

# Tutor . 김민정

## 멋쟁이 사자처럼 at 이화여대

### Git Basic 5 / 8 (화)



LIKE LION

# Part 1 . VCS



LIKE LION

---

버전 관리 시스템

Version Control System



LIKE LION

---

최종\_시안.png

최종\_시안\_(1).png

진짜최종.png

Final.png

이게\_마지막\_시아ㄴ.png

이게\_진짜\_끝.png

끝.png

최종(2).png

진짜진짜\_진짜\_최종.png

위에거\_다아니고\_이게\_최종.png



LIKE LION

---

기능 개선, 버그 수정, SW 커스터마이징



LIKE LION

---

---

기능 개선, 버그 수정, SW 커스터마이징

의미 있는 변화 저장



LIKE LION

---

# 1. 프로젝트의 변경 관리 : 코드 관리

최종\_시안.png

최종\_시안\_(1).png

진짜최종.png

Final.png

이게\_마지막\_시아ㄴ.png

이게\_진짜\_끝.png

끝.png

최종(2).png

진짜진짜\_진짜\_최종.png

위에거\_다아니고\_이게\_최종.png



LIKE LION

## 2. 프로젝트의 버전 관리 : 되돌아가기

최종\_시안.png

최종\_시안\_(1).png

진짜최종.png

Final.png

이게\_마지막\_시아ㄴ.png

이게\_진짜\_끝.png

끝.png

최종(2).png

진짜진짜\_진짜\_최종.png

위에거\_다아니고\_이게\_최종.png



앗?! 이 버전으로 다시 돌아가서 작업할래!



LIKE LION



### 3. 프로젝트의 협업 : 팀 프로젝트

최종\_시안.png

최종\_시안\_(1).png

진짜최종.png

Final.png

이게\_마지막\_시아ㄴ.png

이게\_진짜\_끝.png

끝.png

최종(2).png

진짜진짜\_진짜\_최종.png

위에거\_다아니고\_이게\_최종.png

민정

서영

준선



LIKE LION



소스 코드의 변화를 관리

Version control system



LIKE LION

근데 아기사자들이 가입한 사이트는?

튜터들이 ★ 달라고 찡찡거린 사이트는?!



LIKE LION



소스 코드의 변화를 웹에서도 관리

Hosting service for git repositories



LIKE LION



버전관리 tool



Git을 사용하는 project를 위한 Service



LIKE LION



명령어를 이용해서 코드들의 버전들을 관리하고!

이런 버전들의 모음을  GitHub에 올려서 공유한다.



LIKE LION

---

<https://github.com/>



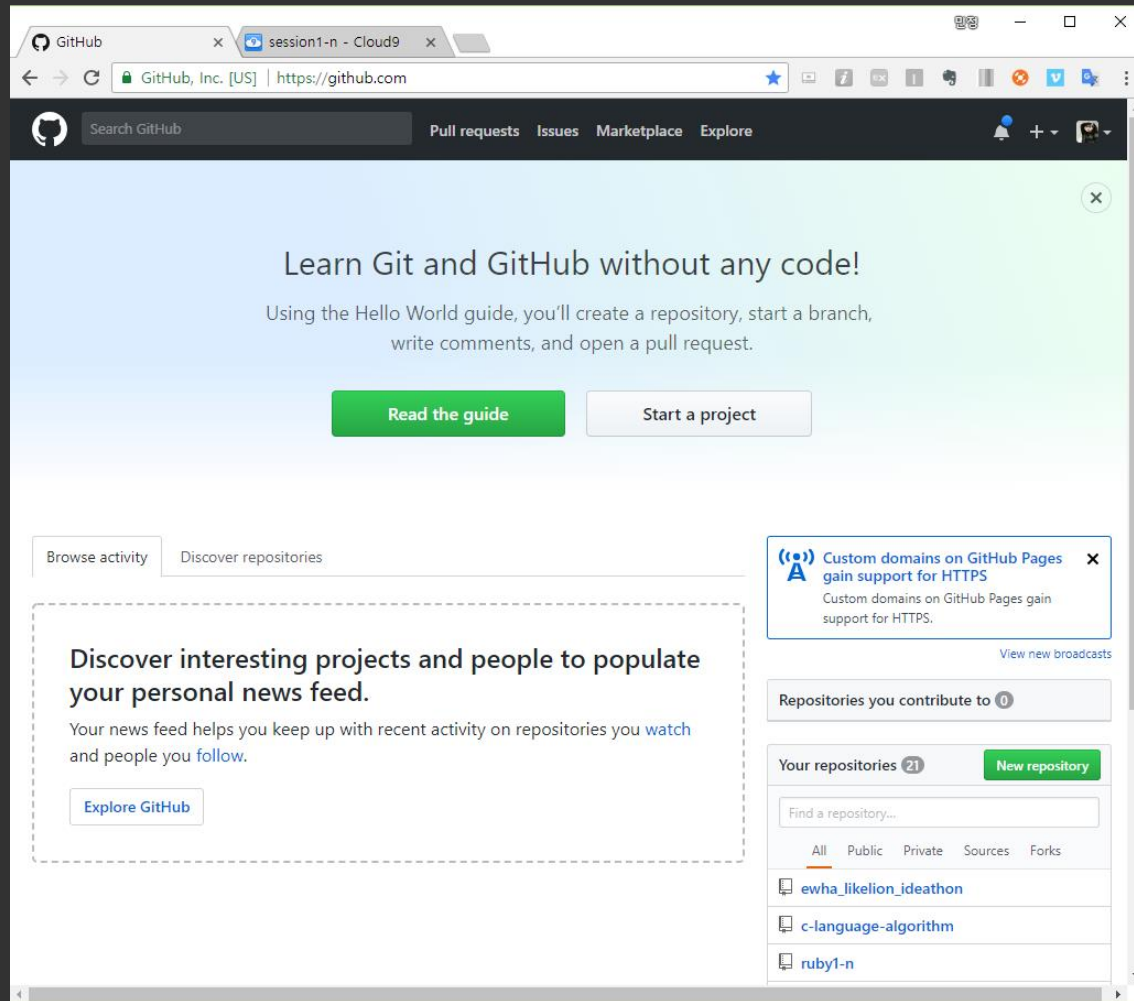
간단한 사용법을 알아보자!



LIKE LION

---

# 1. 저장소 만들기 (repository)



LIKE LION



# 1. 저장소 만들기 (repository)

The image shows a screenshot of the GitHub homepage. Two red rectangular boxes are overlaid on the page, highlighting specific sections. The left box highlights the 'Your repositories' section, which includes a search bar, filters for 'All', 'Public', 'Private', 'Sources', and 'Forks', and a list of repositories: 'ewha\_likelion\_ideathon', 'c-language-algorithm', and 'ruby1-n'. The right box highlights the 'Repositories you contribute to' section, which shows a count of 0 and a 'New repository' button. Both boxes also contain a notification banner about 'Custom domains on GitHub Pages gain support for HTTPS'.

Custom domains on GitHub Pages gain support for HTTPS

Custom domains on GitHub Pages gain support for HTTPS.

View new broadcasts

Repositories you contribute to 0

Your repositories 21

New repository

Find a repository...

All Public Private Sources Forks

ewha\_likelion\_ideathon

c-language-algorithm

ruby1-n

Start a project

Hub without any code!

You'll create a repository, start a branch, and open a pull request.

Repositories you watch

ewha\_likelion\_ideathon

c-language-algorithm

ruby1-n



LIKE LION

# 1. 저장소 만들기 (repository)



repository



C9의 workspace

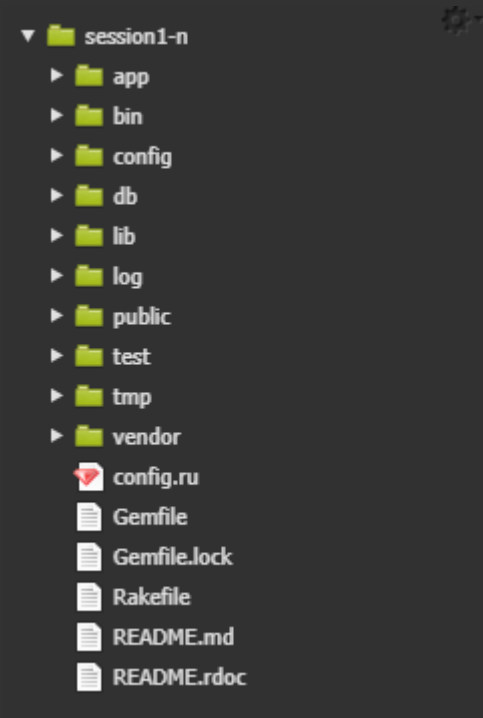


LIKE LION

# 1. 저장소 만들기 (repository)



repository



Project의 directory



LIKE LION

# 1. 저장소 만들기 (repository)

즉! 하나의 프로젝트 안에 들어가는  
소스코드들의 **최상위 폴더!**



LIKE LION

# 1. 저장소 만들기 (repository)

The image shows a screenshot of the GitHub homepage. Two red rectangular boxes are overlaid on the page. The left box highlights the 'New repository' button (a green button with white text) and the list of repositories under 'Your repositories' (21). The right box highlights the same 'New repository' button and repository list, but it is slightly offset to the right. A red arrow points from the 'New repository' button in the left box to the 'New repository' button in the right box. The background shows the GitHub homepage with a navigation bar, a hero section with the text 'Hub without any code!', and a sidebar with a search bar and filters.

Custom domains on GitHub Pages gain support for HTTPS

Custom domains on GitHub Pages gain support for HTTPS.

View new broadcasts

Repositories you contribute to 0

Your repositories 21

New repository

Find a repository...

All Public Private Sources Forks

ewha\_likelion\_ideathon

c-language-algorithm

ruby1-n

Hub without any code!

You'll create a repository, start a branch, and open a pull request.

Start a project

Repositories you watch

ewha\_likelion\_ideathon

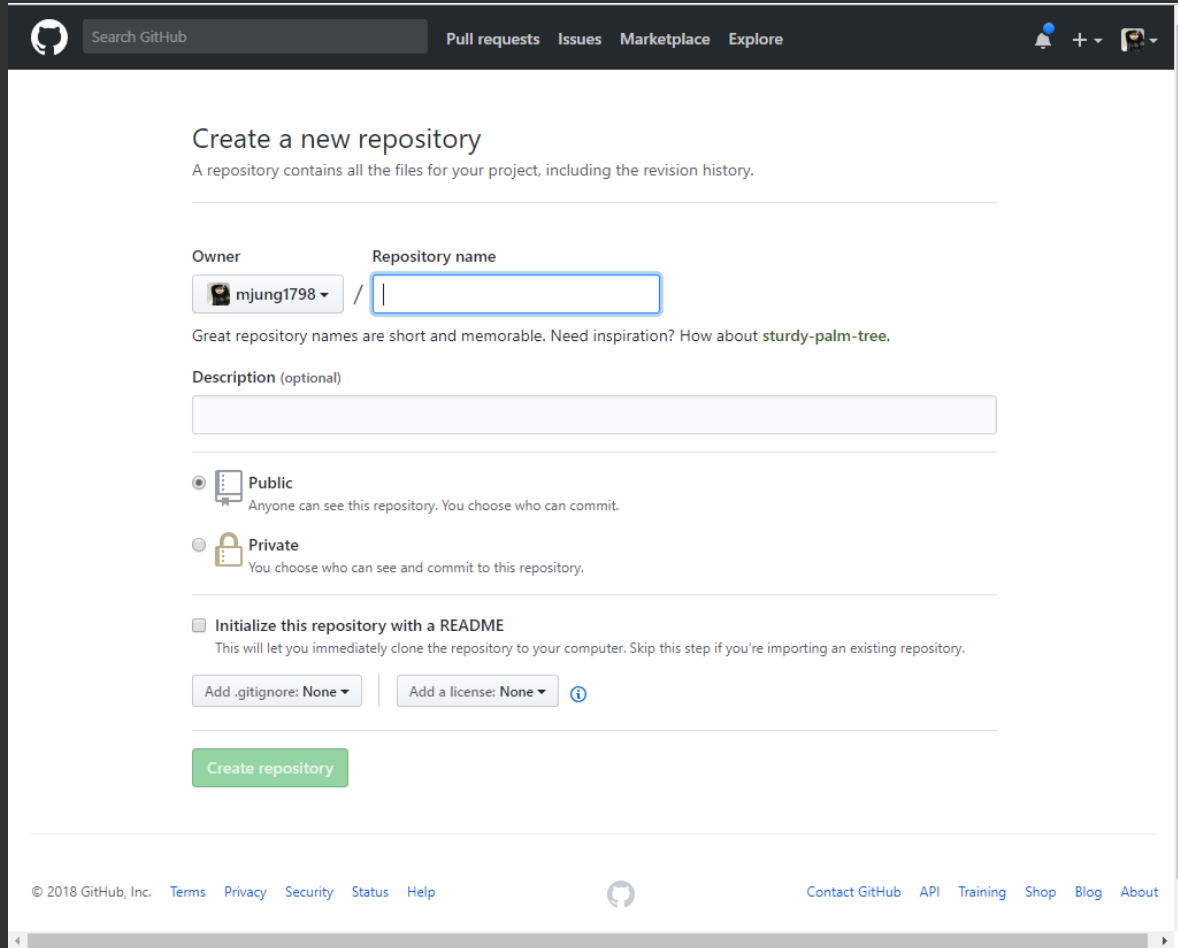
c-language-algorithm

ruby1-n



LIKE LION

# 1. 저장소 만들기 (repository)



The screenshot shows the GitHub interface for creating a new repository. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. The main heading is 'Create a new repository' with a subtext: 'A repository contains all the files for your project, including the revision history.' Below this, there are two input fields: 'Owner' (showing 'mjung1798') and 'Repository name' (an empty box). A hint text says: 'Great repository names are short and memorable. Need inspiration? How about sturdy-palm-tree.' There's a 'Description (optional)' text area. Under 'Visibility', 'Public' is selected with the note 'Anyone can see this repository. You choose who can commit.' 'Private' is also an option with the note 'You choose who can see and commit to this repository.' There's a checkbox for 'Initialize this repository with a README' with the note 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' Below these are dropdowns for 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the bottom. The footer contains copyright information, links to Terms, Privacy, Security, Status, Help, and a GitHub logo, followed by links to Contact GitHub, API, Training, Shop, Blog, and About.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mjung1798 / Repository name:

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

Description (optional):

☒ 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

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About




LIKE LION

# 1. 저장소 만들기 (repository)



Repository name

 mjung1798 / github\_practice ✓

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

Description (optional)

0508 깃허브 세션 (git\_basic)





Workspace name	Description
github_practice	git hub   세션 준비



LIKE LION


# 1. 저장소 만들기 (repository)

☒  **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.

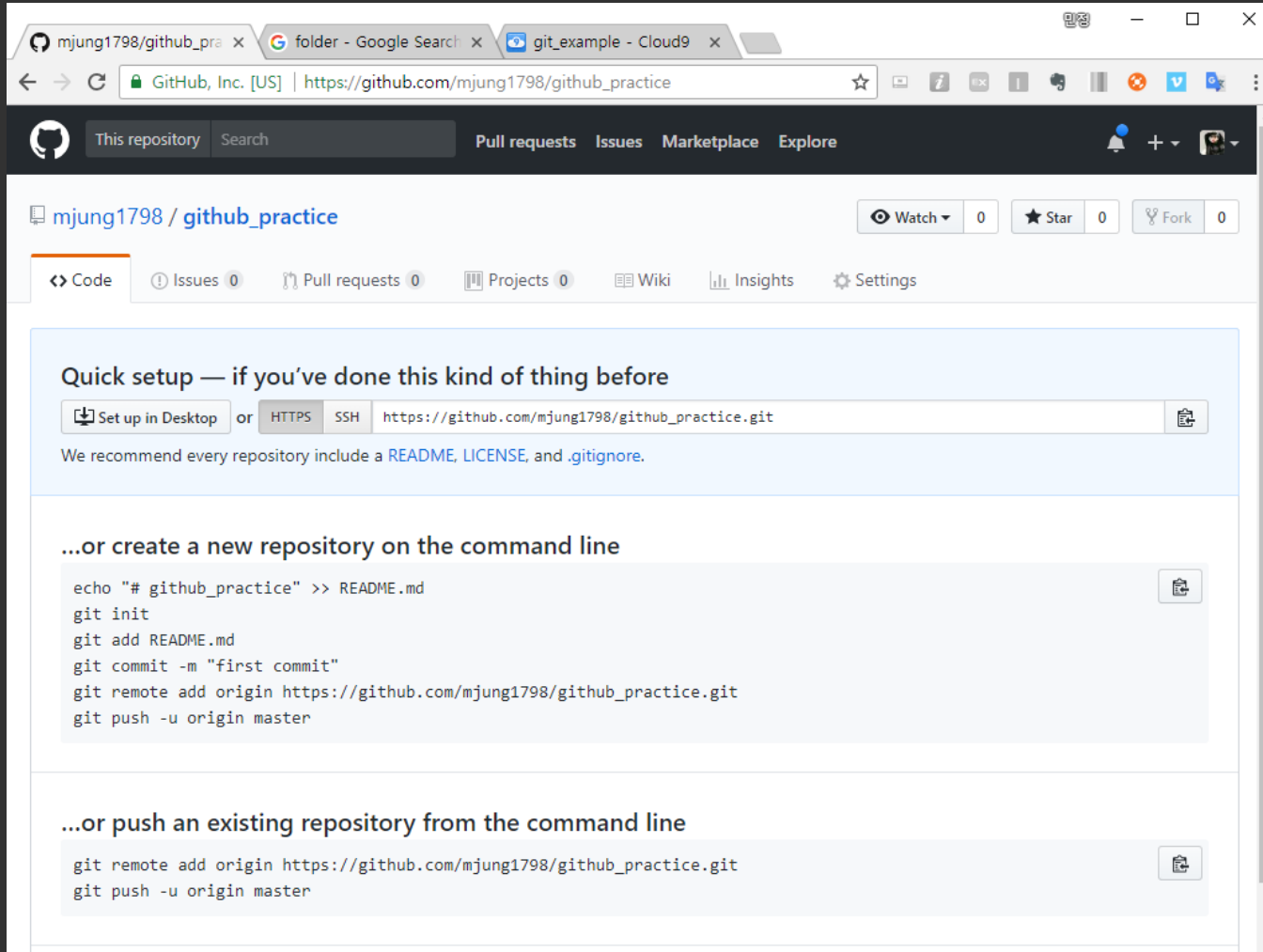
Add .gitignore: **None** ▼ | Add a license: **None** ▼ 



LIKE LION



# 1. 저장소 만들기 (repository)



The screenshot shows the GitHub web interface for a repository named 'mjung1798 / github\_practice'. The browser tabs include 'mjung1798/github\_pre', 'folder - Google Search', and 'git\_example - Cloud9'. The address bar shows the URL 'https://github.com/mjung1798/github\_practice'. The repository page features a navigation bar with 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the repository name 'mjung1798 / github\_practice' is displayed, along with 'Watch' (0), 'Star' (0), and 'Fork' (0) buttons. A tab bar shows 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area provides 'Quick setup' instructions for users who have done this before, offering options to 'Set up in Desktop', 'HTTPS', or 'SSH' with the repository URL. It also includes a recommendation to include a README, LICENSE, and .gitignore. Below this, it shows how to 'create a new repository on the command line' and 'push an existing repository from the command line' with specific git commands.

Quick setup — if you've done this kind of thing before

or   `https://github.com/mjung1798/github_practice.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# github_practice" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mjung1798/github_practice.git
git push -u origin master
```

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

```
git remote add origin https://github.com/mjung1798/github_practice.git
git push -u origin master
```



LIKE LION

# 1. 저장소 만들기 (repository)

나의 프로젝트 (c9에 있는 소스파일들)  
Github의 repository에 올리는 방법!

bash에 입력!



LIKE LION

# 1. 저장소 만들기 (repository)

1:N or CRUD 프로젝트를  
Github에 올려보자!



LIKE LION

**CRUD 프로젝트를  
Github에 올려보자!**



**LIKE LION**

---

**git init**

**git add .**

**git commit -m "first commit"**

**git remote add origin 주소**

**git push -u origin master**



**LIKE LION**

---

```
mjung1798:~/workspace $ git init
Initialized empty Git repository in /home/ubuntu/workspace/.git/
```

```
mjung1798:~/workspace (master) $ git add .
```


```
mjung1798:~/workspace (master) $ git commit -m "first commit"
[master (root-commit) 257f25f] first commit
73 files changed, 1396 insertions(+)
create mode 100644 .gitignore
create mode 100644 Gemfile
create mode 100644 Gemfile.lock
create mode 100644 README.md
create mode 100644 README.rdoc
create mode 100644 Rakefile
create mode 100644 app/assets/images/.keep
create mode 100644 app/assets/javascripts/application.js
create mode 100644 app/assets/javascripts/posts.coffee
create mode 100644 app/assets/stylesheets/application.css
create mode 100644 app/assets/stylesheets/posts.scss
create mode 100644 app/controllers/application_controller.rb
create mode 100644 app/controllers/concerns/.keep
create mode 100644 app/controllers/posts_controller.rb
create mode 100644 app/helpers/application_helper.rb
create mode 100644 app/helpers/posts_helper.rb
create mode 100644 app/mailers/.keep
create mode 100644 app/models/.keep
create mode 100644 app/models/concerns/.keep
create mode 100644 app/models/post.rb
create mode 100644 app/views/layouts/application.html.erb
```



# LIKE LION

```
mjung1798:~/workspace (master) $ git remote add origin https://github.com/mjung1798/github_practice.git
```

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS SSH

[https://github.com/mjung1798/github\\_practice.git](https://github.com/mjung1798/github_practice.git)



We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

주소는 여기서 복사!

```
mjung1798:~/workspace (master) $ git push -u origin master
Username for 'https://github.com': mor2222@naver.com
Password for 'https://mor2222@naver.com@github.com':
Counting objects: 85, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (77/77), done.
Writing objects: 100% (85/85), 22.99 KiB | 0 bytes/s, done.
Total 85 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/mjung1798/github_practice.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

자신의 github 이메일 password 입력하기!



LIKE LION

Browser tabs: mjung1798/github\_pr... x unilion\_work - Cloud9 x

Address bar: GitHub, Inc. [US] | https://github.com/mjung1798/github\_practice

Repository: mjung1798 / github\_practice

Actions: Watch 0, Star 0, Fork 0

Navigation: Code, Issues 0, Pull requests 0, Projects 0, Wiki, Insights, Settings

Repository name: 0508 깃허브 세션 (git\_basic) Edit

Add topics

Stats: 1 commit, 1 branch, 0 releases, 1 contributor

**⚠ We found potential security vulnerabilities in your dependencies.** Dismiss

Some of the dependencies defined in `./Gemfile.lock` have known security vulnerabilities and should be updated.

[Review vulnerable dependencies](#)

Only the owner of this repository can see this message.  
[Learn more about vulnerability alerts](#)

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

mjung1798 first commit		Latest commit 257f25f 8 minutes ago
app	first commit	8 minutes ago
bin	first commit	8 minutes ago
config	first commit	8 minutes ago
db	first commit	8 minutes ago
lib	first commit	8 minutes ago
log	first commit	8 minutes ago
public	first commit	8 minutes ago
test	first commit	8 minutes ago





1:n 에서 튜터가 만들어둔 project를  
복사해서 수정했던 것처럼!



LIKE LION

이제 내 project의 진행사항을 web (github) 에  
저장했으니 코드가 날라가도 괜찮아!



LIKE LION

근데 이게 무슨 명령어지?



LIKE LION

---

**git init**

**git add .**

**git commit -m "first commit"**

**git remote add origin 주소**

**git push -u origin master**



**LIKE LION**

---

## 2. 깃 저장소 초기화 (init)

```
$ git init
```

```
# Starting a working area
```

```
# 현재 작업장에 작업을 진행하겠다는 것을 git에게 알려줘!
```

```
# 이 명령어 입력 전까지는 일반 폴더! git과 관계가 없다.
```

```
# git init 이후에야 추가적인 깃 명령어들을 줄 수 있다.
```



LIKE LION

---

git init

git add .

git commit -m "first commit"

git remote add origin 주소

git push -u origin master



LIKE LION

---

### 3. Commit 대기 (add)

```
$ git add .
```

# commit의 대기상태

# 내가 commit하고 싶은, 버전에 반영하고 싶은 파일만! Add

# commit 전에 add가 우선!

# add index.html.erb      => 이 파일만 commit 대기

# add .                      => 모든 파일 commit 대기



LIKE LION

---

```
git init
```

```
git add .
```

```
git commit -m "first commit"
```

```
git remote add origin 주소
```

```
git push -u origin master
```



LIKE LION

---



## 4. 버전 만들기 (commit)

```
$ git commit -m 'first commit'
```

# add 파일 (변경 내용)을 first commit이란 버전이름 설정

# commit하면서 기억하고 싶은 message를 함께 적자!

# 스냅 샷!

```
$ git commit -m '내용'
```



LIKE LION

---

git init

git add .

git commit -m "first commit"

git remote add origin 주소

git push -u origin master



LIKE LION

---

## 5. 업로드할 github 주소! (remote)

\$ git remote add origin 주소

# 현재 작업하는 workspace를 연결한다!

# 나의 github repository에!

# 그 github repository에 origin이란 별명을 부여!

내 workspace와 주로 동기화하는 메인 repository를  
관습적으로 origin이라고 부른다



LIKE LION

git init

git add .

git commit -m "first commit"

git remote add origin 주소

git push -u origin master



LIKE LION

## 6. Github에 올리기! (push)

```
$ git push -u origin master
```

# 내가 아까 remote한 repository는 origin!

# 이 origin을 push!

# 즉 나의 repository의 내용을 업로드 하겠다!



LIKE LION

## 2. 깃 저장소 초기화 (add)

```
$ git add .
```

# commit의 대기상태

# 내가 commit하고 싶은, 버전에 반영하고 싶은 파일만! Add

# commit 전에 add가 우선!

# add index.html.erb      => 이 파일만 commit 대기

# add .                    => 모든 파일 commit 대기

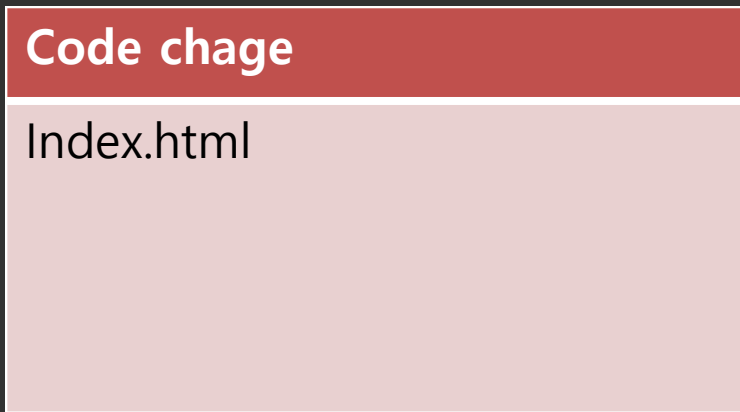
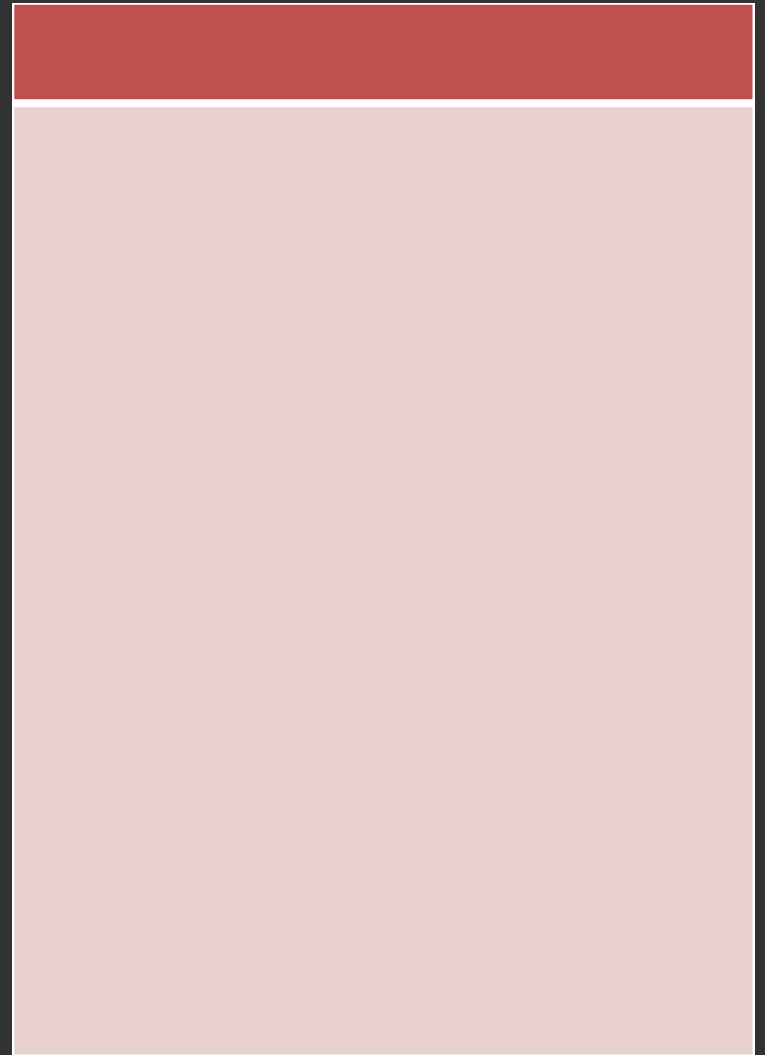


LIKE LION

무슨 소리지 ...



LIKE LION



LIKE LION



**git add**

Index.html



\$git add index.html

**Code chage**

**git commit**



**LIKE LION**

git add

histroy (git commit)

"index 수정!"



git commit -m "index 수정!"

Code chage



LIKE LION

**git add**

**histroy (git commit)**

"index 수정!"

**Code chage**

"create.html.erb"  
"home\_controller.rb"



**LIKE LION**

## git add

```
"create.html.erb"  
"home_controller.rb"
```



\$git add .

## Code chage

## histroy (git commit)

"index 수정!"



LIKE LION

git add

histroy (git commit)

"CRUD의 C완료!"  
"index 수정!"



git commit -m "CRUD의 C완료!"

Code chage



LIKE LION

그러면 한번에 commit하면  
되는거아냐?? add를 왜 해?



LIKE LION

한 파일에 1commit  
내가 add한 파일들에 1commit



LIKE LION

---

# git add

버전에 포함시킬 파일과 포함시키지 않을  
파일을 구분짓기 위해서!



LIKE LION

---



**git add**

**histroy (git commit)**

"CRUD의 C완료!"  
"index 수정!"

**Code chage**

index.html.erb  
edit.html.erb  
home\_controller.rb  
comment\_controller.rb



**LIKE LION**

## git add

index.html.erb  
edit.html.erb  
home\_controller.rb



\$git add index.html.rb edit.html.erb home\_controller.rb

## Code chage

comment\_controller.rb

## histroy (git commit)

"CRUD의 C완료!"  
"index 수정!"



LIKE LION

**git add**

**histroy (git commit)**

"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"

**Code chage**

comment\_controller.rb



**LIKE LION**

**git add**

**histroy (git commit)**

"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"

**Code chage**

comment\_controller.rb



**LIKE LION**

**git add**

comment\_controller.rb



\$git add comment\_controller.rb

**Code chage**

**histroy (git commit)**

"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"



**LIKE LION**

git add

histroy (git commit)

"1:n 완료!"  
"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"

git commit -m "1:n 완료!"

Code chage



LIKE LION

큰 작업에 따라 내가 add시키는  
파일에 따라서 버전을 분리!



LIKE LION

remote, push



LIKE LION



내가 commit한 내용을  
github에도 올릴래!



LIKE LION

## histroy (git commit)

"1:n 완료!"  
"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"



## Repository

- Html
- Rails
- Bootstrap
- **Board**

\$ git remote add origin board

이 내용을 담은 repository는 Board야!!  
이 board의 별명을 origin이라고 해줘!



## histroy (git commit)

"1:n 완료!"  
"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"



## Repository

- Html
- Rails
- Bootstrap
- **Board : origin**

\$ git push -u origin master

나의 commit내용을 repository에 업로드!

## Board

"1:n 완료!"  
"CRUD 완료!"  
"CRUD의 C완료!"  
"index 수정!"



## 2. 깃 저장소 초기화 (init)

```
$ git init
```

```
# Starting a working area
```

```
# 현재 작업장에 작업을 진행하겠다는 것을 git에게 알려줘!
```

```
# 이 명령어 입력 전까지는 일반 폴더! git과 관계가 없다.
```

```
# git init 이후에야 추가적인 깃 명령어들을 줄 수 있다.
```



LIKE LION

### 3. Commit 대기 (add)

`$ git add .`

# commit의 대기상태

# 내가 commit하고 싶은, 버전에 반영하고 싶은 파일만! Add

# commit 전에 add가 우선!

# add index.html.erb      => 이 파일만 commit 대기

# add .                      => 모든 파일 commit 대기



LIKE LION

## 4. 버전 만들기 (commit)

```
$ git commit -m 'first commit'
```

# add 파일 (변경 내용)을 first commit이란 버전이름 설정

# commit하면서 기억하고 싶은 message를 함께 적자!

# 스냅 샷!

```
$ git commit -m '내용'
```



LIKE LION

## 5. 업로드할 github 주소! (remote)

\$ git remote add origin 주소

# 현재 작업하는 workspace를 연결한다!

# 나의 github repository에!

# 그 github repository에 origin이란 별명을 부여!

내 workspace와 주로 동기화하는 메인 repository를  
관습적으로 origin이라고 부른다



LIKE LION

## 6. Github에 올리기! (push)

```
$ git push -u origin master
```

# 내가 아까 remote한 repository는 origin!

# 이 origin을 push!

# 즉 나의 repository의 내용을 업로드 하겠다!



LIKE LION



[실 습]



LIKE LION

새로운 workspace와  
Repository 만들자!



LIKE LION





Create a New Repository

Mjung

GitHub, Inc. [US] | https://github.com/new

Search GitHub

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


   

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 mjung1798


/

github\_basic ✓


Great repository names are short and memorable. Need inspiration? How about **potential-giggle**.

Description (optional)

github\_basic

☒  **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

© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)



Create a New Repository

Create a New Workspace

← → ↻ 안전함 | https://c9.io/new

Create a new workspace

Workspace name

github\_basic

Description

github\_basic

Team

이대멋사

Hosted workspace

Clone workspace

Remote SSH workspace

Salesforce

☐ Private

This is a workspace for your eyes only

☐ Public

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

HTML5

node.js

PHP, Apache & ...

Python

django

Rails

C++

Wordpress

Rails Tutorial

Blank

Harvard's CS50

Create workspace

The logo for 'LIKE LION' features a stylized green lion head inside a circular frame, followed by the text 'LIKE LION' in a bold, white, sans-serif font.

1. 파일 지우기
2. index.html 파일 만들기
3. \$git
4. \$git init
5. \$git add index.html
6. \$git commit -m "index.html 파일 생성"
7. \$git log : 나의 commit내역 살펴보기!
8. \$git remote add origin 주소
9. \$push -u origin master



10. index.html 수정

11. Git commit

12. Readme md 추가

13. Git commit

14. Git push



# \$Git : 깃 명령어 도움말

```
mjung1798:~/workspace $ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <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
  reset      Reset current HEAD to the specified state
  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
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status


grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG


collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```



# LIKE LION

# \$git log : 나의 commit내역 살펴보기

```
commit 6f980889d6588e380ff91f3f20ea0b74e760fd75 (HEAD -> master)
Author: Mjung <mor2222@naver.com>
Date:   Tue May 8 04:05:49 2018 +0000

    README.md 생성

commit 6d742a565cb025c4c2fa333c523f88d994ab4110
Author: Mjung <mor2222@naver.com>
Date:   Tue May 8 04:01:15 2018 +0000

    index.html body생성

commit 8ca8d69626d6b9de80fd44f8e80014c012031ce2 (origin/master)
Author: Mjung <mor2222@naver.com>
Date:   Tue May 8 03:54:52 2018 +0000

    index.html파일 생성
```

\$q로 나가요!



LIKE LION



\$git log -p : commit사이 달라진 내용!

\$git log -2 : 최근 2개의 commit만

```
mjung1798:~/workspace (master) $ git log -p
commit 6f980889d6588e380ff91f3f20ea0b74e760fd75 (HEAD -> master)
Author: Mjung <mor2222@naver.com>
Date: Tue May 8 04:05:49 2018 +0000

    README.md 생성

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..ac5d2f7
--- /dev/null
+++ b/README.md
@@ -0,0 +1 @@
+commit을 이용한 버전관리
\ No newline at end of file
```

```
commit 6d742a565cb025c4c2fa333c523f88d994ab4110
Author: Mjung <mor2222@naver.com>
Date: Tue May 8 04:01:15 2018 +0000
```

index.html body생성

```
diff --git a/index.html b/index.html
index 5c1870a..052a117 100644
--- a/index.html
+++ b/index.html
@@ -1,3 +1,5 @@
<html>
-
+  <body>
+  [REDACTED]
+  </body>
</html>
\ No newline at end of file
```

```
commit 8ca8d69626d6b9de80fd44f8e80014c012031ce2 (origin/master)
Author: Mjung <mor2222@naver.com>
```



LIKE LION

---

\$git log를 보기 위한 좀더 편한 방법!

<https://git-scm.com/book/en/v2/GitBasics-Viewing-the-Commit-History>



**LIKE LION**

---

# Github에서 확인하기!

The screenshot shows the GitHub interface for the repository 'mjung1798 / github\_basic'. At the top, there's a navigation bar with 'This repository', a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the right, there are icons for notifications, a dropdown menu, and a user profile. Below the navigation bar, the repository name 'mjung1798 / github\_basic' is displayed, followed by 'Watch' (0), 'Star' (0), and 'Fork' (0) buttons. A secondary navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area shows the repository name 'github\_basic' with an 'Edit' button. Below this, a summary bar highlights '3 commits' (boxed in red), '1 branch', '0 releases', and '1 contributor'. A row of buttons includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table follows, with columns for the commit message, the file changed, and the time ago. The first commit is 'mjung1798 README.md 생성' (12 minutes ago), followed by 'README.md 생성' (12 minutes ago) and 'index.html body생성' (17 minutes ago). Below the table, the 'README.md' file content is shown, which reads 'commit을 이용한 버전관리'.

This repository

Search

Pull requests Issues Marketplace Explore

mjung1798 / github\_basic

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

github\_basic Edit

Add topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

commit	file	time
mjung1798 README.md 생성		Latest commit 6f98088 12 minutes ago
README.md	README.md 생성	12 minutes ago
index.html	index.html body생성	17 minutes ago

README.md

commit을 이용한 버전관리



LIKE LION

# Github에서 확인하기!

This repository Search Pull requests Issues Marketplace Explore

mjung1798 / github\_basic Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master

Commits on May 8, 2018

- README.md 생성  
mjung1798 committed 14 minutes ago
- index.html body생성  
mjung1798 committed 18 minutes ago
- index.html파일 생성  
mjung1798 committed 25 minutes ago

6f98088

6d742a5

8ca8d69

이 commit에서 달라진 점 (git log)

이 commit에서의 repository 목록 보기



LIKE LION



<https://git-scm.com/doc>



LIKE LION

수고하셨습니다



LIKE LION