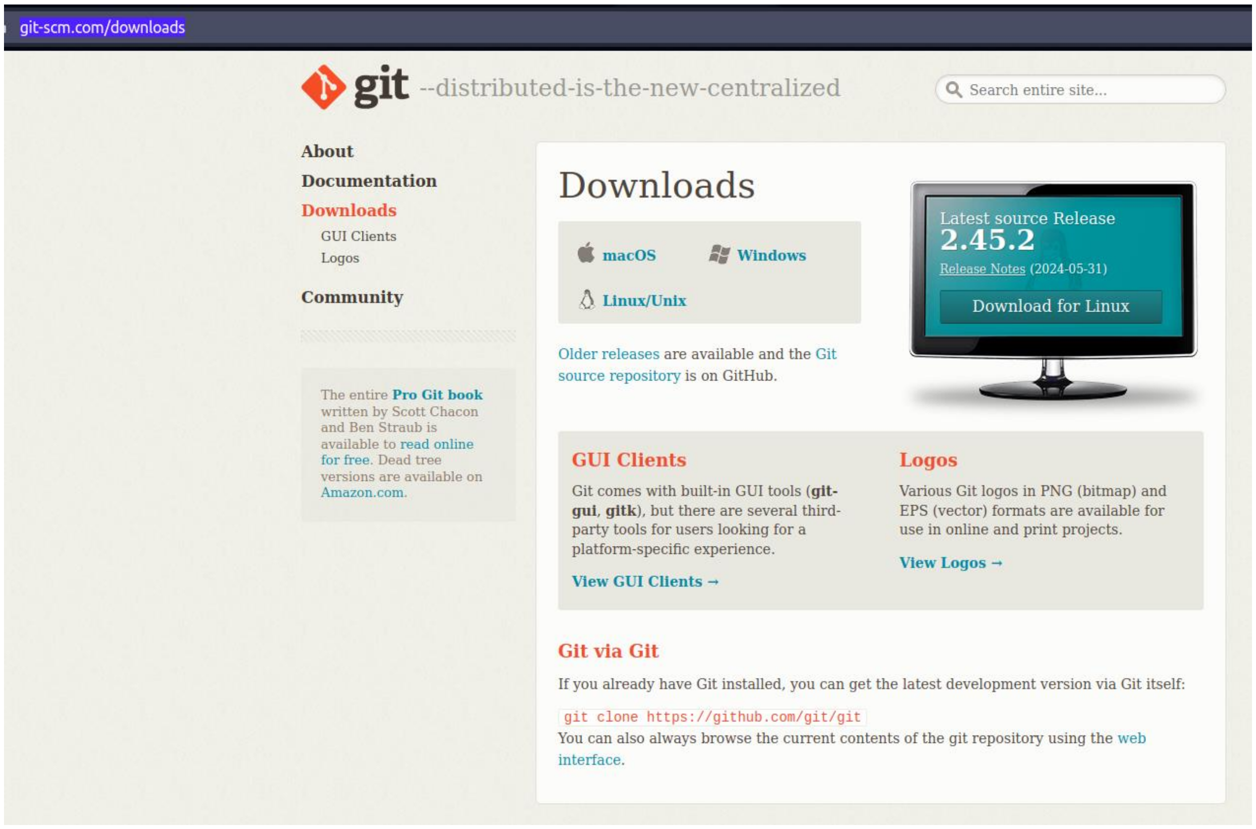
Linux and MacOS are likely to ship with git pre-installed. You can open a terminal and try command git -



```
(base) mahabib@ahsan-X1:~$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
```

If the output says error such as unknown command, then you should install git using the following ways.

Go to website https://git-scm.com/downloads, which looks like below



git-scm.com/downloads

git --distributed-is-the-new-centralized

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.45.2
Release Notes (2024-05-31)
Download for Linux

GUI Clients

Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

`git clone https://github.com/git/git`

You can also always browse the current contents of the git repository using the web interface.

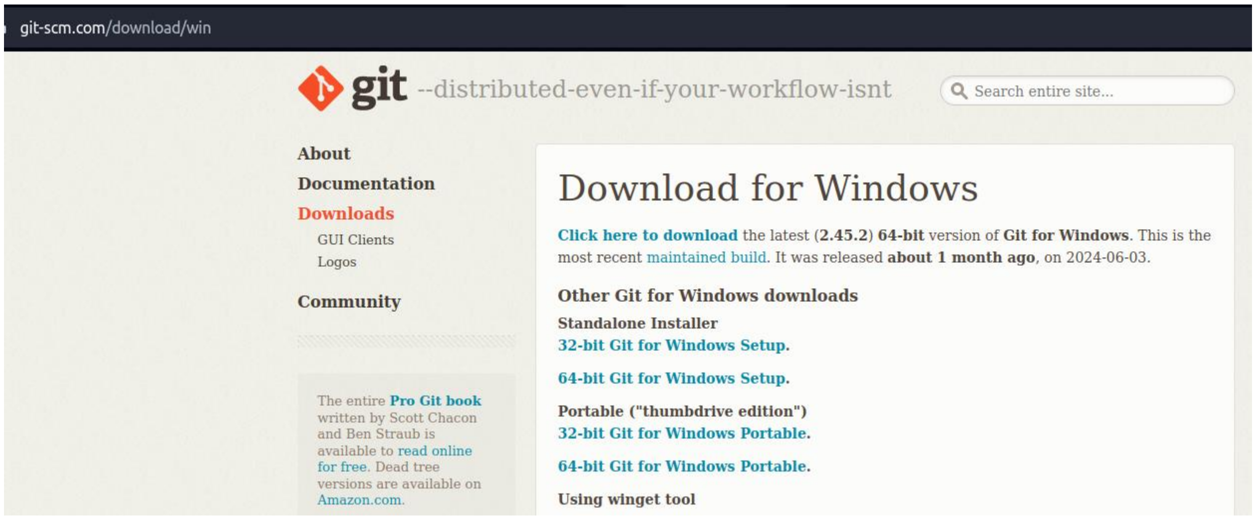
Go to macOS and issue the bellow commands -

```
$ brew install git
```

If you are a Linux user (Ubuntu or other distributions), use the below command in a terminal -

```
$ sudo apt-get install git
```

For windows users, click on the download link at the top of the page as shown below to download the latest version (2.45.2) 64-bit version of Git for Windows.



The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Download for Windows

Click here to download the latest (**2.45.2**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **about 1 month ago**, on 2024-06-03.

Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.

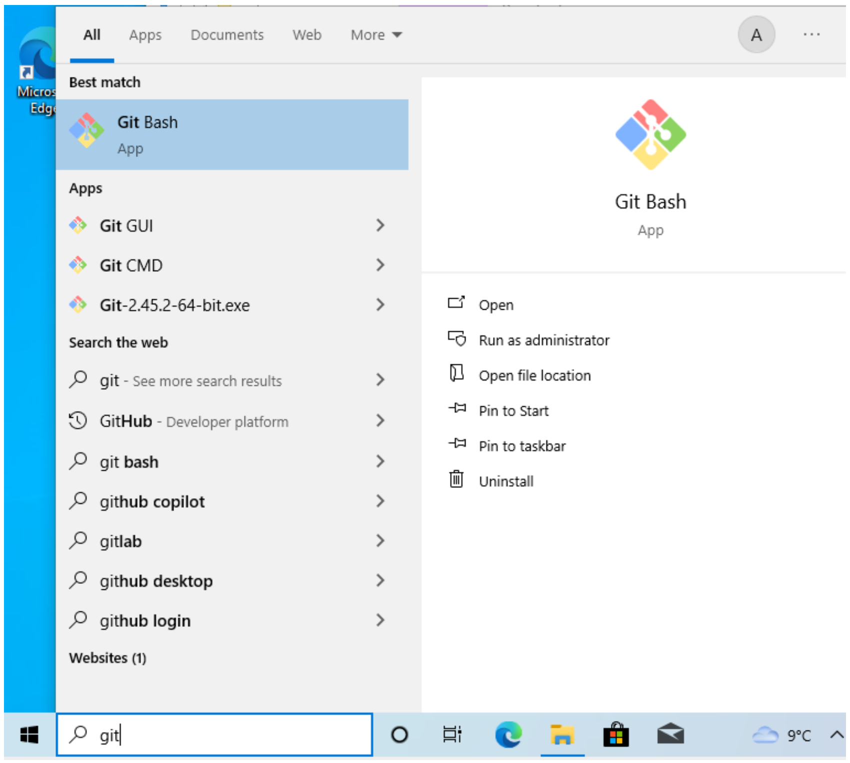
64-bit Git for Windows Setup.

Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

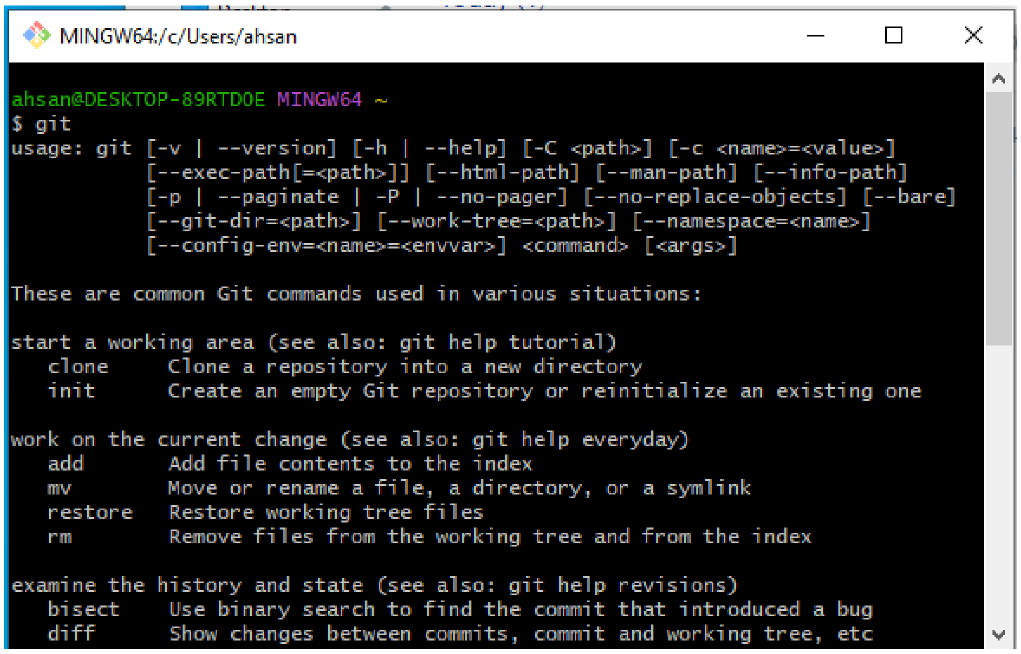
64-bit Git for Windows Portable.

Using winget tool

Follow all the default options to complete the installation. After installing is done, you will be given a Git Bash.



The Git Bask looks like a regular terminal (or command line) as found in macOS or Ubuntu Linux. You can type a command git and see similar output in shown in Linux terminal above.



```
ahsan@DESKTOP-89RTD0E MINGW64 ~
$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
```

GitHub account and developer token

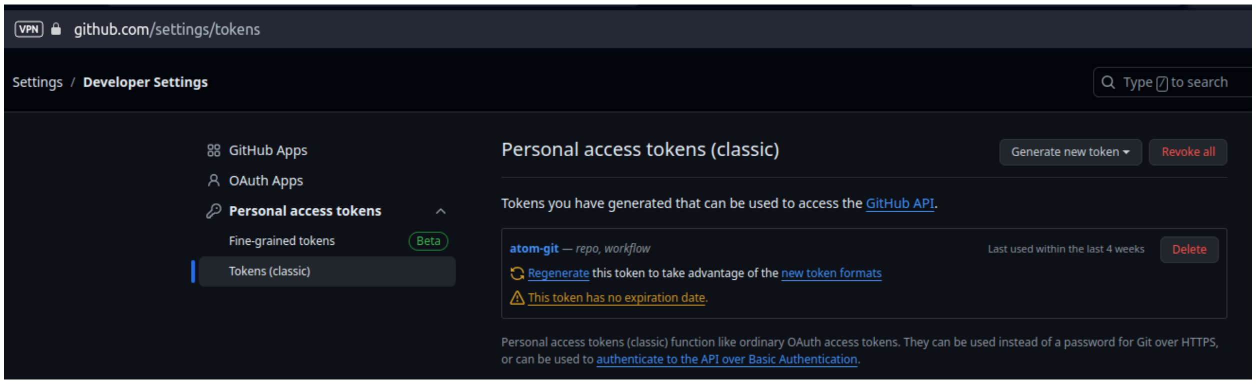
Create an account in GitHub if you are new to github.com.

Remember your GitHub password which you need to login github.com. However, this password can NOT be used from applications such as the git bash tool you have just installed. This is a new norm adopted by GitHub, similar to Google or other as part of a security standard. You need to create a personal token.

GitHub developer personal token:

Login to GitHub.

Click on the user icon in the top right corner, a menu pops up, select Settings – which gives you below page -



Personal access tokens (classic)

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

atom-git — repo, workflow   Last used within the last 4 weeks

Regenerate this token to take advantage of the new token formats

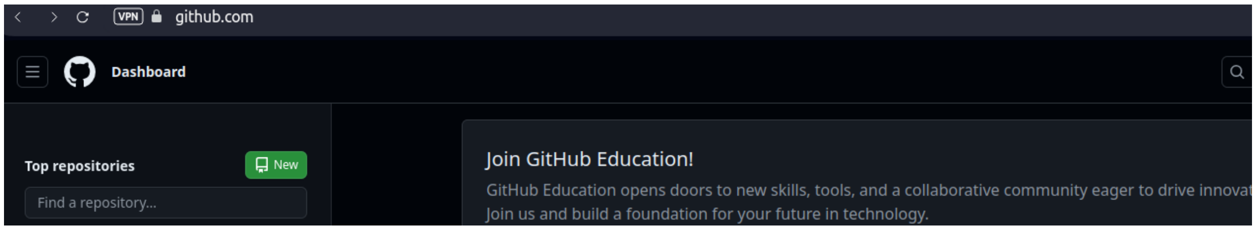
This token has no expiration date.

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.

Select Develop Settings and then Tokens (classic). At top right, Generate new tokens > Generate new token (classic). This will create a passcode for you to use to login to your account instead of using password.

Create a new repository in GitHub:

Login to github.com and create a new repository by clicking the New button.

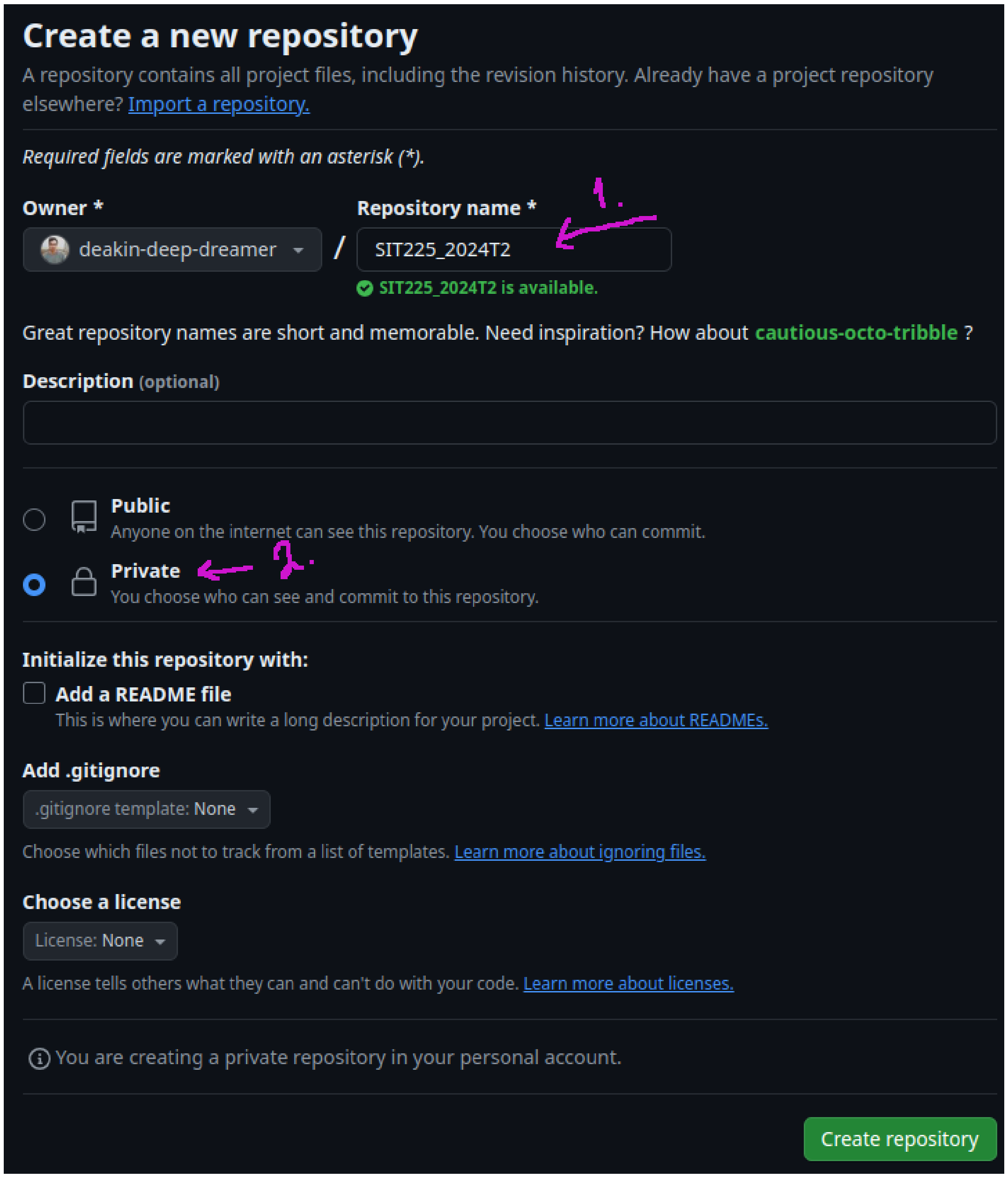


Join GitHub Education!

GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovat

Join us and build a foundation for your future in technology.

Create a new repository SIT225_2024T2. Make it private. This is important you create a private repository to keep your codes sharable only to your tutor.


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 * deakin-deep-dreamer / **Repository name *** SIT225_2024T2 ← 1.

SIT225_2024T2 is available.

Great repository names are short and memorable. Need inspiration? How about **cautious-octo-tribble** ?

Description (optional)

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

● **Private** ← 2. 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.

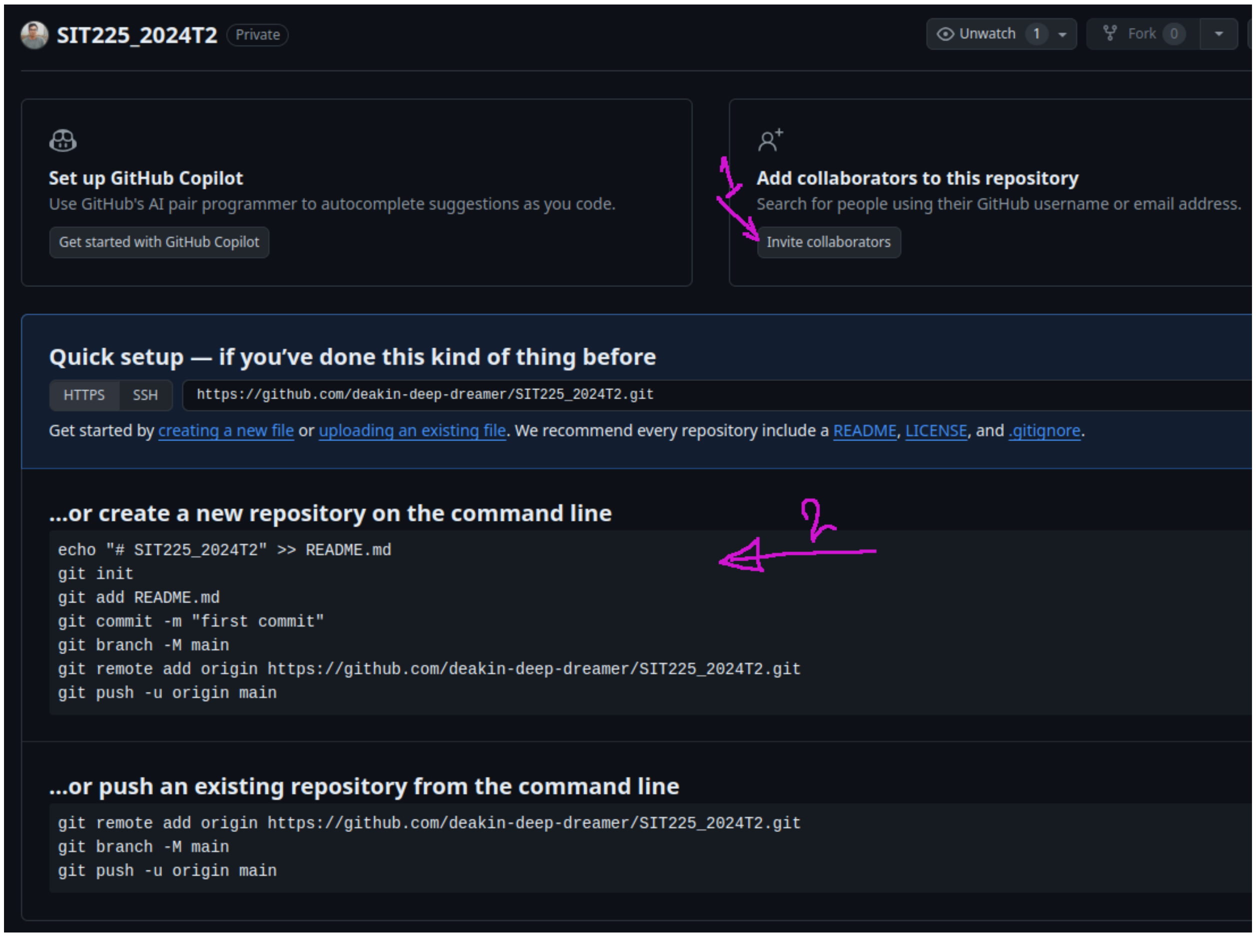
Choose a license

License: None

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

You are creating a private repository in your personal account.

Click on Create repository, this will create a new repository with instructions what to do next.


SIT225_2024T2
Private
Unwatch 1
Fork 0
Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators
1
2
Quick setup — if you've done this kind of thing before
HTTPS
SSH

```
https://github.com/deakin-deep-dreamer/SIT225_2024T2.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# SIT225_2024T2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/deakin-deep-dreamer/SIT225_2024T2.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/deakin-deep-dreamer/SIT225_2024T2.git
git branch -M main
git push -u origin main
```

Step 1 shows you can add collaborator, your tutor, by clicking on Invite collaborators button.

Step 2 shows list of git commands you need to issue from your Git Bash (windows) or terminal (macOS or Linux).

Open a terminal and go to a folder (say, source_code) where you want to create a new repository with provided instructions. Create a new folder SIT225_2024T2 and go to that folder. Issue the instructions in order –

Initialise git in current folder -

$ git init

Create a temporary file (say, README.md, you can do it from your file explorer)

$ touch README.md

Now add all updated items (files or folder within SIT225_2024T2) to version control. This prepares git to list down items to save.

$ git add *

Commit or save the content to git repository. Note that it requires you to provide a comment what is the purpose of this save.

$ git commit –m “first commit”

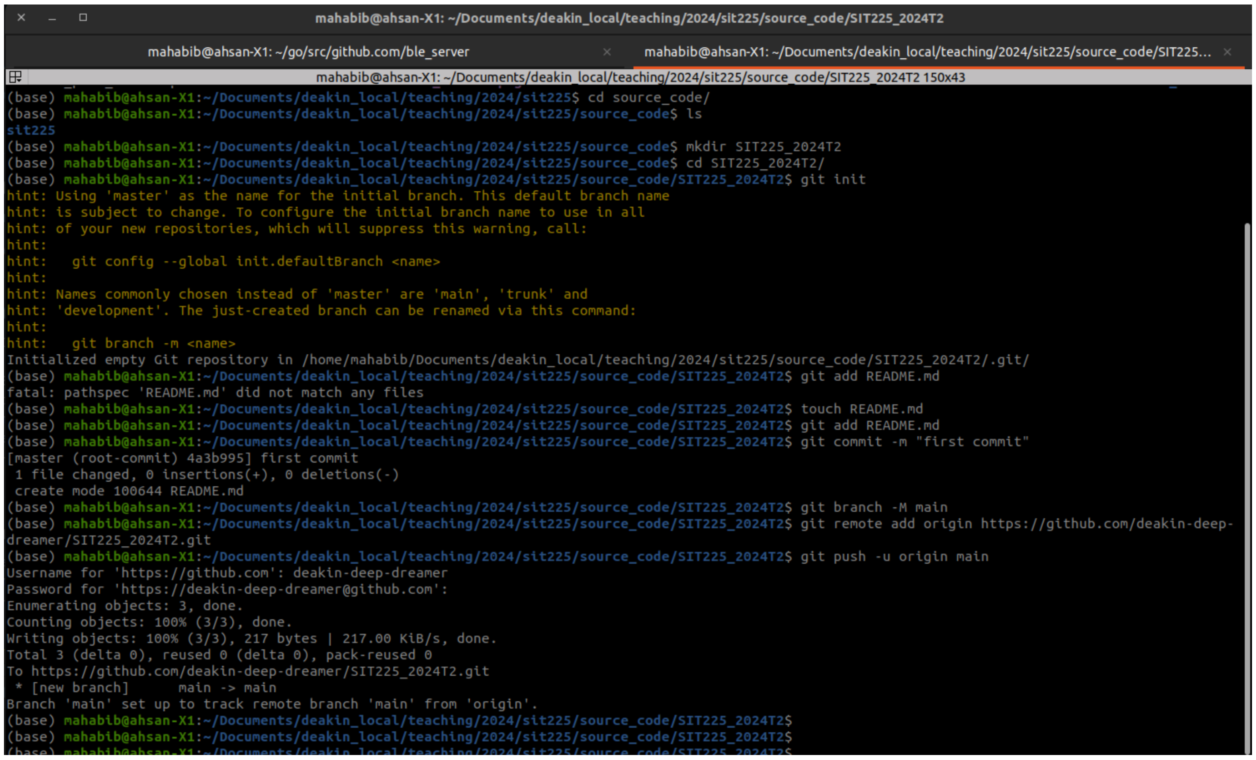
Below 2 commands branch and remote are for the first time only, which does not requires next time when you commit something (say, you have created week_1 folder and your Arduino sketch and Python scripts in it). The branch command informs git to save in the main trunk of the repository. The “remote add origin” links your local git to the GitHub repository which your local git installation will remember for the next time.

$ git branch –M main

$ git remote add origin https://github.com/deakin-deep-dreamer/SIT225_2024T2.git

Now you need to upload your git saved content to GitHub online repository. You will use the git push command following the git commit next time, without needing git branch and git remote command anymore. Git push command will ask your username and password. Use your GitHub username and developer token (classic) for login.

$ git push –u origin main


mahabib@ahsan-X1: ~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2
mahabib@ahsan-X1: ~/go/src/github.com/ble_server
mahabib@ahsan-X1: ~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225...
mahabib@ahsan-X1: ~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2 150x43

```
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225$ cd source_code/
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code$ ls
sit225
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code$ mkdir SIT225_2024T2
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code$ cd SIT225_2024T2/
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/mahabib/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2/.git/
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git add README.md
fatal: pathspec 'README.md' did not match any files
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ touch README.md
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git add README.md
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git commit -m "first commit"
[master (root-commit) 4a3b995] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git branch -M main
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git remote add origin https://github.com/deakin-deep-
dreamer/SIT225_2024T2.git
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$ git push -u origin main
Username for 'https://github.com': deakin-deep-dreamer
Password for 'https://deakin-deep-dreamer@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 217 bytes | 217.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/deakin-deep-dreamer/SIT225_2024T2.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$
(base) mahabib@ahsan-X1:~/Documents/deakin_local/teaching/2024/sit225/source_code/SIT225_2024T2$
```

The above terminal is in Ubuntu which may not be the same for your case but consider above and match with the instructions above for better understanding.

Refresh your github page and you will see the content you have created in your computer.

What is next?

Create a week_1 folder in parent folder SIT225_2024T2 using your file explorer. Open terminal and go to SIT225_2024T2 folder and issue below commands in sequence -

```
$ git add *
```

```
$ git commot –m "week 1 content"
```

```
$ git push –u origin main
```

You are good to go!

Welcome to git !!!