Assignment 1: Git & GitHub

▼ 1. Create a repo on GitHub

https://github.com/princekili/iOS_Class_Remote_Assignments

▼ 2. Git commands

> 💡 Here are a few git and GitHub commands we usually use in software development, please explain the meanings and use cases of them.

▼ **git status**

→ It shows the state of the working directory and the staging area. It also shows which changes have been staged, which haven't, and which files aren't being tracked by git.

→ When you make some changes for your project, you should know the state of the working directory and the staging area before you add or commit them.

▼ **git add**

→ To add changes to the staging area from the working directory.

→ You have to add changes to staging area before committing them.

▼ **git commit**

→ To create a commit.

→ A commit is a point in history that you can then reference and access later on. It makes revision control of projects easier.

▼ **git log**

→ It shows the history of commits that we'd created for the repo.

→ It's good to check the git log after committing.

▼ **git push [Repo_name] [Branch_name]**

→ To sync the repo on GitHub with the local one.

→ When you want to share the latest repo to others after committing.

▼ **git remote -v**

→ It shows the URLs of remote repos when listing your current remote connections. If there are no remotes set up, you'll see nothing.

→ To check the link of remote repo and local one.

▼ **git branch**

→ To check the state of branches list or to create a new branch

→ When you want to know which branch you're on or to create a new branch

▼ **fork**

→ It means you just create a copy of the main repository of a project source code to your own GitHub profile.

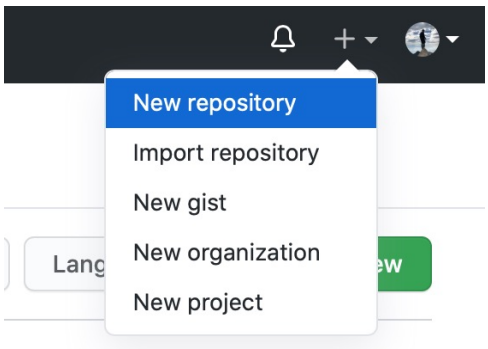
→ You can make changes and create a Pull Request to the main repository branch. If the Main Repository owners like your changes they will merge it to the main repository.

▼ 3. Establish a GitHub repo and upload the local projects

💡 Please describe how to establish a github repo and how to upload the local projects to github. Try to explain it as detail as possible.

▼ 3-1. Create a new repo on GitHub.com

→ New repository



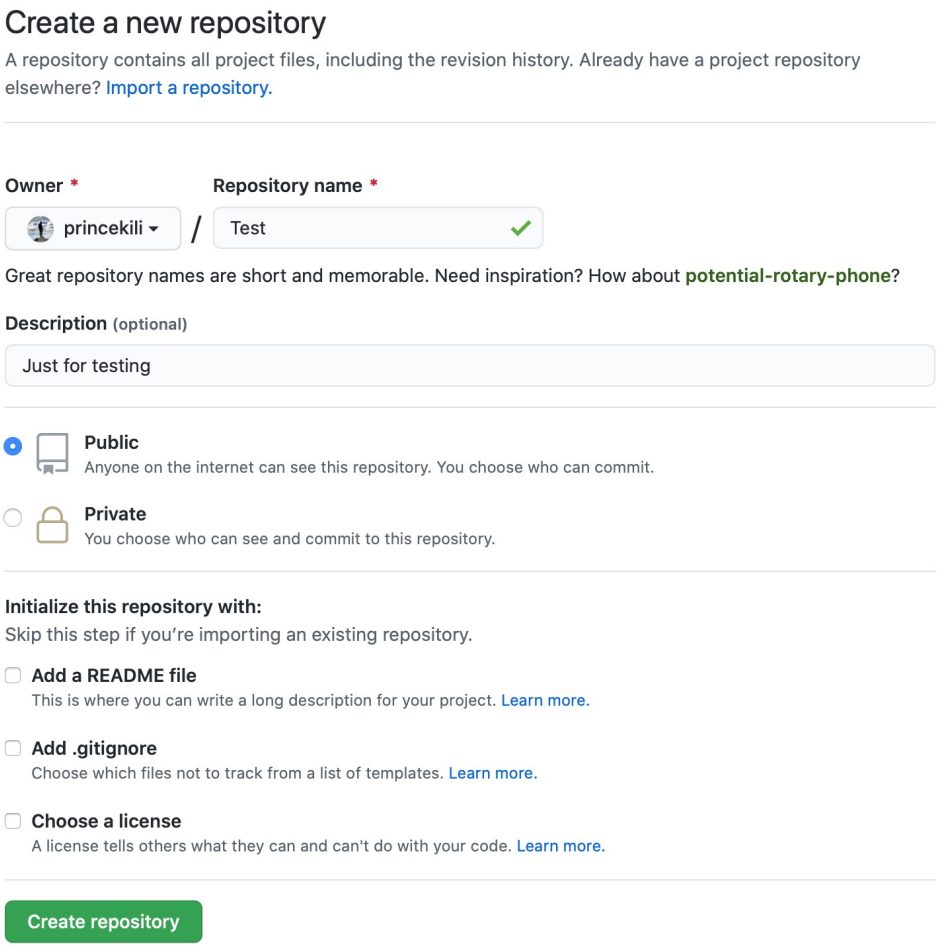
→ Give it a name

→ Add a README file *(optional)*

→ Add .gitignore *(optional)*

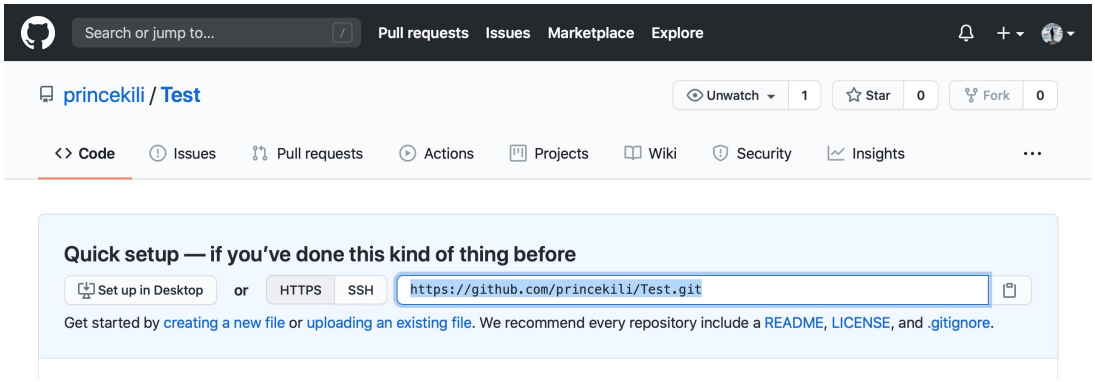
→ Choose a license *(optional)*

→ No need if your local repo already has them.



→ Copy the URL

→ Go back to command line



▼ 3-2. Upload the local projects

💡 Local repo ← 🔗 → Remote repo

▼ `cd Test`

→ Go to the directory of your local project

```
~/Desktop
> cd Test

~/Desktop/Test
>
```

▼ `git init`

→ Initialized empty Git repo

```
~/Desktop/Test
> git init
Initialized empty Git repository in /Users/prince/Desktop/Test/.git/
```

▼ `git remote add origin https://github.com/princekili/Test.git`

→ Link the local repo and the remote repo via URL

```
~/Desktop/Test
> git remote add origin https://github.com/princekili/Test.git
```

▼ `git remote -v`

→ Check the link of remote repo

```
~/Desktop/Test
> git remote -v
origin  https://github.com/princekili/Test.git (fetch)
origin  https://github.com/princekili/Test.git (push)
```

▼ `git status`

→ Check if there are untracked files

```
~/Desktop/Test
> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Images/
        Videos/

nothing added to commit but untracked files present (use "git add" to
 track)
```

▼ `git add .` + `git status`

→ Add changes to the Staging Area and check the git status

```
~/Desktop/Test
> git add .

~/Desktop/Test
> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Images/.keep
        new file:   Videos/.keep
```

▼ `git commit -m "Upload the local repo"` + `git status`

→ Create a commit and check the git status

```
~/Desktop/Test
> git commit -m "Upload the local repo"
[master (root-commit) 6a1188e] Upload the local repo
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Images/.keep
 create mode 100644 Videos/.keep

~/Desktop/Test
> git status
On branch master
nothing to commit, working tree clean
```

▼ `git push --set-upstream origin master`

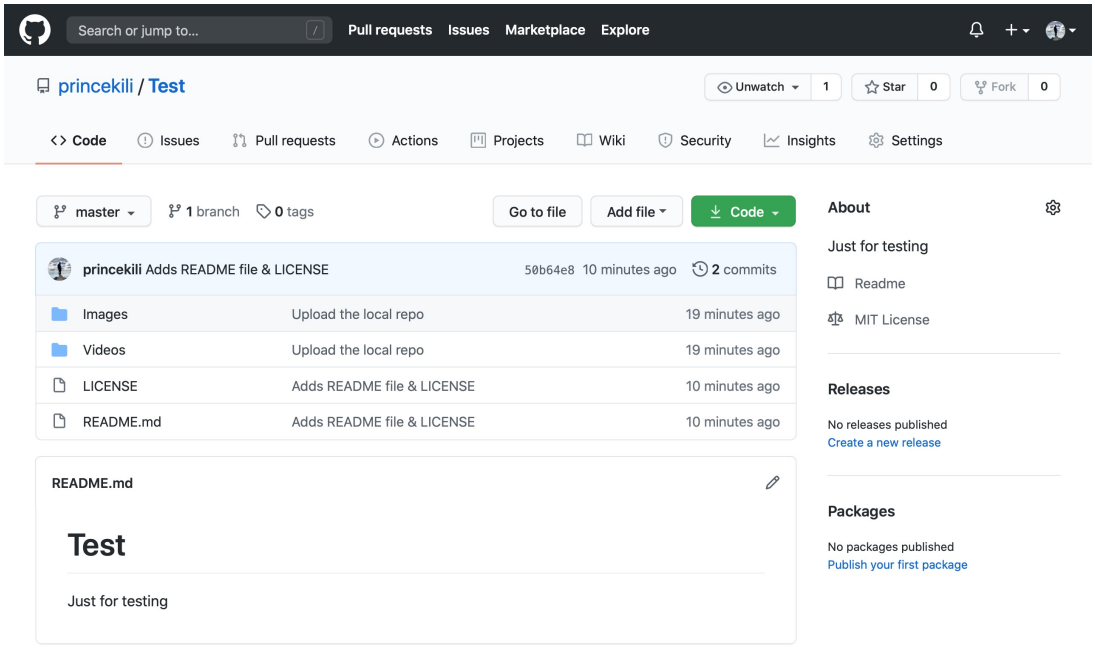
→ The set upstream argument ensures that my local branch will track the remote master branch.

→ Enter Username & Password for GitHub for the first time

```
~/Desktop/Test
> git push --set-upstream origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.20 KiB | 1.20 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/princekili/Test.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

▼ 3-3. Refresh the GitHub repo page

→ You'll see the updated repo


Search or jump to...
Pull requests
Issues
Marketplace
Explore
princekili / Test
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
master
1 branch
0 tags
Go to file
Add file
Code
princekili Adds README file & LICENSE
50b64e8 10 minutes ago
2 commits
Images Upload the local repo 19 minutes ago
Videos Upload the local repo 19 minutes ago
LICENSE Adds README file & LICENSE 10 minutes ago
README.md Adds README file & LICENSE 10 minutes ago
README.md
Test
Just for testing
About
Just for testing
Readme
MIT License
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package