

# 오픈소스 소프트웨어 실습

Open-Source Software Lab

## #12

# 실습 담당 조교 연락처

◆실습 조교 : 김민곤

◆연구실 : 학연산클러스터 601호

◆이메일 : [phenix235@hanyang.ac.kr](mailto:phenix235@hanyang.ac.kr)

메일양식: [오픈소스]학번\_수업일(20190000)\_이름

# Github

깃허브(GitHub)는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이다. 2011년의 조사에서는 가장 인기있는 오픈 소스 코드 저장소로 뽑혔다. - 위키백과 -



# Github

- GitHub는 Git을 지원하는 서비스 중 가장 유명 하기 때문에, 유명한 오픈 소스 라이브러리들이 많이 올라옴
- Ex)TensorFlow
  1. 저장소에 팀원 초대는 무제한
  2. 개인 저장소(Private Repository, 공개되지 않는 저장소)는 유료  
(무료로 개인 저장소를 제공하는 다른 Git 웹 서비스들이 존재)
  3. Pull -> Pull Request 형식

# Github 계정 만들기

## Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

Use at least one letter, one numeral, and seven characters.


**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

<https://github.com/>

# Github 계정 만들기

1. 이름(nickname)과 이메일 주소(인증가능한), 비밀번호 입력
2. Verify 통과
3. 무료버전 선택



**Free**

The basics of GitHub for every developer

**\$0**  
per month

**Includes:**

- ∞ Unlimited public and private repositories
- ✓ 3 collaborators for private repositories
- ✓ Issues and bug tracking
- ✓ Project management

Are you a [student](#)? Get access to the best developer tools for free with the [GitHub Student Developer Pack](#).



## Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.  
An email containing verification instructions was sent to **phenix235@hanyang.ac.kr**.

# Github 시작하기

## 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

<https://github.com/phenix0805/hello-git.git>

Owner

 phenix0805 ▾

Repository name \*

hellogit ✓

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

Description (optional)

test|



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

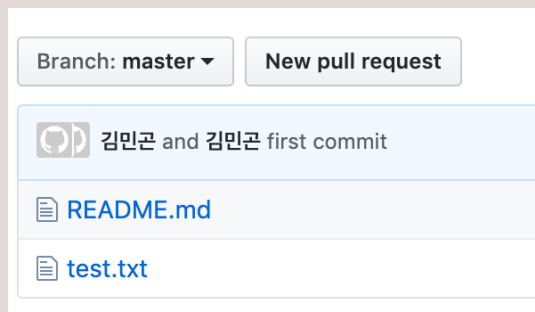
Add a license: **None** ▾



Create repository

# Github 시작하기

1. `echo "test" >> test.txt` : test.txt 라는 파일을 만들려고 하는데 이 파일의 내용은 test
2. `git init` : git 저장소를 생성
3. `git add test.txt` : test.txt라는 파일을 커밋 영역에 넣어둘 준비 (작업공간과 저장소 사이의 대기실 index 영역 또는, stage 영역)
4. `git commit -m "first commit"` : 지금 대기실(index or stage area)에 있는 파일(들)을 커밋장소에 들어가게되는데 지금 first commit 라고 지칭
5. `git remote add test https://github.com/phenix0805/hellogit.git` : 이 저장소 (https://github.com/phenix0805/hellogit.git)의 별칭을 test로 지정, clone시 test라는 이름을 확인 할 수 있음
6. `git push -u test master` : 커밋상태에 있는 내용들을 원격저장소로 밀어보냄
7. Github에 push 확인





# Github 시작하기

```
(base) gimmingon-ui-MacBookPro:testgit phenix$ git add test.txt  
(base) gimmingon-ui-MacBookPro:testgit phenix$ git commit -m "first commit"
```

```
[master 20f554e] first commit
```

```
[Committer: 김민곤 <phenix@gimmingon-ui-MacBookPro.local>
```

```
Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.  
You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 test.txt
```

```
(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote add test https://github.com/phenix0805/hellogit.git
```

```
(base) gimmingon-ui-MacBookPro:testgit phenix$ git push -u test +master
```

```
Enumerating objects: 5, done.
```

```
Counting objects: 100% (5/5), done.
```

```
Delta compression using up to 4 threads
```

```
[Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (5/5), 483 bytes | 241.00 KiB/s, done.
```

```
Total 5 (delta 0), reused 0 (delta 0)
```

```
To https://github.com/phenix0805/hellogit.git
```

```
+ 005e4b9...20f554e master -> master (forced update)
```

```
Branch 'master' set up to track remote branch 'master' from 'test'.
```

# Github 명령어

- **git remote (저장소 URL 설정하기)** : git remote는 git url을 설정할 때 주로 사용.

```
[(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote -v
hello      https://github.com/phenix0805/hellogit.git (fetch)
hello      https://github.com/phenix0805/hellogit.git (push)
origin     https://github.com/phenix0805/hellogit/test.git (fetch)
origin     https://github.com/phenix0805/hellogit/test.git (push)
test       https://github.com/phenix0805/hellogit.git (fetch)
test       https://github.com/phenix0805/hellogit.git (push)
```

# Github 명령어

- **git remote add (name) (저장소 URL 설정하기)** : git remote add로 git url을 추가할 때 주로 사용.

```
[(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote add test2 https://git.test2.com/hello.git  
[(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote -v  
hello      https://github.com/phenix0805/hello.git (fetch)  
hello      https://github.com/phenix0805/hello.git (push)  
origin     https://github.com/phenix0805/hello/test.git (fetch)  
origin     https://github.com/phenix0805/hello/test.git (push)  
test       https://github.com/phenix0805/hello.git (fetch)  
test       https://github.com/phenix0805/hello.git (push)  
test2      https://git.test2.com/hello.git (fetch)  
test2      https://git.test2.com/hello.git (push)
```

# Github 명령어

- **git remote set-url (name) (url 설정하기)** : git remote set-url로 git url을 수정할 때 주로 사용.

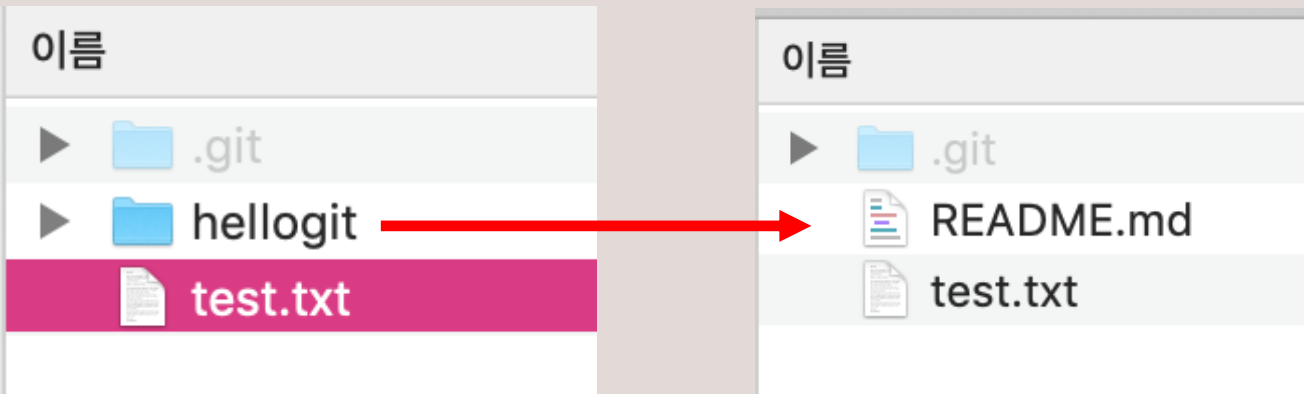
```
(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote set-url hello https://github.com/phenix0805/opensource.git
(base) gimmingon-ui-MacBookPro:testgit phenix$ git remote -v
hello      https://github.com/phenix0805/opensource.git (fetch)
hello      https://github.com/phenix0805/opensource.git (push)
origin     https://github.com/phenix0805/hellogit/test.git (fetch)
origin     https://github.com/phenix0805/hellogit/test.git (push)
test       https://github.com/phenix0805/hellogit.git (fetch)
test       https://github.com/phenix0805/hellogit.git (push)
```



# Github 명령어

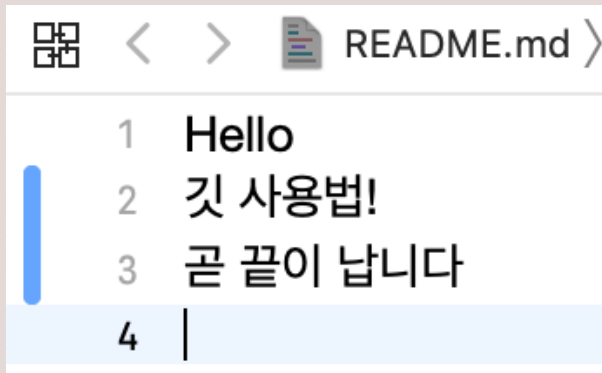
- **git clone** : github 저장소 가져오기

```
(base) gimmingon-ui-MacBookPro:testgit phenix$ git clone https://github.com/phenix0805/hellogit.git  
Cloning into 'hellogit'...  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0  
Unpacking objects: 100% (5/5), done.
```



# Github 명령어

- **git status** : 현재 저장소에 있는 내용과 로컬 컴퓨터에 있는 내용이 얼마나 다른지에 대하여 알려주는 명령어



```
(base) gimmingon-ui-MacBookPro:hellogit phenix$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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

# Github 명령어

- **git status** : 현재 저장소에 있는 내용과 로컬 컴퓨터에 있는 내용이 얼마나 다른지에 대하여 알려주는 명령어

```
(base) gimmingon-ui-MacBookPro:hellogit phenix$ git add .
(base) gimmingon-ui-MacBookPro:hellogit phenix$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

       modified:   README.md
```

# Github 명령어

- **git status** : 현재 저장소에 있는 내용과 로컬 컴퓨터에 있는 내용이 얼마나 다른지에 대하여 알려주는 명령어

```
(base) gimmingon-ui-MacBookPro:hellogit phenix$ git commit -m "modify README"
[master d1bcefc] modify README
Committer: 김민곤 <phenix@gimmingon-ui-MacBookPro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

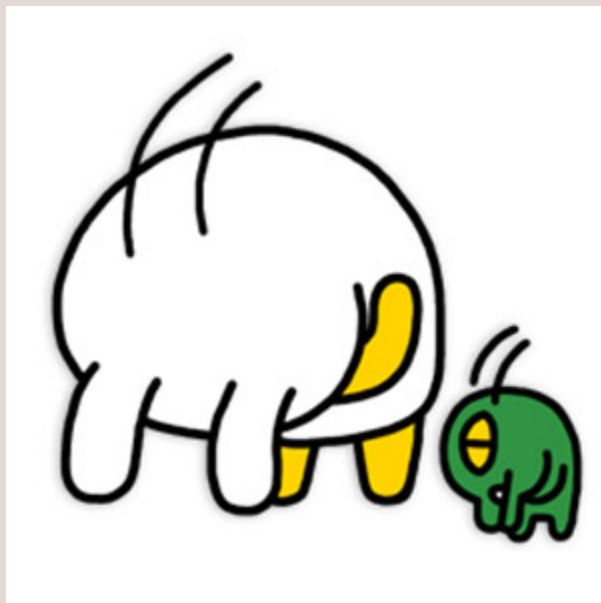
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+)
```





끝