



2020 INYRO UN-TACT SEMINAR

1부 깃허브 명령어 사용법과 개인 포트폴리오 관리

상명대학교 휴먼지능정보공학과 19학번 임재영

목차

Introduction

발표자 소개

Git & Github 에 대해 알아보기

About Git

About Github

Github 실제로 사용해보기

Repository 생성 해보기

명령창에서 Commit 하기

Portfolio 관리법

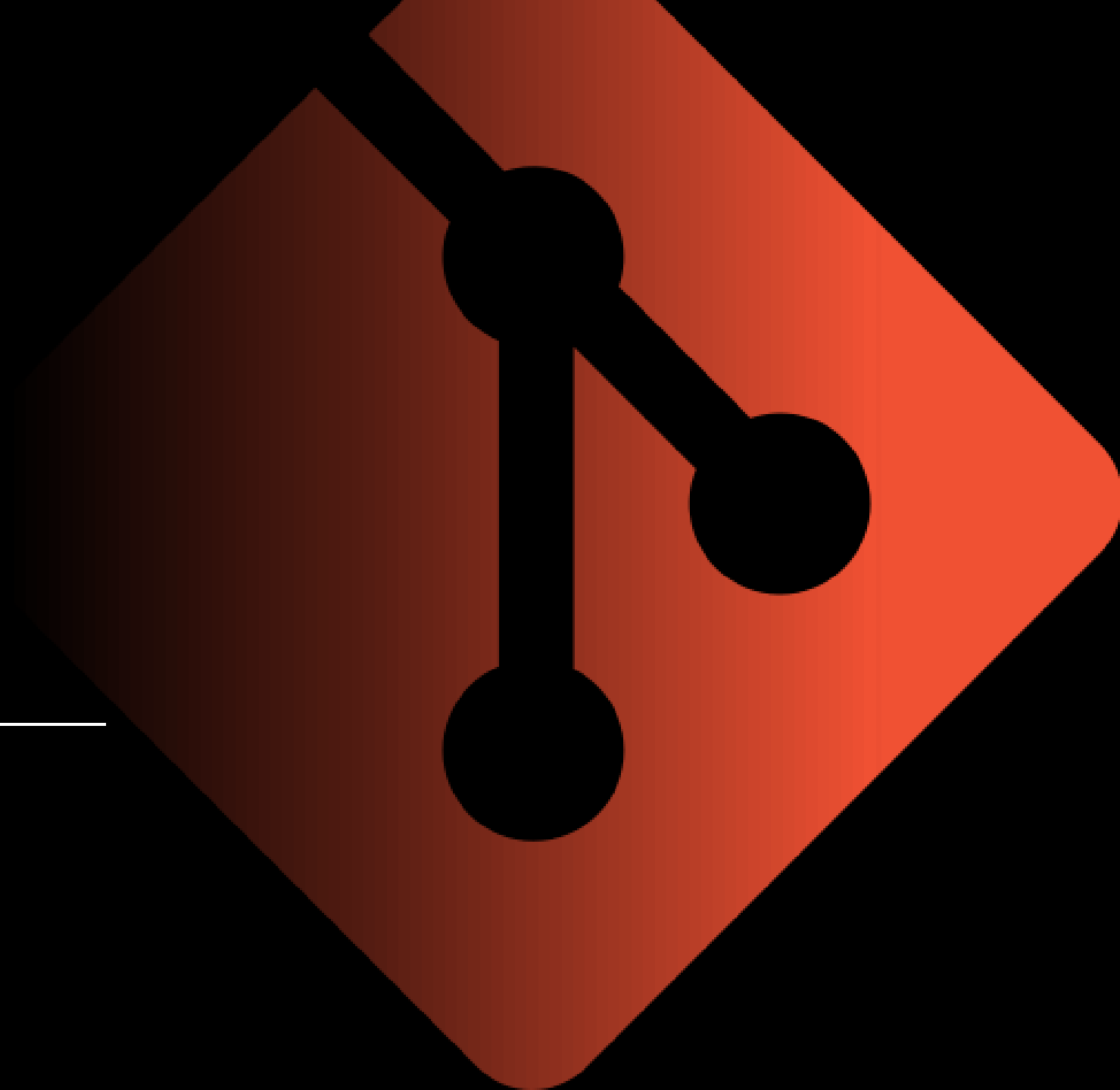
발표자 소개

- 현 이니로 부회장
 - 동아리 내 주요 활동 구상 및 관리
 - 동아리 문서 관리 및 다양한 행사 추진
- 2020학년도 C프로그래밍 멘토
- 다수의 프로젝트 및 스터디 진행
 - Android, 딥러닝 관련하여 3개의 프로젝트 진행
 - 동기 및 선배들과 다양한 스터디 진행 및 멘토링 이수
- 깃허브 링크 : <https://github.com/limjustin>





About Git



About Git



1. Version Control System

Version 관리를 위한 소프트웨어

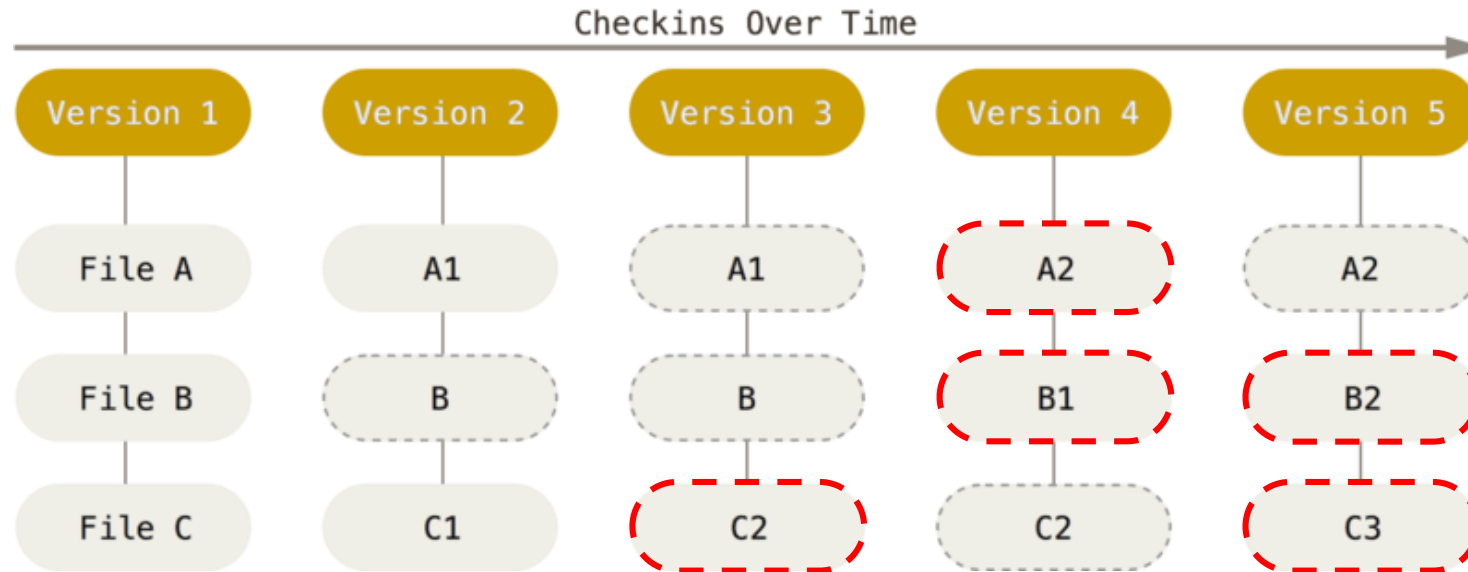
각 Version의 변경 사항만 기록

2. Collaboration

같은 파일을 가지고 여러 명과 함께 작업 가능

어느정도 규모가 있는 프로젝트에서 사용하기 편리

About Git

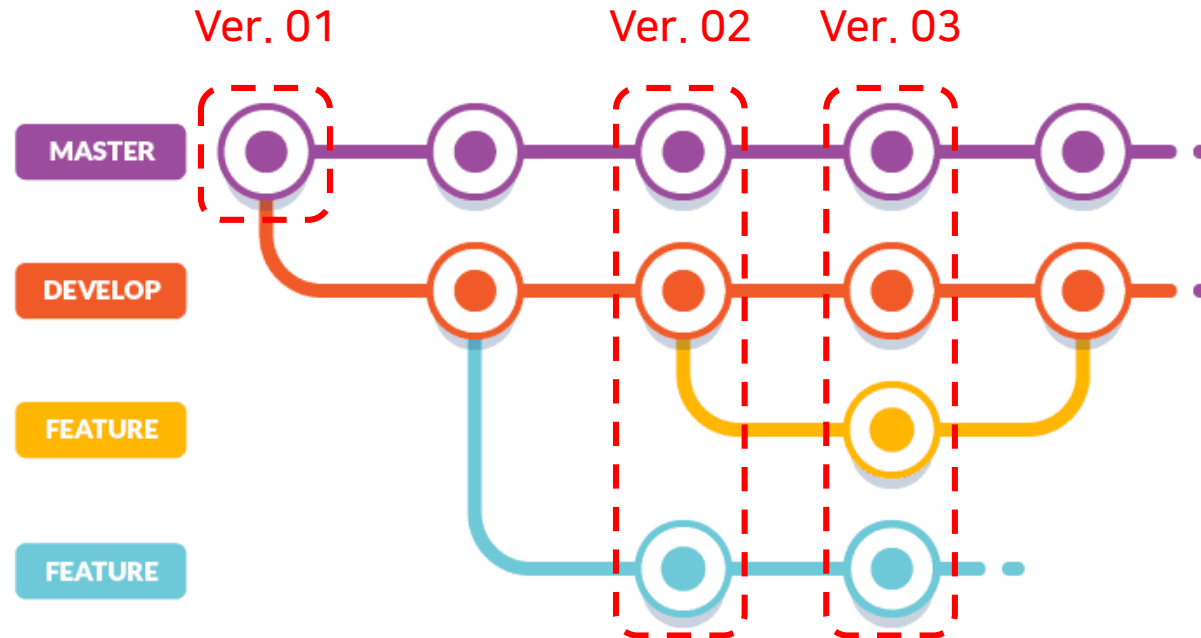


1. Version Control System

Version 관리를 위한 소프트웨어

각 Version의 변경 사항만 기록

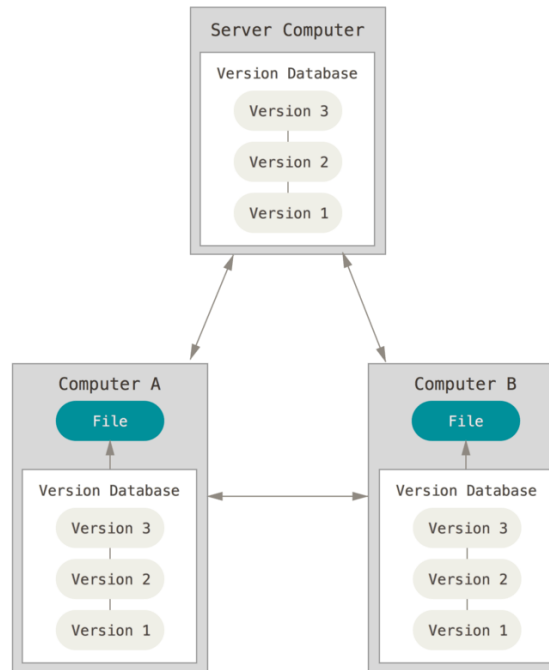
About Git



1. Version Control System

Version 관리를 위한 소프트웨어
각 Version의 변경 사항만 기록

About Git



2. Collaboration

같은 파일을 가지고 여러 명과 함께 작업 가능
어느정도 규모가 있는 프로젝트에서 사용하기 편리



About Github

GitHub

About Github



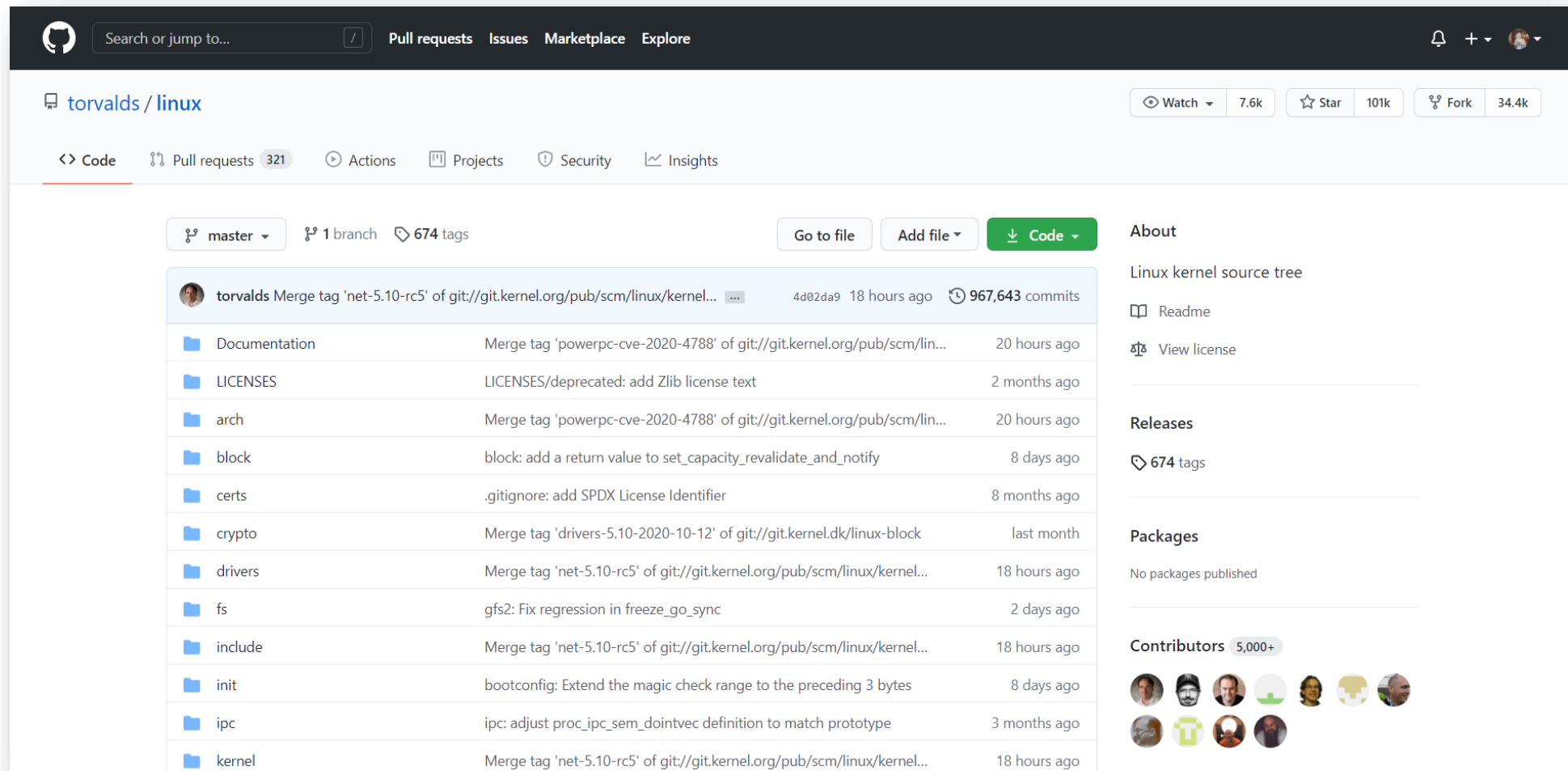
Git으로 저장되어 원격 전송된 파일들이 저장되는 공간을 제공하는 서비스

클라우드에서 Git을 제공한다고 이해하면 편리

변화 내용을 올리고 공유 가능

공개된 모든 오픈소스 프로젝트들을 공유 가능

About Github



The screenshot shows the Github interface for the `torvalds/linux` repository. The top navigation bar includes the Github logo, a search bar, and links to Pull requests, Issues, Marketplace, and Explore. The repository name `torvalds/linux` is displayed, along with statistics: 7.6k Watchers, 101k Stars, and 34.4k Forks. The main content area shows the repository structure with a table of files and folders, including Documentation, LICENSES, arch, block, certs, crypto, drivers, fs, include, init, ipc, and kernel. The right sidebar contains sections for About (Linux kernel source tree), Releases (674 tags), Packages (No packages published), and Contributors (5,000+).

Repository: torvalds/linux

Watch 7.6k Star 101k Fork 34.4k

Code Pull requests 321 Actions Projects Security Insights

master 1 branch 674 tags

Go to file Add file Code

File/Folder	Commit Message	Commit Hash	Time Ago
torvalds	Merge tag 'net-5.10-rc5' of git://git.kernel.org/pub/scm/linux/kernel...	4d02da9	18 hours ago
Documentation	Merge tag 'powerpc-cve-2020-4788' of git://git.kernel.org/pub/scm/lin...		20 hours ago
LICENSES	LICENSES/deprecated: add Zlib license text		2 months ago
arch	Merge tag 'powerpc-cve-2020-4788' of git://git.kernel.org/pub/scm/lin...		20 hours ago
block	block: add a return value to set_capacity_revalidate_and_notify		8 days ago
certs	.gitignore: add SPDX License Identifier		8 months ago
crypto	Merge tag 'drivers-5.10-2020-10-12' of git://git.kernel.dk/linux-block		last month
drivers	Merge tag 'net-5.10-rc5' of git://git.kernel.org/pub/scm/linux/kernel...		18 hours ago
fs	gfs2: Fix regression in freeze_go_sync		2 days ago
include	Merge tag 'net-5.10-rc5' of git://git.kernel.org/pub/scm/linux/kernel...		18 hours ago
init	bootconfig: Extend the magic check range to the preceding 3 bytes		8 days ago
ipc	ipc: adjust proc_ipc_sem_dointvec definition to match prototype		3 months ago
kernel	Merge tag 'net-5.10-rc5' of git://git.kernel.org/pub/scm/linux/kernel...		18 hours ago

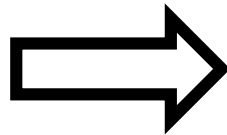
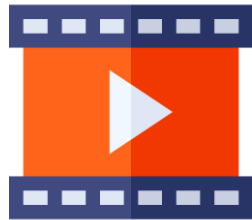
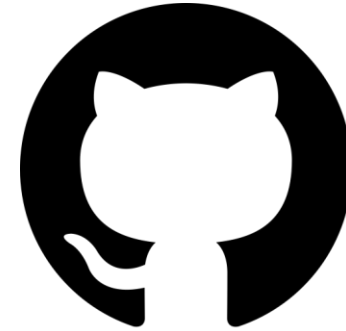
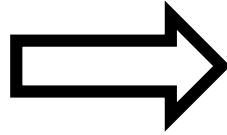
About
Linux kernel source tree
Readme
View license

Releases
674 tags

Packages
No packages published

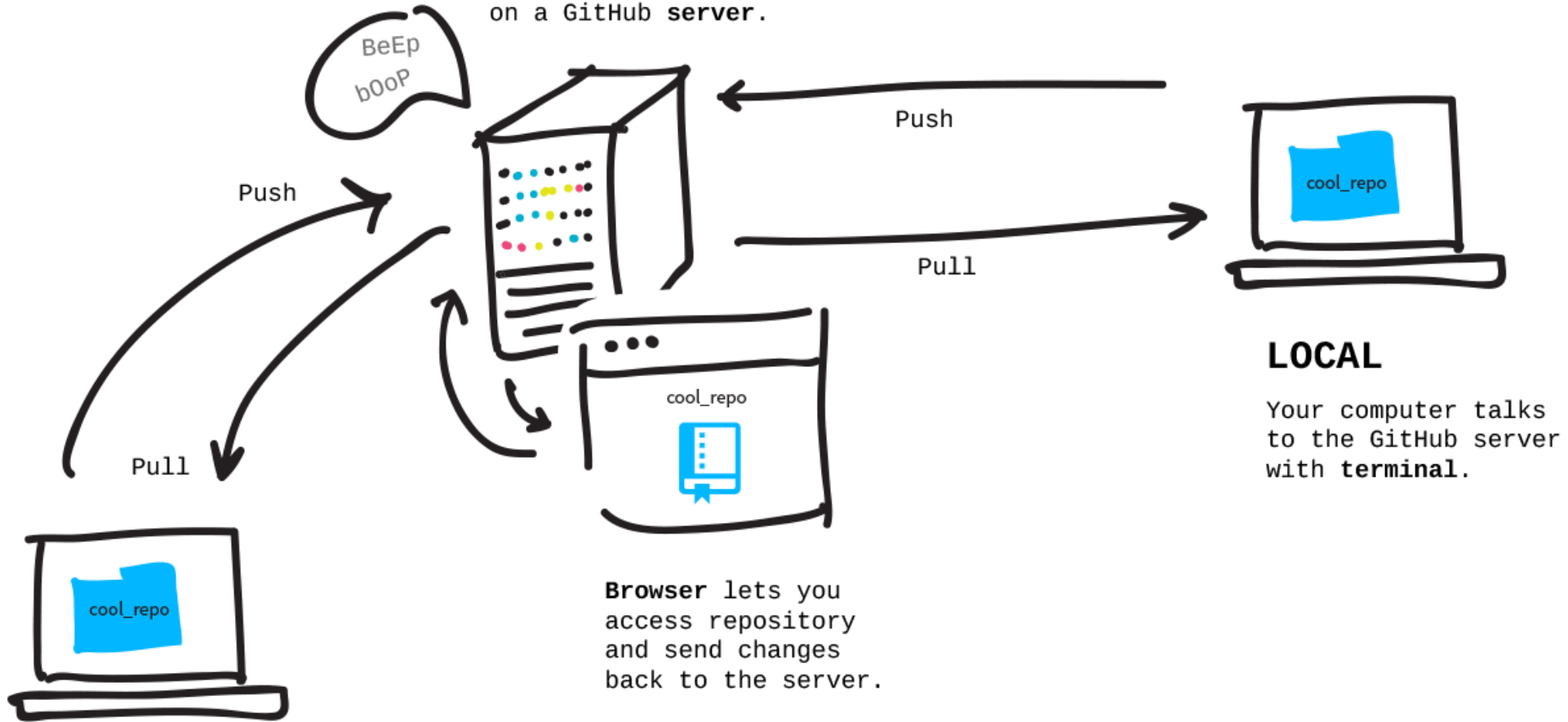
Contributors 5,000+
11,000 contributors

Git과 Github에 대한 이해



REMOTE

Repositories live on a GitHub server.



LOCAL

Your computer talks to the GitHub server with **terminal**.

LOCAL

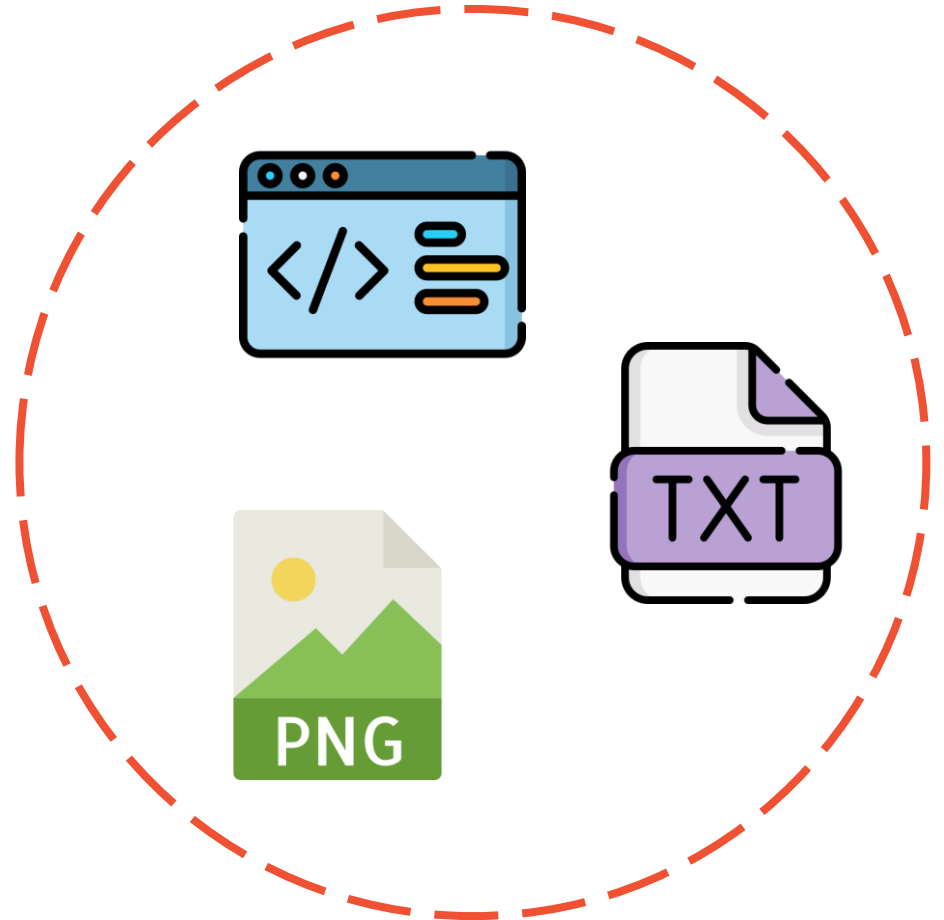
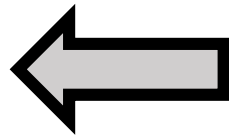
Someone else's computer talks to the GitHub server.



Repository 생성 해보기

Repository 생성하기

Repository는 '저장소' 라는 뜻!!







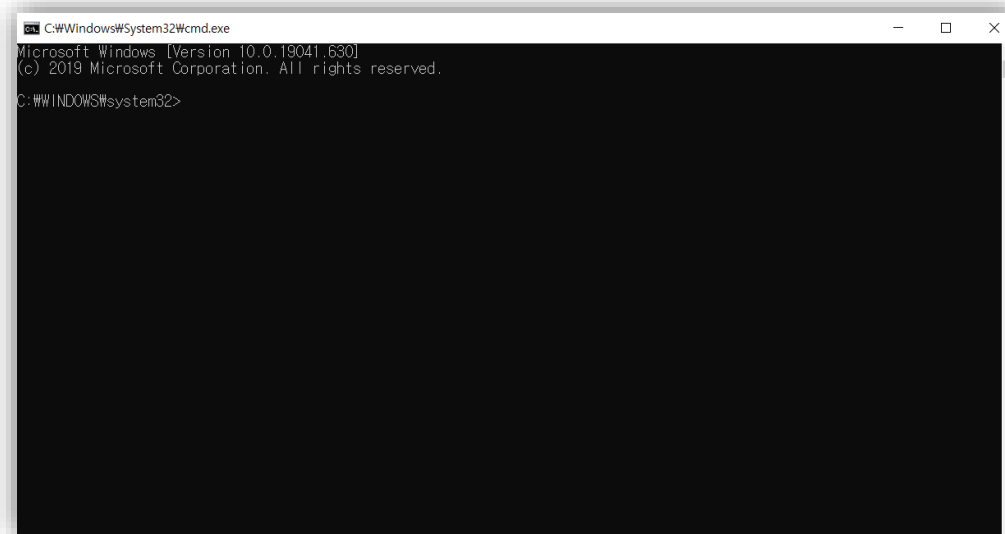
명령창에서 Commit 하기

(Windows 운영체제 기준)

Step 0. 명령창 띄우기

 + R 키를 누르거나,  키를 누르고 'cmd' 라고 입력하면
명령창이 생성되는 것을 볼 수 있다.

이를 CMD라고 부르는 이유는, Command의 약자이기 때문이다.



Step 1. 작업하고자 하는 디렉토리로 이동하기

작업하고자 하는 디렉토리로 이동하기 위해서는,

디렉토리를 이동할 수 있게 하는 `cd` 명령어가 필요하다.

```
C:\Users\Wuser\Desktop>cd seminar
```

위 명령어를 쳤다고 가정하면, 바탕화면에서
'seminar' 라는 디렉토리로 이동할 수 있다.

Step 2. 기존 디렉토리를 Git 저장소로 만들기

`git init`이라는 명령을 실행하면 `.git`이라는 하위 디렉토리가 만들어진다.

```
C:\Users\User\Desktop\seminar>git init  
Initialized empty Git repository in C:/Users/user/Desktop/seminar/.git/
```

`.git` 디렉토리에는 저장소에 필요한 뼈대 파일이 들어있다.

이 명령만으로는 아직 프로젝트의 어떤 파일도 관리하지 않는다.

`.git`은 숨김 폴더로 이를 확인하기 위해서는 `dir /ah` 명령어를 사용하면 된다.

Step 3. Github에서 만든 Repository 연동

`git remote add` 명령어를 사용하면

Github에서 생성한 Repository와 연동할 수 있다.

이를 '원격(remote) 저장소'라고 부른다. origin은 원격 저장소의 이름이다.

```
C:\Users\user\Desktop\seminar>git remote add origin https://github.com/limjustin/Git_Practice.git
```

```
C:\Users\user\Desktop\seminar>git remote -v  
origin https://github.com/limjustin/Git_Practice.git (fetch)  
origin https://github.com/limjustin/Git_Practice.git (push)
```

`git remote -v` 명령어를 사용하여 현재 연결되어 있는 원격 저장소를 확인할 수 있다.

기본 브랜치 변경 사항

all: replace usages of whitelist/blacklist and master/slave

There's been plenty of discussion on the usage of these terms in tech. I'm not trying to have yet another debate. It's clear that there are people who are hurt by them and who are made to feel unwelcome by their use due not to technical reasons but to their historical and social context. That's simply enough reason to replace them.

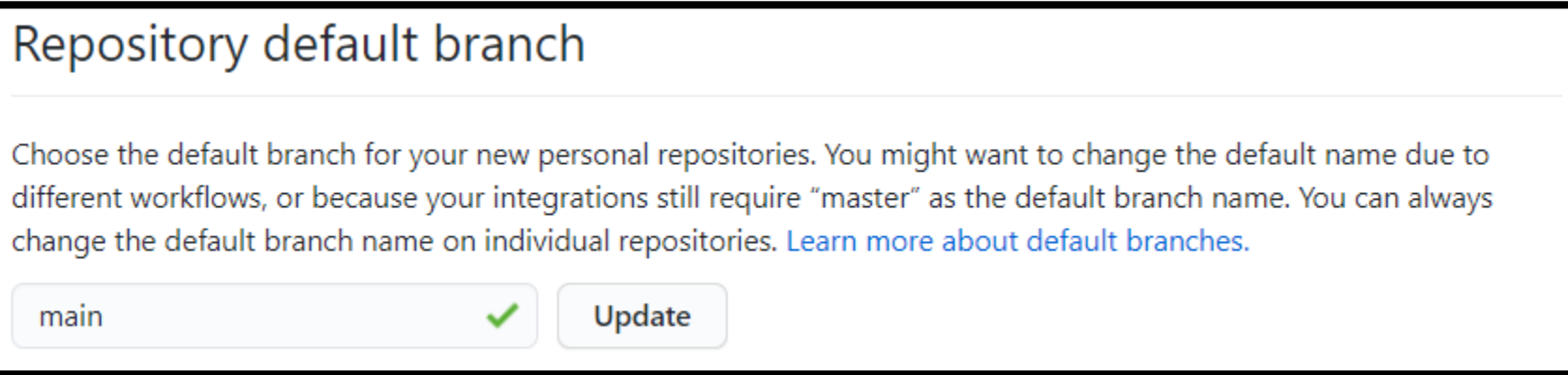
Anyway, allowlist and blocklist are more self-explanatory than whitelist and blacklist, so this change has negative cost.

Didn't change vendored, bundled, and minified files. Nearly all changes are tests or comments, with a couple renames in cmd/link and cmd/oldlink which are extremely safe. This should be fine to land during the freeze without even asking for an exception.

올해 6월 Go 언어가 인종차별적 요소나 주종 관계의 의미를 담고 있는
whitelist/blacklist와 master/slave라는 용어를 프로젝트에서 제거하기로 결정.

이후 **master**를 기본 브랜치로 사용하던 **Git**에서도 이에 대한 논의가 이루어졌고,
2020년 10월부터 **기본 브랜치**를 **master**에서 **main**으로 **변경**하기로 하였다.

기본 브랜치 변경 사항



따라서 현재 기본 브랜치가 **main**으로 되어 있을텐데,
master로 설명하는 것이 뜻의 전달도 잘 되고, 설명하기에도 편리하여
이 이후의 내용에도 **master**로 설명하는 것이 좋다고 판단하여
오늘 세미나에서는 **master**를 기준으로 설명할 것이다.

이는 Settings > Repositories 에 가면 위 그림과 같이 수정할 수 있는 부분이 나온다.

저기에 쓰여 있는 **main**을, **master**로 바꿔보도록 하겠다.

Step 4. 최신 작업물 받아오기

`git fetch` 명령어를 사용하면 원격 저장소의 최신 이력을 확인할 수 있다.

`git pull` 명령어를 사용하면 원격 저장소에 있는 내용을 가져와 자동으로 병합한다.

```
C:\Users\User\Desktop\seminar>git fetch
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 591 bytes | 15.00 KiB/s, done.
From https://github.com/limjustin/Git_Practice
* [new branch] master -> origin/master

C:\Users\User\Desktop\seminar>git pull origin master
From https://github.com/limjustin/Git_Practice
* branch master -> FETCH_HEAD
```

Step 5. 파일 상태 확인하기

`git status` 명령어를 사용하면 **파일의 상태를 확인**할 수 있다.

아직 넣은 파일이 없거나, 최신 상태라면 아래와 같은 문구가 뜬다.

```
C:\Users\User\Desktop\seminar>git status
On branch master
nothing to commit, working tree clean
```

반면, 어떤 파일을 디렉토리에 가져다 놓았다면 아래와 같은 문구가 뜬다.

```
C:\Users\User\Desktop\seminar>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.cpp

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


Step 6. 파일을 로컬 저장소에 추가하기

`git add` 명령어를 사용하면 로컬 저장소에 파일을 추가할 수 있다.

이는 Github에 완전히 파일을 올린 상태가 아니다.

```
C:\Users\User\Desktop\seminar>git add *
```

파일을 올리고 난 후, 다시 `git status` 명령어를 사용하게 되면

```
C:\Users\User\Desktop\seminar>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test.cpp
```

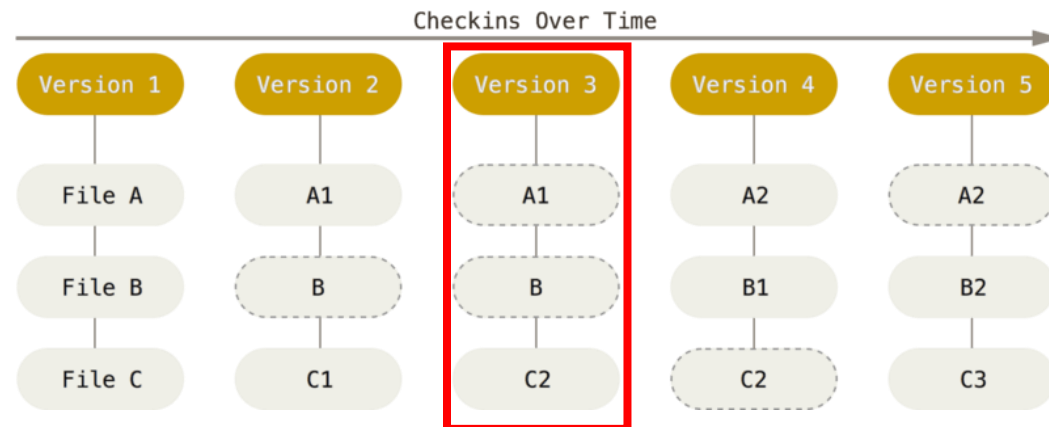
새로운 파일이 추가되었다는 문구를 확인할 수 있다.

Step 7. 커밋 내용 기재하기

`git commit -m` 명령어를 사용하면 커밋 내용과 함께 업로드 내용을 저장할 수 있다.

```
C:\Users\User\Desktop\seminar>git commit -m "Initial commit"
[master b3c7feb] Initial commit
1 file changed, 6 insertions(+)
create mode 100644 test.cpp
```

이제 이 커밋은 하나의 기록으로 남겨지게 된다.



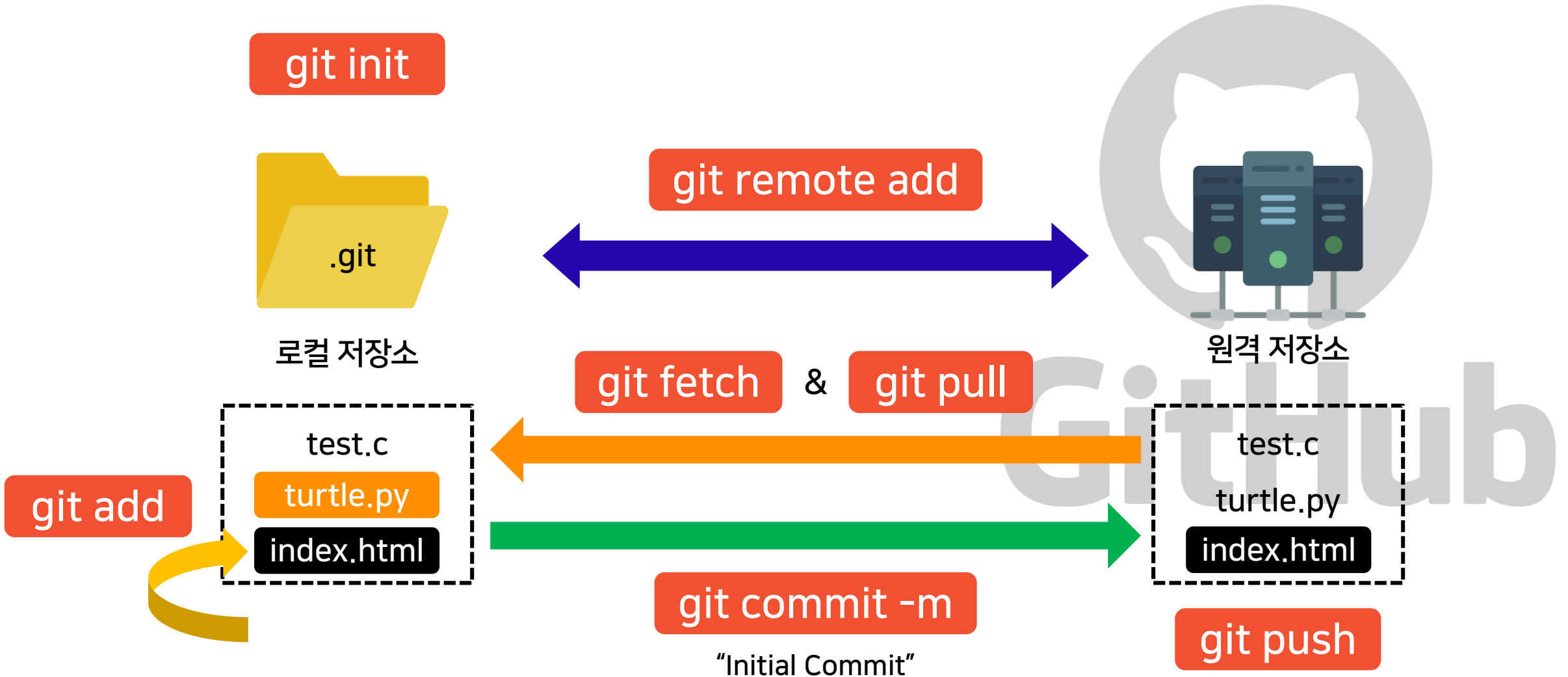
Step 8. 로컬 저장소에 있는 것들을 올리기

`git push` 명령어를 사용하면 로컬 저장소에 있는 내용들을

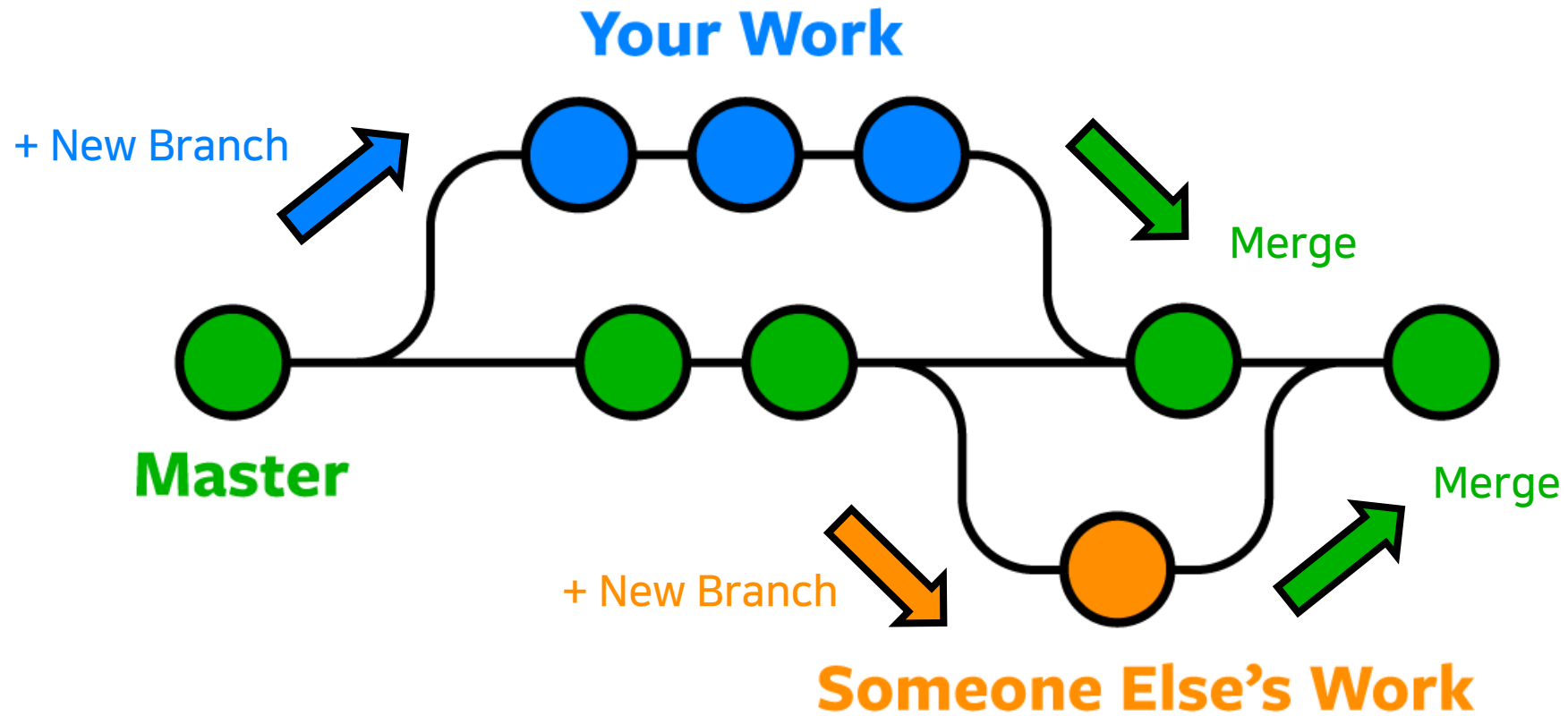
원격 저장소에 추가할 수 있게 된다.

```
C:\Users\User\Desktop\seminar>git push origin master
Username for 'https://github.com': limjustin
Password for 'https://limjustin@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 382 bytes | 127.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/limjustin/Git_Practice.git
 0b98b8c..1e84cd9 master -> master
```

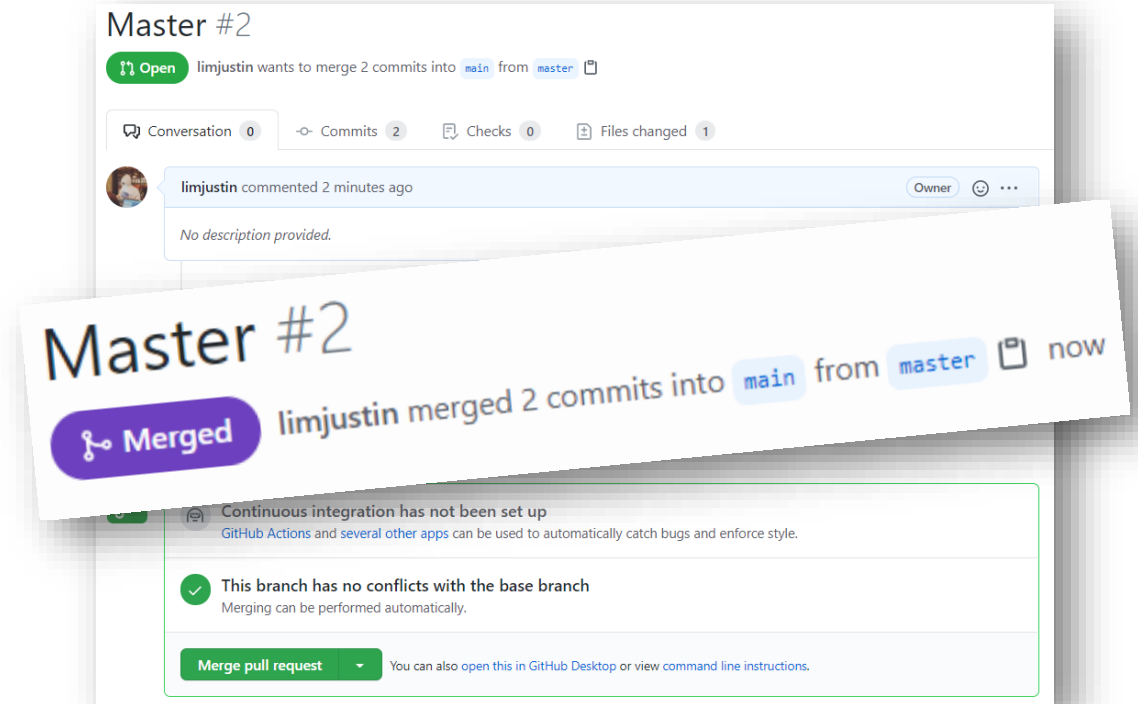
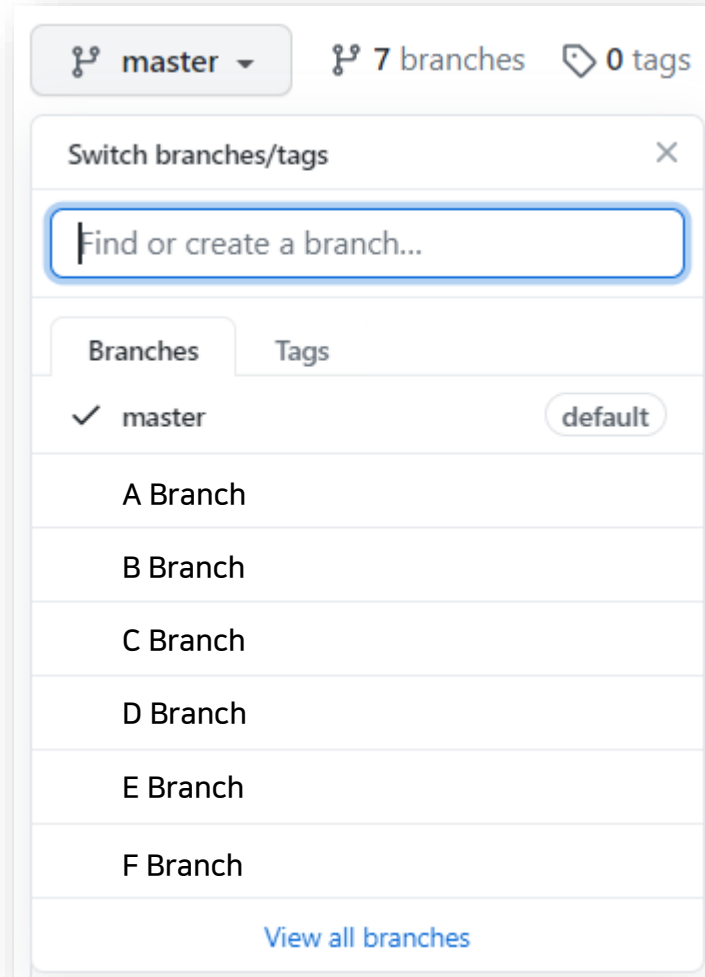
Commit 과정 총정리



Branch & Merge



Branch & Merge



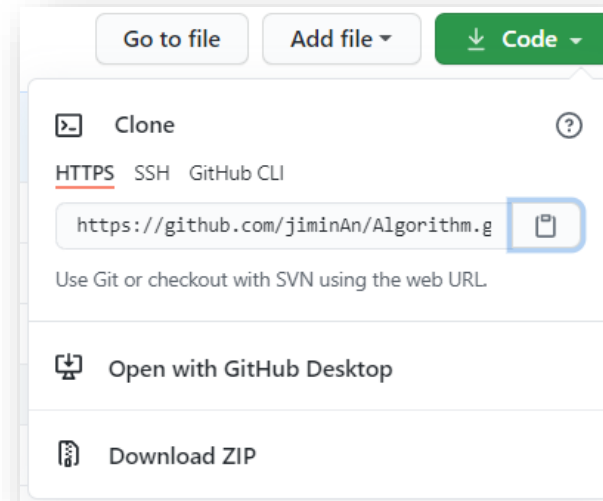
Branch를 능숙하게 잘 사용한다면 **상황에 맞게**
여러 작업을 동시에 진행할 수 있다는 장점이 있다!!
그런 다음 Merge를 통해서 **Branch를 통합하자!!**

Clone



Local

```
C:\Users\User\Desktop>git clone https://github.com/jiminAn/Algorithm.git
Cloning into 'Algorithm'...
remote: Enumerating objects: 235, done.
remote: Counting objects: 100% (235/235), done.
remote: Compressing objects: 100% (232/232), done.
Receiving objects: 83% (640/770)used 2 (delta 1), pack-reused 535Receiving objects: 82% (632/770)
Receiving objects: 100% (770/770), 182.47 KiB | 510.00 KiB/s, done.
Resolving deltas: 100% (396/396), done.
```



Portfolio

활용법

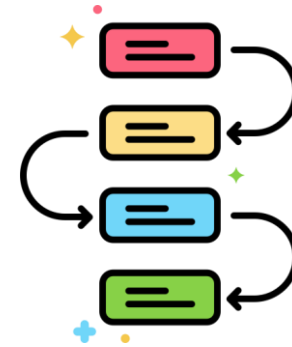


Portfolio 활용법

1. 프로젝트 내 README 파일로 프로젝트 내용 정리하기

다른 사람이 봤을 때 해당 프로젝트가 어떤 내용이고

구체적인 실행화면은 어떻게 나오는지 정리하면 좋다!!

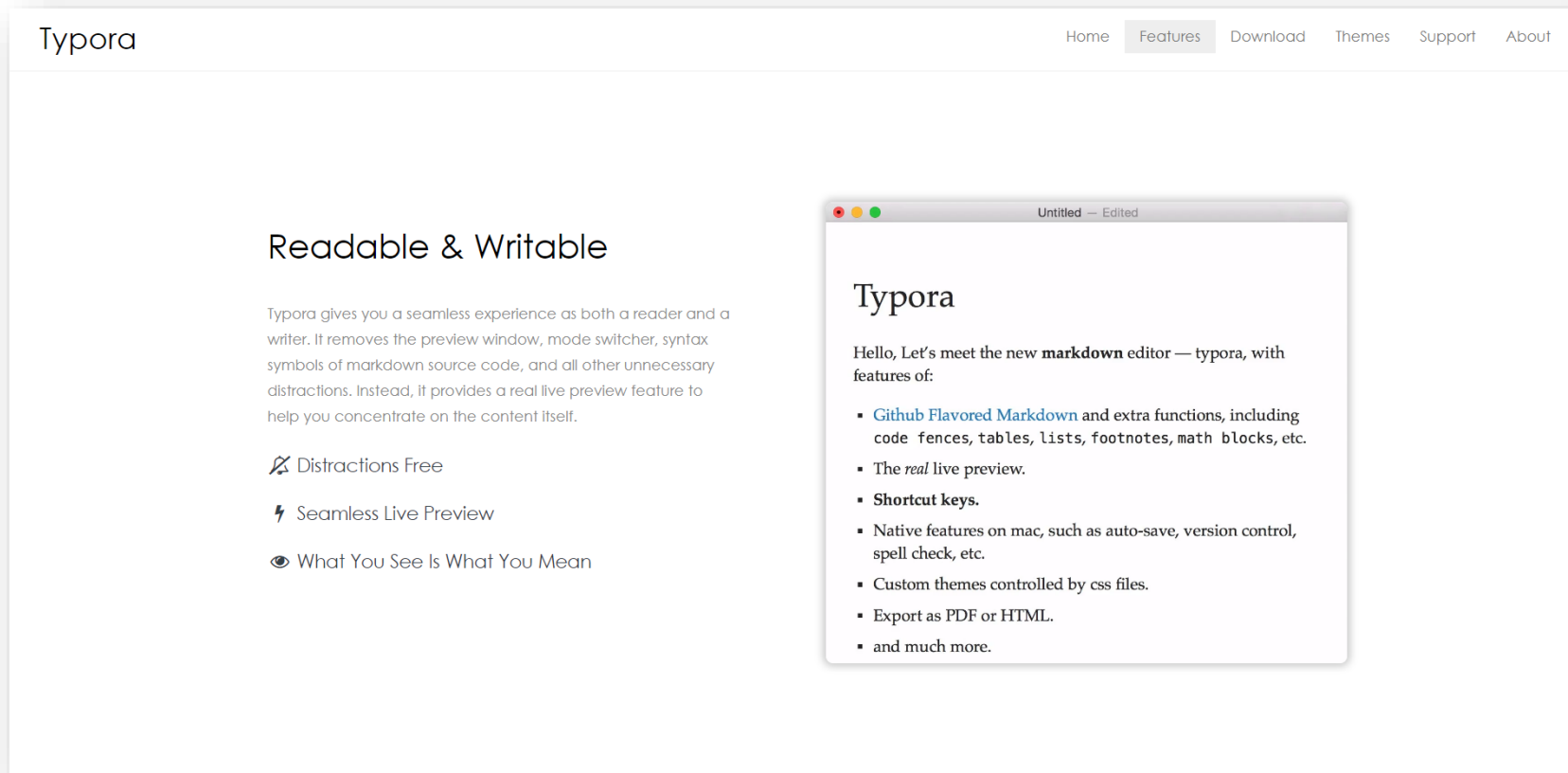


2. 자신이 했던 활동들 꾸준히 정리하기

Github에는 코드만 올릴 수 있는 것이 아니다!!

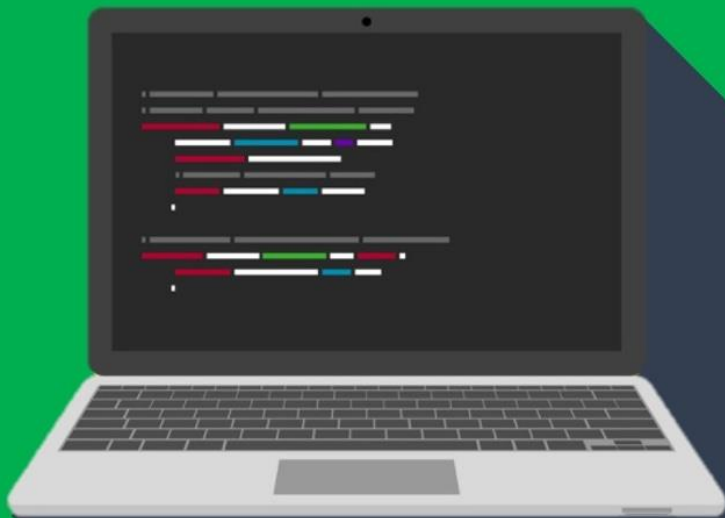
자신이 했던 활동들을 꾸준히 Commit해 나가는 습관 가지기

Portfolio 활용법



<https://typora.io/>

상명대학교 중앙동아리



INYRO

감사합니다!

@SMU_INYRO