



It's Your Life

with





git

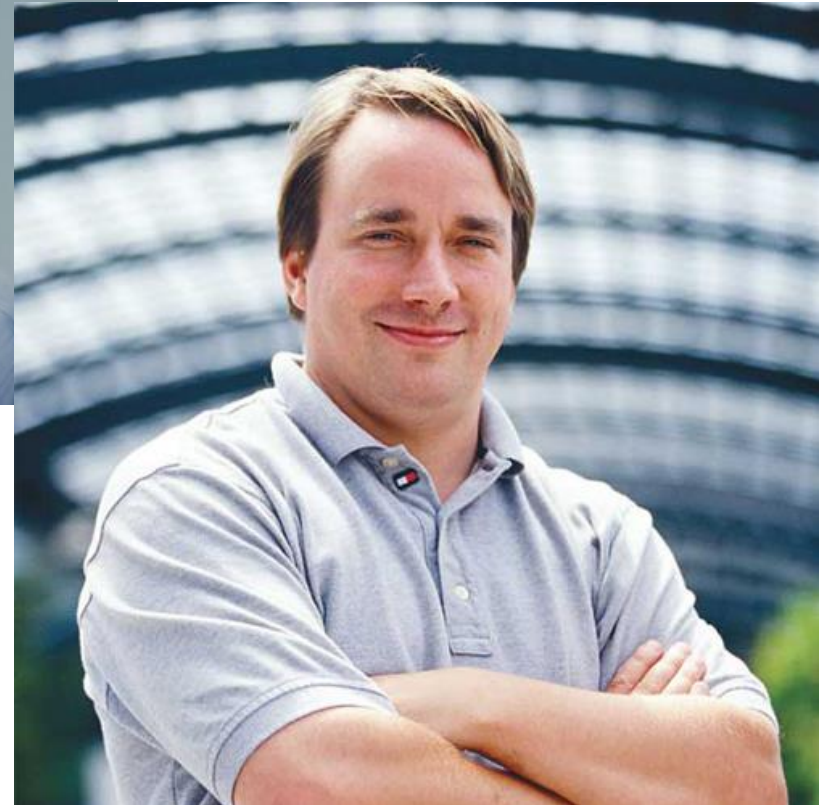


Git(깃)은 컴퓨터 파일의 변경사항을 추적하고
여러 사용자들 간에 해당 파일 작업을 조율하기 위한
대표적인 버전 관리 시스템(VCS)입니다.

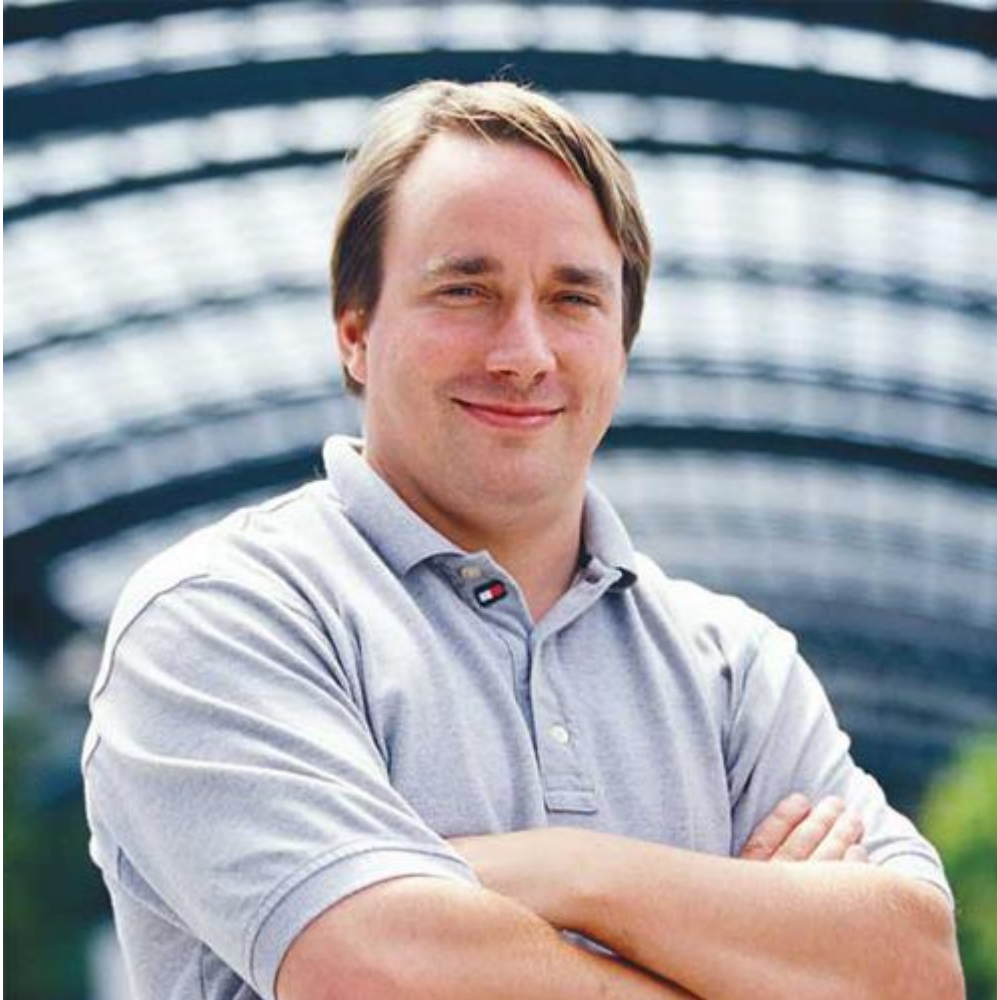
VCS(Version Control System)



세계 3대 개발자!?



Linus Benedict Torvalds



Linux(Linus + unix)



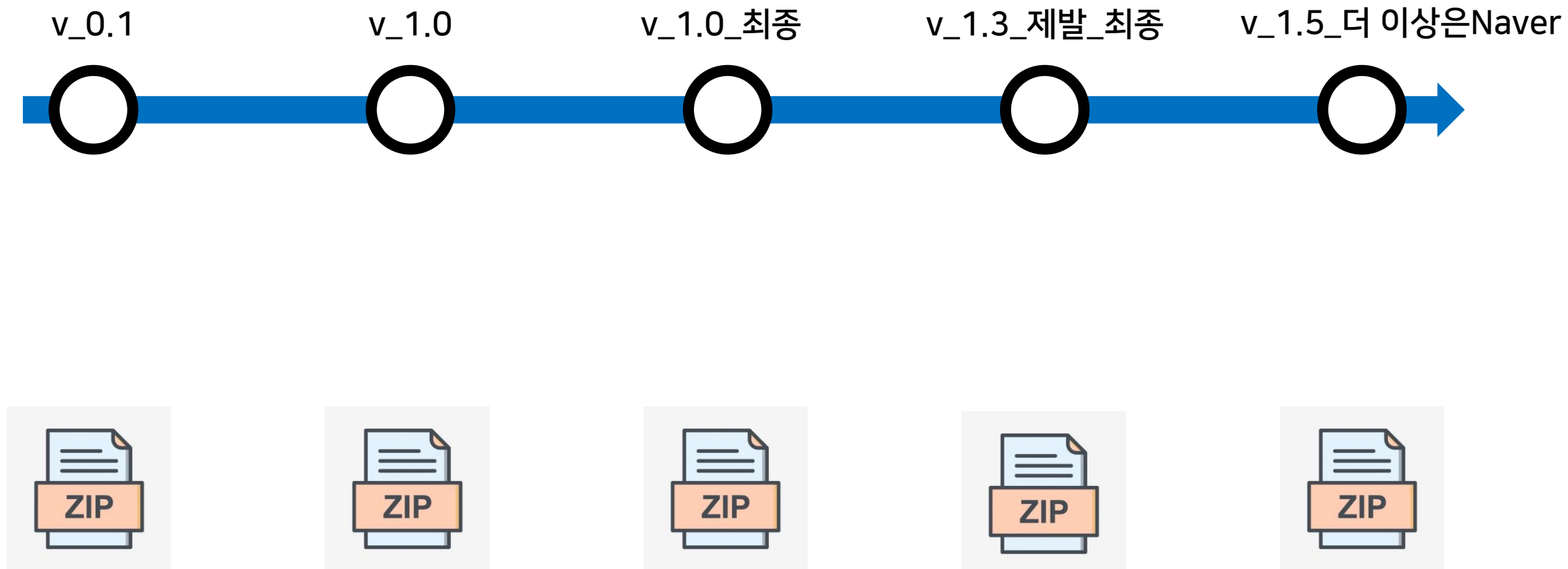
VCS(Version Control System)



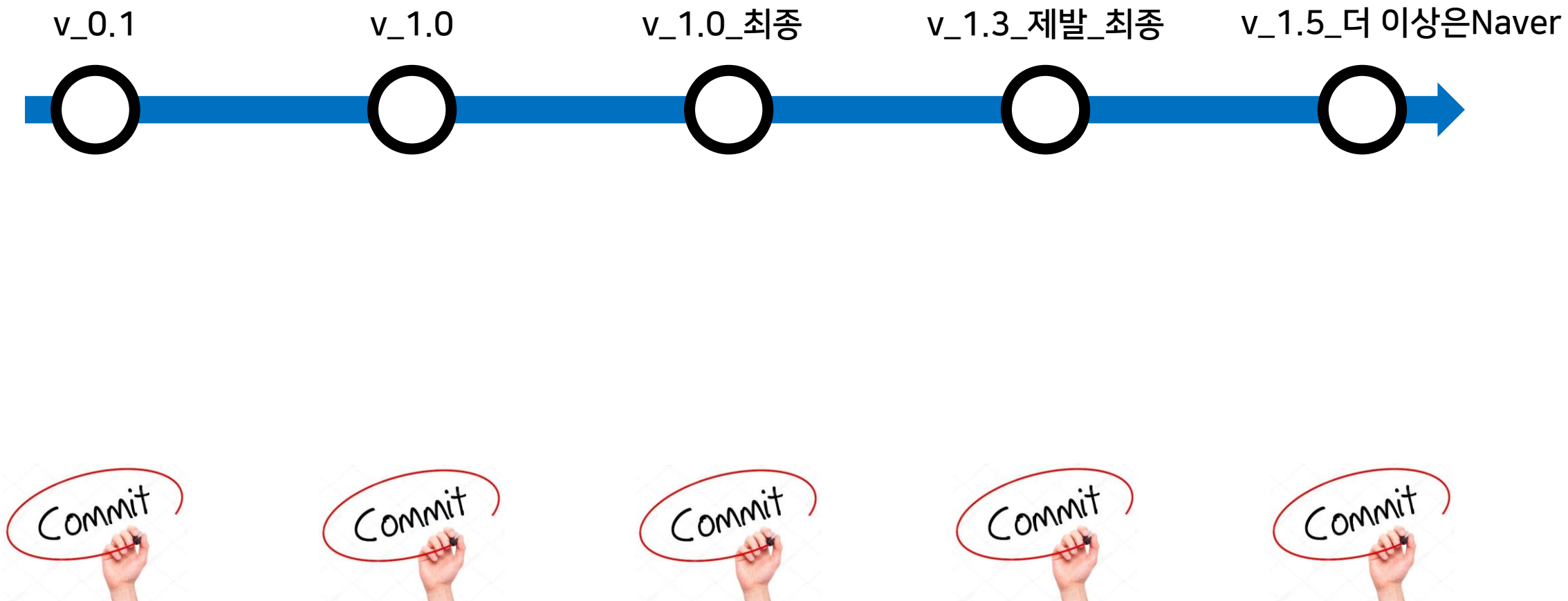
Git이 하는 일은!



버전 관리



버전 관리



Git의 특징!



- 폴더 단위로 관리! → 하나의 프로젝트 단위로 관리
- Git에 대한 모든 정보는 .git 폴더에 있습니다
 - 일종의 블랙 박스!
- 커밋은 로컬(= 여러분의 컴퓨터)에서만 일어나는 행위입니다!
- 따라서, 깃 허브는 여러분의 로컬에 있는 깃 정보를 온라인에 저장해 주는 서비스를 제공하는 것입니다!
- 깃 허브의 리포지토리는 여러분 로컬의 폴더의 개념과 같아요!



깃 설치하기!



<https://github.com/>



Desktop

<https://desktop.github.com/>

Git, 설치하기(Windows)



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

Download for Windows

[Click here to download](#) the latest (2.36.1) 64-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **24 days ago**, on 2022-05-09.

Other Git for Windows downloads

Standalone Installer

[32-bit Git for Windows Setup.](#)

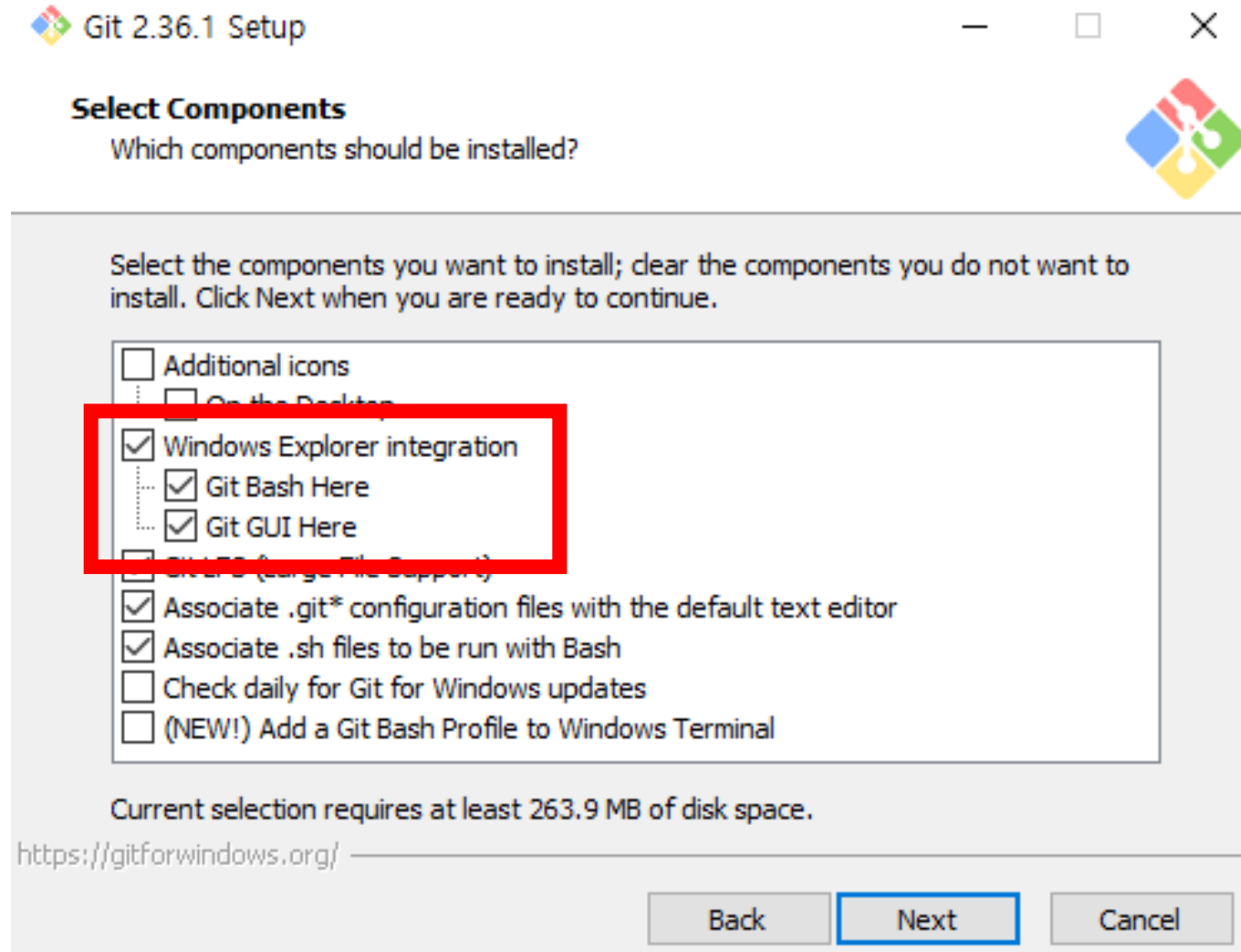
[64-bit Git for Windows Setup.](#)

Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

Git, 설치하기(Windows)



Git, 설치하기(Mac)



https://brew.sh/index_ko

brew install git



Commit?



project



index.html



main.css



favicon.png

로컬에 설치



git



사용자
(local, 로컬)

```
$ git init
```

```
# 현재 프로젝트에서 변경사항 추적(버전 관리)을 시작.
```

master



index.html



main.css



favicon.png



변경사항 추적 중..
(stage)



사용자
(local, 로컬)



```
$ git add index.html
```

변경사항을 추적할 특정 파일(index.html)을 지정.



사용자
(local, 로컬)

master



index.html



main.css



favicon.png

index.html

변경사항 추적 중..
(stage)



```
$ git add .
```

```
# 모든 파일의 변경사항을 추적하도록 지정.
```



사용자
(local, 로컬)

master



index.html



main.css



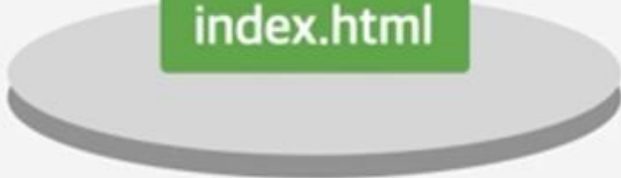
favicon.png



favicon.png

main.css

index.html



변경사항 추적 중..
(stage)




```
$ git commit -m '프로젝트 생성'  
# 메시지(-m)와 함께 버전을 생성.
```



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



프로젝트 생성



Git, 커밋 메시지!?



\$



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



main.js

파일 생성



프로젝트 생성



```
$ git add .  
# 모든 파일의 변경사항을 추적하도록 지정.
```



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



main.js



프로젝트 생성



```
$ git commit -m 'main.js 추가'
# 메시지(-m)와 함께 버전을 생성.
```



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



main.js



main.js 추가



프로젝트 생성



\$



사용자
(local, 로컬)

master



index.html

수정함



main.css

수정함



favicon.png



main.js



main.js 추가



프로젝트 생성



```
$ git add .
```

```
# 모든 파일의 변경사항을 추적하도록 지정.
```



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



main.js



main.js 추가



프로젝트 생성



```
$ git commit -m 'index.html 수정'
# 메시지(-m)와 함께 버전을 생성.
```



사용자
(local, 로컬)

master



index.html



main.css



favicon.png



main.js

- index.html 수정
- main.js 추가
- 프로젝트 생성

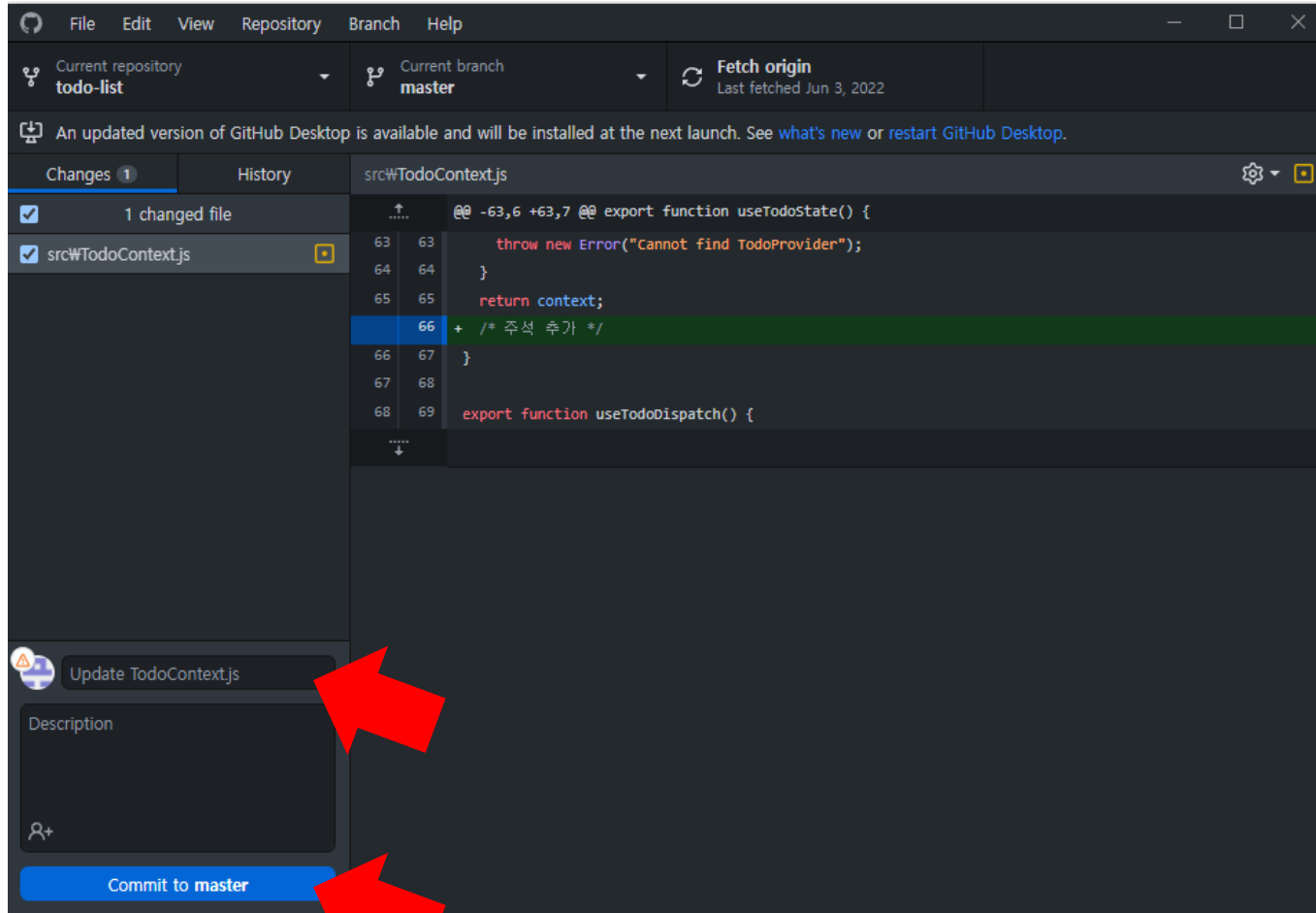


Git, 커밋하기 on CLI



- git status
- Untracked files?
- git add 가 추천 되네요?
- git add .
- git status
- git commit -m “커밋 메세지”
- git log

Git, 커밋하기 on GUI





Push!?



```
...login.credentials (logged:#input.new(c
// script src= address [statu
[lock.command]#>>access:denial // scri
then script src=[true] {?unkno
ction logged:#input.false function logged:#
script src=[true] {?unknown) m#4:80a?
script src=[true] local.config
delay.input <cham>= {d fg#6 m#4:h61l0
//script src= address [status?] code<
name[>#>>access:denial // script src=[error]
script src=[true] {?unknown) m#4:80a?/
status, omm
(245,23,068,789,
nname<img>=spa
put.new(create))
atus?] code< [tr
t src=[erro
icl statu
onfig sc
onfig sc
```







GitHub



GitHub
(remote, 원격)

원격 저장소
(Repository)



사용자
(local, 로컬)



GitHub
(remote, 원격)

```
$ git remote add origin https://github.c...  
# origin이란 별칭으로 원격 저장소를 연결.
```

master



index.html



main.css



favicon.png



main.js



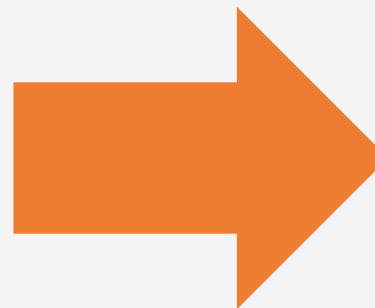
index.html 수정



main.js 추가



프로젝트 생성



원격 저장소
(Repository)



사용자
(local, 로컬)



GitHub
(remote, 원격)

```
$ git push origin master
```

origin이란 별칭의 원격 저장소로 버전 내역 전송.

master

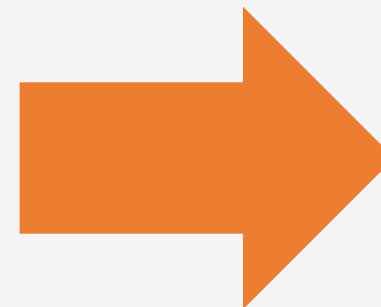
 index.html

 main.css

 favicon.png

 main.js

● index.html 수정
● main.js 추가
● 프로젝트 생성



원격 저장소
(Repository)

● index.html 수정
● main.js 추가
● 프로젝트 생성

Push, 온라인에 깃을 저장하자!



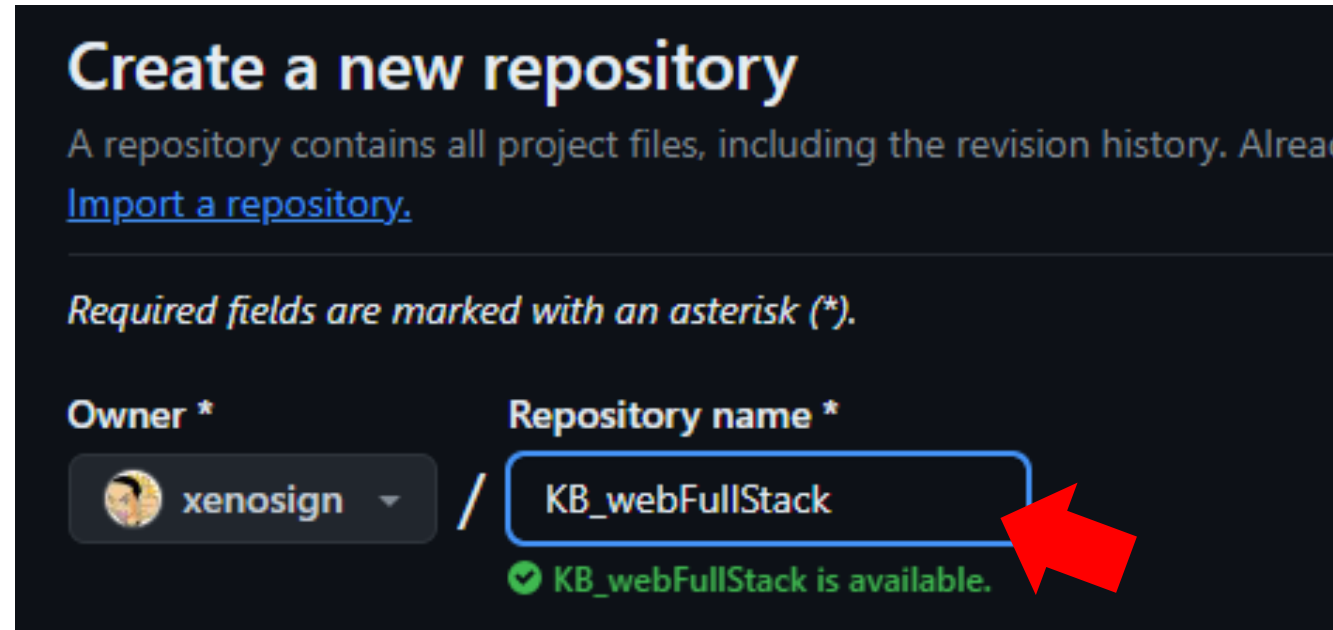
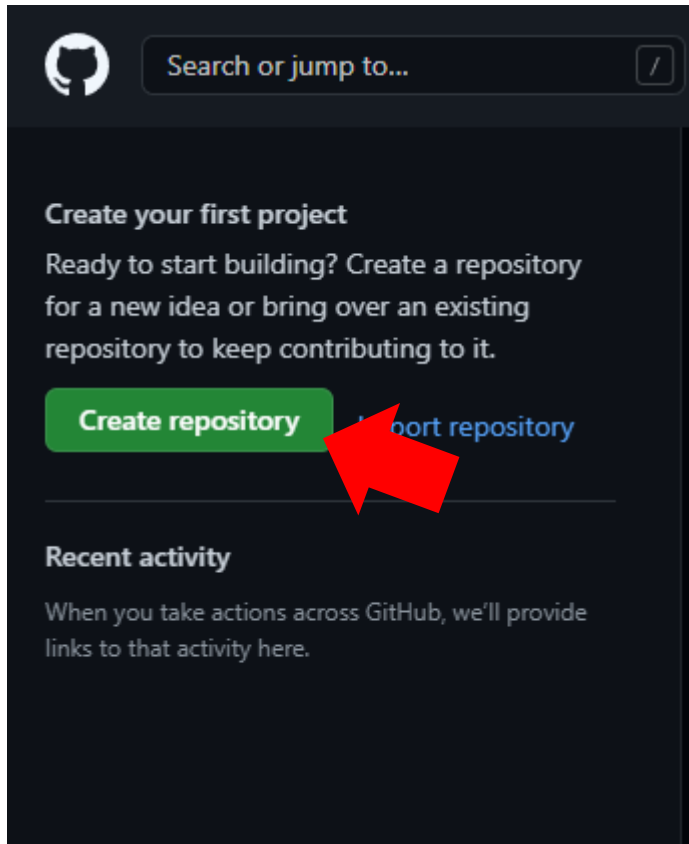
- Commit 은 여러분의 컴퓨터에서만 일어나는 일입니다!
- 즉, 온라인에 해당 내역을 올리지 않으면 github은 아무것도 모르는 거죠

Push, 온라인에 깃을 저장하자!



- Github에 여러분의 커밋 내역을 올리려면 반드시 push를 해주어야 합니다!
- Git이 관리하는 폴더 하나(=프로젝트) → github 리포지토리 하나
 - 두 개를 하나에 올리면, 코드의 변화를 추적하는 깃의 특성상 황당하겠죠?
 - 자동차 2대의 블랙 박스를 하나의 블랙 박스에 넣는 것과 비슷합니다!

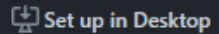
Github 에 올리기! - Repo 생성



Github 에 올리기! - 안내 따라하기!



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



Set up in Desktop

or

HTTPS

SSH

https://github.com/xenosign/KB_webFullStack.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 "# KB_webFullStack" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/xenosign/KB_webFullStack.git
git push -u origin main
```



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

```
git remote add origin https://github.com/xenosign/KB_webFullStack.git
git branch -M main
git push -u origin main
```



- 상황에 맞는 코드를 복사해서 Vscode 에 붙여 넣기!

Github 에 올리기!(Push)



- `echo "# ss" >> README.md`
- `git init`
- `git add .`
- `git commit -m "커밋 메세지"`
- `git branch -M main`
- `git remote add origin https://github.com/TetzSpreatics/ss.git`
- `git push -u origin main`

블랙 박스와 비슷합니다!






- 하나의 프로젝트 = 하나의 폴더 = 한대의 차
- A 라는 차가 사고가 났을 때, B 라는 차의 블랙 박스를 확인하면 될까요?
- 또, 급하다고 해서 C 라는 차의 블랙 박스 데이터를 D 라는 차에 덮어 씌워도 될까요?




모를 심어 봅시다!





Signed in as
TetzSpreatics

 Set status

Your profile

Your repositories

Your codespaces


Your organizations

Your projects

Your stars

Your gists

Upgrade

Feature preview 

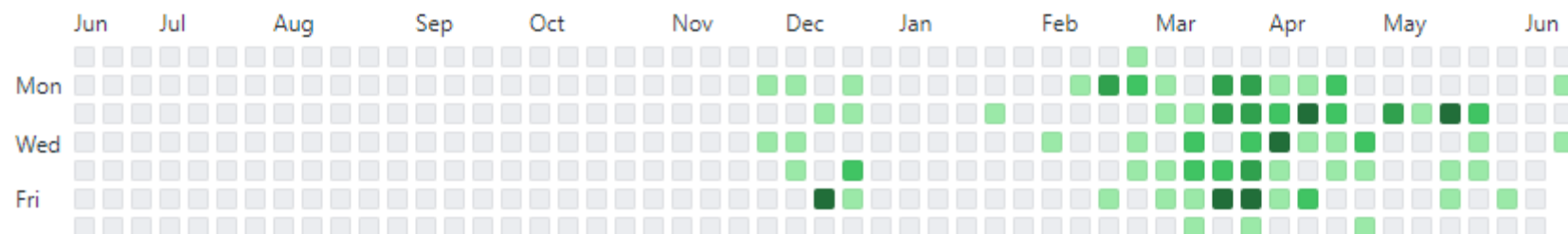
Help

Settings

Sign out

238 contributions in the last year

Contribution settings ▾



[Learn how we count contributions](#)

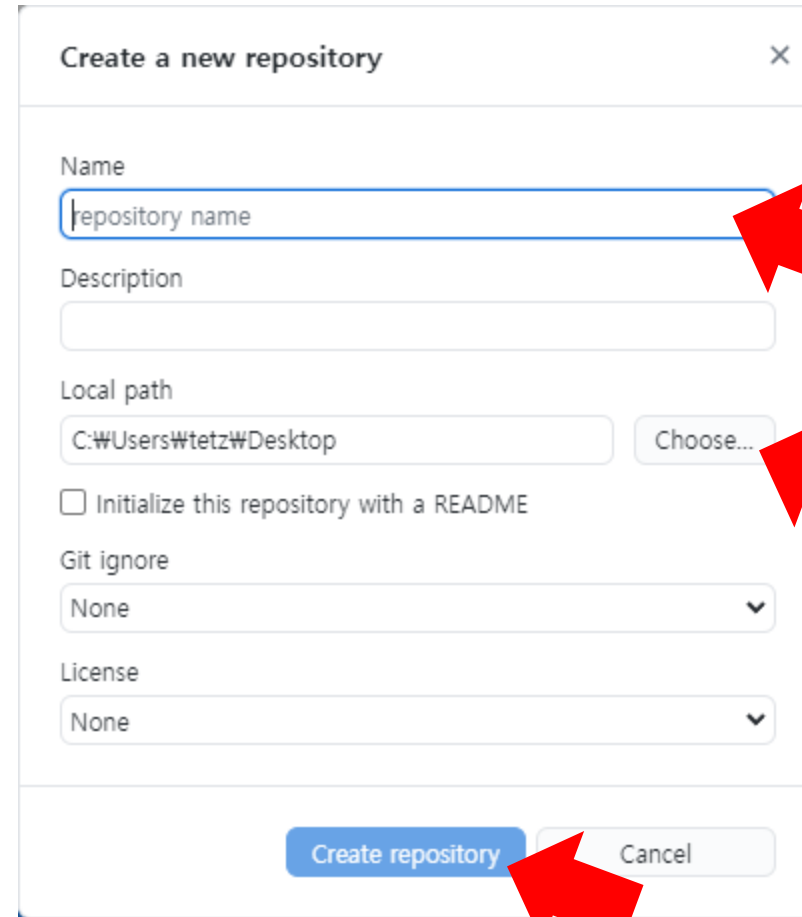
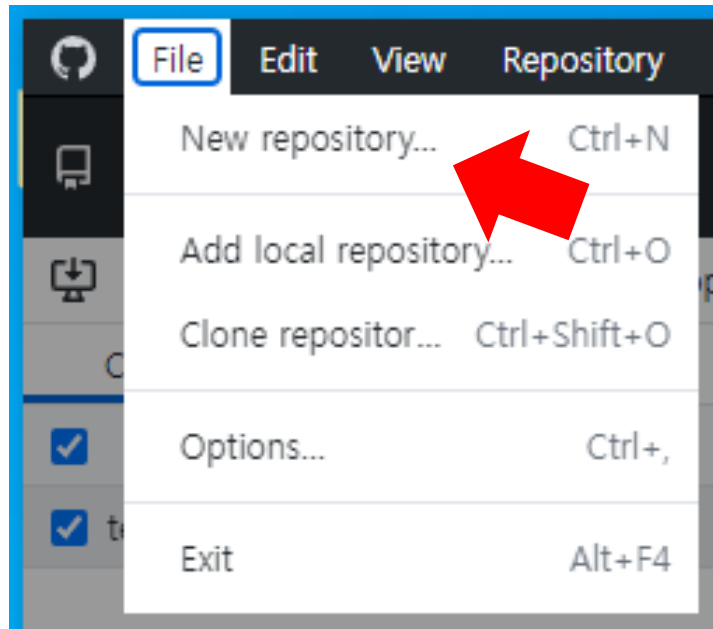
Less     More

NEW! [View your contributions in 3D, VR and IRL!](#)



귀찮으면

Github desktop





Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

Always available in the toolbar for local repositories or **Ctrl** **P**

Publish repository



Publish repository



GitHub.com

GitHub Enterprise

Name

KB_webFullStack_GUI

Description



Keep this code private

Publish repository

Cancel





Open the repository in your external editor

Select your editor in [Options](#)

Repository menu or `Ctrl` `Shift` `A`

Open in Visual Studio Code

