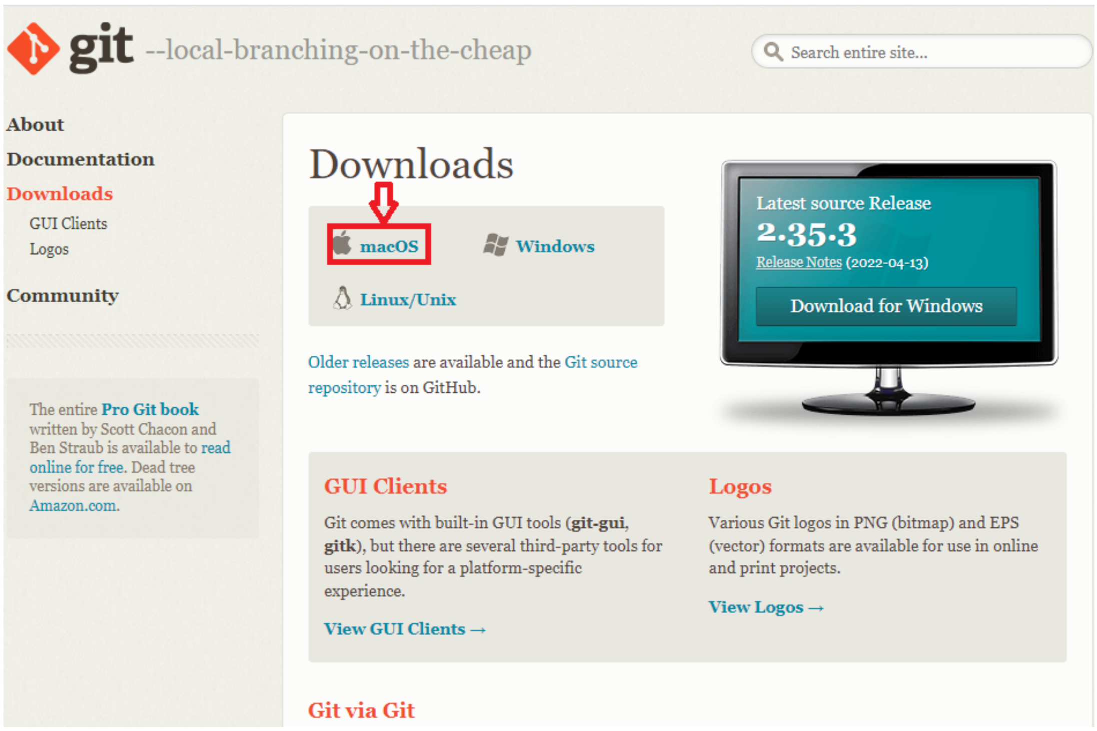
Git-MacOS

Download Git for MacOS

There are many different ways to set up Git on Mac. If you prefer using a GUI, Git offers a simple installation using the installer for Mac. On the other hand, you can install Git using the terminal with a couple of simple commands.

Install Git Using Git Installer

1. The easiest way to set up Git is to use the Git installer for Mac.
2. Navigate to the website https://git-scm.com/downloads
3. Click the download link for macOS and allow the download to complete.



git --local-branching-on-the-cheap

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
2.35.3
Release Notes (2022-04-13)
Download for Windows

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 →

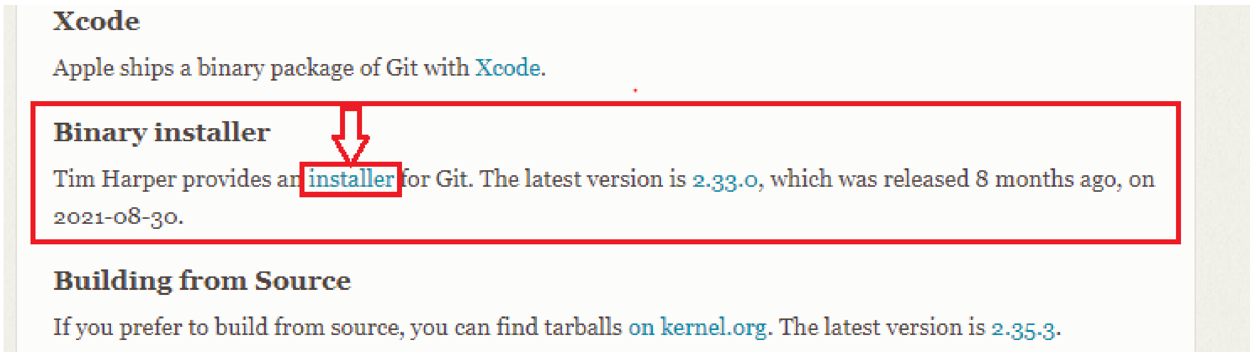
Logos

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

View Logos →

Git via Git

Next page it opens up several options for installing Git. Select Binary Installer and navigate to installer link



Xcode

Apple ships a binary package of Git with Xcode.

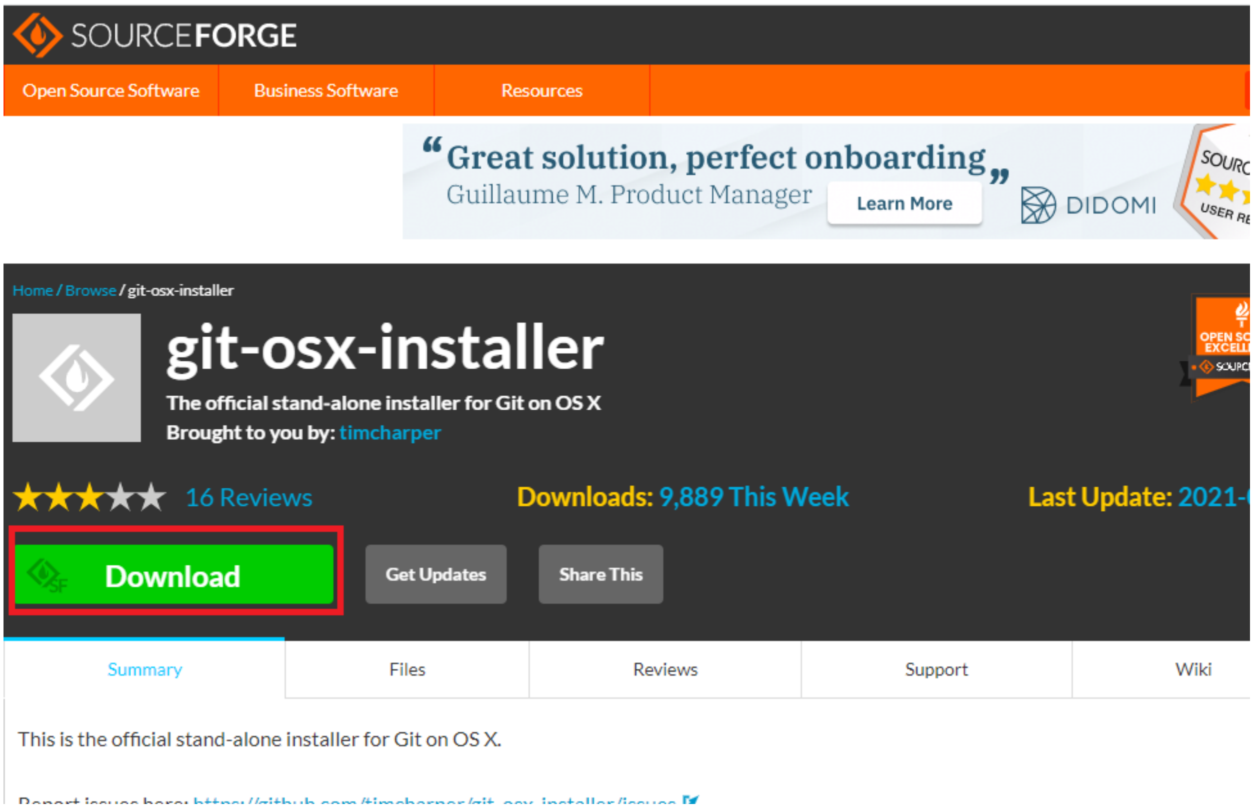
Binary installer

Tim Harper provides an installer for Git. The latest version is 2.33.0, which was released 8 months ago, on 2021-08-30.

Building from Source

If you prefer to build from source, you can find tarballs on kernel.org. The latest version is 2.35.3.

It opens SourceForge built by Tim Harper. Click on the download button.


SOURCEFORGE
Open Source Software
Business Software
Resources
"Great solution, perfect onboarding"
Guillaume M. Product Manager
Learn More
DIDOMI
Home / Browse / git-osx-installer

git-osx-installer

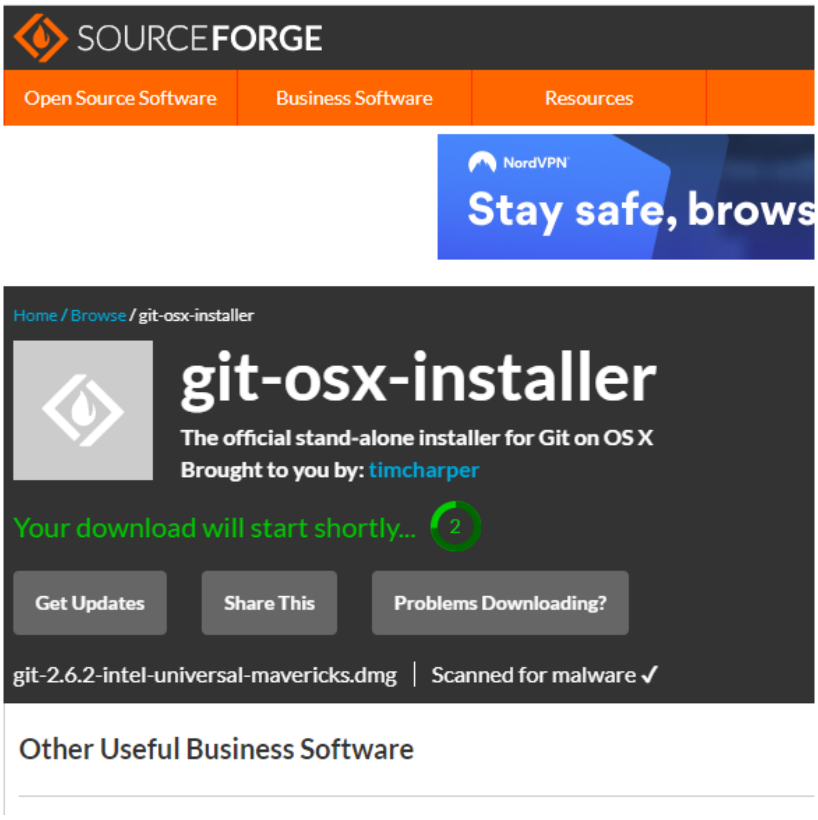
The official stand-alone installer for Git on OS X

Brought to you by: timcharper

16 Reviews Downloads: 9,889 This Week Last Update: 2021-

This is the official stand-alone installer for Git on OS X.

Wait for the download to start



git-osx-installer

The official stand-alone installer for Git on OS X

Brought to you by: timcharper

Your download will start shortly... 2

git-2.6.2-intel-universal-mavericks.dmg | Scanned for malware ✓

Other Useful Business Software

4. Find the package and double-click to open the Git installer.
5. Follow the installation wizard and configure Git to suit your development needs. If you are new to version control systems, the best option would be to leave the default settings.
6. Click Install and type in your password if necessary.
7. Confirm once again by clicking Install Software.

Install Git Using Homebrew

1. Open Terminal and run the following command
   "brew install git"
2. If you don't have Homebrew installed you can follow the steps in the website https://brew.sh/ to install it.
3. To check if git is installed properly, go to the Terminal and type "*git --version*" then you should see the git version

Install Git Using Xcode

1. If you prefer the terminal, using Xcode is the fastest and easiest way to start working with Git. Its command-line tools include Git in the package.
2. Users who don't have Xcode can install it with a single command:
   "xcode-select --install"
3. With Xcode running on your Mac, you can check whether Git is also available by prompting for the Git version: "*git --version*"
4. The output should display the latest Git release, as in the example below.
   "git version 2.25.0 (Apple Git-66)"
5. If you do not have Git, it automatically asks you whether you want to install it. Confirm the installation, and Xcode sets up Git.

Install Git Using MacPorts

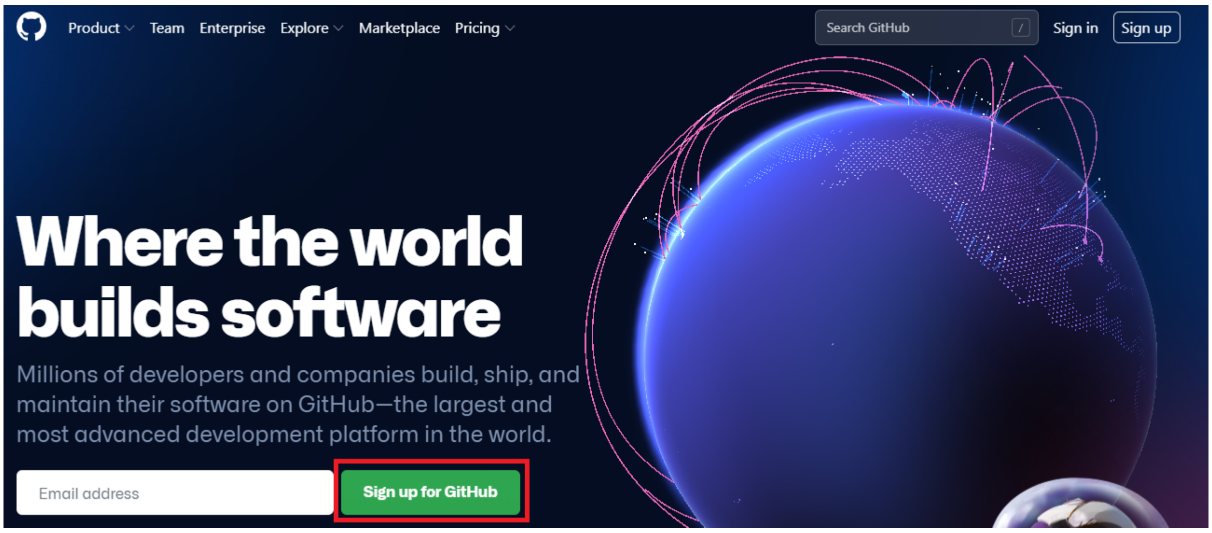
1. If you are using MacPorts to manage your packages on the system, you can use the port command to set up Git.
2. Open up your Terminal, Start by updating MacPorts with the command:
   "sudo port selfupdate".
3. Search for and install the newest Git ports and variants by running the following two commands:
   "port search git"
   "port variants git"
4. And then install git with the command
   "sudo port install git"
5. To check if git is installed properly, go to the Terminal and type "*git --version*" then you should see the git version

Creating Github Account

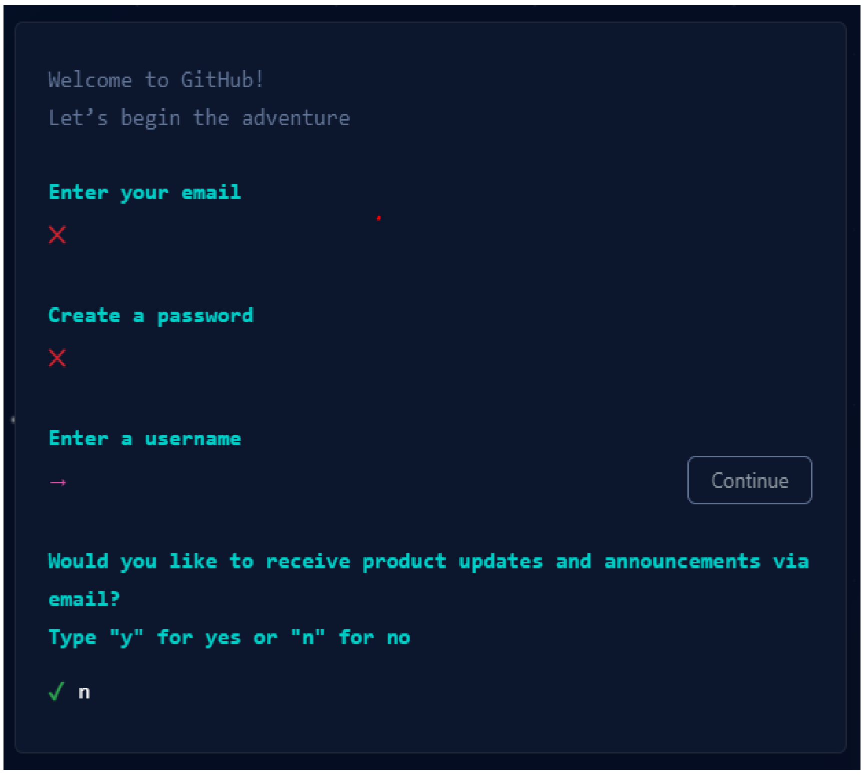
1. Now you have to create a github account. GitHub is a software development platform that allows you to save, track, and collaborate on software projects online. It allows developers to upload their own code files and work on open-source projects with other developers.

2. Navigate to the website https://github.com

3. If you already have an account you can directly **Sign in,** Otherwise to create a new account choose **Sign Up for Github**.

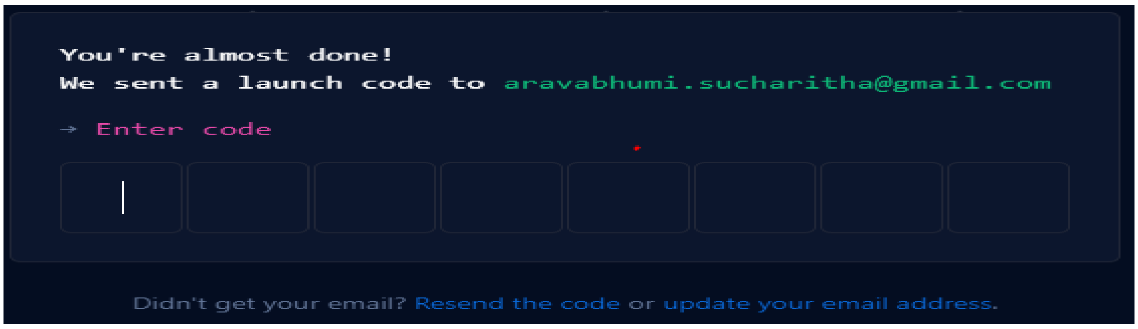

Product
Team
Enterprise
Explore
Marketplace
Pricing
Search GitHub
Sign in
Sign up
Where the world builds software
Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.
Email address
Sign up for GitHub

Fill all the details



```
Welcome to GitHub!
Let's begin the adventure

Enter your email
✗

Create a password
✗

Enter a username
→                                                     Continue

Would you like to receive product updates and announcements via
email?
Type "y" for yes or "n" for no

✓ n
```

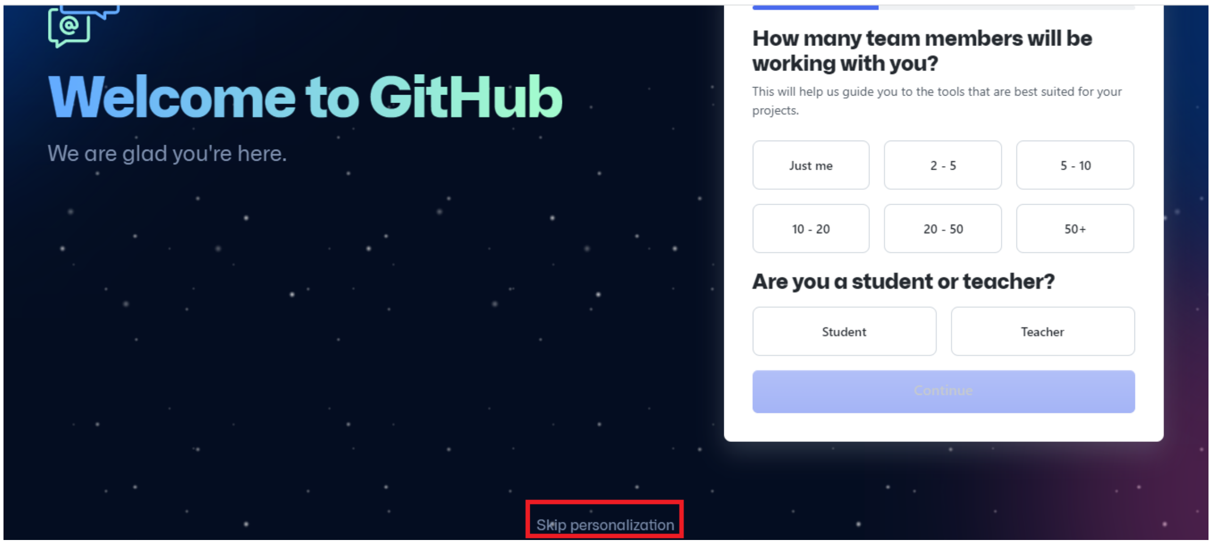
4. You will get a registration code to the email used for registering to github



```
You're almost done!
We sent a launch code to aravabhumi.sucharitha@gmail.com

→ Enter code
```

Didn't get your email? Resend the code or update your email address.

5. You have to answer a few personalization questions or you can choose skip personalization .



Welcome to GitHub

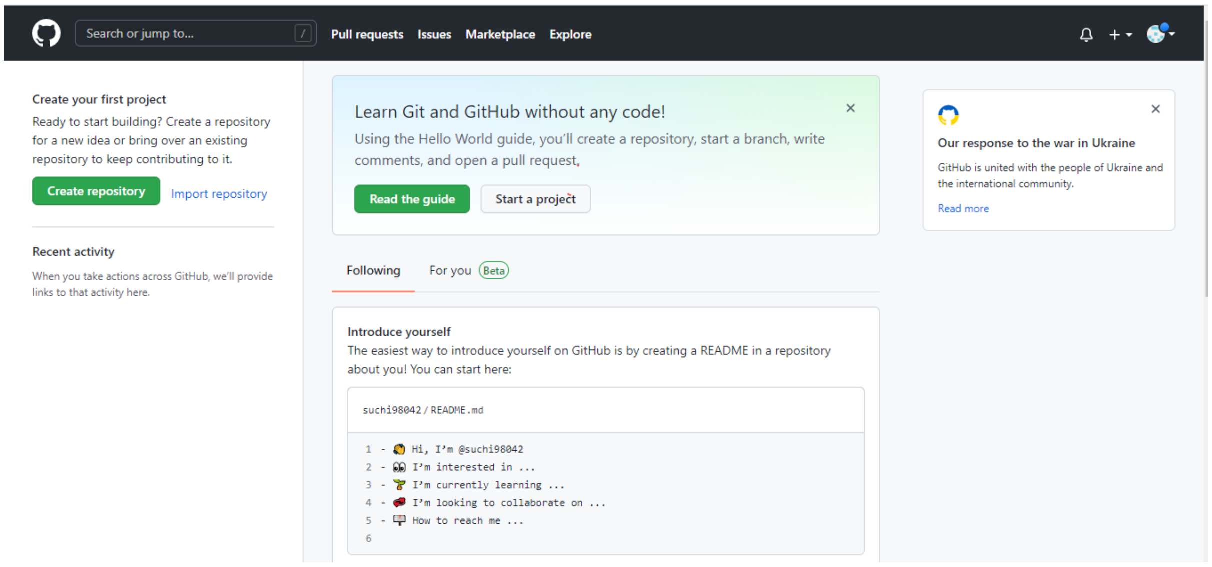
We are glad you're here.

How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Are you a student or teacher?

6. You are all set to work on github, after you're done registering you will be redirected to the dashboard page of github.



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

suchi98042 / README.md

```
- 👋 Hi, I’m @suchi98042
- 👀 I’m interested in ...
- 🌱 I’m currently learning ...
- 💞️ I’m looking to collaborate on ...
- 📫 How to reach me ...
```

Our response to the war in Ukraine

GitHub is united with the people of Ukraine and the international community.

Git Configuration On MacOS

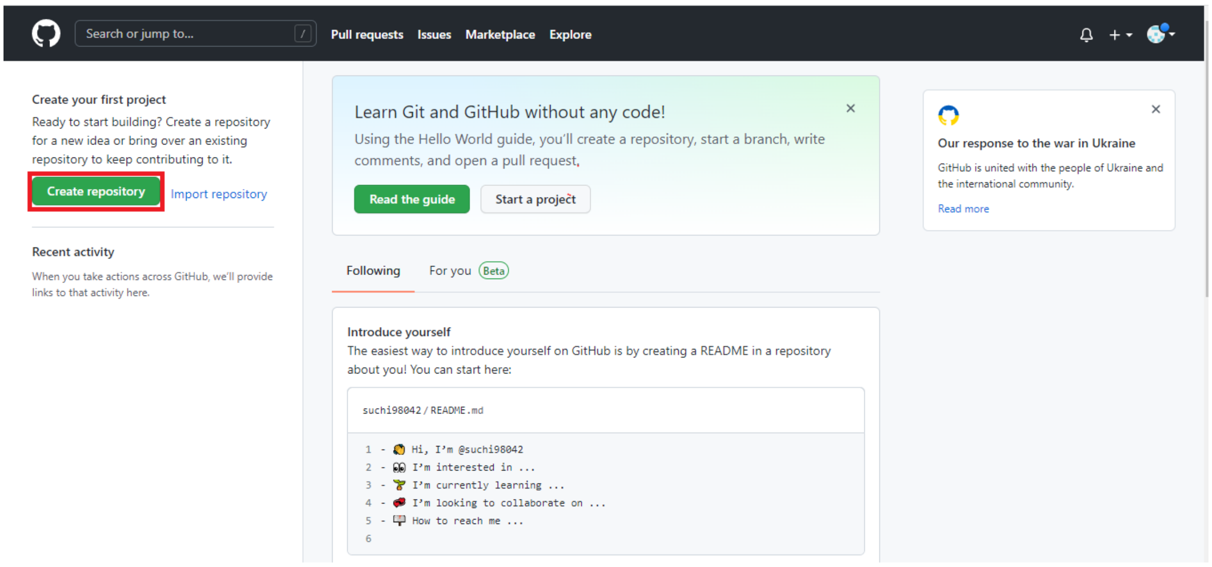
7. You have git for MacOSand a github account. You have to configure the git before using it. Open Terminal and type the following commands.

   git config --global user.name "your username here"

   git config --global user.email "your email here"

Creating a Git Repository-

1. Navigate to the website https://github.com
2. **Sign In** with your credentials and select the option **Create repository.**



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository   Import repository

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide   Start a project

Our response to the war in Ukraine

GitHub is united with the people of Ukraine and the international community.

Read more

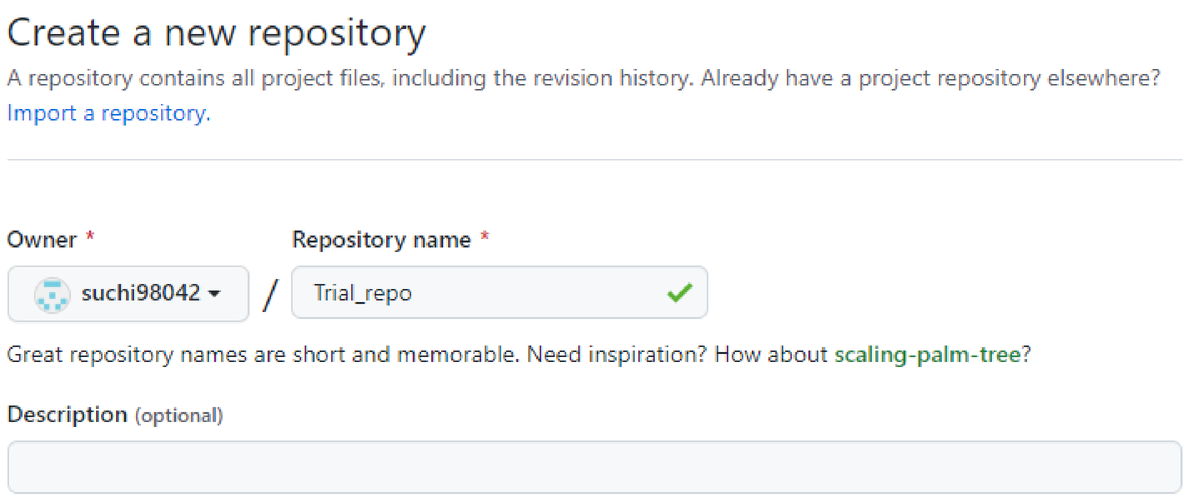
Following   For you (Beta)

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

```
suchi98042 / README.md

- 👋 Hi, I'm @suchi98042
- 👀 I'm interested in ...
- 🌱 I'm currently learning ...
- 💞️ I'm looking to collaborate on ...
- 📫 How to reach me ...
```

3. Type the name of your repository and description is optional.



Create a new repository

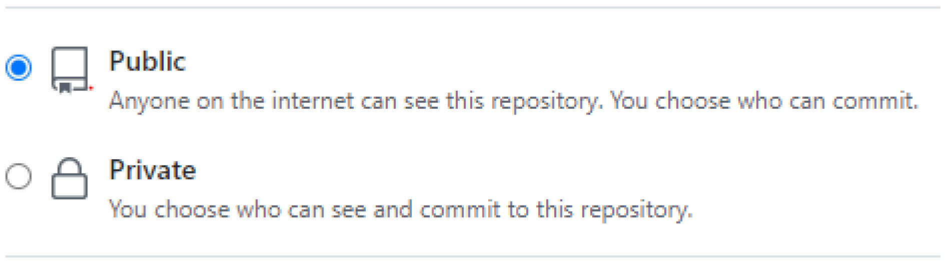
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner *		Repository name *
suchi98042 ▾	/	Trial_repo ✓

Great repository names are short and memorable. Need inspiration? How about scaling-palm-tree?

Description (optional)

4. Next section you choose if your repository is public or private. If you choose public, anyone on the internet can see your code but code commits you can decide. If you choose private, only people you choose can see the code and make any commits

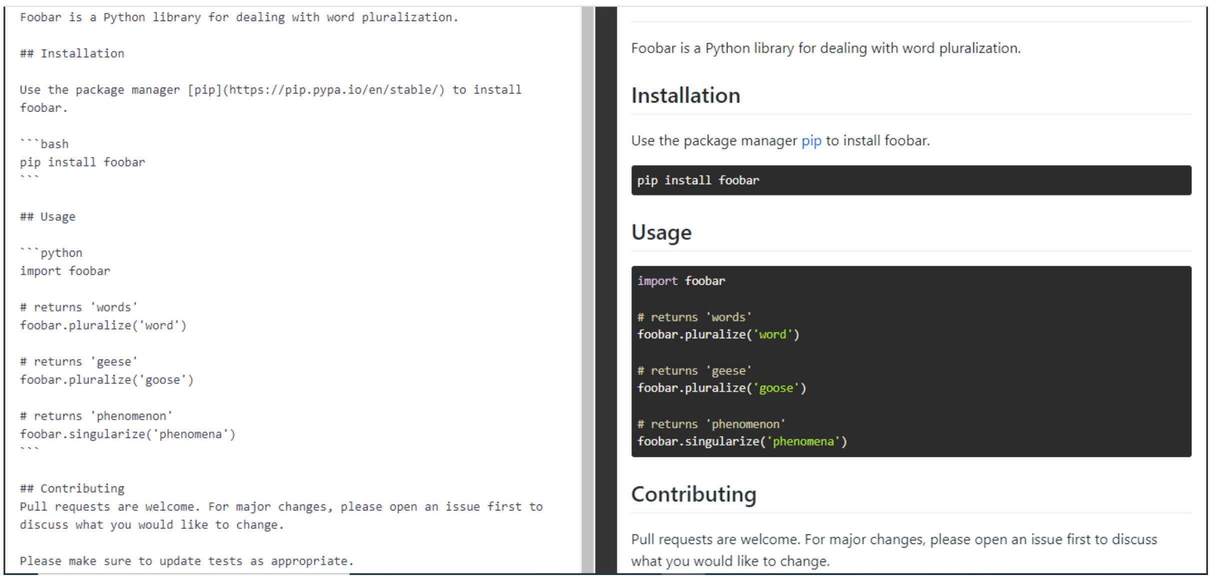


[✓ Public]
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

5. Next step you will be shown various options

   Add a README file- This option allows you to add a README file for your repository. README is a markdown file where you write a long description for your project and add any instructions or documentation that you want to share with others. Use Markdown to format headings, lists, links, etc., I suggest you add this file when creating the repository as you can keep editing this file as you add more functionalities to your project. Example of README.md file- left side shows how the text file looks, right side shows how it is displayed on github.



~~~
Foobar is a Python library for dealing with word pluralization.

## Installation

Use the package manager [pip](https://pip.pypa.io/en/stable/) to install
foobar.

```bash
pip install foobar
~~~

Usage

```python
import foobar

# returns 'words'
foobar.pluralize('word')

# returns 'geese'
foobar.pluralize('goose')

# returns 'phenomenon'
foobar.singularize('phenomena')
```

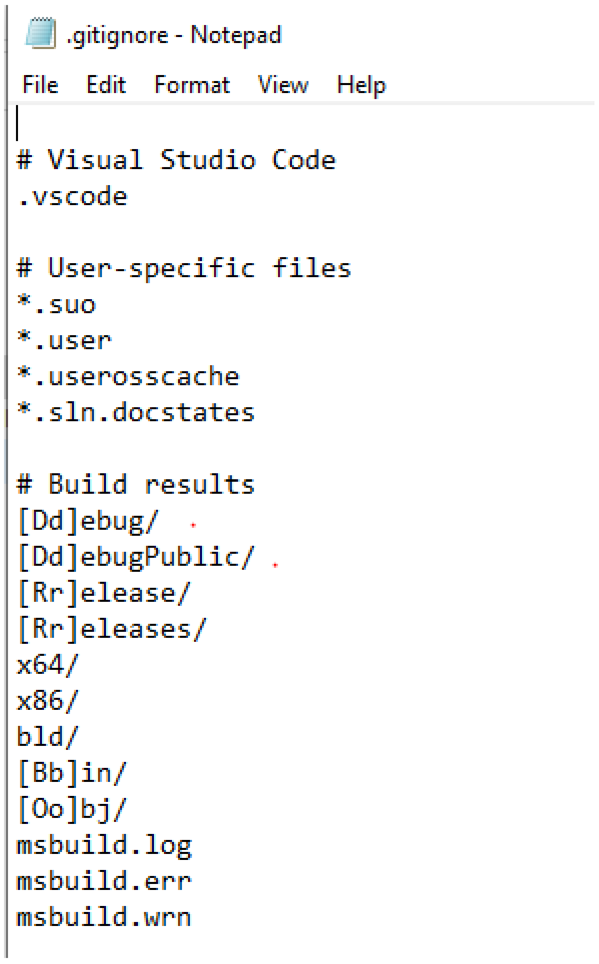
Contributing
Pull requests are welcome. For major changes, please open an issue first to
discuss what you would like to change.

Please make sure to update tests as appropriate.
```

Foobar is a Python library for dealing with word pluralization.

Installation

Use the package manager pip to install foobar.

```
pip install foobar
```

Usage

```
import foobar

# returns 'words'
foobar.pluralize('word')

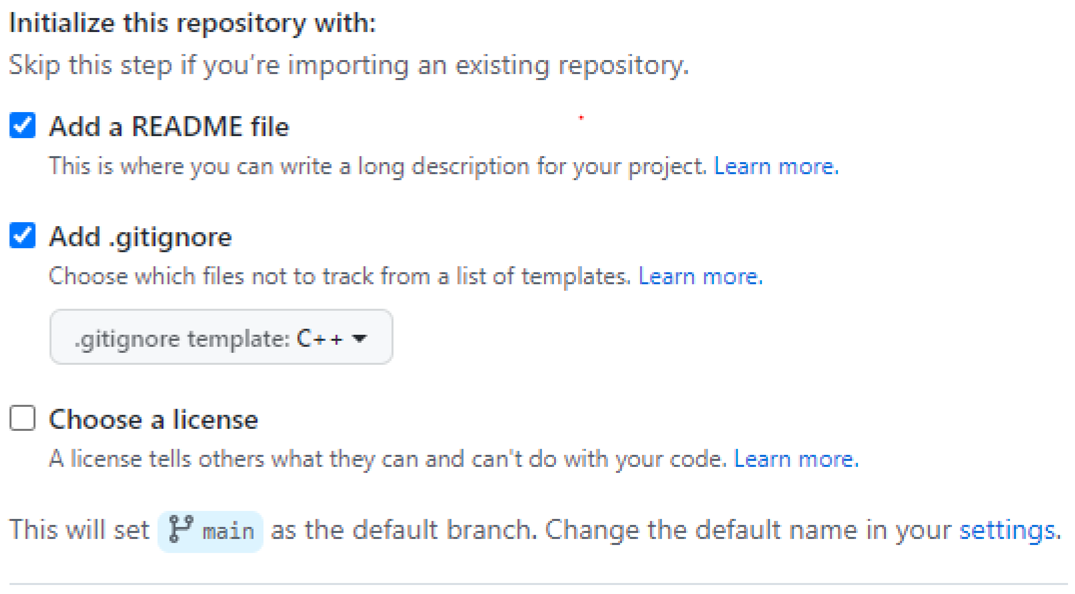
# returns 'geese'
foobar.pluralize('goose')

# returns 'phenomenon'
foobar.singularize('phenomena')
```

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

   Add .gitignore- This option allows you to add a .gitignore file for your repository. The .gitignore file is a text file that tells Git which files or folders to ignore in a project.I suggest you add this file when creating the repository. You can choose a template from github or you can write on your own. Example of .gitignore file
```


.gitignore - Notepad

```
# Visual Studio Code
.vscode

# User-specific files
*.suo
*.user
*.userosscache
*.sln.docstates

# Build results
[Dd]ebug/
[Dd]ebugPublic/
[Rr]elease/
[Rr]eleases/
x64/
x86/
bld/
[Bb]in/
[Oo]bj/
msbuild.log
msbuild.err
msbuild.wrn
```

Choose a license- A license tells others what they can and can't do with your code.


Initialize this repository with:

Skip this step if you're importing an existing repository.

☑ Add a README file

This is where you can write a long description for your project. Learn more.

☑ Add .gitignore

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

.gitignore template: C++

☐ Choose a license

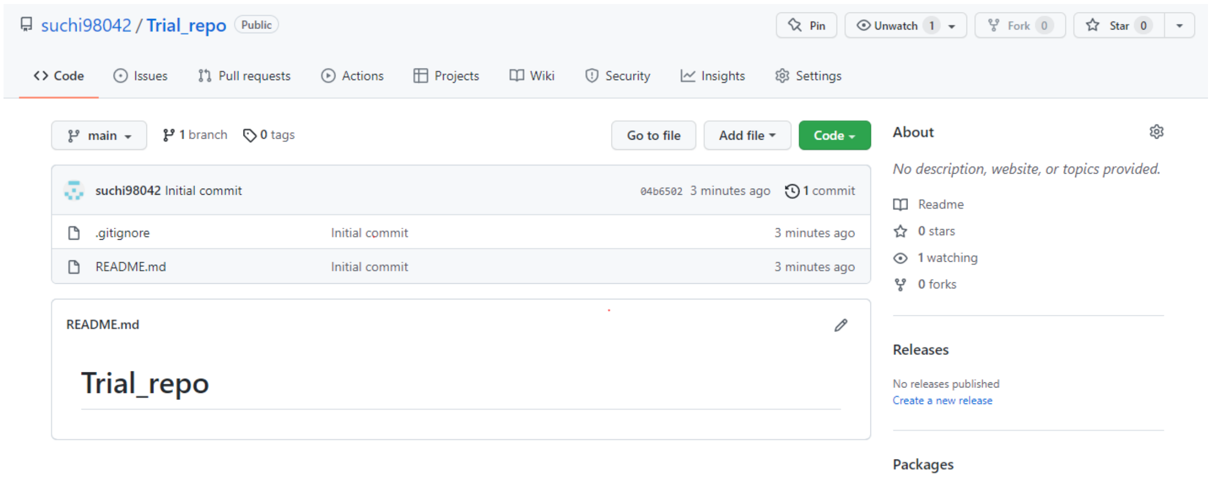
A license tells others what they can and can't do with your code. Learn more.

This will set main as the default branch. Change the default name in your settings.

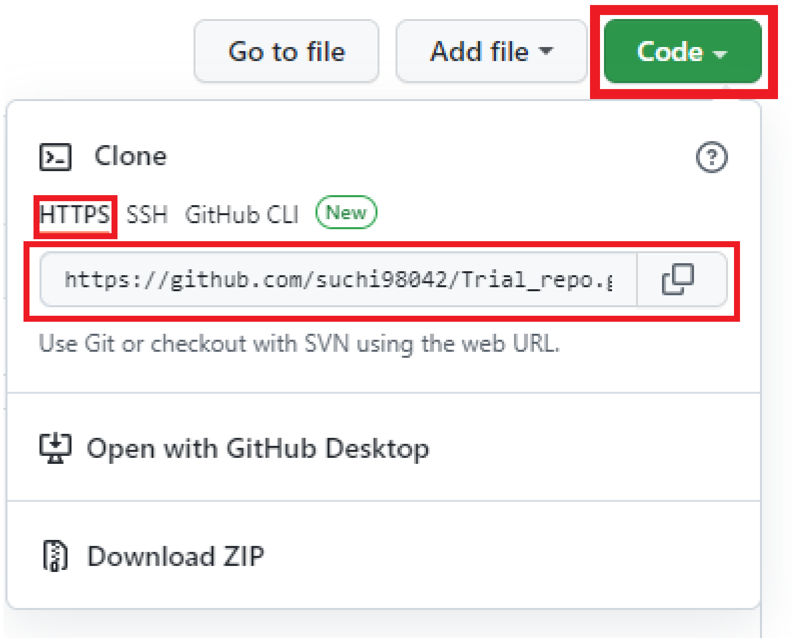
6. Then select **Create repository**. By default it will create a repository with one branch which is named either master or main branch.

Clone repository from github-

1. Navigate to the repository you would like to clone.A repository will look like this-



2. Select the option **code** on the right hand side corner, select **HTTPS** and copy the link. The default option is **HTTPS**.

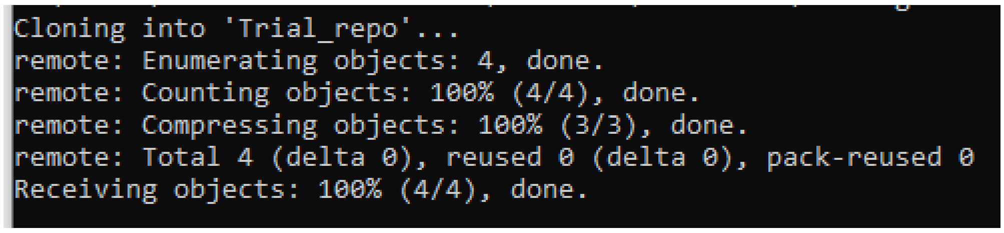

Go to file
Add file
Code
Clone
HTTPS
SSH
GitHub CLI
New
https://github.com/suchi98042/Trial_repo.g
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP

3. Open command prompt and navigate to the location on your computer in which you want to clone the repository. Type command *git clone path to repository.*

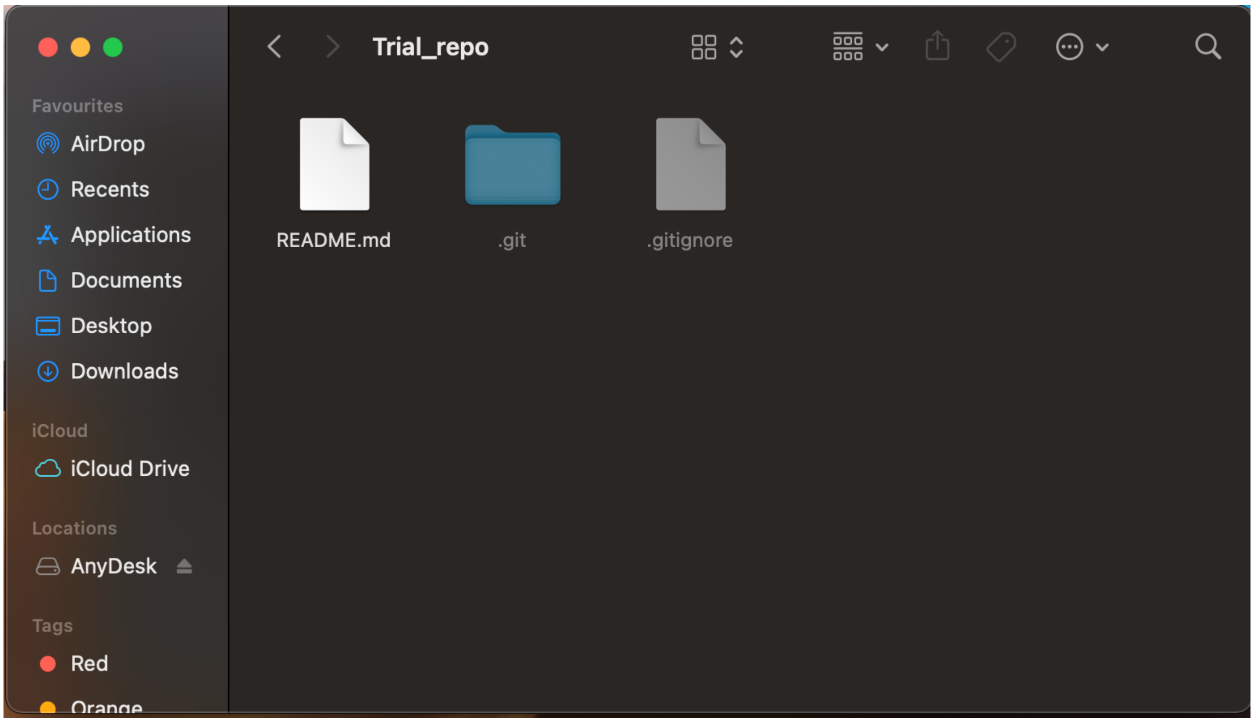
```
git clone https://github.com/suchi98042/Trial_repo.git
```

4. To clone a particular branch *git clone -b branch-name path to repository*.'

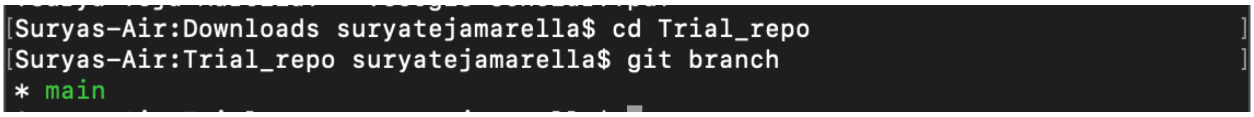
5. After cloning it creates a folder with your repo name. Navigate to the folder you can see all the files in your repository in the folder. You can work on the files as you work regularly.



```
Cloning into 'Trial_repo'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```



6. *git branch* command shows the branch which you cloned into the local repository.



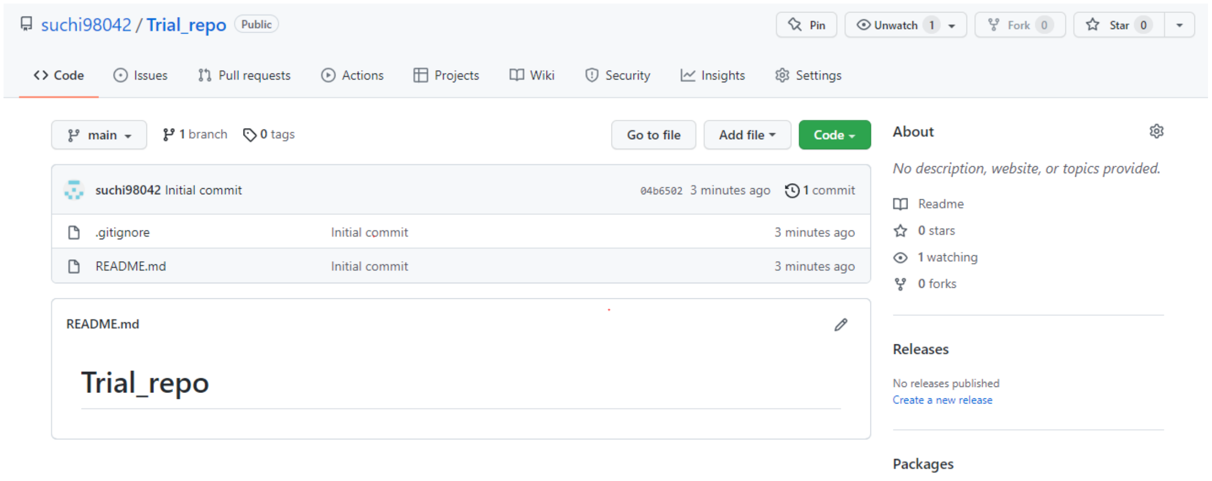
```
[Suryas-Air:Downloads suryatejamarella$ cd Trial_repo
[Suryas-Air:Trial_repo suryatejamarella$ git branch
* main
```

Creating a new branch-

There are two ways to create a branch from github and the second way is from command prompt.

From github:

1. Navigate to the repository you would like to create a branch.A repository will look like this-



suchi98042 / Trial_repo Public

suchi98042 Initial commit		04b6502 3 minutes ago	1 commit
.gitignore	Initial commit	3 minutes ago	
README.md	Initial commit	3 minutes ago	

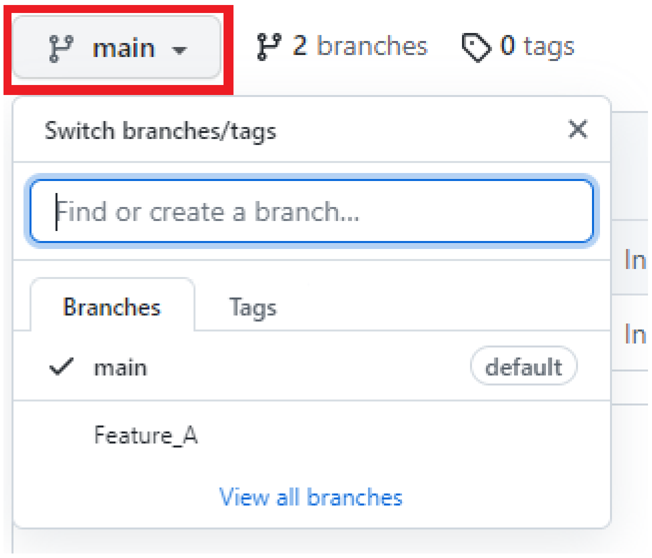
README.md

Trial_repo

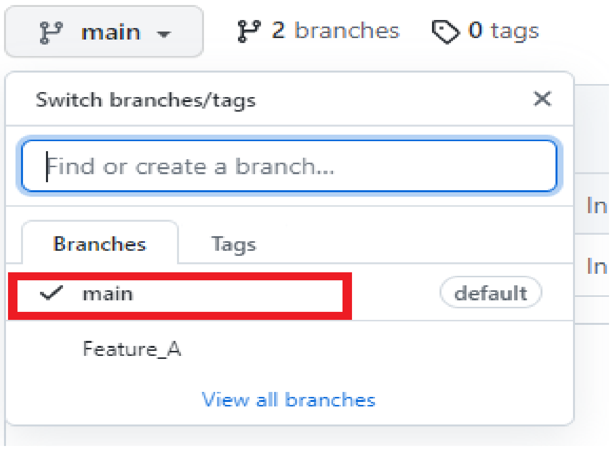
About

No description, website, or topics provided.

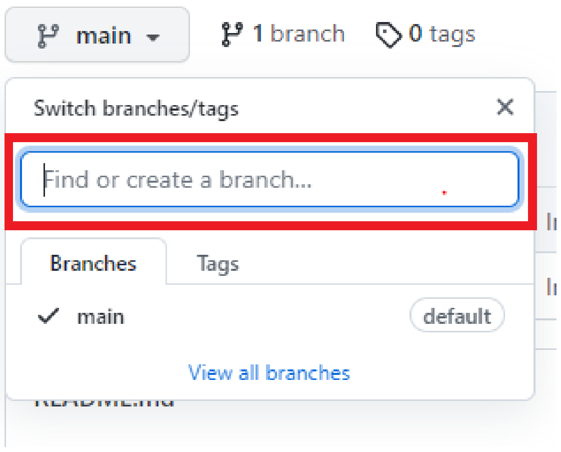
2. Select the branch list dropdown from the left hand side of your dashboard.



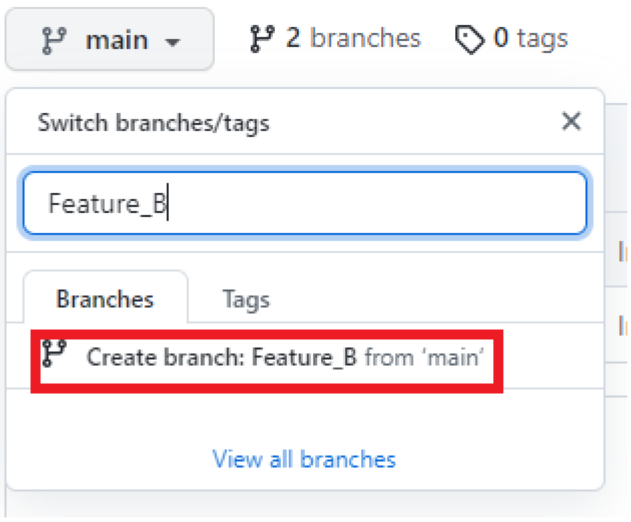
3. Select the branch from which you want to make a copy from. I selected the main branch.



4. Start typing the new branch name in the textbox.

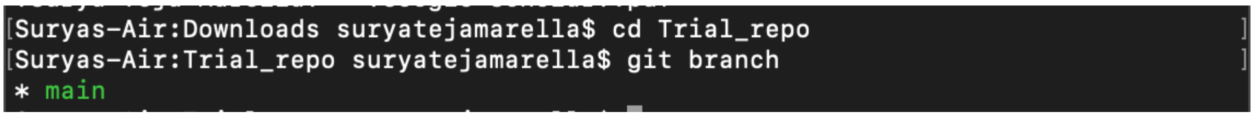


5. An option to create a branch will appear, select that option and the branch will be created.



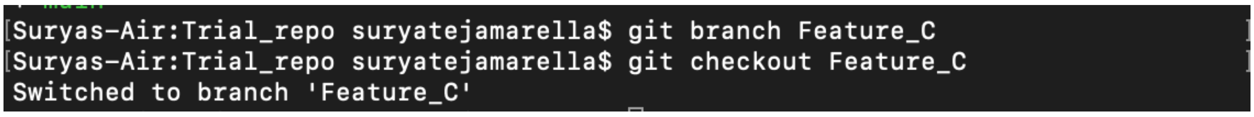
From Command Prompt:

1. *git branch* command shows the branch which you cloned into the local repository.



```
[Suryas-Air:Downloads suryatejamarella$ cd Trial_repo
[Suryas-Air:Trial_repo suryatejamarella$ git branch
* main
```

2. *git branch new_branch_name* creates new branch. *git checkout new_branch_name* changes local working directory.

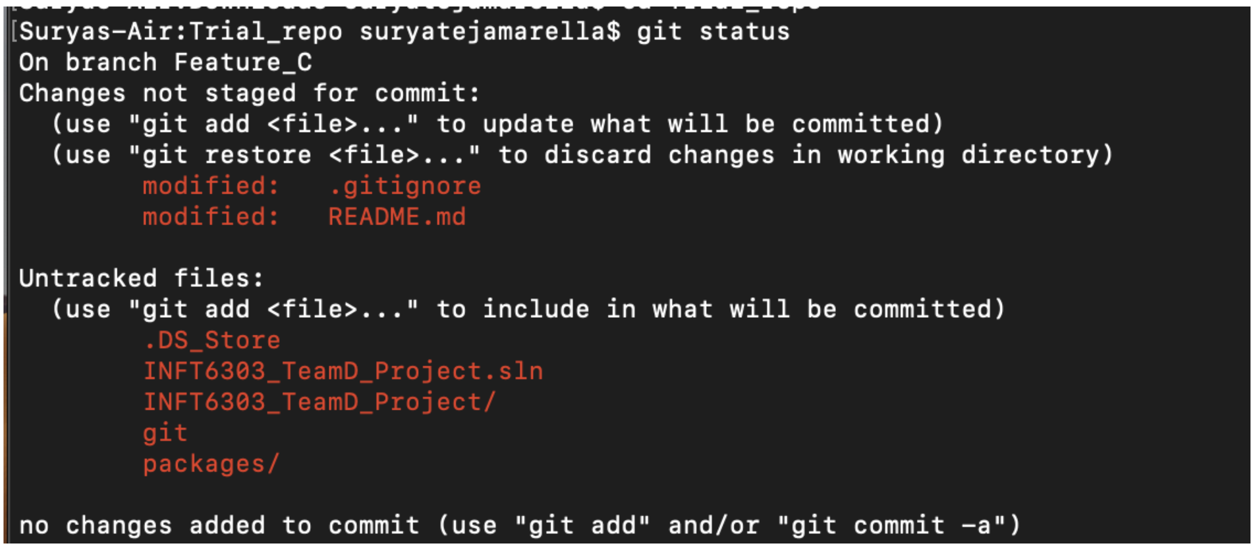


```
[Suryas-Air:Trial_repo suryatejamarella$ git branch Feature_C
[Suryas-Air:Trial_repo suryatejamarella$ git checkout Feature_C
Switched to branch 'Feature_C'
```

3. *git checkout -b new_branch_name* is shorthand command for above both commands.

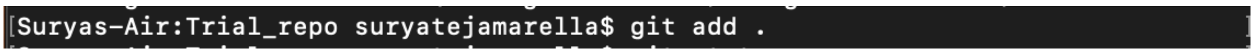
Push code to github:

1. *git status* shows the list of tracked, untracked files and changes.



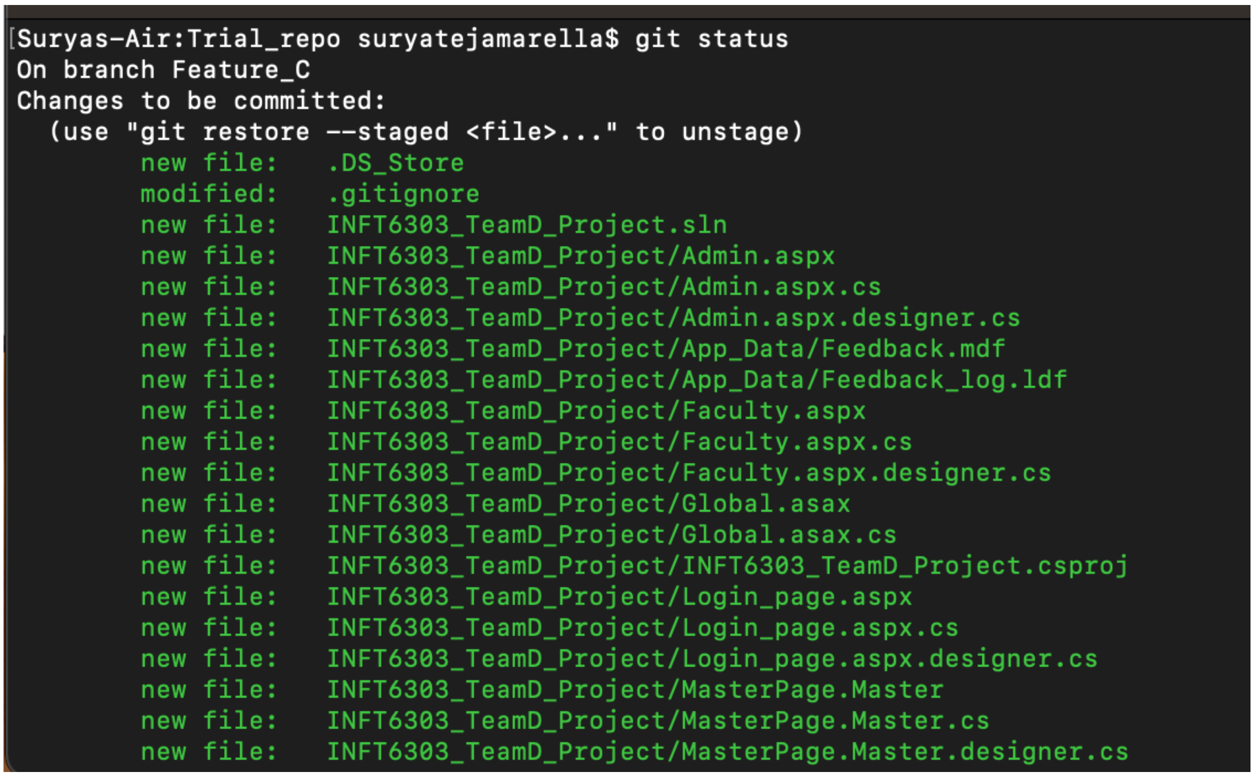
```
[Suryas-Air:Trial_repo suryatejamarella$ git status
On branch Feature_C
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        INFT6303_TeamD_Project.sln
        INFT6303_TeamD_Project/
        git
        packages/

no changes added to commit (use "git add" and/or "git commit -a")
```

2. *git add <filename>* to add a new or changed file in your working directory to the Git staging area. *git add .* or *git add* * to add all files to the staging area.



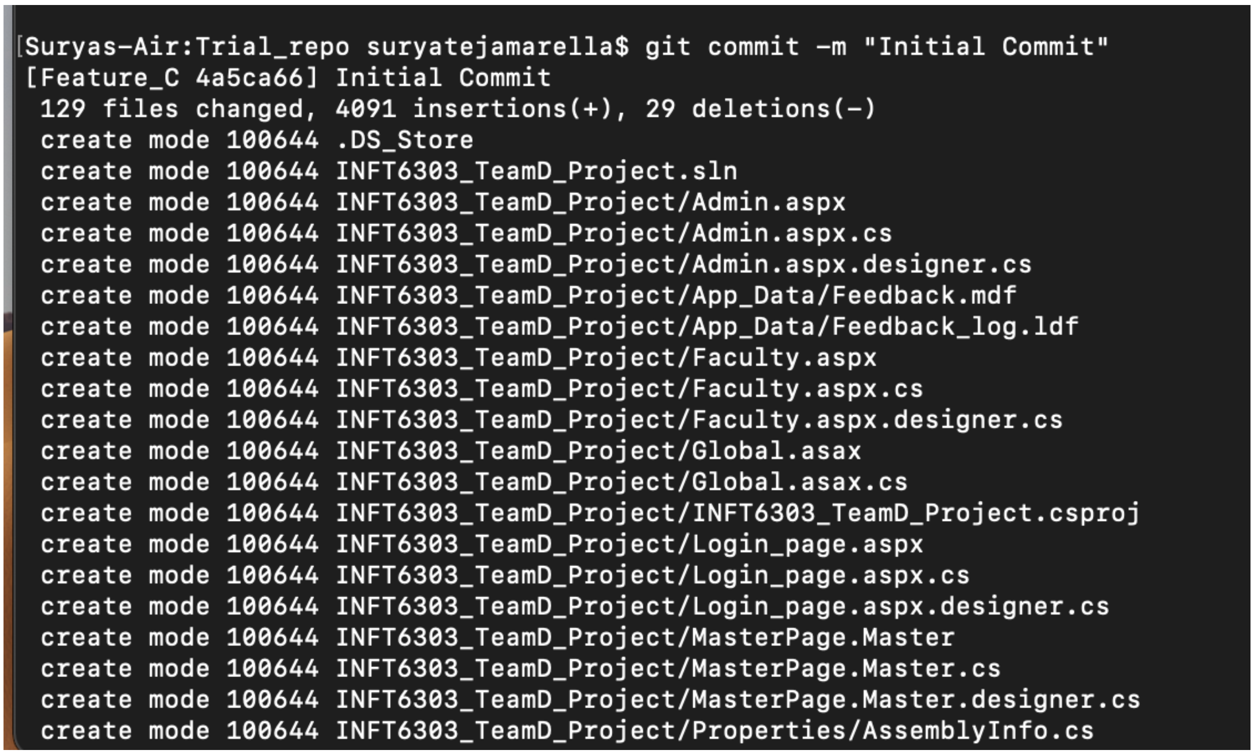
```
[Suryas-Air:Trial_repo suryatejamarella$ git add .
```

3. *git status* now you can see all files are in green color and added to the staging area.



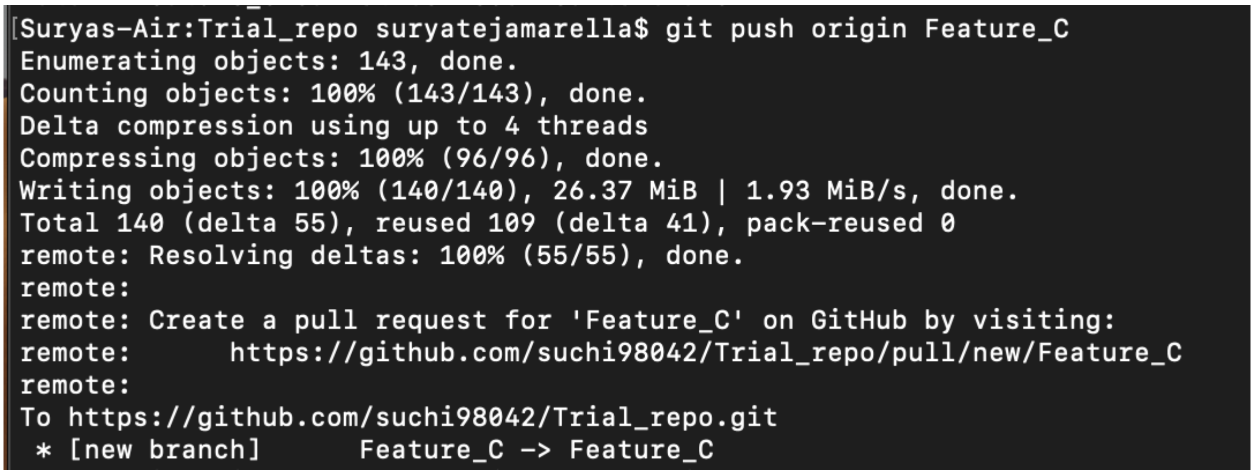
```
[Suryas-Air:Trial_repo suryatejamarella$ git status
On branch Feature_C
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .DS_Store
        modified:   .gitignore
        new file:   INFT6303_TeamD_Project.sln
        new file:   INFT6303_TeamD_Project/Admin.aspx
        new file:   INFT6303_TeamD_Project/Admin.aspx.cs
        new file:   INFT6303_TeamD_Project/Admin.aspx.designer.cs
        new file:   INFT6303_TeamD_Project/App_Data/Feedback.mdf
        new file:   INFT6303_TeamD_Project/App_Data/Feedback_log.ldf
        new file:   INFT6303_TeamD_Project/Faculty.aspx
        new file:   INFT6303_TeamD_Project/Faculty.aspx.cs
        new file:   INFT6303_TeamD_Project/Faculty.aspx.designer.cs
        new file:   INFT6303_TeamD_Project/Global.asax
        new file:   INFT6303_TeamD_Project/Global.asax.cs
        new file:   INFT6303_TeamD_Project/INFT6303_TeamD_Project.csproj
        new file:   INFT6303_TeamD_Project/Login_page.aspx
        new file:   INFT6303_TeamD_Project/Login_page.aspx.cs
        new file:   INFT6303_TeamD_Project/Login_page.aspx.designer.cs
        new file:   INFT6303_TeamD_Project/MasterPage.Master
        new file:   INFT6303_TeamD_Project/MasterPage.Master.cs
        new file:   INFT6303_TeamD_Project/MasterPage.Master.designer.cs
```

4. *git commit -m “commit message”* Adding commits keep track of our progress and changes as we work. Git considers each commit change point or "save point". It is a point in the project you can go back to if you find a bug, or want to make a change.



```
[Suryas-Air:Trial_repo suryatejamarella$ git commit -m "Initial Commit"
[Feature_C 4a5ca66] Initial Commit
 129 files changed, 4091 insertions(+), 29 deletions(-)
 create mode 100644 .DS_Store
 create mode 100644 INFT6303_TeamD_Project.sln
 create mode 100644 INFT6303_TeamD_Project/Admin.aspx
 create mode 100644 INFT6303_TeamD_Project/Admin.aspx.cs
 create mode 100644 INFT6303_TeamD_Project/Admin.aspx.designer.cs
 create mode 100644 INFT6303_TeamD_Project/App_Data/Feedback.mdf
 create mode 100644 INFT6303_TeamD_Project/App_Data/Feedback_log.ldf
 create mode 100644 INFT6303_TeamD_Project/Faculty.aspx
 create mode 100644 INFT6303_TeamD_Project/Faculty.aspx.cs
 create mode 100644 INFT6303_TeamD_Project/Faculty.aspx.designer.cs
 create mode 100644 INFT6303_TeamD_Project/Global.asax
 create mode 100644 INFT6303_TeamD_Project/Global.asax.cs
 create mode 100644 INFT6303_TeamD_Project/INFT6303_TeamD_Project.csproj
 create mode 100644 INFT6303_TeamD_Project/Login_page.aspx
 create mode 100644 INFT6303_TeamD_Project/Login_page.aspx.cs
 create mode 100644 INFT6303_TeamD_Project/Login_page.aspx.designer.cs
 create mode 100644 INFT6303_TeamD_Project/MasterPage.Master
 create mode 100644 INFT6303_TeamD_Project/MasterPage.Master.cs
 create mode 100644 INFT6303_TeamD_Project/MasterPage.Master.designer.cs
 create mode 100644 INFT6303_TeamD_Project/Properties/AssemblyInfo.cs
```

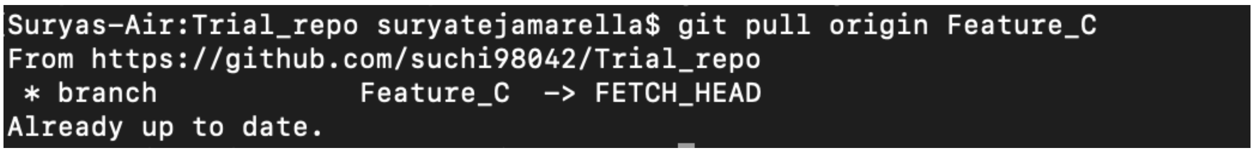
5. *git push origin branch_name* to push the code to remote repository.



```
[Suryas-Air:Trial_repo suryatejamarella$ git push origin Feature_C
Enumerating objects: 143, done.
Counting objects: 100% (143/143), done.
Delta compression using up to 4 threads
Compressing objects: 100% (96/96), done.
Writing objects: 100% (140/140), 26.37 MiB | 1.93 MiB/s, done.
Total 140 (delta 55), reused 109 (delta 41), pack-reused 0
remote: Resolving deltas: 100% (55/55), done.
remote:
remote: Create a pull request for 'Feature_C' on GitHub by visiting:
remote:      https://github.com/suchi98042/Trial_repo/pull/new/Feature_C
remote:
To https://github.com/suchi98042/Trial_repo.git
 * [new branch]      Feature_C -> Feature_C
```

Pull code from git:

git pull <remote> branch_name used to fetch and download content from a remote repository and immediately update the local repository to match that content.



```
Suryas-Air:Trial_repo suryatejamarella$ git pull origin Feature_C
From https://github.com/suchi98042/Trial_repo
 * branch            Feature_C  -> FETCH_HEAD
Already up to date.
```

Merging two branches:

It’s preferred to change/switch to the master branch before any branch needs to be merged with it. This will merge the specified branch with our master branch.

git merge <branch_name>

Deleting a branch:

git branch -d <branch_name>