

Github Instruction

1. Create a GitHub account (it's important since I need add your Email-Address so you can also upload files to Github)

The screenshot shows the GitHub 'Join GitHub' page. At the top, it says 'Join GitHub' and 'The best way to design, build, and ship software.' Below this are three steps: Step 1: Create personal account, Step 2: Choose your plan, and Step 3: Tailor your experience. The 'Create your personal account' section is active, showing a form with three fields: Username, Email, and Password. The Username field contains 'WorgenZhang' and has a red error message 'Username is already taken'. The Email field contains 'frank08081993@gmail.com' and has a red error message 'Email is invalid or already taken'. The Password field is empty and has a red error message 'Use at least one lowercase letter, one numeral, and seven characters.' To the right of the form is a box titled 'You'll love GitHub' with the following text: 'Unlimited collaborators', 'Unlimited public repositories', 'Great communication', 'Frictionless development', and 'Open source community'.

2. Use Terminal command “cd ...” to change directory to any place you want in your local computer. (My path is to desktop)

```
FanZhang:597-Group-Project Frank$ cd /Users/Frank/Desktop
FanZhang:Desktop Frank$
```

3. Git clone <https://github.com/WorgenZhang/597-Group-Project.git> (**Please ignore this step if you already did this before**)

```
FanZhang\Desktop Frank$ git clone https://github.com/WorgenZhang/597-Group-Project.git
```

Everytime if you guys want to edit the code or upload files, please do this first:

1. Cd 597-Group-Project/

```
FanZhang\Desktop Frank$ cd 597-Group-Project/  
FanZhang:597-Group-Project Frank$
```

2. Git pull <https://github.com/WorgenZhang/597-Group-Project.git> (Rember, this code fetch the folder from github and replace your previous files in 597-Group-Project)

After you change the files in folder and want to upload to Github so we can see, do these steps:

3. Git init

4. Git add .

5. Git commit -m 'name it whatever you want'

6. Git push -u origin master

Done!

[Importing your projects to GitHub](#) / Adding an existing project to GitHub using the comm...

How can we help?



Adding an existing project to GitHub using the command line

MAC | WINDOWS | LINUX

Putting your existing work on GitHub can let you share and collaborate in lots of great ways.

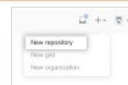
Tip: If you're most comfortable with a point-and-click user interface, try adding your project with GitHub Desktop. For more information, see "Adding a repository from your local computer to GitHub Desktop" in the *GitHub Desktop Help*.

Warning: Never `git add`, `commit`, or `push` sensitive information to a remote repository. Sensitive information can include, but is not limited to:

- › Passwords
- › SSH keys
- › [AWS access keys](#)
- › API keys
- › Credit card numbers
- › PIN numbers

For more information, see "[Removing sensitive data from a repository](#)."

- 1 [Create a new repository](#) on GitHub. To avoid errors, do not initialize the new repository with `README`, license, or `.gitignore` files. You can add these files after your project has been pushed to GitHub.



- 2 Open Terminal.
- 3 Change the current working directory to your local project.
- 4 Initialize the local directory as a Git repository.


```
$ git init
```

- 5 Add the files in your new local repository. This stages them for the first commit.

```
$ git add .  
# Adds the files in the local repository and stages them for commit. To unstage a file, use 'git reset HEAD  
YOUR-FILE'.
```

- 6 Commit the files that you've staged in your local repository.

```
$ git commit -m "First commit"  
# Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this  
commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.
```

- 7 At the top of your GitHub repository's Quick Setup page, click  to copy the remote repository URL.



- 8 In Terminal, [add the URL for the remote repository](#) where your local repository will be pushed.

```
$ git remote add origin remote repository URL  
# Sets the new remote  
$ git remote -v  
# Verifies the new remote URL
```

- 9 [Push the changes](#) in your local repository to GitHub.

```
$ git push -u origin master  
# Pushes the changes in your local repository up to the remote repository you specified as the origin
```

Article versions

[GitHub.com](#)
[GitHub Enterprise 2.8](#)
[GitHub Enterprise 2.7](#)
[GitHub Enterprise 2.6](#)
[GitHub Enterprise 2.5](#)
[GitHub Enterprise 2.4](#)