



@Vietnam

3-1주차

Git-Github

Github 배포

CONTENTS

1. Git
2. Github
3. Github 배포

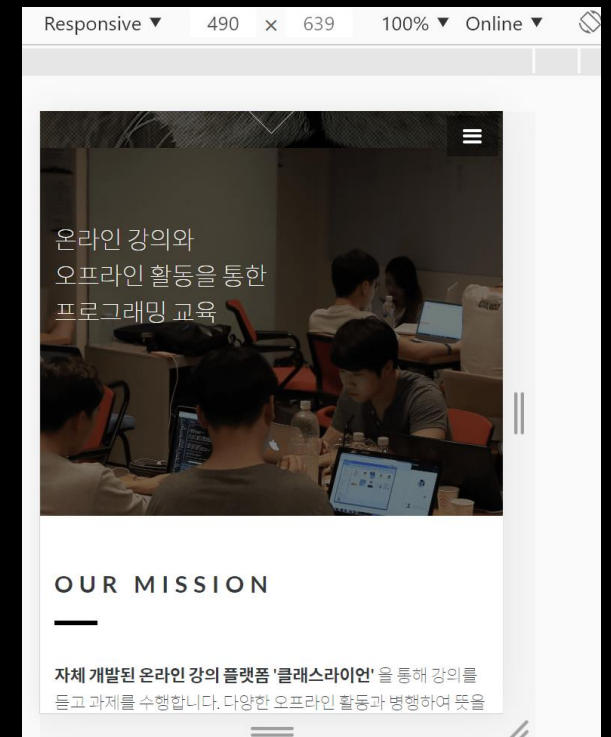
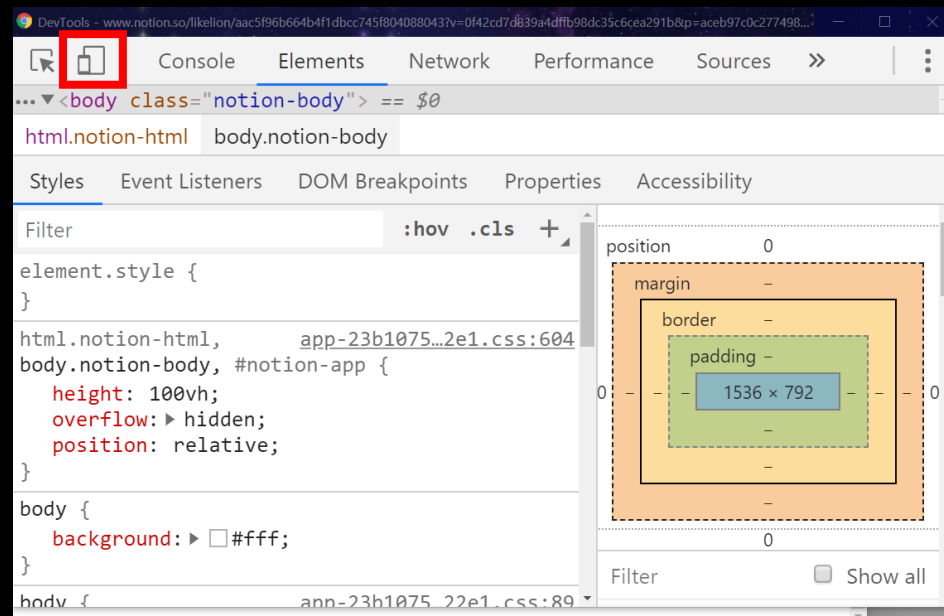
공지사항

- 복습 영상 강의도 같이 올라오니 참고해 주세요!

추가사항

- 반응형 웹 디자인 : 크롬 개발자도구에서 반응형 디자인이 잘 적용 되는 지 확인하는 방법

1. 개발자 도구를 켜다
2. 빨간 사각형 부분을 클릭하고
3. 브라우저 화면에서 원하는 값으로 조정해본다.



이번시간에 배운 내용

- 지금까지 배웠던 CSS에 대한 복습
- CSS에서 반응형 웹 디자인을 활용하여 웹 브라우저의 화면이 변할 때 그에 따라 디자인이 반응적으로 변경되도록 구성

이번시간 학습목표

- 깃(git)과 깃허브(github)라는 도구를 이해
- 깃(git)과 깃허브(github)를 활용해 버전관리하는 방법의 기초를 이해

깃(git)?



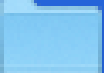

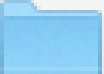


깃 (소프트웨어)

위키백과, 우리 모두의 백과사전.
(Git (소프트웨어)에서 넘어옴)

깃(Git /git/^[5])은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다. 소프트웨어 개발에서 소스 코드 관리에 주로 사용되지만^[6] 어떠한 집합의 파일의 변경사항을 지속적으로 추적하기 위해 사용될 수 있다. 기하학적 불변 이론을 바탕으로 설계됐고, 분산 버전 관리 시스템으로서 빠른 수행 속도에 중점을 두고 있는 것^[7]이 특징이며 데이터 무결성,^[8] 분산, 비선형 워크플로를 지원한다.^[9]

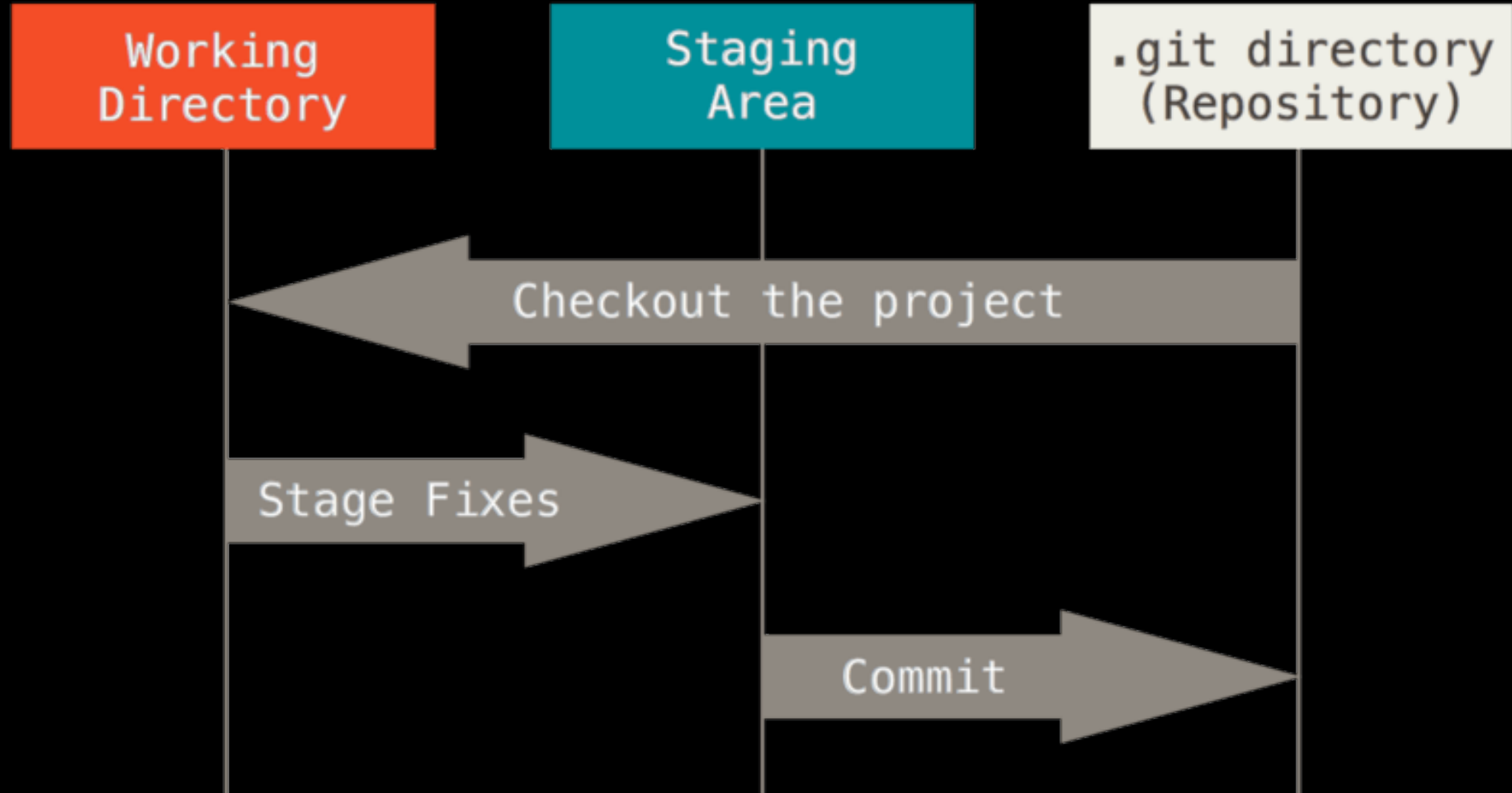


버전관리??

- ▶  bootcamp version final final
- ▶  bootcamp version final
- ▶  bootcamp version edited
- ▶  bootcamp version0.1
- ▶  bootcamp

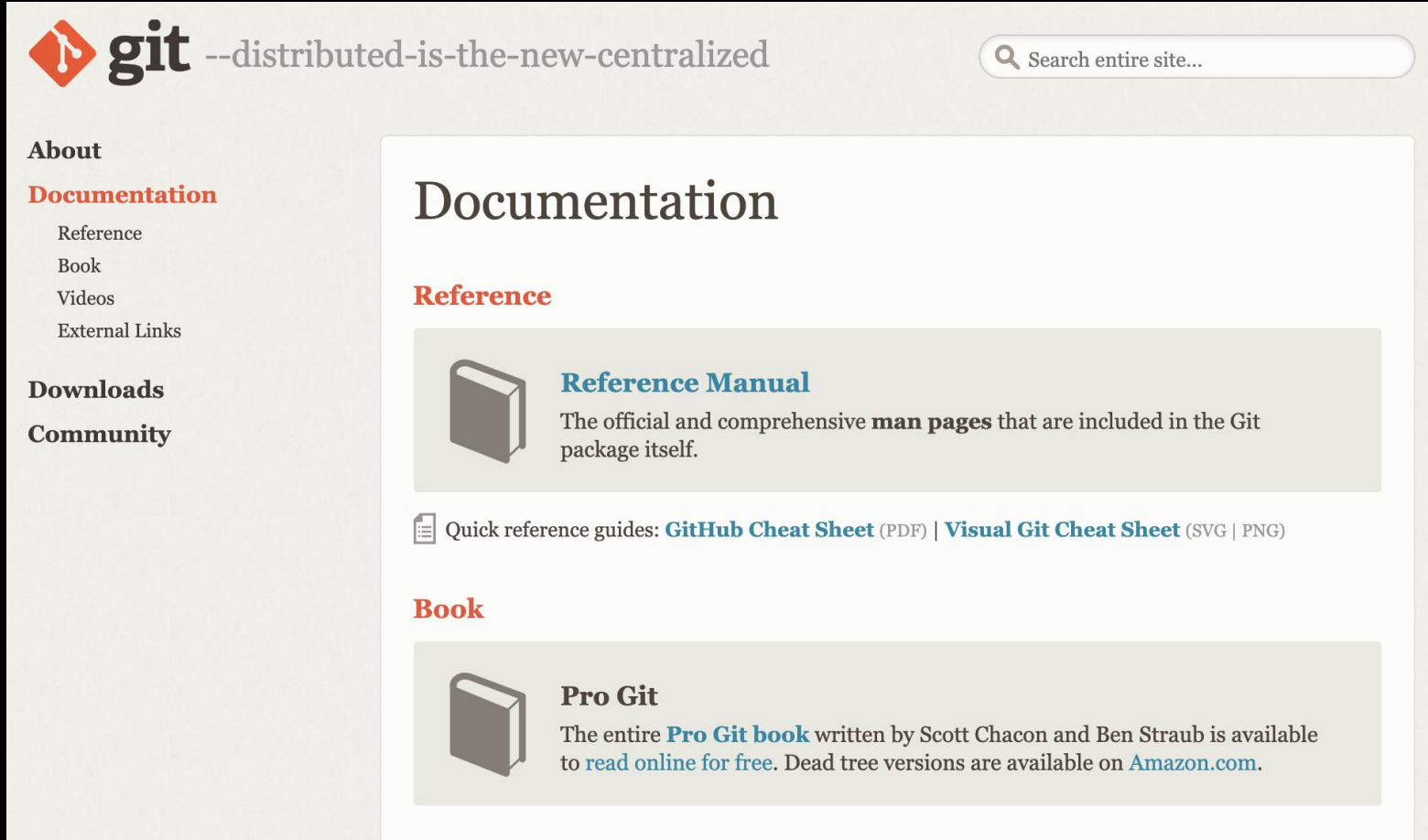
얼마나 많은 “final”파일을 만들어 보셨나요??

깃의 작동방식



깃은 버전관리를 하는데 유용하다.

깃 설치



The screenshot shows the Git documentation website. At the top, the Git logo is followed by the tagline "--distributed-is-the-new-centralized". A search bar is located in the top right corner. On the left side, there is a navigation menu with links for "About", "Documentation" (highlighted in red), "Reference", "Book", "Videos", and "External Links". Below the navigation menu, there are sections for "Downloads" and "Community". The main content area is titled "Documentation" and contains a "Reference" section with a book icon and the text "Reference Manual". Below this, there is a section for "Quick reference guides" with links to "GitHub Cheat Sheet (PDF)" and "Visual Git Cheat Sheet (SVG | PNG)". The "Book" section follows, featuring a book icon and the text "Pro Git".

git --distributed-is-the-new-centralized

Search entire site...

About

Documentation


- Reference
- Book
- Videos
- External Links

Downloads


Community

Documentation


Reference

 **Reference Manual**

The official and comprehensive **man pages** that are included in the Git package itself.

 Quick reference guides: [GitHub Cheat Sheet](#) (PDF) | [Visual Git Cheat Sheet](#) (SVG | PNG)

Book

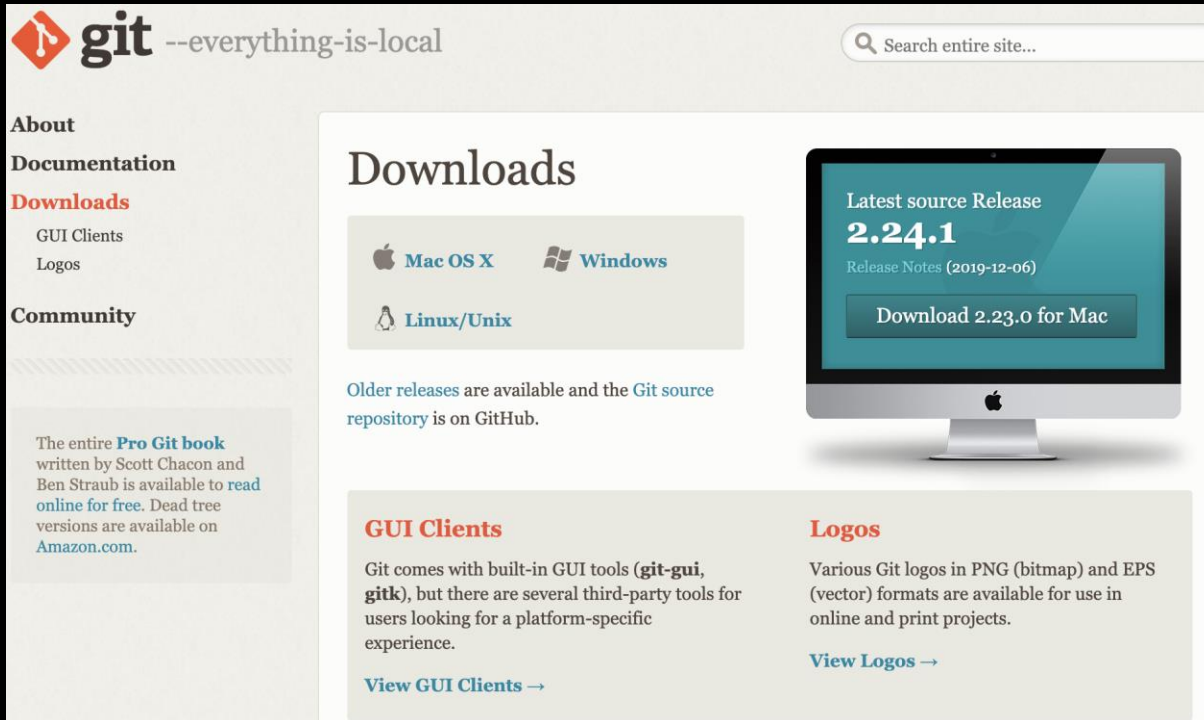
 **Pro Git**

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](#).

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

맥에서 깃 설치 방법

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



The screenshot shows the Git website's Downloads page. The header includes the Git logo and the tagline "--everything-is-local". A search bar is present. The left sidebar has links for "About", "Documentation", "Downloads" (highlighted), "GUI Clients", "Logos", and "Community". The main content area is titled "Downloads" and features a large monitor graphic displaying the "Latest source Release 2.24.1" and a "Download 2.23.0 for Mac" button. Below the monitor, there are sections for "GUI Clients", "Logos", and "View GUI Clients →". The footer mentions the "Pro Git book" and its availability on Amazon.com.

Downloading Git



Your download is starting...

You are downloading version **2.23.0** of Git for the **Mac** platform. This is the most recent **maintained build** for this platform. It was released **3 months ago**, on 2019-09-03.

If your download hasn't started, [click here to download manually](#).

The current source code release is version **2.24.1**. If you want the newer version, you can build it from [the source code](#).

Now What?

Now that you have downloaded Git, it's time to start using it.



Read the Book

Dive into the Pro Git book and learn at your own pace.



Download a GUI

Several free and commercial GUI tools are available for the Mac platform.

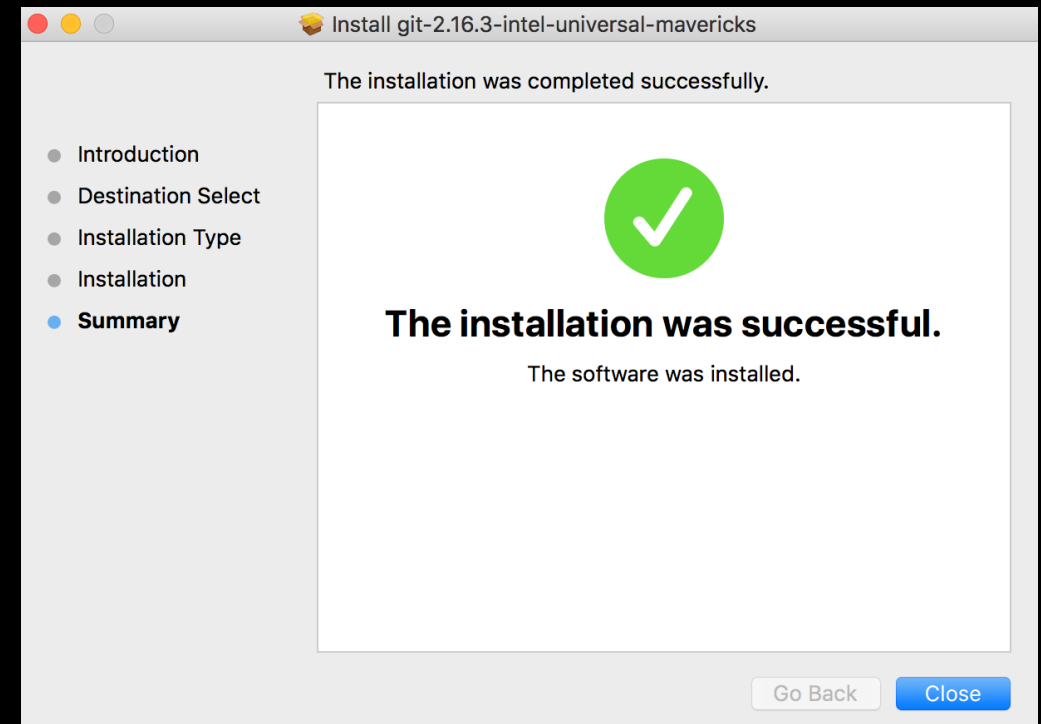
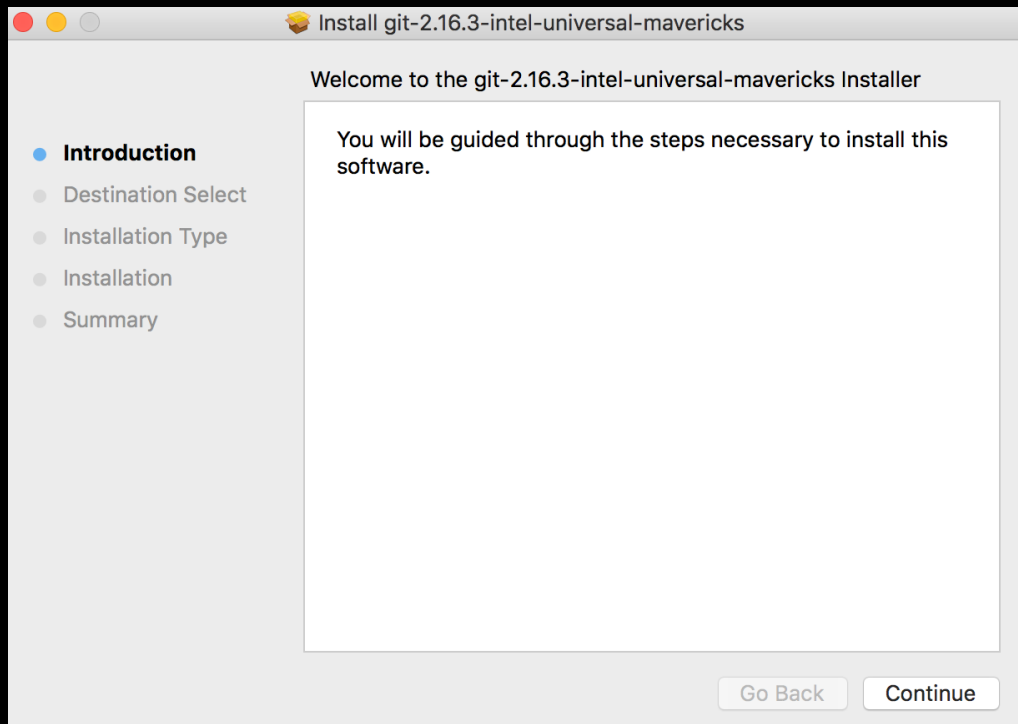


Get Involved

A knowledgeable Git community is available to answer your questions.

깃 설치

맥에서 깃 설치 방법



과정을 따라가보자

맥에서 깃 설치 방법

Command + space -> terminal이라고 입력 후 클릭
'git --version'을 쳐서 깃이 제대로 설치되었는 지 확인해보자.

```
likelion-vietnamui-MacBookPro:~ likelion_vietnam$ git --version  
git version 2.23.0  
likelion-vietnamui-MacBookPro:~ likelion_vietnam$
```

설치가 완료되면, 깃 버전을 확인할 수 있다.

윈도우에서의 깃 설치 방법



Your download is starting...

You are downloading the latest (**2.24.1**) **32-bit** version of **Git for Windows**. This is the most recent **maintained build**. It was released **5 days ago**, on 2019-12-10.

If your download hasn't started, [click here to download manually](#).

Other Git for Windows downloads

Git for Windows Setup

32-bit Git for Windows Setup.

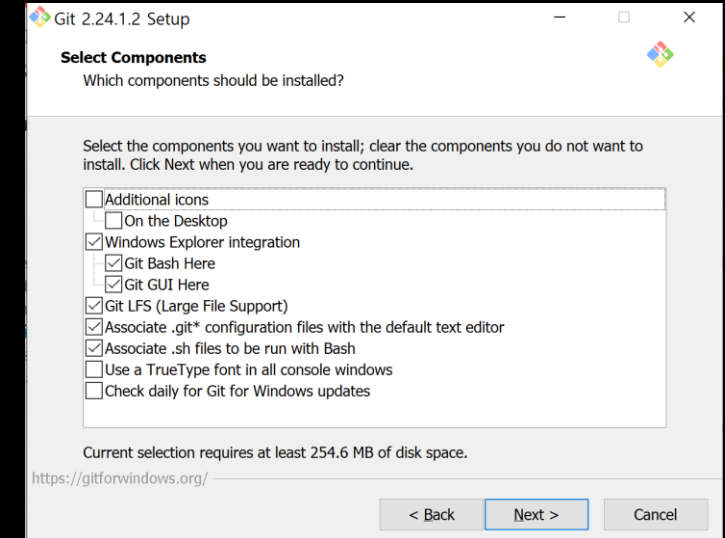
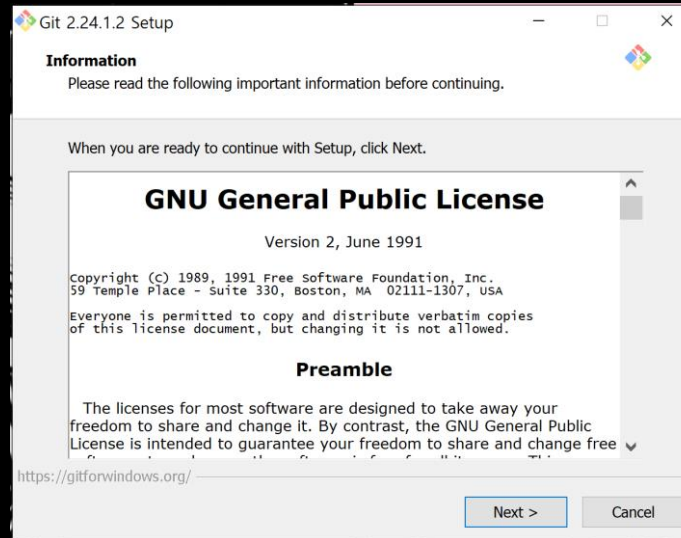
64-bit Git for Windows Setup.

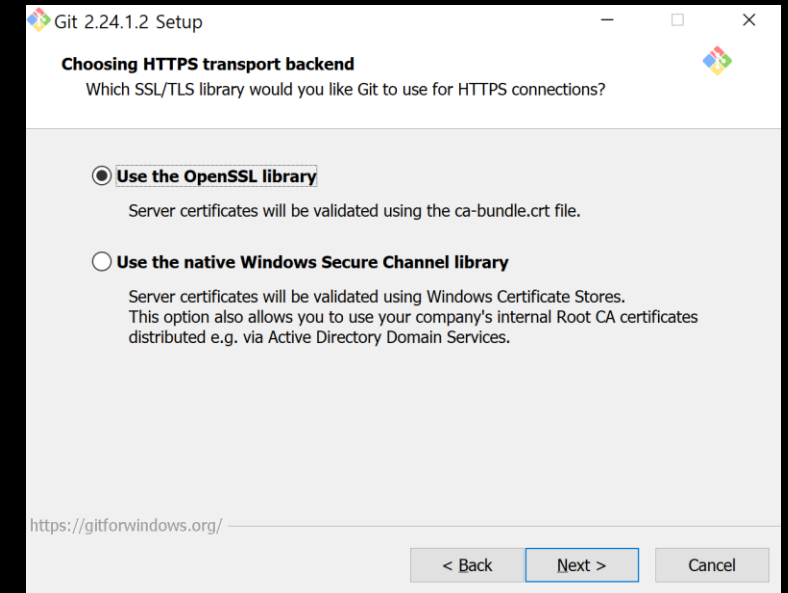
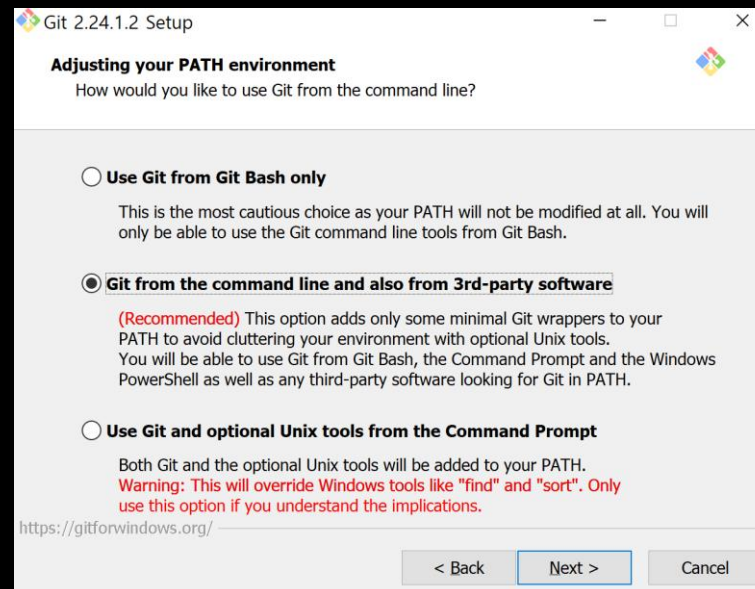
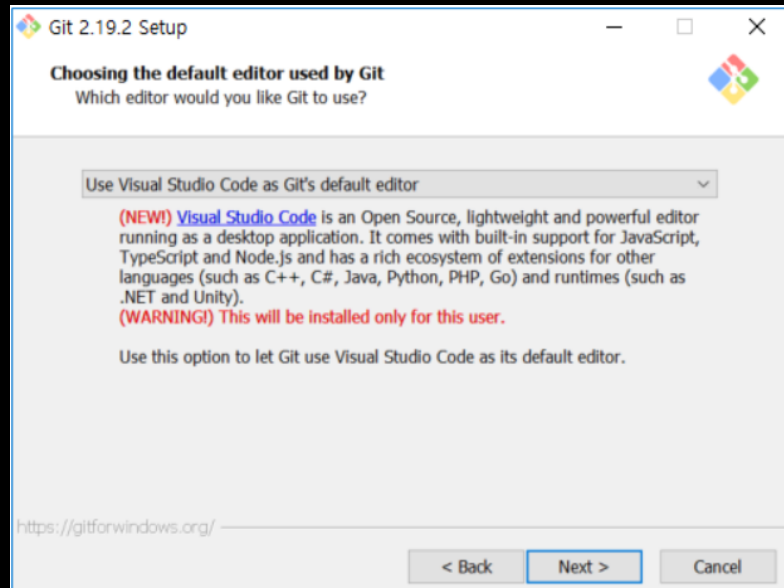
Git for Windows Portable ("thumbdrive edition")

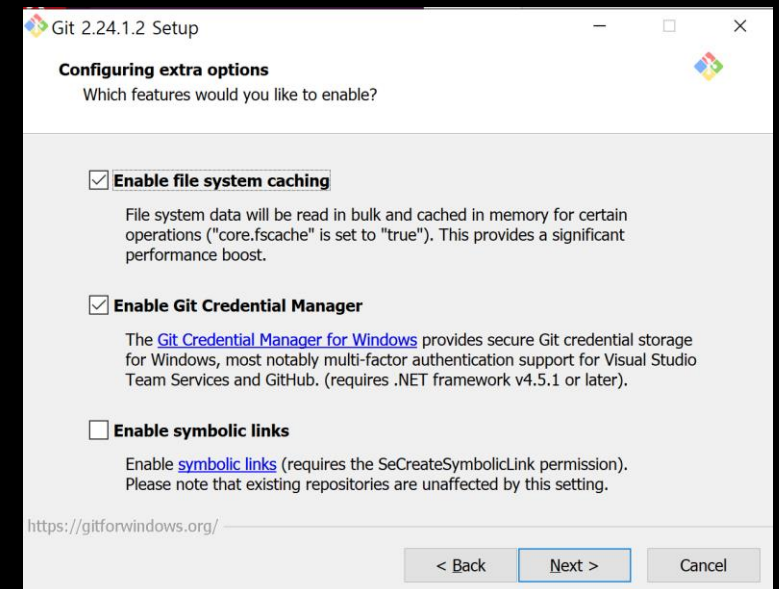
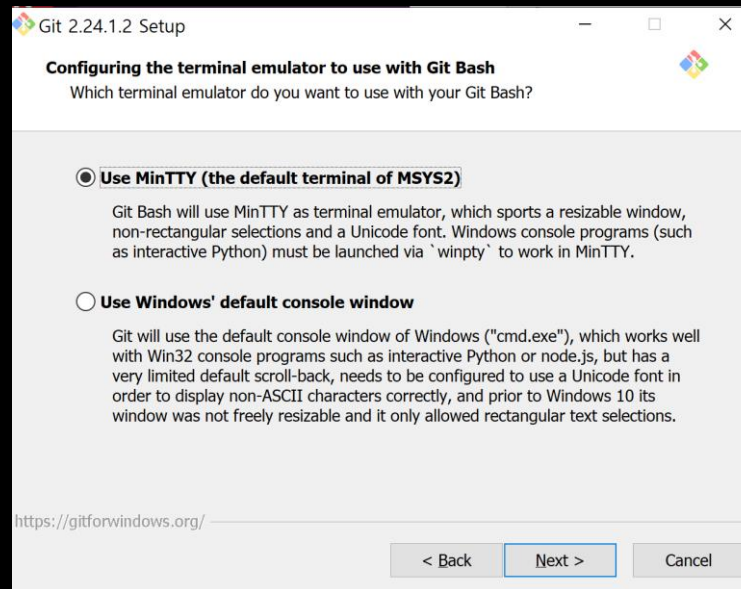
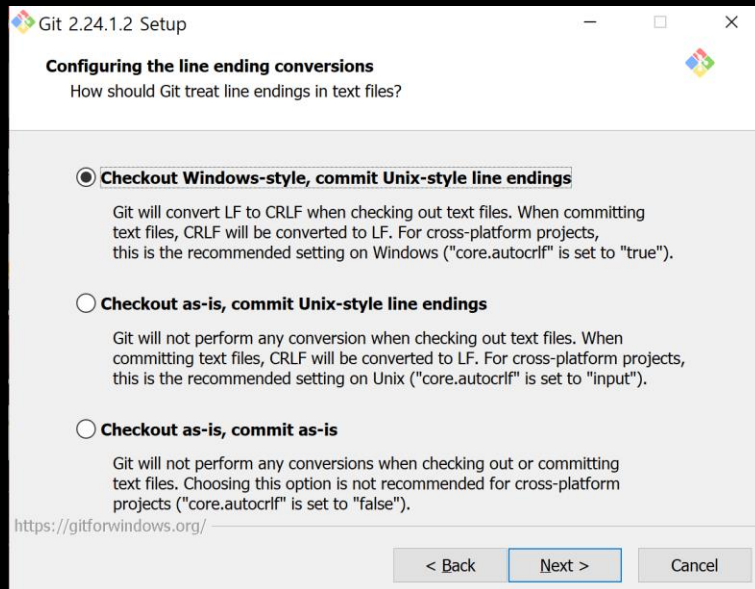
32-bit Git for Windows Portable.

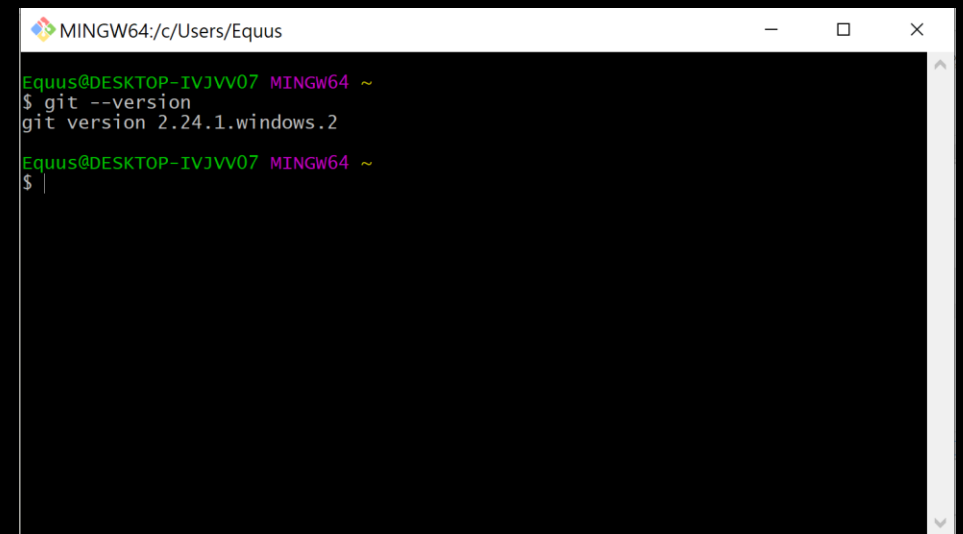
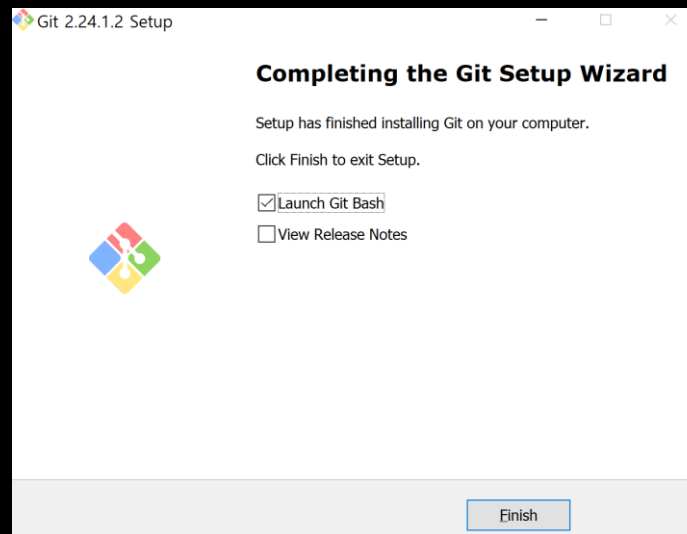
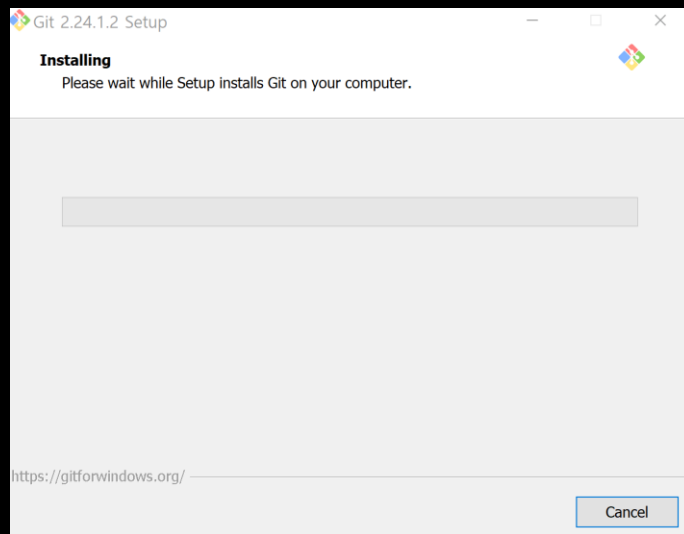
64-bit Git for Windows Portable.

The current source code release is version **2.24.1**. If you want the newer version, you can build it from [the source code](#).

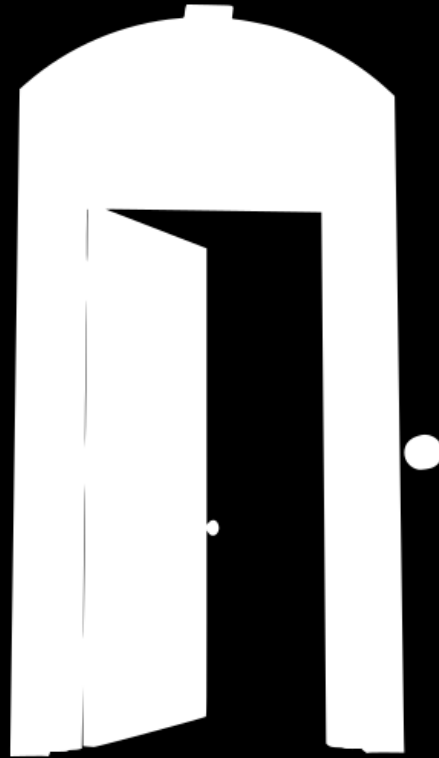






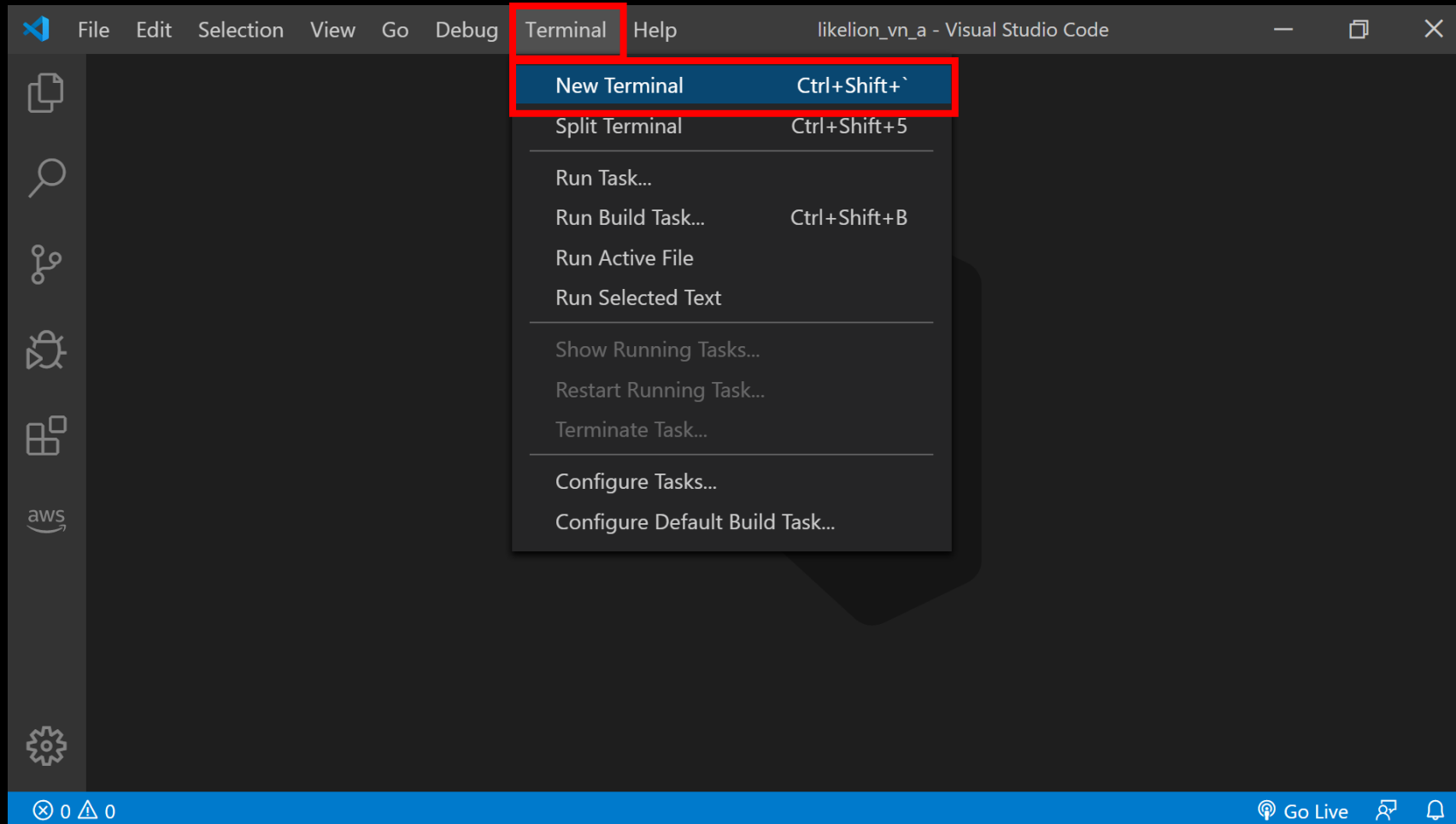


터미널(terminal)이란?



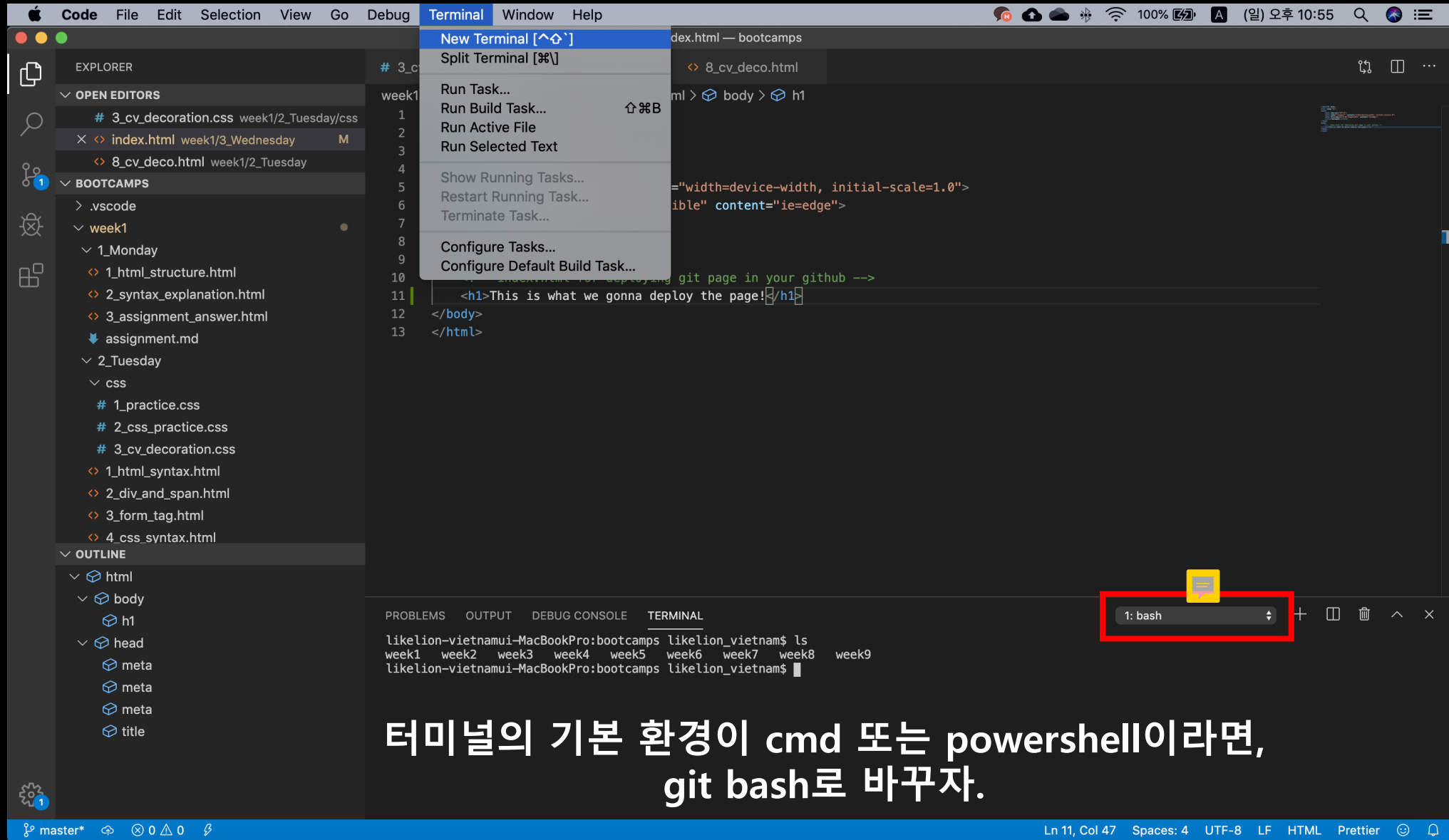
터미널은 컴퓨터와 소통하기 위한 통로

터미널켜기



단축키 : **ctrl + j** or **command + j**

터미널 세팅



<> 2_div_and_span.html

Command Palette... ⌘P

Settings [⌘,]
 Online Services Settings
 Extensions ⌘X
 Keyboard Shortcuts [⌘K ⌘S]
 Keymaps [⌘K ⌘M]
 User Snippets
 Color Theme [⌘K ⌘T]
 File Icon Theme
 Restart to Update (1)



>Shell

Terminal: Select Default **Shell** recently used
 Docker: Attach **Shell** other commands
 Terminal: Allow Workspace **Shell** Configuration
 Terminal: Disallow Workspace **Shell** Configuration

Select your preferred terminal shell, you can change this later in your settings

Command Prompt C:\WINDOWS\System32\cmd.exe
 PowerShell C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
 WSL Bash C:\WINDOWS\System32\wsl.exe
 Git Bash C:\Program Files\Git\bin\bash.exe

터미널의 기본 환경이 cmd 또는 powershell이라면, git bash
로 바꾸자.

절대경로 vs 상대경로

절대경로 : 최상위 폴더로부터 경유한 경로를 전부 적는 방식

(ex : C:\Users\UserID\Desktop\likelion_vn)

상대경로 : 현재 폴더를 기준으로 작성된 경로

터미널 명령어

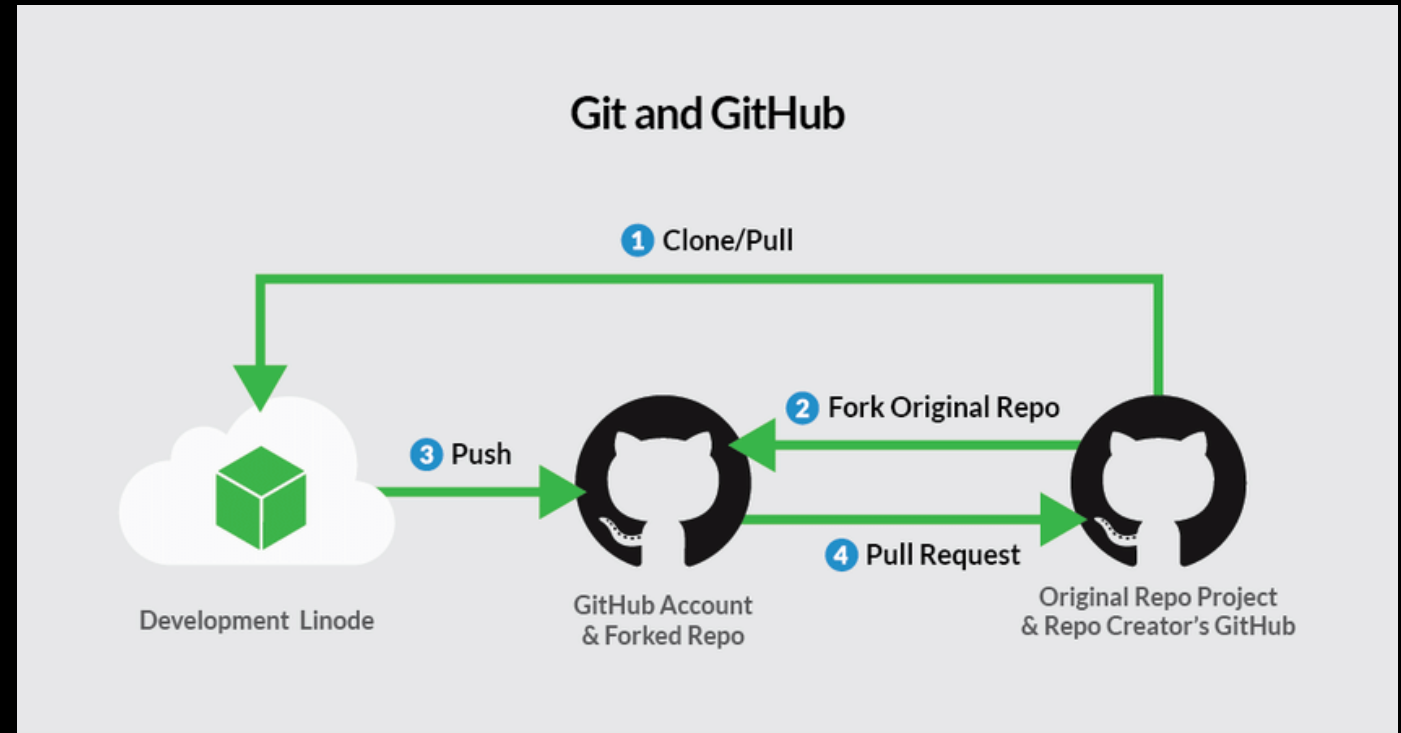
1. **cd** – 디렉토리(directory) 변경(change)
 1. cd directoryname (하위 디렉토리로 이동)
 2. cd .. (상위 디렉토리로 이동)
 3. cd + tab : tab 키는 디렉토리의 이름을 자동완성하는데 활용된다.
2. **mkdir foldername** – 디렉토리(directory) 생성(make)
3. **touch filename** – 새로운 파일 생성
4. **clear** – 터미널 화면 정리
5. **ls** – 해당 디렉토리 내부에 있는 파일이나 디렉토리 확인 🗨
6. **pwd** – 현재 작업하고 있는 공간을 확인

* 디렉토리 = 폴더

명령어 연습


1. week 3-1 폴더(디렉토리)를 만들고
2. 내부에 index.html이라는 이름의 html파일을 만들고,
3. blog.css파일을 만들자.
4. 여러분이 2-1주차 과제로 완성한 부트스트랩 과제의 html파일의 코드와 css파일의 코드를 각각의 파일에 복사 + 붙여넣기 해보자.

Github(깃허브)?



인터넷 상에서 언제든지 코드를 관리하고 백업하자!

왜 개발자들은 깃허브를 사용할까?



infominer


Information Discovery Research Driven Content

@infominer33

infominer@protonmail.com

My Projects


GitHub repositories that I'm working on.



github-portfolio

A place to track the repositories of infominer.


HTML ★ 0 0



web-work

Resources for Independent Web-Workers: SEO, Publishing, Git, GitHub Pages, Linux, Static Site Generators, Development.


CSS ★ 0 1



DiDecentralized

Decentralized, Self-Sovereign, and Blockchain Identity resources


JavaScript ★ 59 12



infominer33.github.io

InfoMine@: Bitcoin History, Self-Sovereign Identity, Blockchain Development and Web Technologies.


JavaScript ★ 1 1



bitcoin-history

Historical Information Related to Bitcoin


JavaScript ★ 4 3



SourceCrypto

Discord Chat Server for Creating a Reasonably Complete Index of Crypto Knowledge.


HTML ★ 3 1



indieweb

GitHub Pages for an Indieweb


JavaScript ★ 0 0



transcripts

crypto transcripts - video \podcasts - for easy reference


CSS ★ 0 0



learn-crypto-trading

Becoming an Epic Repository of Crypto-Trading and Finance Info.

JavaScript ★ 10 0




indieweb-old

A Jekyll quick start to getting up and going with the IndieWeb


HTML ★ 0 0

My Interests


Topics that I want to learn more about.



Bitcoin




Decentralized Identity



Linux


깃허브는 개발 커리어에서 포트폴리오가 된다.


첫번째 레포지토리를 만들어보자!


 **RyanKor** ▼


Repositories


Find a repository...


 [RyanKor/60DB](#)


 [RyanKor/pople-pi](#)

 [RyanKor/vietnam_bootcamp](#)

 [RyanKor/koreahacks](#)

 [RyanKor/ryankor.github.io](#)

 [sanghaC/Mango_Survey](#)

 [RyanKor/PopleWeb](#)



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 *



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

Description (optional)

all about likelion project repository

☒ **Public**

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

☐ **Private**

You choose who can see and commit to this repository.

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

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

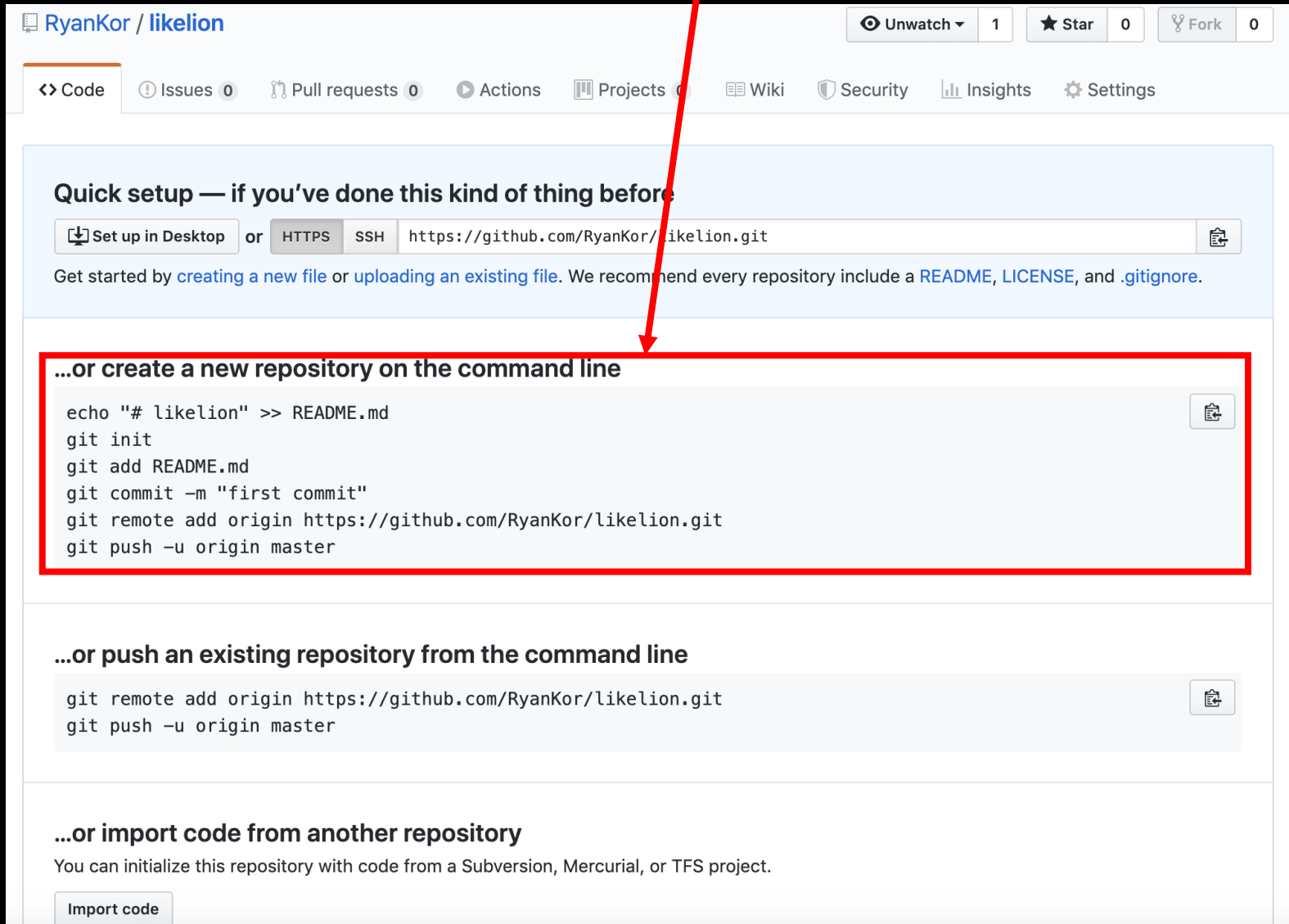
Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

우리가 터미널에 입력할 명령어들



RyanKor / likelion

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects Wiki Security Insights Settings

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

Set up in Desktop or HTTPS SSH `https://github.com/RyanKor/likelion.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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

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

```
git remote add origin https://github.com/RyanKor/likelion.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

깃의 작동방식

컴퓨터의 작업공간

커밋 이전의 상황을
보여주는 공간

1. git init

Working
Directory

Staging
Area

.git directory
(Repository)

Checkout the project

Stage Fixes

2. git add

3. git commit

Commit

↳ 우리 컴퓨터 내부의 로컬 저장소
(master 브랜치가 자동으로 생김)

4. git remote~ : git에게 원격서버주소(외부레포지토리 주소)를 알려준다.

5. git push~ : 3번의 과정을 외부의 레포지토리로 이동하는데 사용된다.

git status : 상태점검

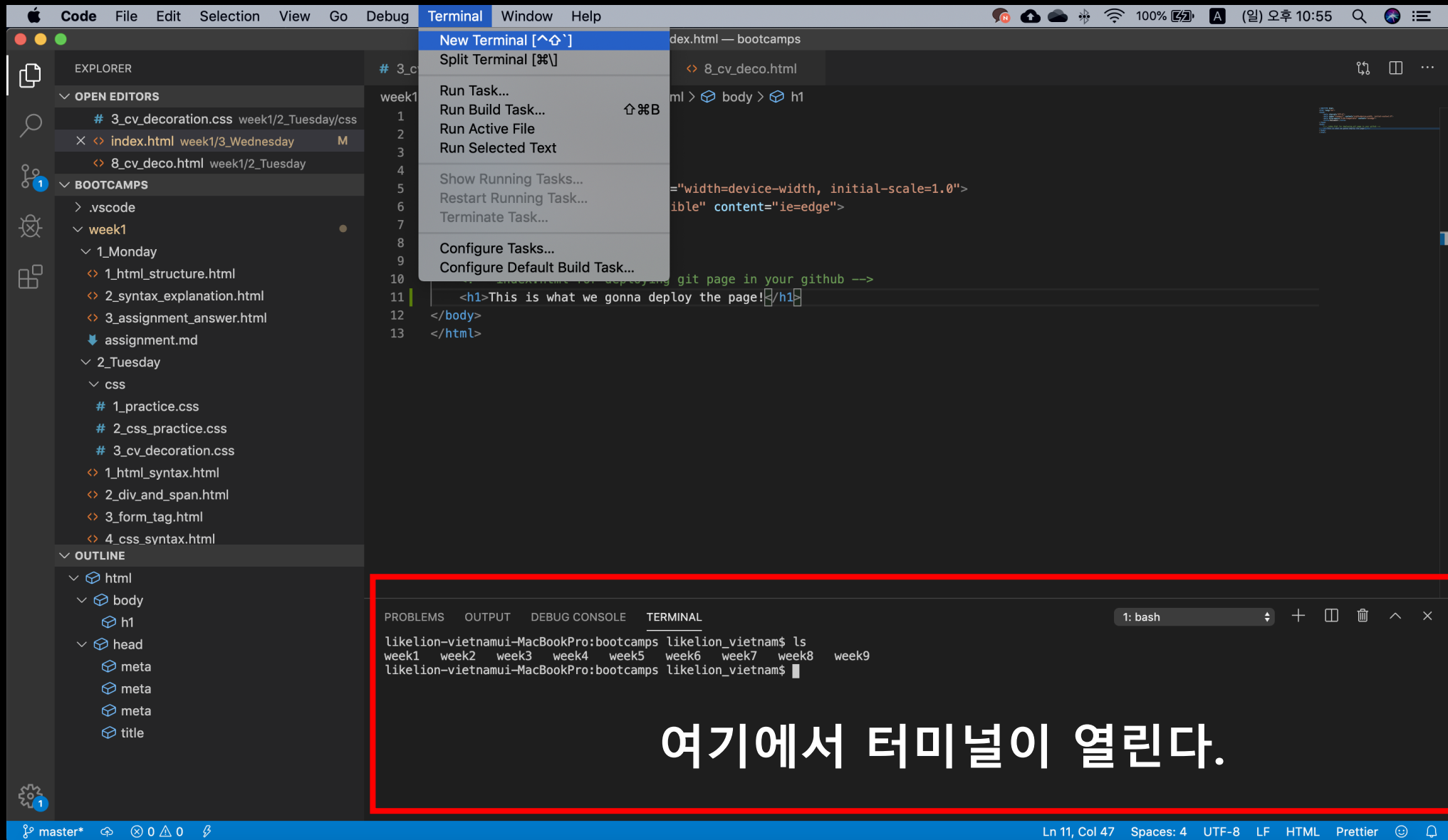
깃의 작동방식

4. `git remote add origin` 외부레포지토리 주소 : git에게 원격서버주소(외부레포지토리 주소)를 알려준다.
5. `git push -u origin master` : 3번의 과정을 원격(외부)의 레포지토리로 이동하는데 사용된다.
-u : 우리 컴퓨터의 master 브랜치와 원격(외부)의 master브랜치를 연결한다.

For more :

<http://ndpsoftware.com/git-cheatsheet.html>

<https://learngitbranching.js.org/>



깃허브 명령어 따라쳐보기



```
likelion-vietnamui-MacBookPro:Programing likelion vietnam$ mkdir likelion
likelion-vietnamui-MacBookPro:Programing likelion vietnam$ cd likelion
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git init
initialized empty git repository in /Users/likelion_vietnam/Documents/Programing/likelion/.git/
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git commit -m "my first commit on github"
On branch master

Initial commit

Untracked files:
  week1/

nothing added to commit but untracked files present
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git add .
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git commit -m "my first commit on github"
[master (root-commit) d12a812] my first commit on github
16 files changed, 619 insertions(+)
create mode 100644 week1/1_Monday/1_html_structure.html
create mode 100644 week1/1_Monday/2_syntax_explanation.html
create mode 100644 week1/1_Monday/3_assignment_answer.html
create mode 100644 week1/1_Monday/assignment.md
create mode 100644 week1/2_Tuesday/1_html_syntax.html
create mode 100644 week1/2_Tuesday/2_div_and_span.html
create mode 100644 week1/2_Tuesday/3_form_tag.html
create mode 100644 week1/2_Tuesday/4_css_syntax.html
create mode 100644 week1/2_Tuesday/5_id_and_class.html
create mode 100644 week1/2_Tuesday/6_programming_disgestion.html
create mode 100644 week1/2_Tuesday/7_css_practice.html
create mode 100644 week1/2_Tuesday/8_cv_deco.html
create mode 100644 week1/2_Tuesday/css/1_practice.css
create mode 100644 week1/2_Tuesday/css/2_css_practice.css
create mode 100644 week1/2_Tuesday/css/3_cv_decoration.css
create mode 100644 week1/3_Wednesday/index.html
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git pull
there is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=<remote>/<branch> master

likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git remote add origin https://github.com/RyanKor/likelion.git
likelion-vietnamui-MacBookPro:likelion likelion_vietnam$ git push -u origin master
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (23/23), 11.34 KiB | 1.89 MiB/s, done.
Total 23 (delta 1), reused 0 (delta 0)
```

깃 명령어 설명

git init은 내 컴퓨터 상에서 새로운 레포지토리를 만들 것이다. 뒤따르는 명령어들이 레포지토리를 만들 것이다.

git clone은 레포지토리를 복사하기 위해 사용된다. 이는 레포지토리가 웹 상에 있을 경우에 사용된다.

git add는 staging area와 서버로 파일을 추가하기위해 사용된다.

git commit은 변화에 대한 스냅샷을 만들것이고 그것을 깃 디렉토리에 저장하는데 사용된다.

git config는 사용자에게 맞는 환경을 구축하기 위해 사용된다. (ex. 이메일, 사용자이름)

→ `git config --global user.email youremail@example.com`

git push는 내 컴퓨터상의 **commit**을 외부의 레포지토리의 마스터 브랜치로 보내는데 사용된다.

git remote는 모든 리모트(remote) 레포지토리를 볼 수 있게 한다.

git pull은 리모트(remote) 레포지토리에 있는 모든 변화를 내 컴퓨터상의 작업 디렉토리로 합친다.

git status : 전체 깃의 상태를 점검할 수 있다.

우리의 웹페이지를 깃허브로 배포해봅시다! LIKE LION



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 *

 aimclee ▾ / aimclee.github.io 

Great repository names are short and memorable. Need inspiration? How about [special-waddle?](#)

Description (optional)

- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

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

- ☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

여러분의 새로운 레포지토리를 만들어봅시다.

형식 : 사용자이름.github.io

저의 경우에, username은 aimclee이기에, 레포지토리의 이름은 aimclee.github.io가 됩니다.


<> index.html

M

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <!-- index.html for deploying git page in your github -->
  <h1>This is what we gonna deploy the page!</h1>
</body>
</html>
```

1. 배포할 html 파일을 '**index.html**'로 만들어봅시다. (왜냐하면 깃허브가 이 파일을 배포할 것이라고 자동으로 인식하기 때문입니다.)
2. 여러분의 resume를 배포해봅시다.
3. 깃허브 배포과정을 그대로 따라해보면 됩니다.

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

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/aimclee/aimclee.github.io.git`



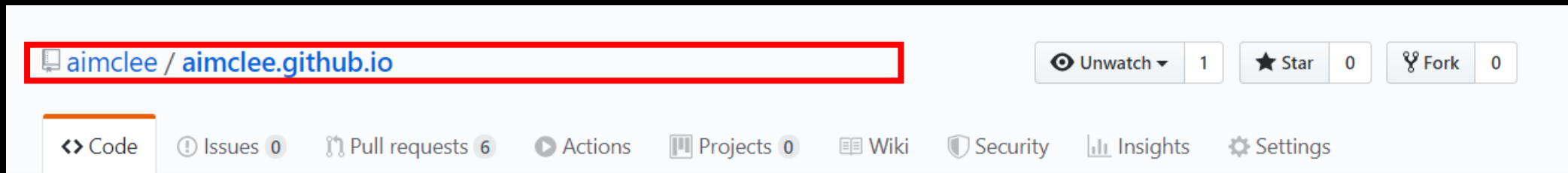
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and `.gitignore`.

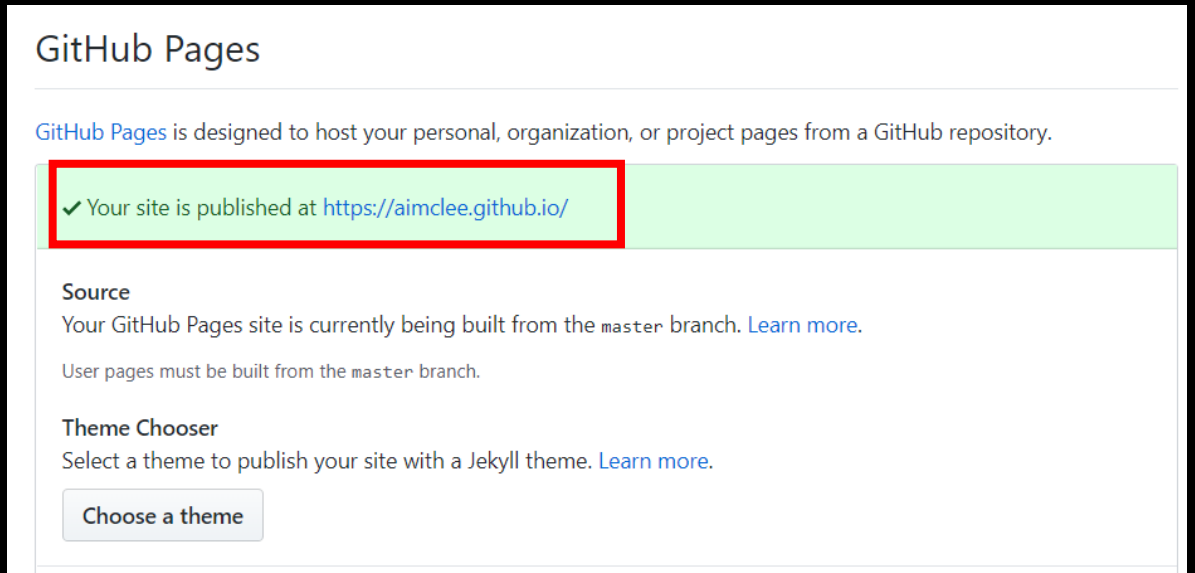
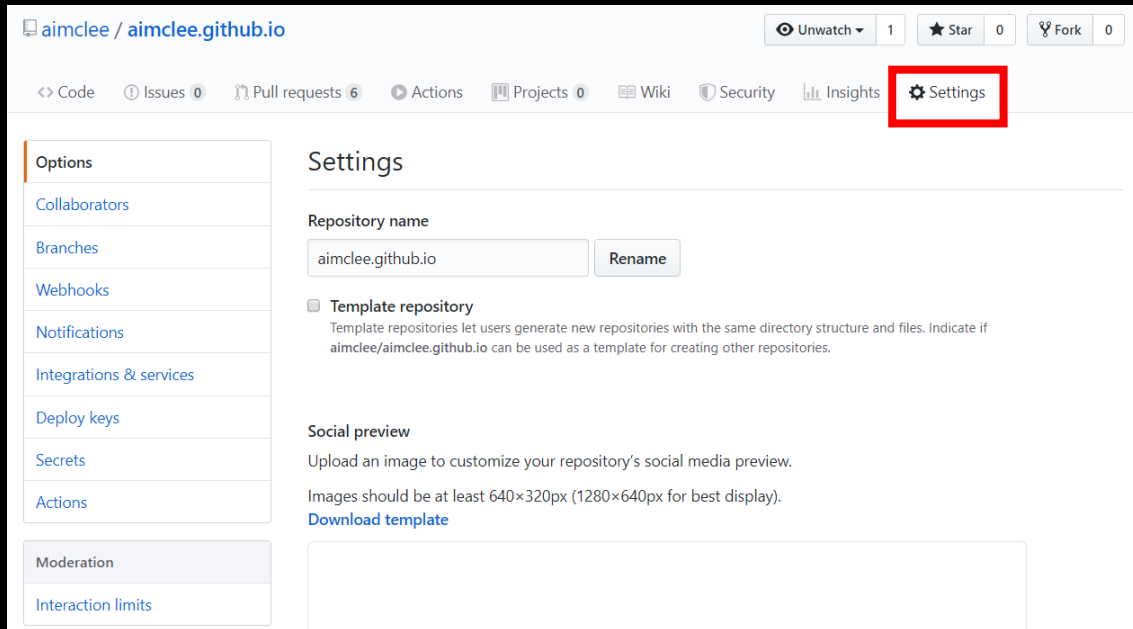
...or create a new repository on the command line

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



깃허브를 활용해서 페이지 배포하기





Settings에 들어가서 여러분의 레포지토리 이름을 다시 한번 확인해봅시다.
그리고 깃허브 페이지를 배포할 수 있는 지 없는 지를 github pages 항목에서 확인해봅시다.

images	page	14 days ago
static	page	14 days ago
CNAME	Create CNAME	14 days ago
asset-manifest.json	page	14 days ago
favicon.ico	page	14 days ago
index.html	page	14 days ago
manifest.json	page	14 days ago
precache-manifest.6d17efe74163eace64cfd3148c43b0ea.js	page	14 days ago
robots.txt	page	14 days ago
service-worker.js	page	14 days ago

깃허브에 index.html파일을 push한 이후에, 깃허브 서버가 파일을 제대로 인식하기까지 1분~10분정도 기다려야 합니다.

2-1주차 과제였던 부트스트랩 과제를 배포해주세요!

깃허브에서 계정 당 하나까지는 공짜로 배포할 수 있게 해줍니다.

Hack Your Life

Boost to learn programming and create your products!

[Apply Now!](#)

Our Services



Curriculum

You can learn programming and create your ideas in 9 weeks

[See Curriculum »](#)

Alumni

LIKE LION alumni is powerful. 6,000 students make 742 services in Korea

[See Alumni »](#)

Mentors

Accumulated 7 year data and developer mentors are waiting you!

[See Mentors »](#)

Join Us

Send us a message and we will get back to you as soon as possible.

[Contact us](#)

오늘의 과제

1. github에 **assignment-hochiminh-1st** 레포지토리 만들기(앞으로의 과제는 여기에!)
2. 저번시간 과제인 인스타그램 반응형 웹디자인 적용해서 최대한 완성해 와보세요!

감사합니다 :)

 LIKE LION

@Vietnam

아이디어톤



팀빌딩 시간

발표된 아이디어에 대해 다른 수강생 분들과
자유롭게 논의해보도록 하겠습니다~!



Icon attribution

- Icon made by geotatah from www.flaticon.com
- Icon made by Freepik from www.flaticon.com