

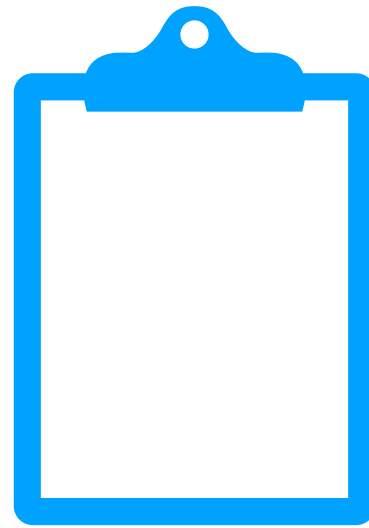
Matlab with Git

Dept of Convergence EE, 너몽

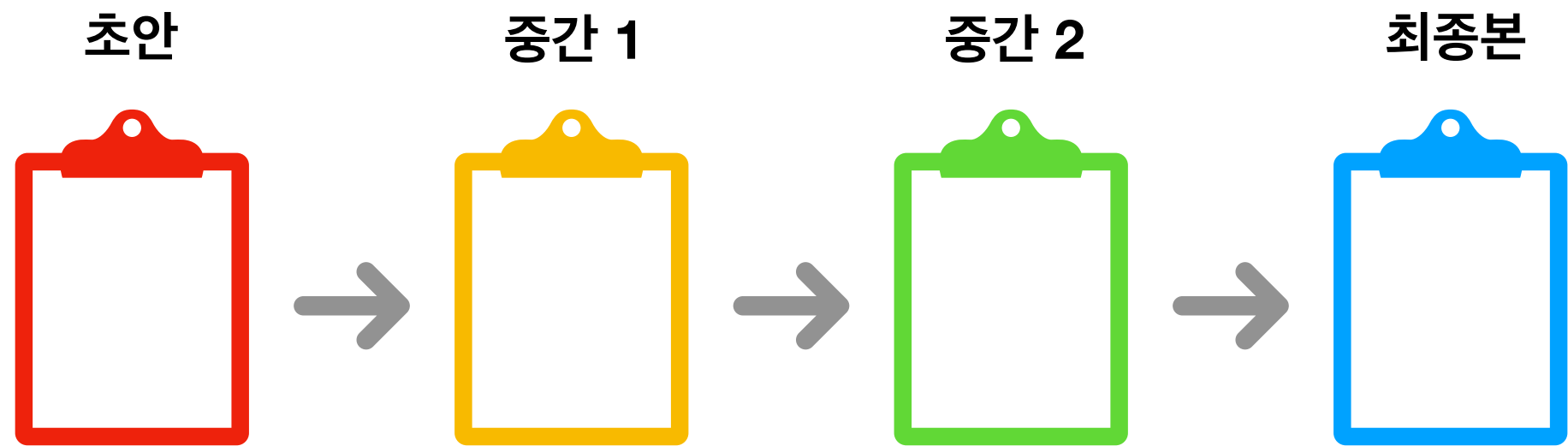
수업자료 모음

<https://github.com/Tanukimong/openSeminar>

Why Git?

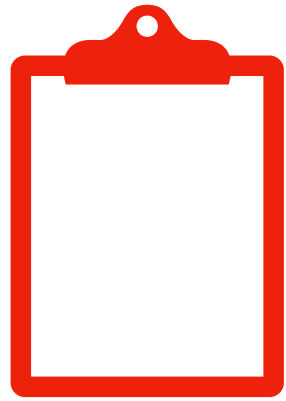


레포트 하나를 작성한다고 가정해봅시다

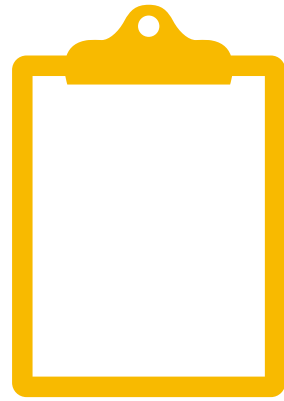


우리가 생각하는 이상

초안



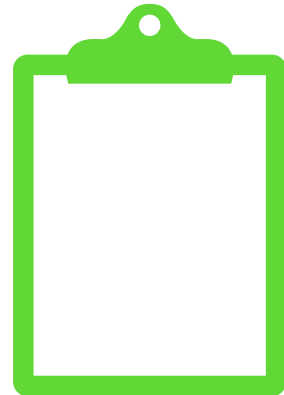
중간 1



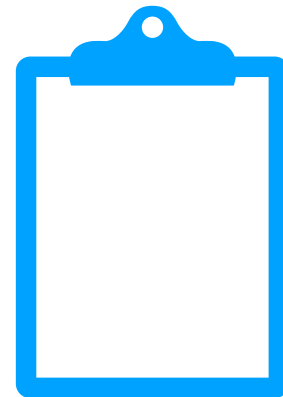
중간 완성



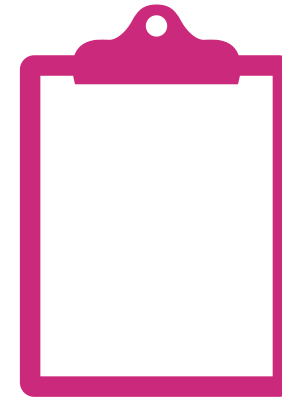
임시 수정 1



임시 수정 최종



임시 수정 최종 마지막



현실

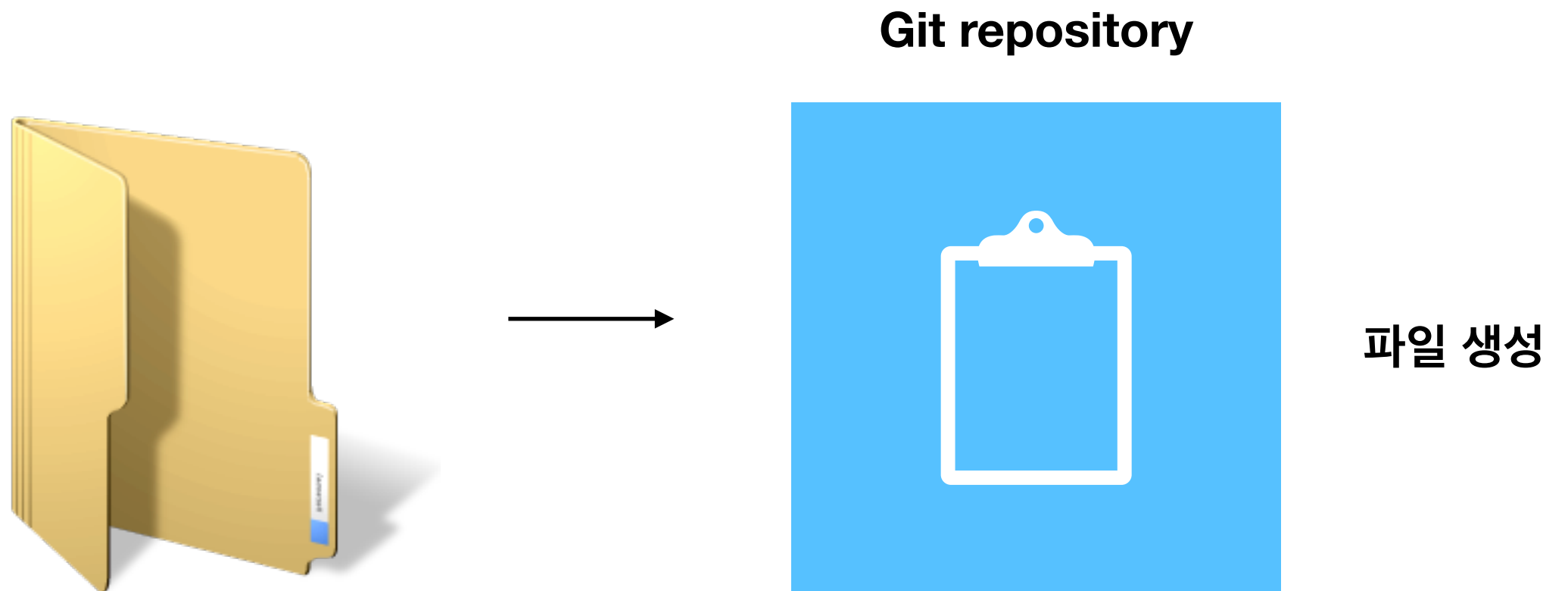
마지막 파일? 차이? 되돌리기?

문제점

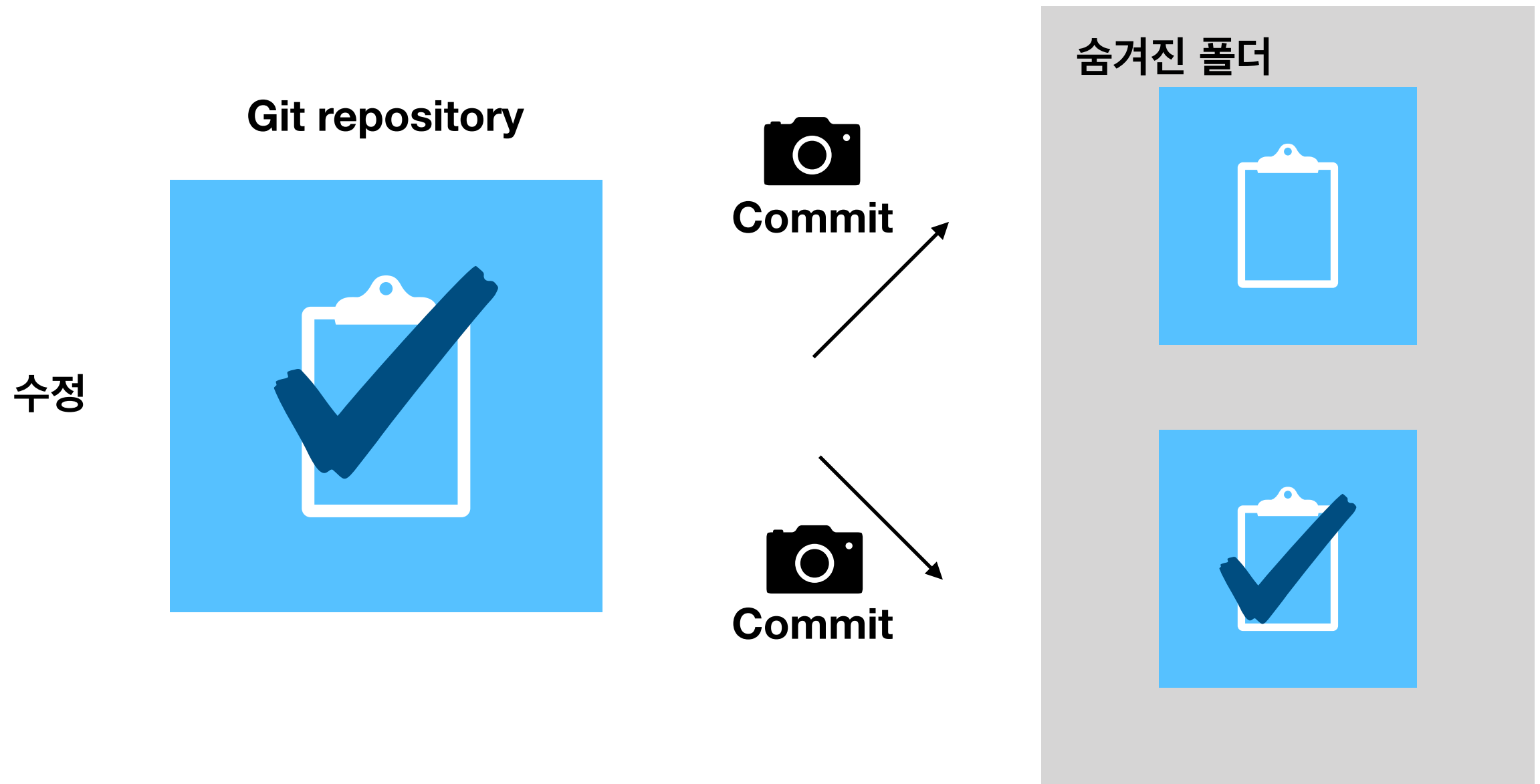
- 어떤 파일이 진짜 마지막 파일?
- 각 파일 별로 차이가 뭐지?
- 되돌리고 싶은데...

Git

How Git operates?



How Git operates?



일반 저장이란 차이가 있나?

Difference

- 헛갈림 방지
- 언제든지 그때 그 시절로
- **Branch**
- **Merge**

The screenshot shows the Git GUI interface. At the top, there are icons for '시' (Show), '페치' (Fetch), '브랜치' (Branch), '병합' (Merge), '스태시' (Stash), '원격 보기' (Remote View), '파인더에서 보기' (View in Finder), and '터미널' (Terminal). Below these are filters for '모든 브랜치' (All branches), '원격 브랜치 표시' (Show remote branches), and '원형 순' (In order). The main area displays a commit history with columns for '그래프' (Graph), '설명' (Description), '커밋' (Commit), and '날짜' (Date). The bottom section shows a diff for 'README.md' with a search bar and a '코드문치 되돌리' (Reset code) button.

그래프	설명	커밋	날짜
	master origin/master Merge branch 'master' of https://github.com...	4642a54	어제 오후 4:21
	수업자료 과정 수정	6f41686	어제 오후 1:21
	후에 Git을 배우는 부분을 제거	54171b4	어제 오후 1:20
	행렬 부분 추가	a569dba	어제 오후 4:20
	용량 최적화	e06bf77	어제 오전 12:33
	Merge branch 'master' of https://github.com/Tanukimong/openSeminar	6a035c5	어제 오전 12:22

Below the commit history, there is a section for 'README.md' showing a diff. The diff includes a search bar and a '코드문치 되돌리' (Reset code) button. The diff content shows a change in the 'README.md' file, with a section titled '1 부분 : 2-10 줄' (1 part : 2-10 lines). The diff content is as follows:

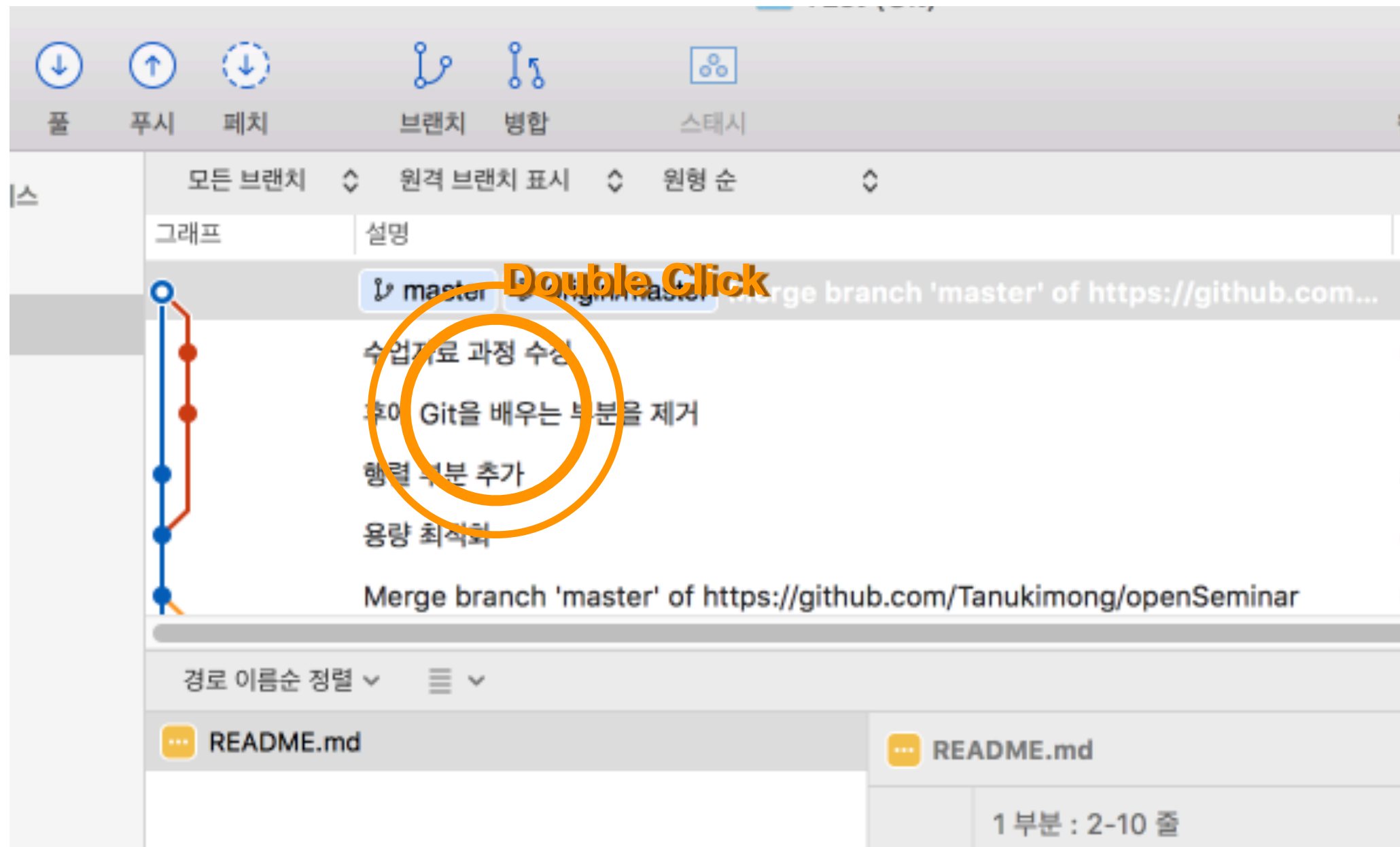
```

2 2
3 3   짝짓 주제 : Matlab & Github
4 4
5 5   - ## 팁

```

헛갈림 방지

파일의 발전 과정을 한눈에



그때 그 시절로

언제든지 원하던 버전으로 복귀 가능

Branch

“뺨셈 기능을 추가하고 싶어!”



뺨셈 계산 프로그램 완성!



“헉 근데 추가하다가 뺨셈도 안되면 어떡하지..”

Case with no Git

Branch



덧셈 계산 프로그램



복제된 프로그램

“뺄셈 기능을 추가하고 싶어!”



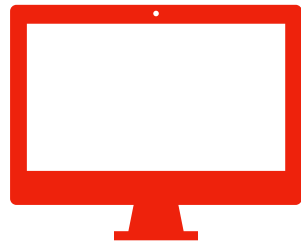
“헉 근데 추가하다가 덧셈도 안되면 어떡하지..”

Case with no Git

Branch



덧셈 계산 프로그램



수정 중인 프로그램

“뺄셈 기능을 추가하고 싶어!”



“헉 근데 추가하다가 덧셈도 안되면 어떡하지..”

Case with no Git

Branch



덧셈 계산 프로그램



덧셈 + 뺄셈 프로그램

“뺄셈 기능을 추가하고 싶어!”



“헉 근데 추가하다가 덧셈도 안되면 어떡하지..”

Case with no Git

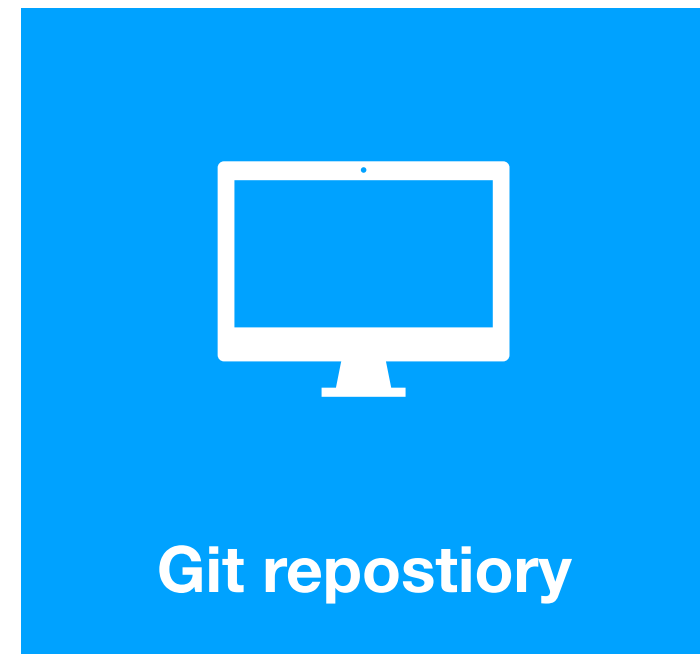
Branch

Branch의 일반적인 의미?
in Computer



[명사] 나뭇가지 , 분기

Branch



Case with Git

Branch

Branch **예비**

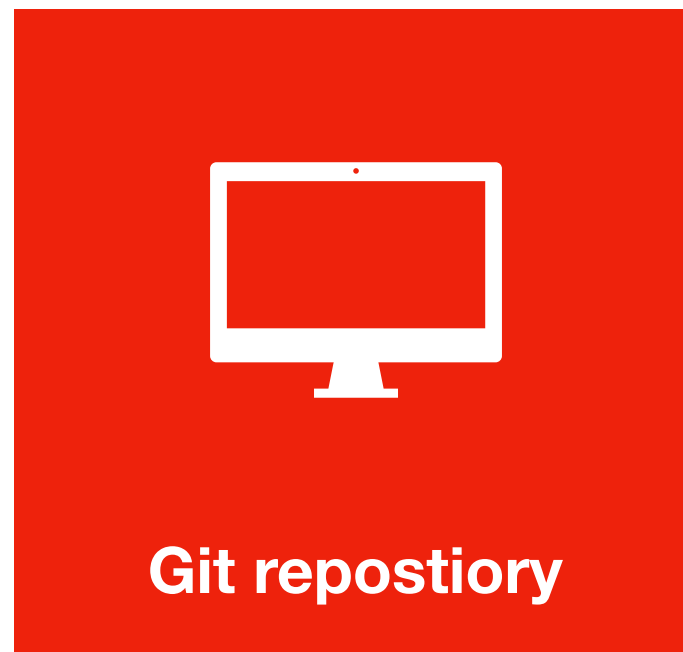


Git repostiory

Case with Git

Branch

checkout 예비



Git repostiory

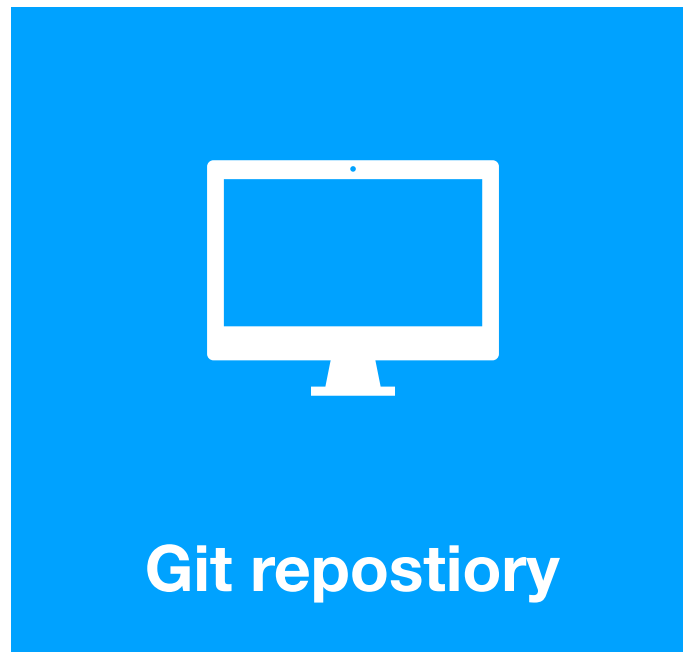
엇 망했다
덧셈도 아예 안되네 ㅠㅠ



Case with Git

Branch

checkout **master**



오 빨짓하기 전으로 돌아왔다



Case with Git

“그냥 되돌리기를 쓰면 안되나?”

두 가지 이유

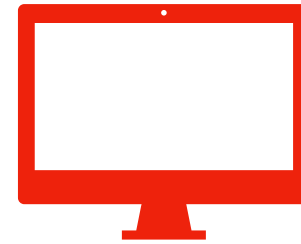
- 아주 많은 기능을 동시에 추가 (Ex. 협업)
- Merge

아-주 많은 기능 추가

in 웹하드



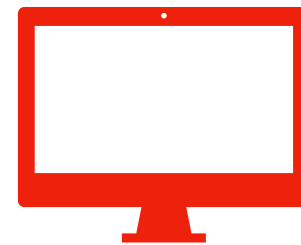
덧셈 계산 프로그램



빨셈 추가 중



곱셈 추가 중



인쇄 기능 구현 중



아-주 많은 기능 추가



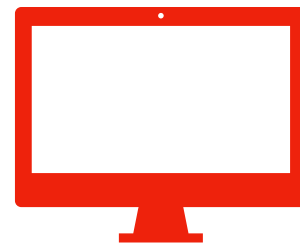
덧셈 계산 프로그램



빨셈 추가 완료!



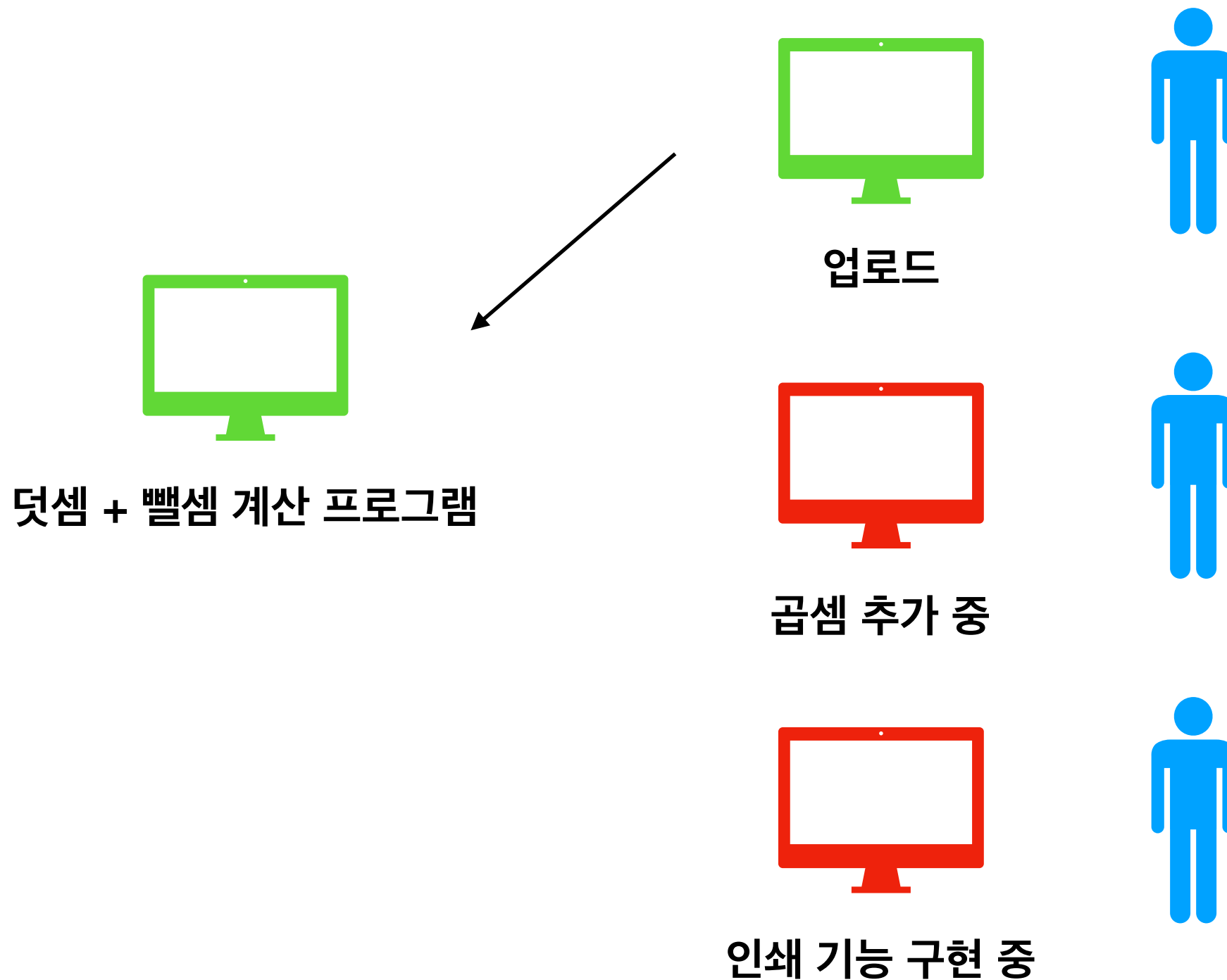
곱셈 추가 중



인쇄 기능 구현 중



아-주 많은 기능 추가



아-주 많은 기능 추가



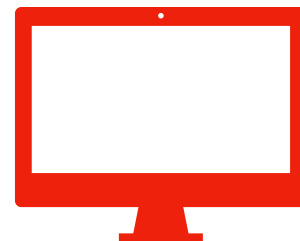
덧셈 + 뺄셈 계산 프로그램



아싸 퇴근 ㅋㅋ



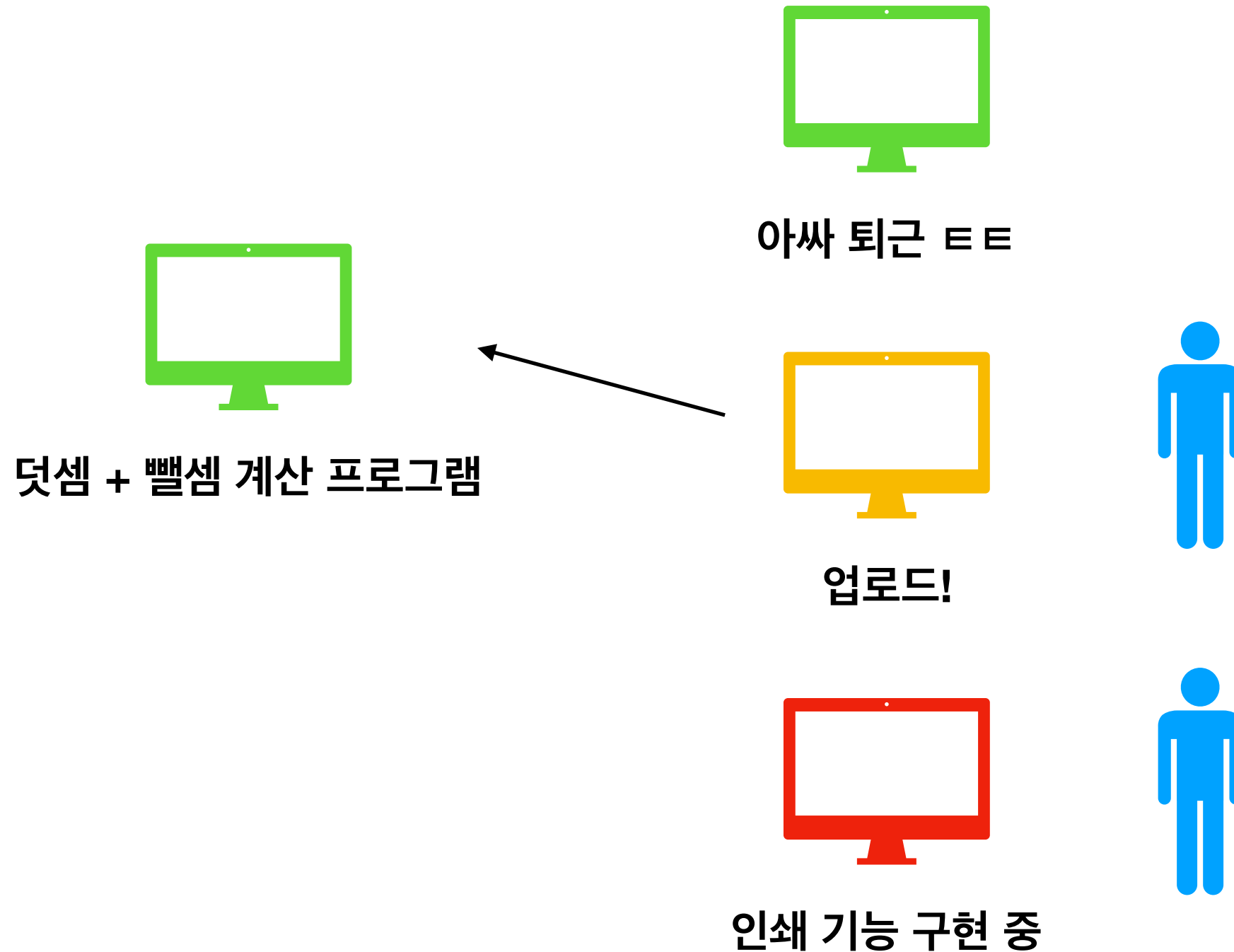
곱셈 추가 완료!



인쇄 기능 구현 중



아-주 많은 기능 추가



아-주 많은 기능 추가

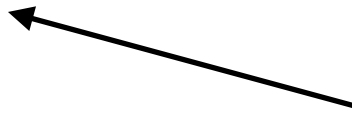
어? 분명 완성해서 올렸는데?



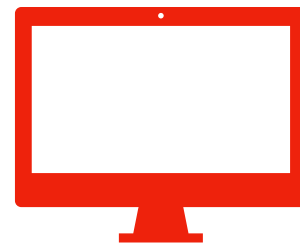
아싸 퇴근 ㅋㅋ



덧셈 + 곱셈 계산 프로그램



업로드!



인쇄 기능 구현 중



Merge

아-주 많은 기능 추가



Remote Git repository
(GitHub)

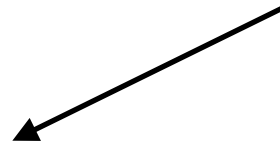


아-주 많은 기능 추가



Remote Git repository
(GitHub)

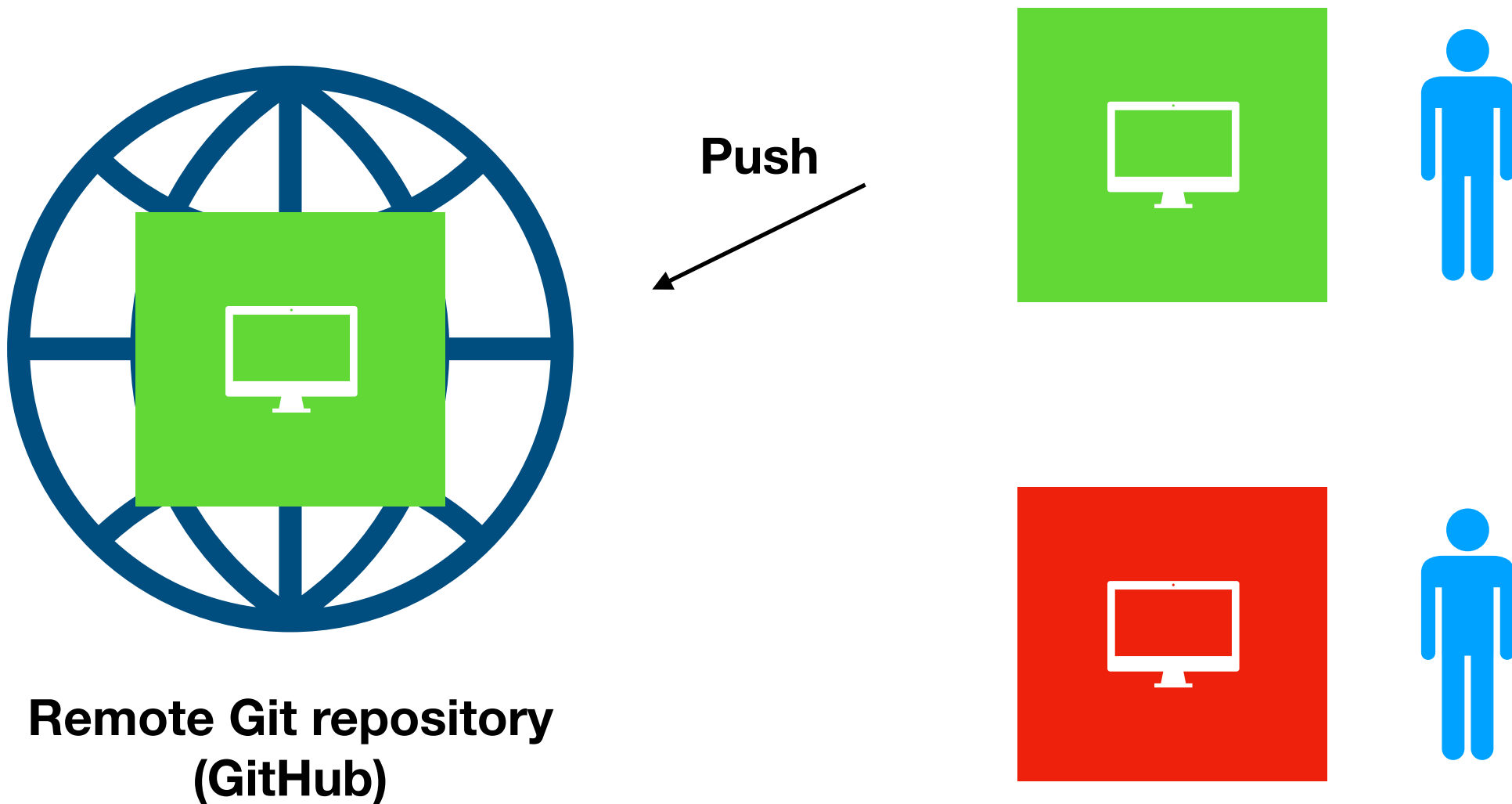
Push



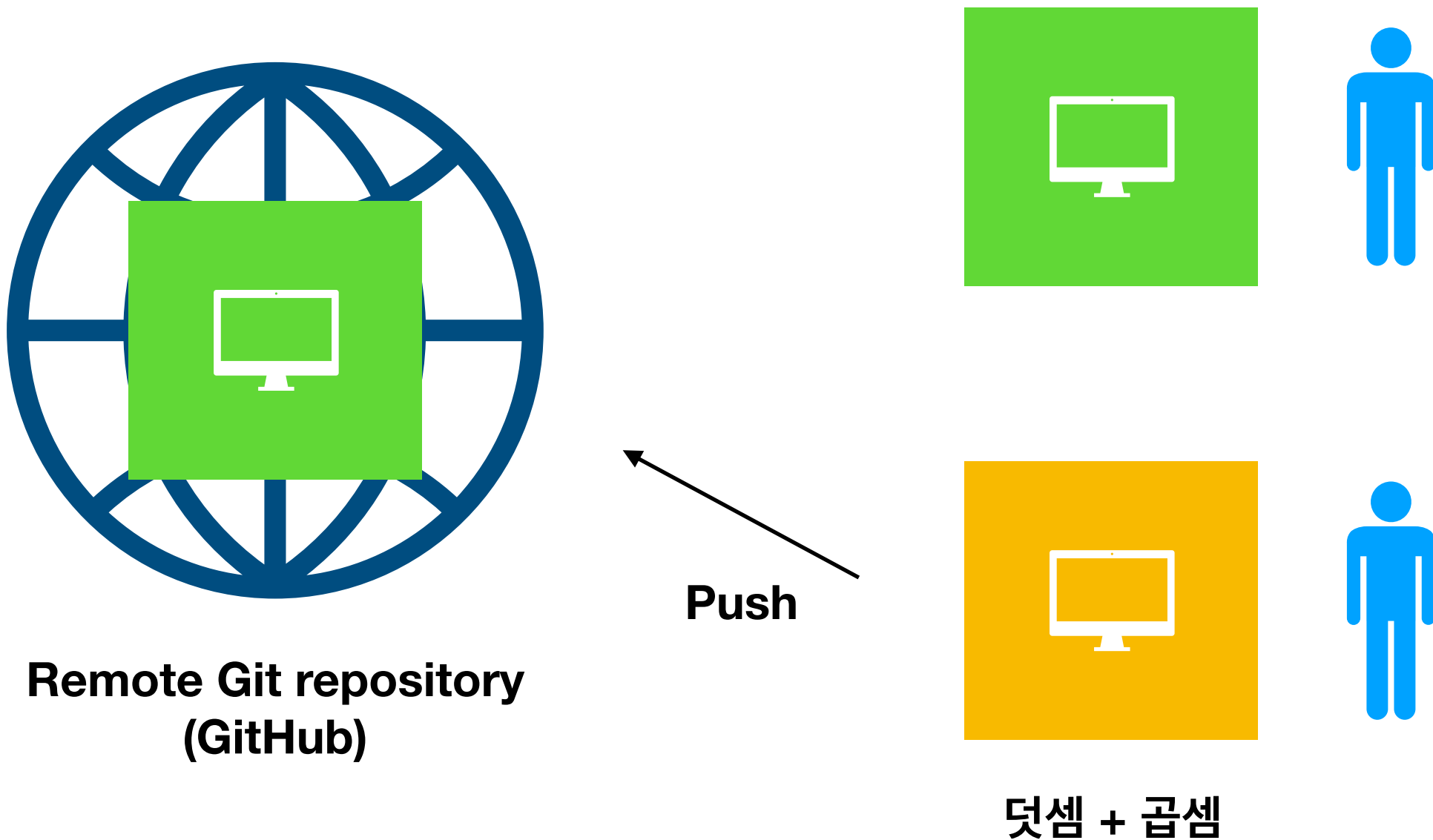
덧셈 + 뺄셈



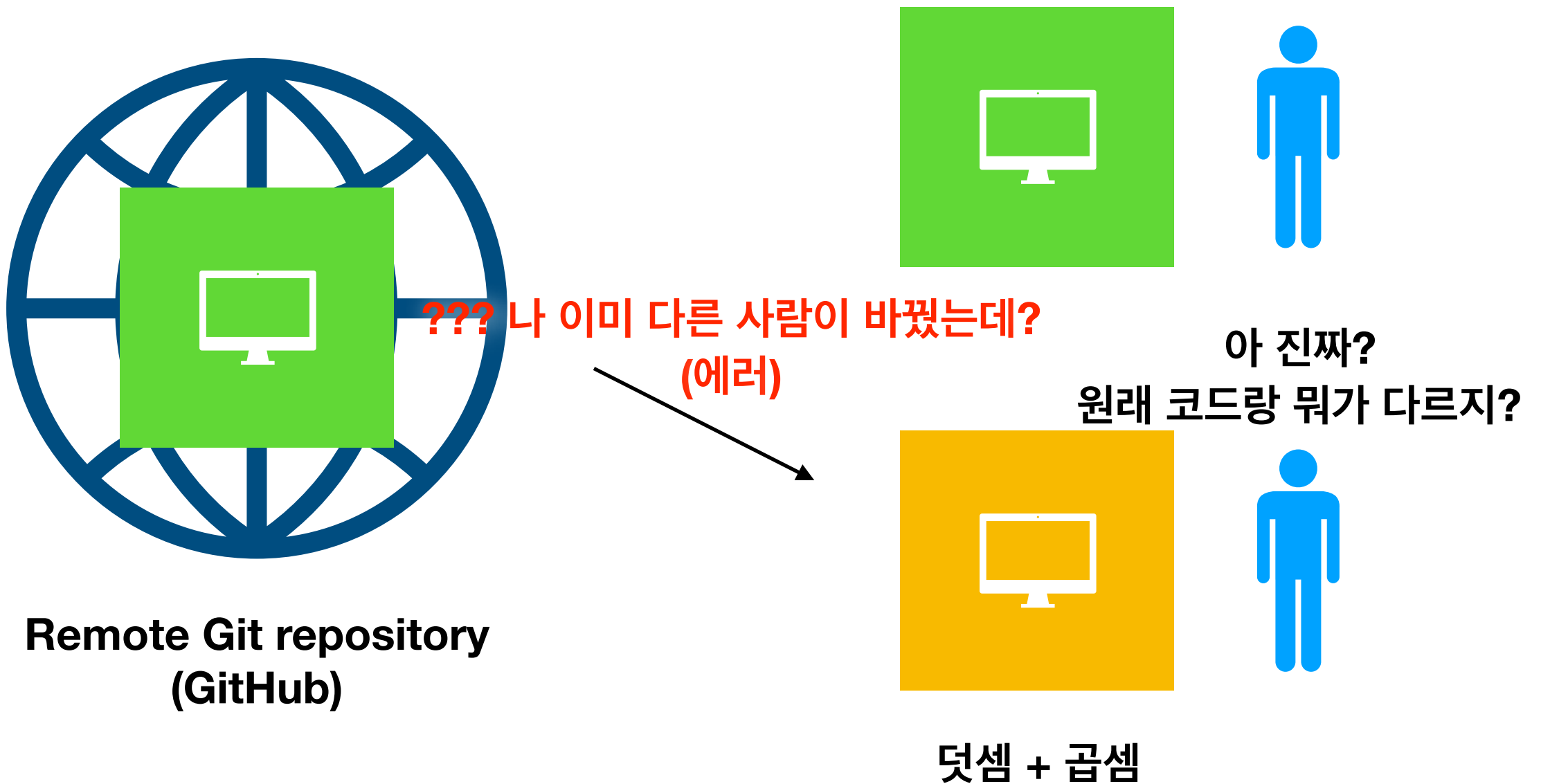
아-주 많은 기능 추가



아-주 많은 기능 추가



아-주 많은 기능 추가



아-주 많은 기능 추가



Remote Git repository
(GitHub)

Fetch
(Download와 비슷)



일단 가져와보자



덧셈 + 곱셈

아-주 많은 기능 추가



Remote Git repository
(GitHub)



Merge



덧셈 + 곱셈

아-주 많은 기능 추가



Remote Git repository
(GitHub)



Merge



Error → 자신의 현재 코드와 다른 부분을 표시해줌

아-주 많은 기능 추가



Remote Git repository
(GitHub)



아. 이렇게 바꿨구나.
이거에 맞게 다시 고쳐야지



Error → 자신의 현재 코드와 다른 부분을 표시해줌

아-주 많은 기능 추가



Remote Git repository
(GitHub)



좋아. 고쳤다.



아-주 많은 기능 추가



Remote Git repository
(GitHub)

Push



좋아. 고쳤다.



아-주 많은 기능 추가

“흠. 이번 건 **빨셈 기능**이 추가된 버전을
토대로 만든게 맞군. Ok. 승인”



Remote Git repository
(GitHub)

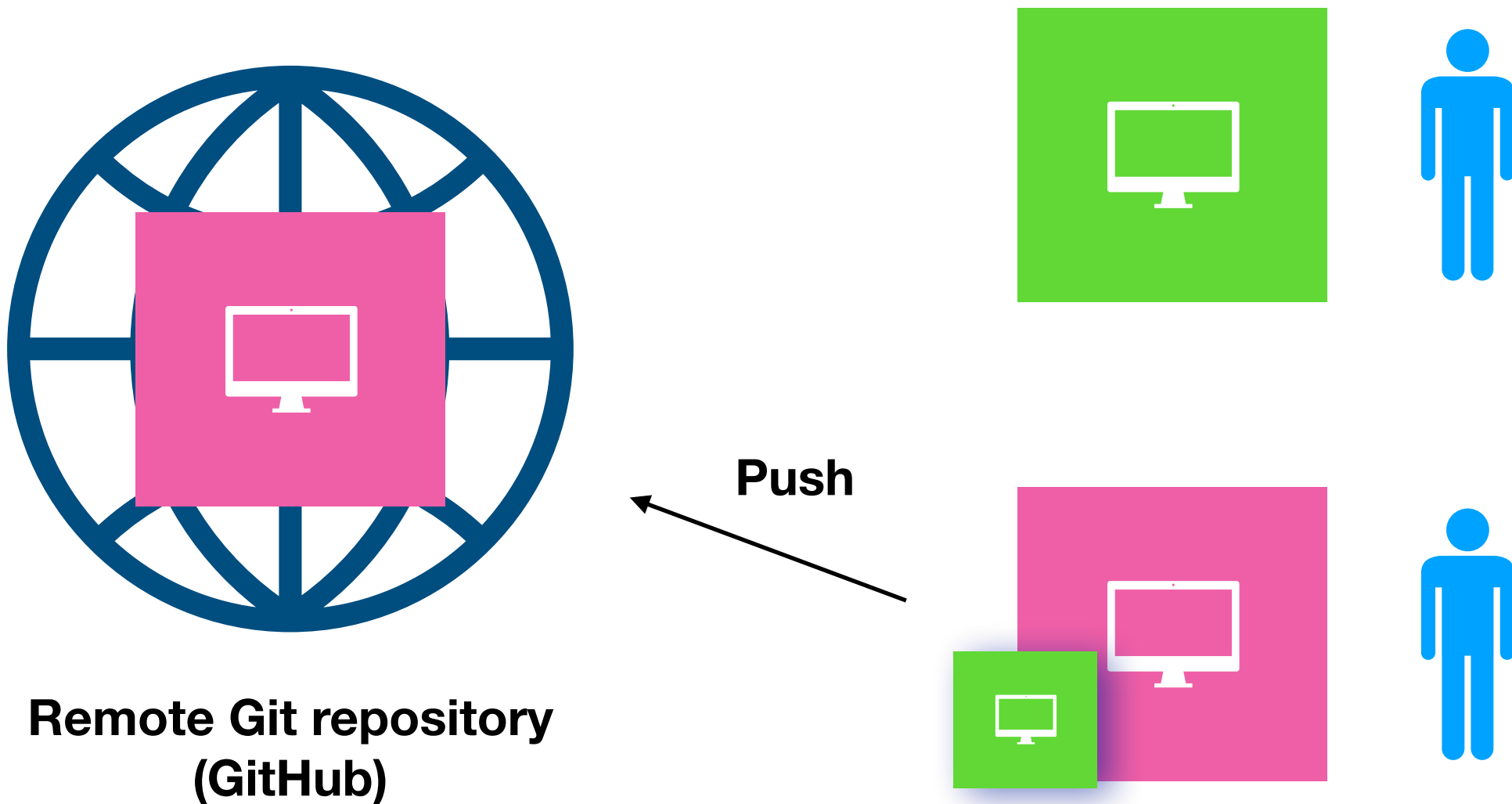
Push



좋아. 고쳤다.



아-주 많은 기능 추가



실습 시작!

- Student 혜택 Github ID 만들기
 - <https://education.github.com/pack>
- Sourcetree 설치
 - <https://www.sourcetreeapp.com>
- Atom 에디터 설치 (옵션)
 - <https://atom.io>

SourceTree 설치

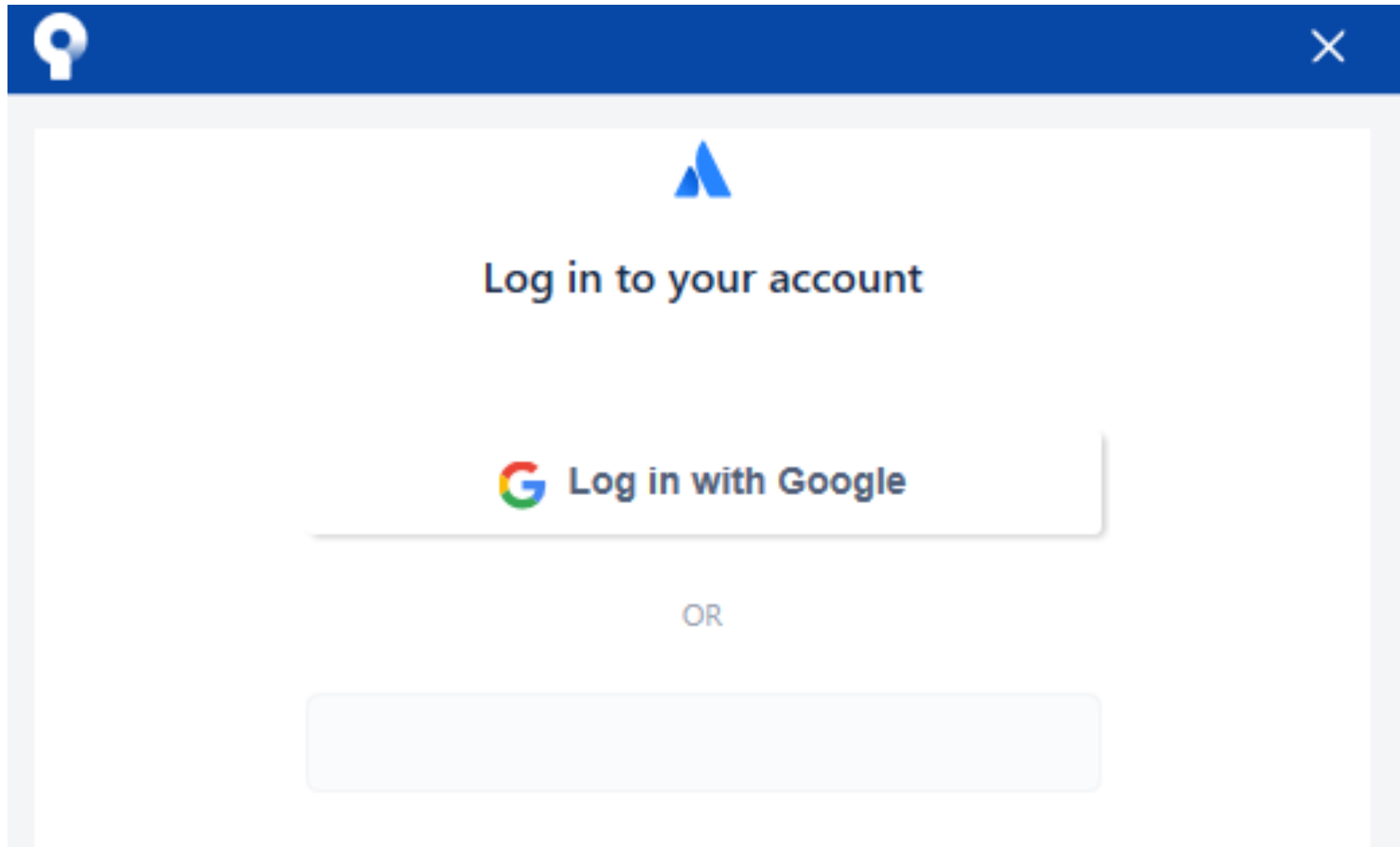
1. <https://www.sourcetreeapp.com> 에서

설치파일을 다운로드 받아 실행

2. 설치를 진행하다가 Bitbucket/Atlassian선택화면이 나오면

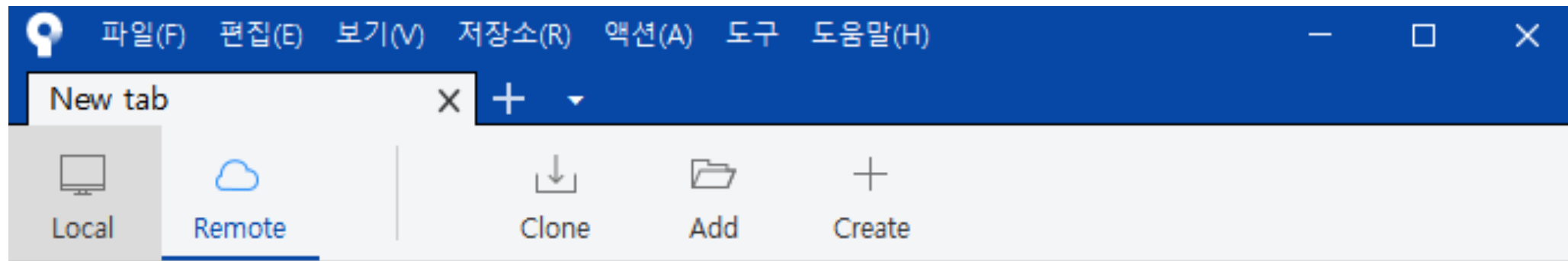
Atlassian 선택

3. Google 계정을 이용하여 로그인



Atlassian 선택 이후

Log in with Google 클릭 후, 안드로이드 계정 이용하여 로그인



원격 저장소

+ 계정 추가...

검색

☒ Show Organization Repos

새로고침

계정 변경

‘계정 추가’

계정 추가를 눌러 Github 계정 등록

호스팅 계정 편집

Host

호스팅 서비스: GitHub

호스트 URL: https://github.com/

선호 프로토콜: HTTPS

Credentials

인증: Basic

사용자명: Tanukimong

비밀번호 새로고침

Authentication failed for GitHub with username - please check your account settings.

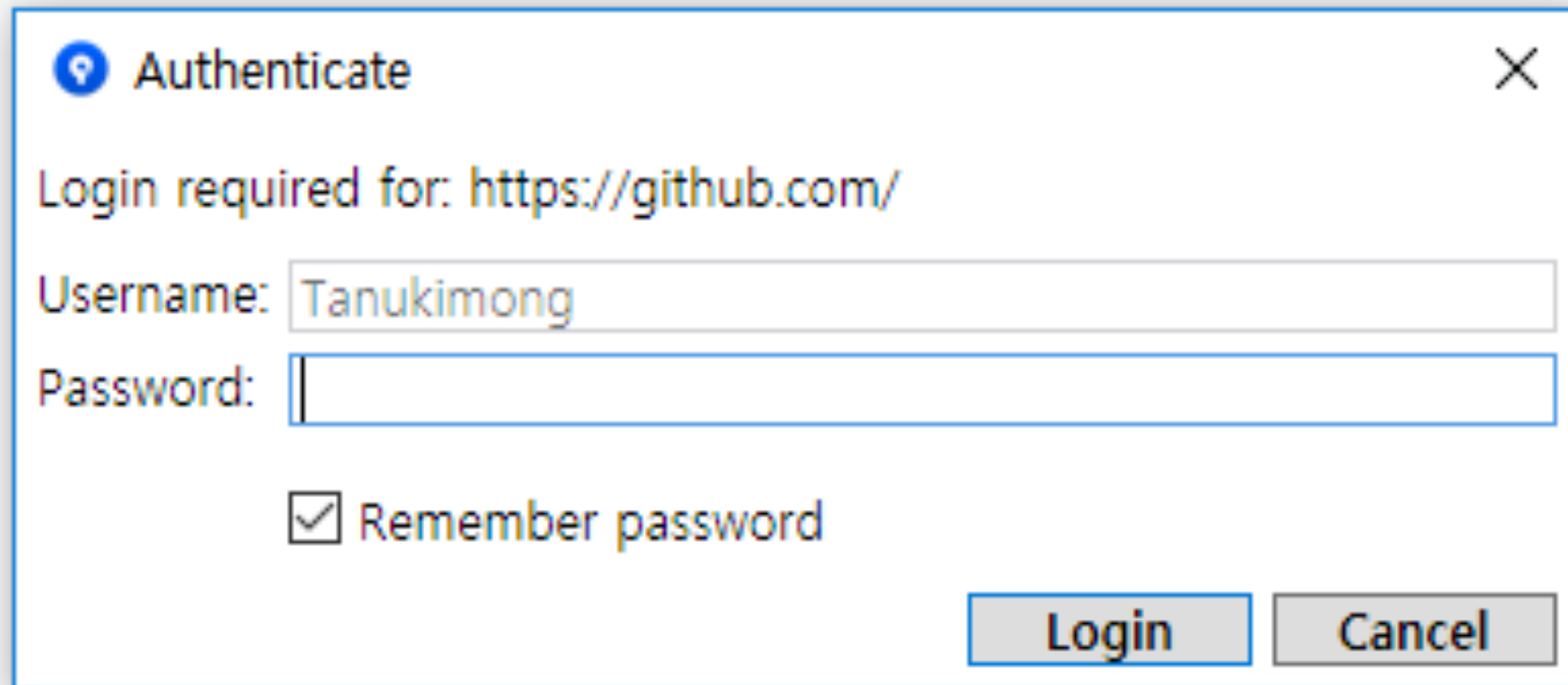
확인 취소

Github 계정 등록

호스팅 서비스 = 'GitHub'

프로토콜 = 'HTTPS'

사용자명 = '이메일 말고 사용자 이름 입력' → '비밀번호 새로 고침' 클릭



An 'Authenticate' dialog box with a blue title bar and a close button (X) in the top right corner. The text 'Login required for: https://github.com/' is displayed. Below this, there are two input fields: 'Username:' containing 'Tanukimong' and 'Password:' which is empty. A checkbox labeled 'Remember password' is checked. At the bottom right, there are 'Login' and 'Cancel' buttons.

Authenticate

Login required for: https://github.com/

Username: Tanukimong

Password:

☒ Remember password

Login Cancel

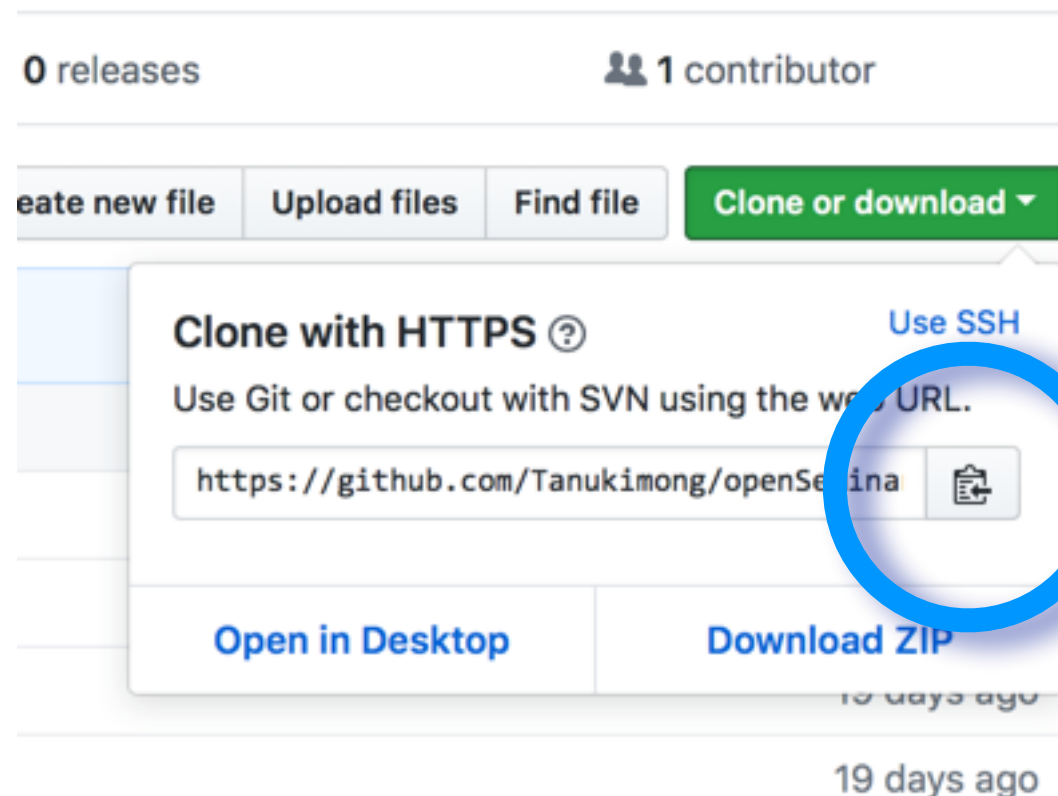
비밀번호 입력

Github 가입시 등록한 비밀번호 입력

Git 저장소 가져오기

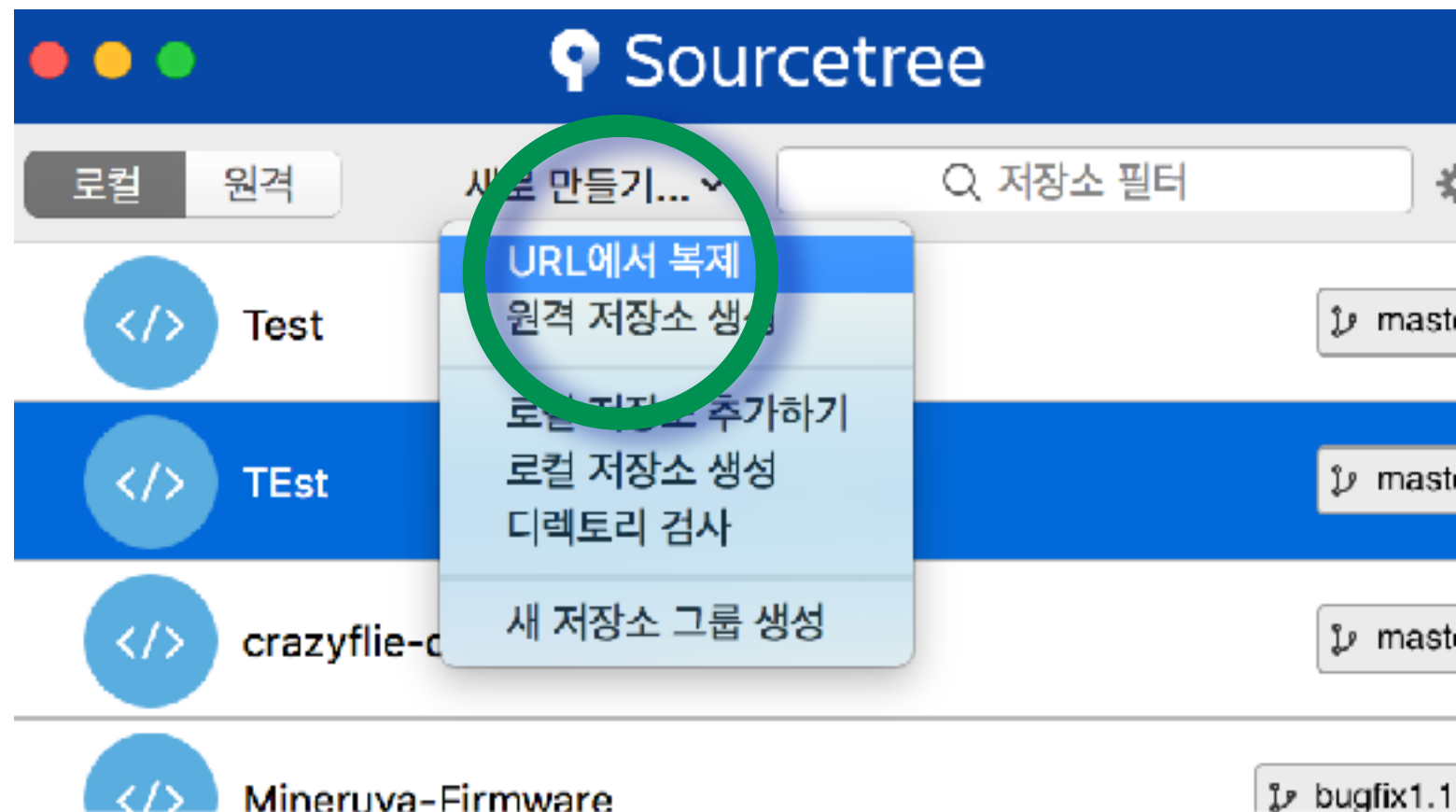
Cloning을 이용한 방법

복사할 Git저장소 찾기 : <https://github.com/Tanukimong/openSeminar> 접속!



Git 저장소 가져오기

Cloning을 이용한 방법



Git 저장소 가져오기

Cloning을 이용한 방법


저장소 복제

원본 URL:

목적지 경로: ...

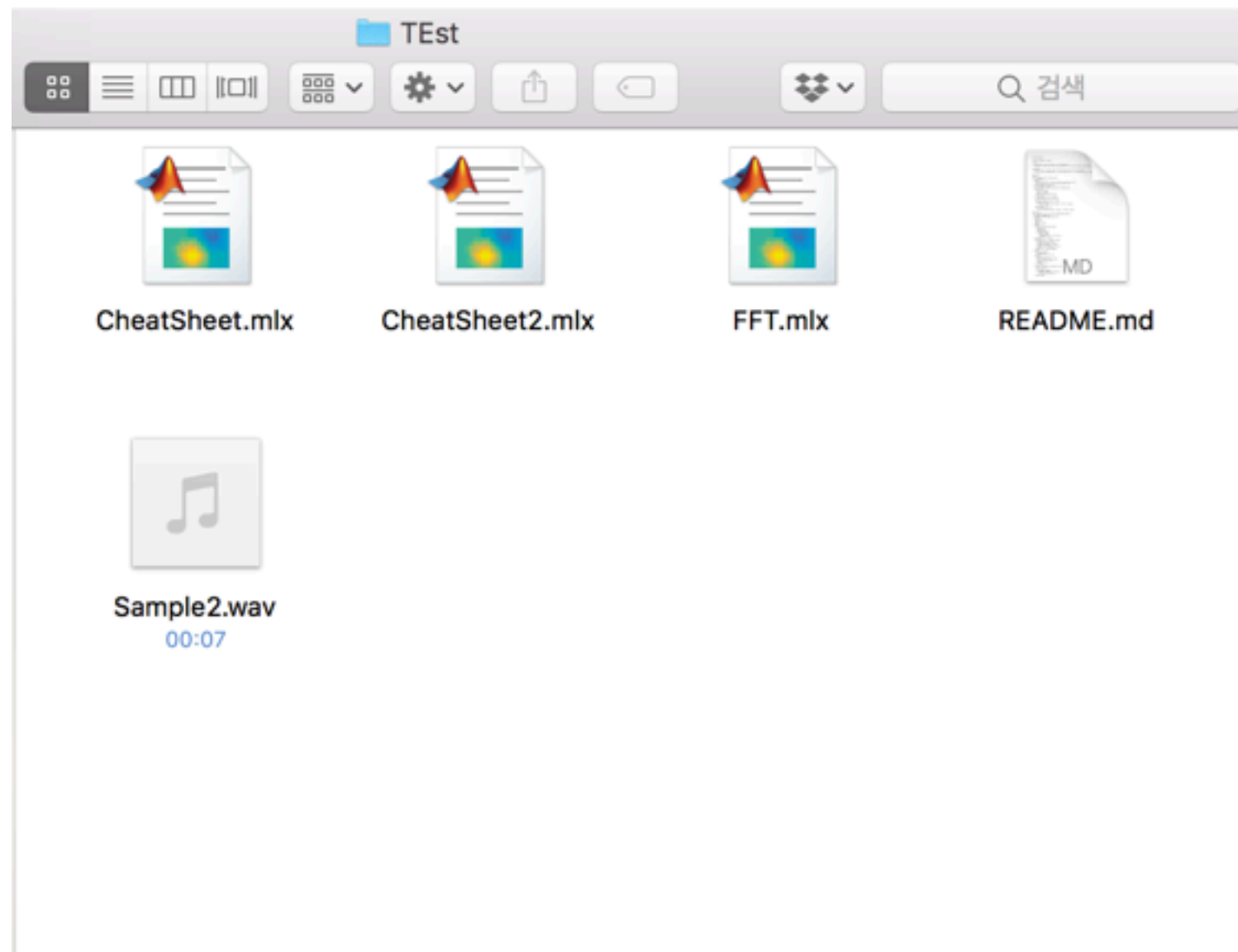
이름:

▶ 고급 옵션

 Git 저장소입니다

Git 저장소 가져오기

Cloning을 이용한 방법



Flow of use

“흠... CSS를 제어할때 일일이 함수를 만들어주기가 귀찮군...”



Flow of use

“다른 사람들이 미리 만들어 놓은 건 없나??”



Flow of use



Google이나 Namu.wiki에 검색하여 만들어진 라이브러리를 찾기

Flow of use

“읽어보니 대략 내가 쓰려는 용도에 맞는거 같은데...”



Flow of use

“자세한 사용법이랑 jQuery의 최신 버전이 필요한데...”

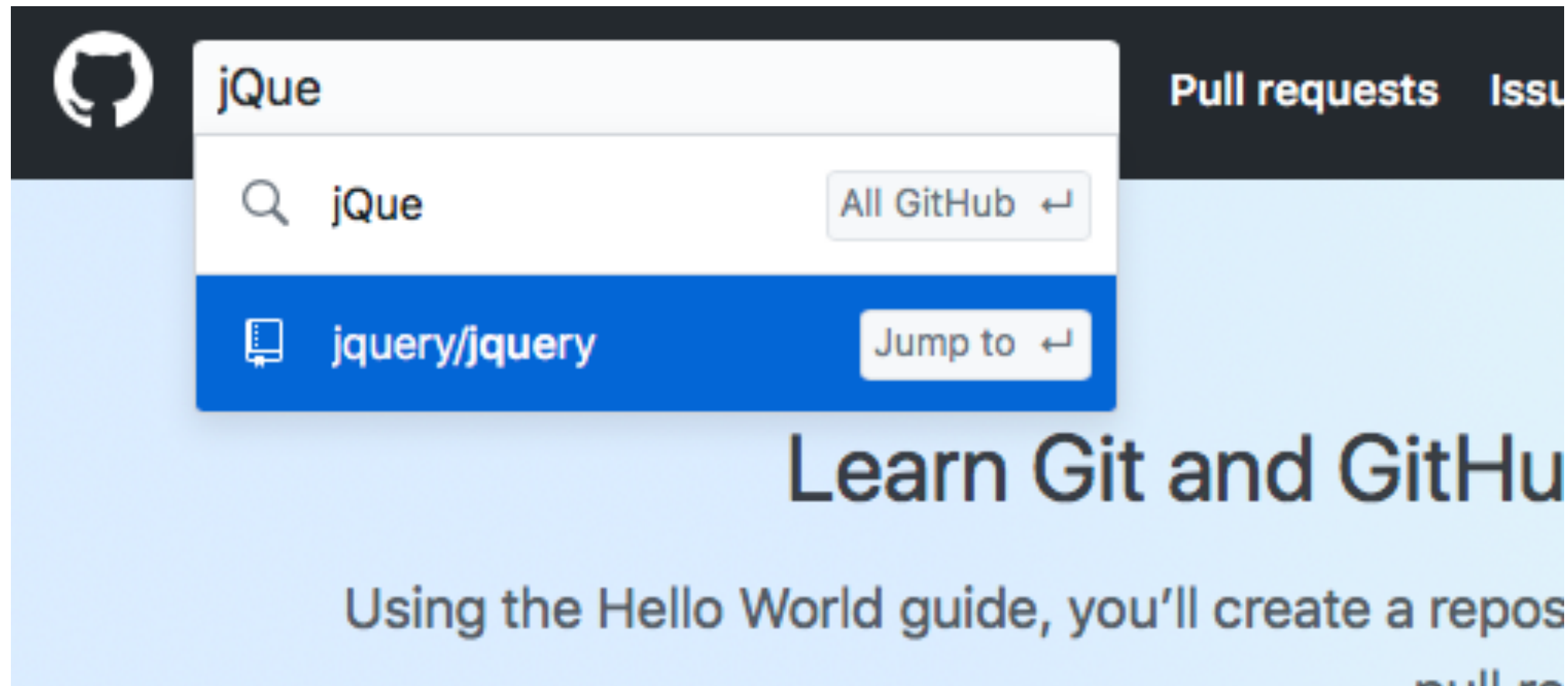


Flow of use

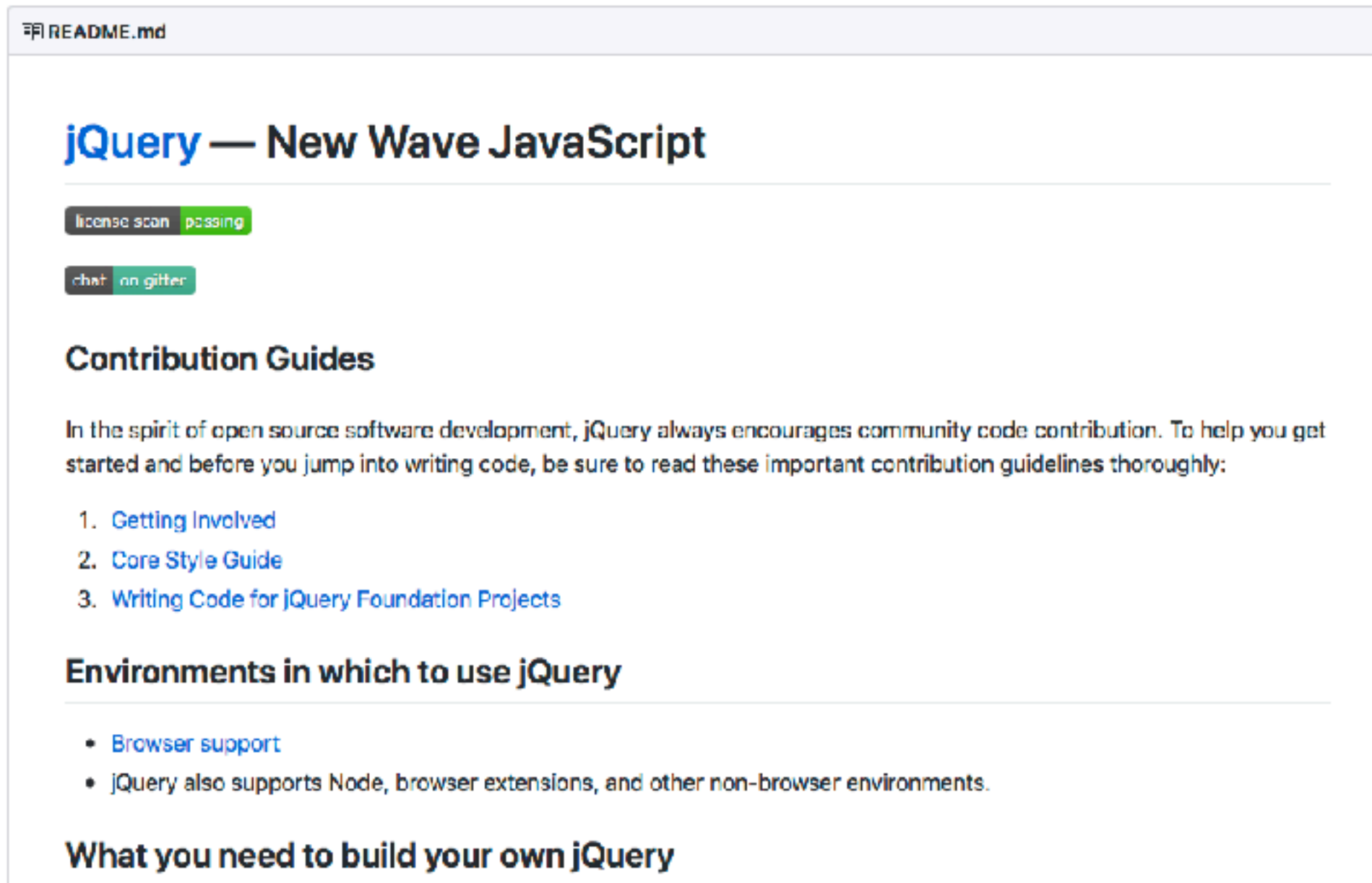
“Github에 있을까?”



Flow of use



Flow of use



들어가면 프로그램에 대한 깔끔한 설명도 볼 수 있고,

Flow of use

The screenshot shows the jQuery GitHub repository page. At the top, the repository name 'jquery / jquery' is displayed. Below it, navigation tabs include 'Code', 'Issues 78', 'Pull requests 17', 'Projects 0', 'Wiki', and 'Insights'. The repository description is 'jQuery JavaScript Library' with a link to 'https://jquery.com/'. A yellow bar highlights the repository statistics: '6,342 commits' (circled in red), '3 branches', '148 releases', '273 contributors', and 'MIT' license. Below this, a green 'Clone or download' button is visible. A dropdown menu is open, showing 'Clone with HTTPS' (selected) and 'Use SSH'. The HTTPS URL 'https://github.com/jquery/jquery.git' is displayed and circled in red. Below the URL are buttons for 'Open in Desktop' and 'Download ZIP'. The background shows a list of recent commits.

jquery / jquery

Watch 3,567 Star 49,832 Fork 16,707

Code Issues 78 Pull requests 17 Projects 0 Wiki Insights

jQuery JavaScript Library <https://jquery.com/>

6,342 commits 3 branches 148 releases 273 contributors MIT

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

Clone with HTTPS Use SSH

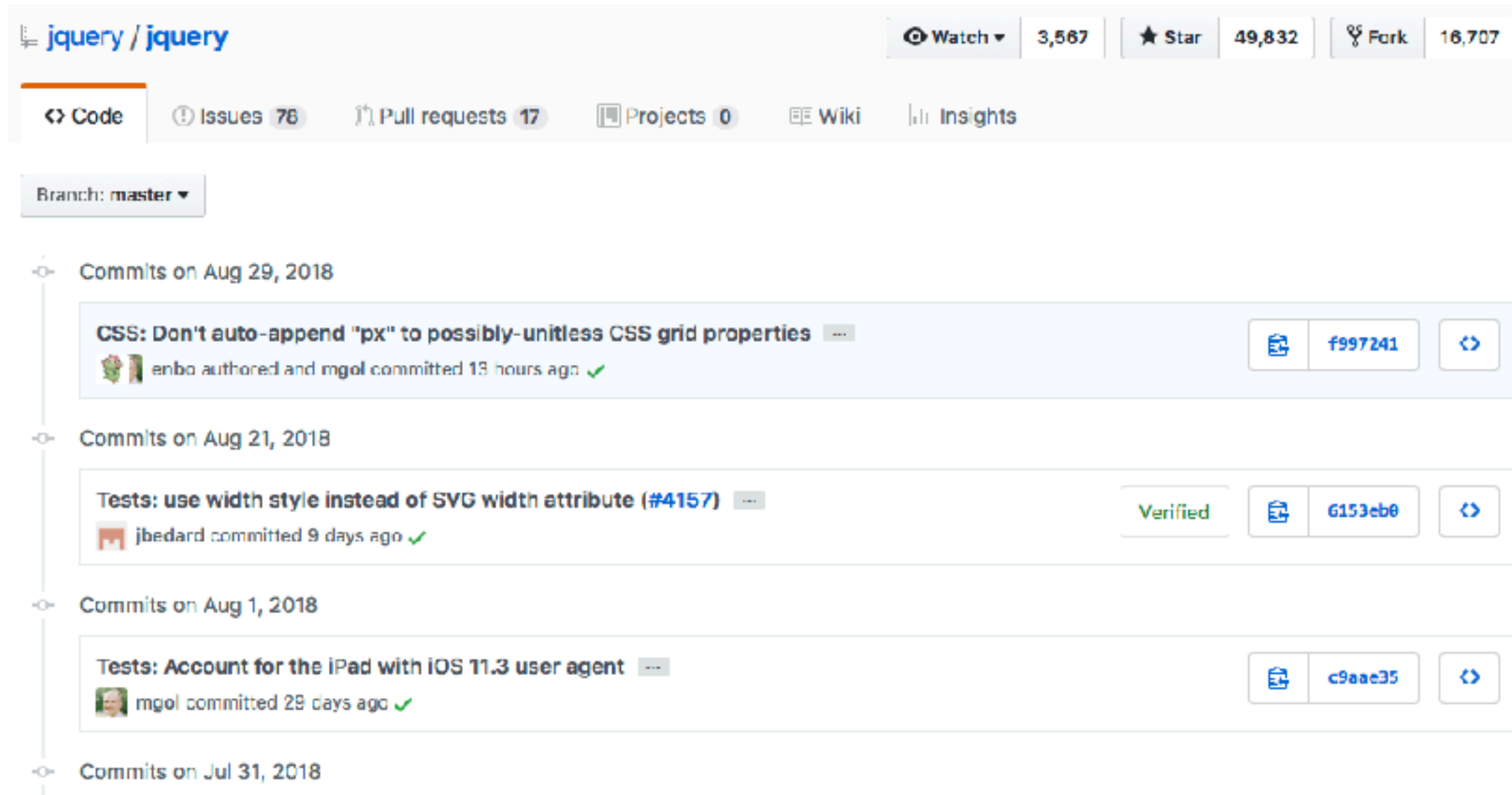
Use Git or checkout with SVN using the web URL.

<https://github.com/jquery/jquery.git>

Open in Desktop Download ZIP

Download 주소

Flow of use



The screenshot displays the jQuery GitHub repository page. At the top, the repository name 'jquery / jquery' is shown alongside statistics: 3,567 Watchers, 49,832 Stars, and 16,707 Forks. Below this, navigation tabs include 'Code', 'Issues 78', 'Pull requests 17', 'Projects 0', 'Wiki', and 'Insights'. A 'Branch: master' dropdown is visible. The commit history is shown on the left, with a vertical timeline. The main area lists three commits:

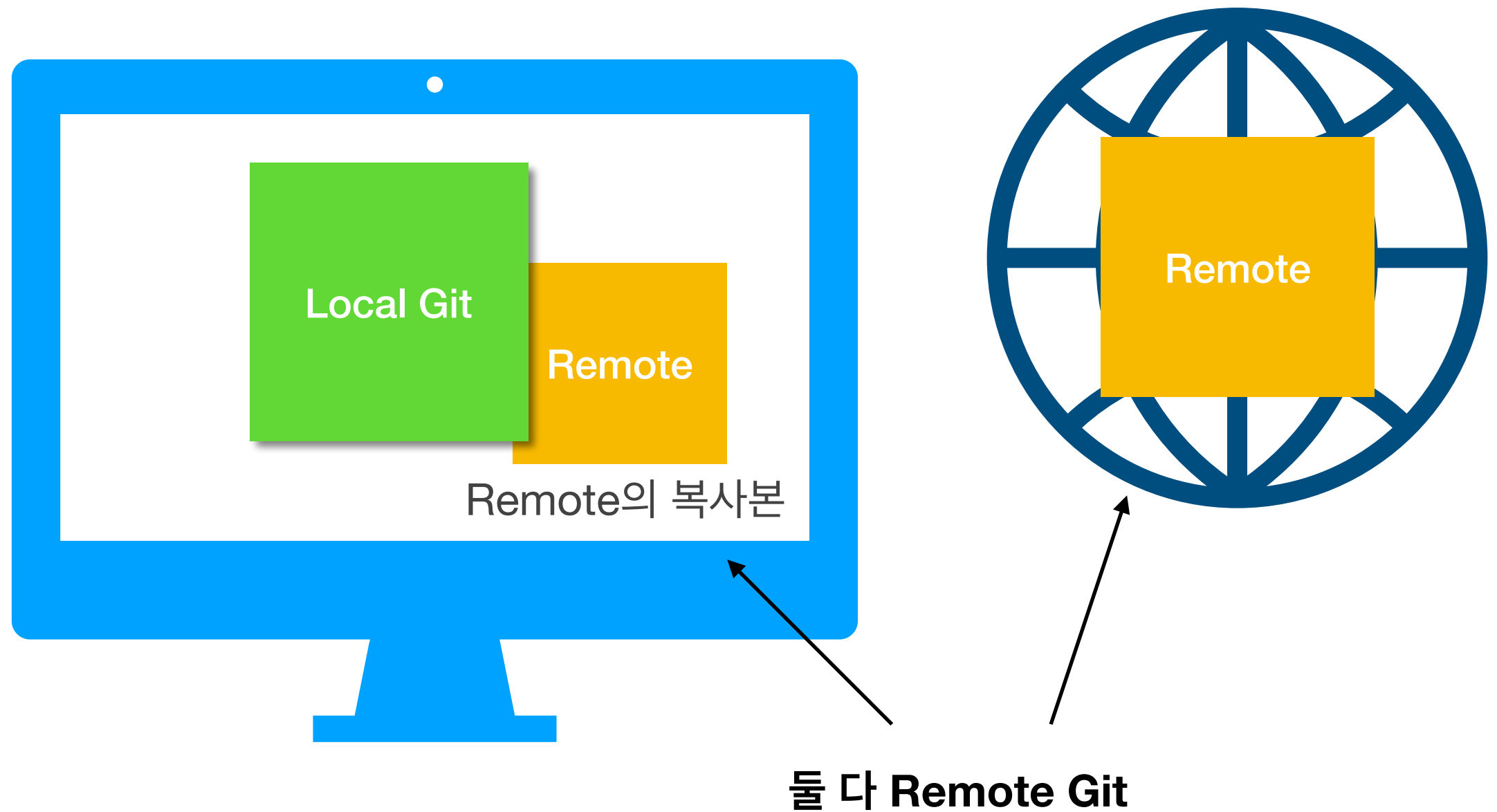
- Commit 1:** 'CSS: Don't auto-append "px" to possibly-unitless CSS grid properties' by enbo, committed 13 hours ago. Commit hash: f997241.
- Commit 2:** 'Tests: use width style instead of SVG width attribute (#4157)' by jbedard, committed 9 days ago. Commit hash: 6153eb0. This commit is marked as 'Verified'.
- Commit 3:** 'Tests: Account for the iPad with iOS 11.3 user agent' by mgol, committed 29 days ago. Commit hash: c9aac35.

Github에서 공동으로 Develop되고 있는 프로그램의 각 버전별 차이도 이해할 수 있다
→ 나온지 얼마 안되는 프로그램의 경우 이 부분을 봄으로써 해결되는 일이 잦다

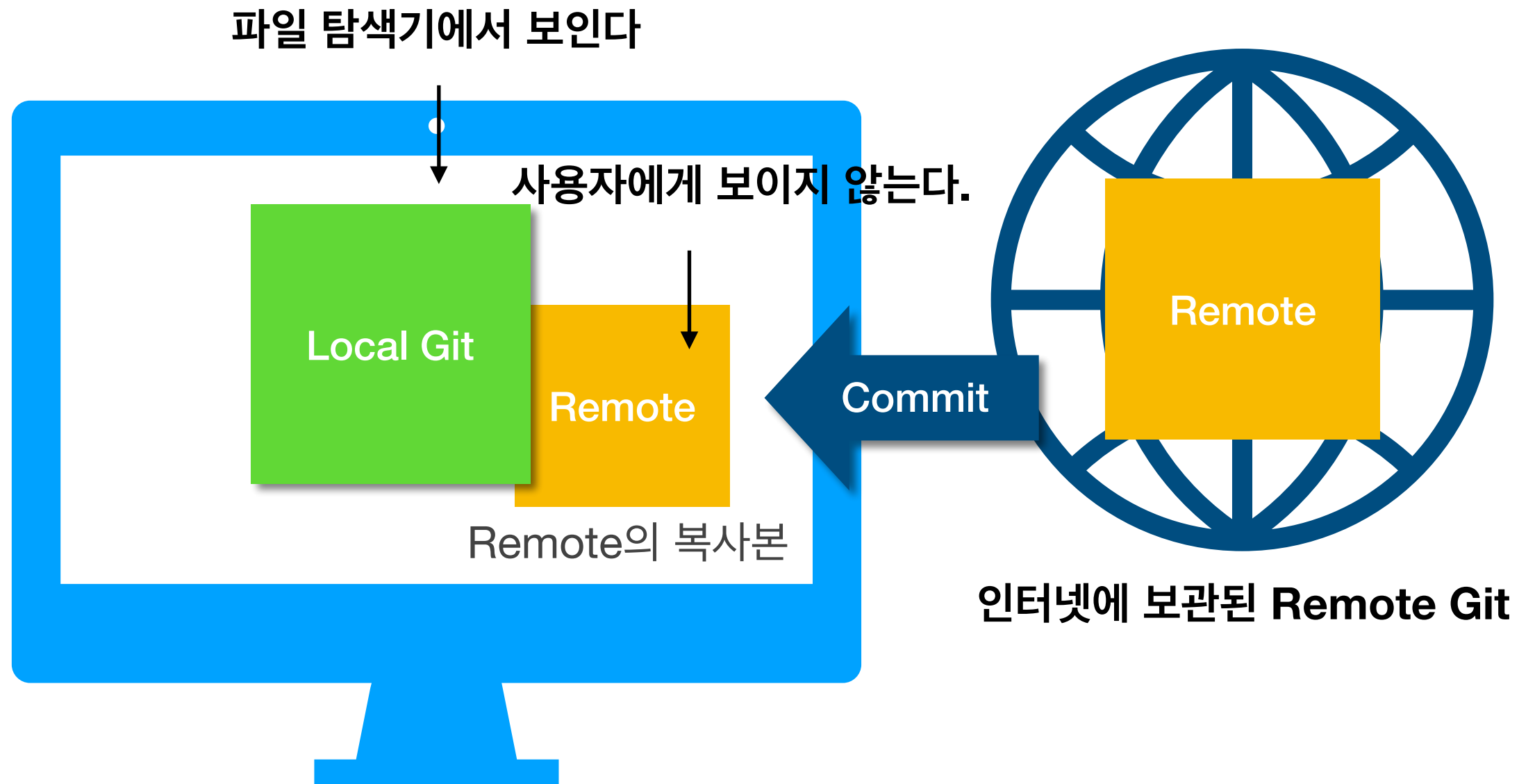
Lv1. Git저장소 생성

- 한 번에 Local 저장소와 이에 연결된 Remote 저장소 생성
 - 생성된 Local 저장소 -> SourceTree에서 확인
 - 생성된 Remote 저장소 -> 자신의 Github계정에 접속하여 확인
- **[설명]** Local vs Remote

Local vs Remote



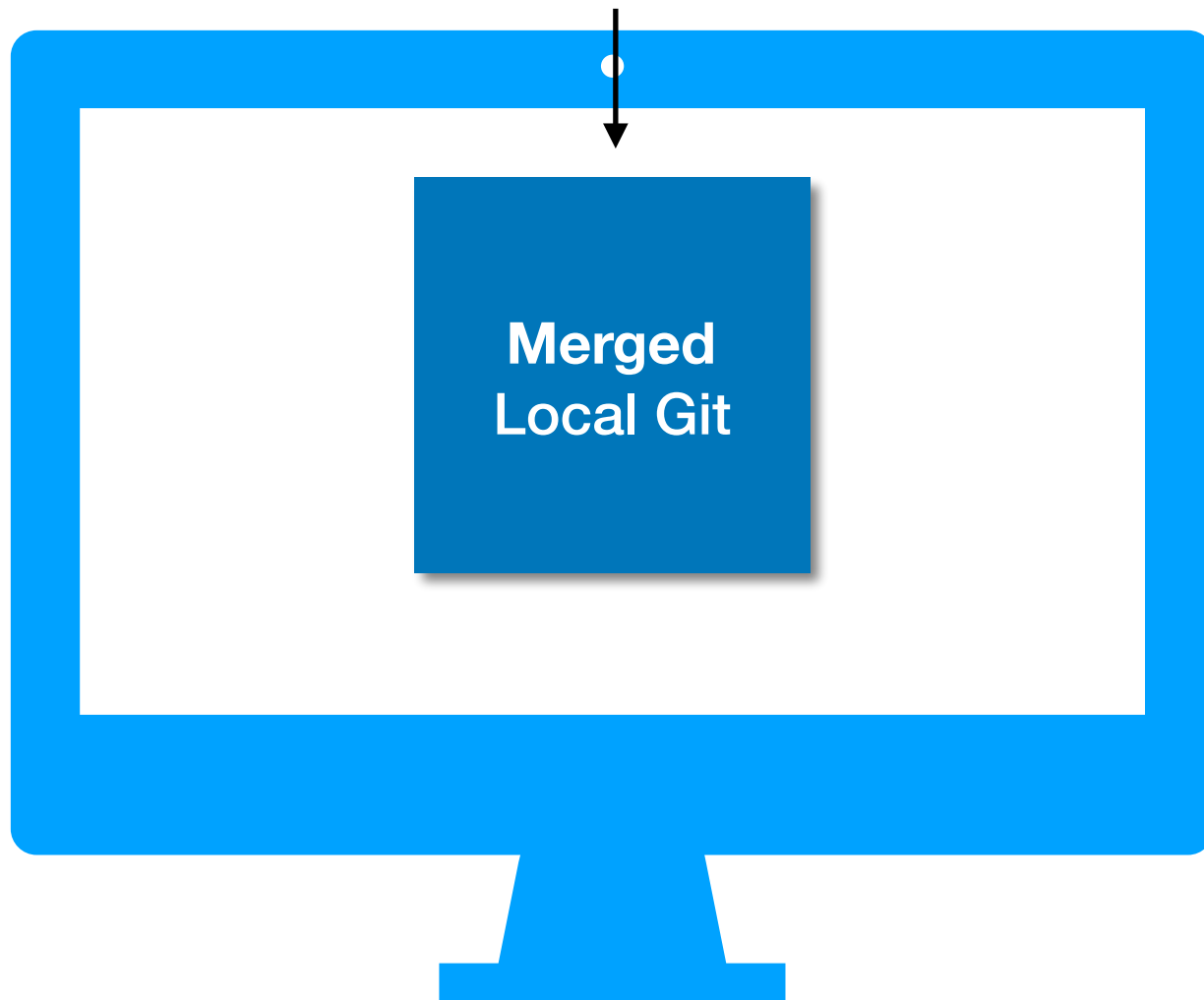
Local vs Remote



Why?

Local vs Remote

Remote와 적절하게 섞인 버전

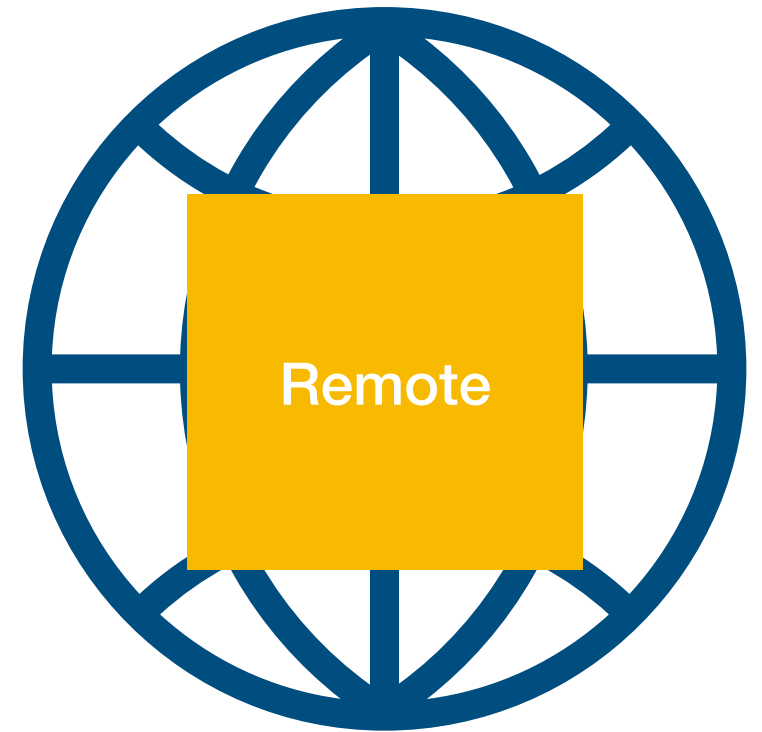
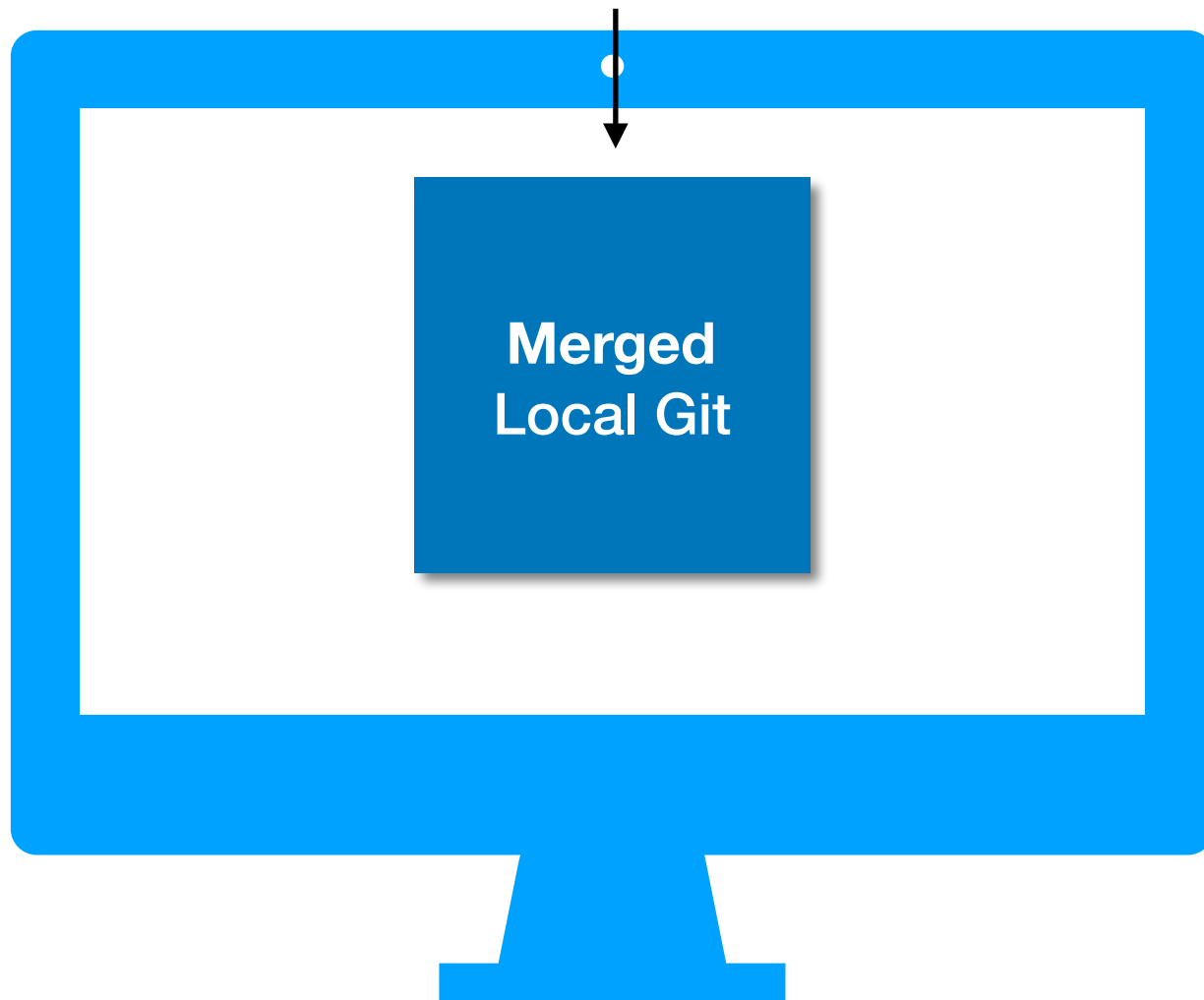


인터넷에 보관된 Remote Git

만약 복사를 안하고, 인터넷에 있는 Remote와 바로 합하면... 원본파일이 손상된다

Local vs Remote

Remote와 적절하게 섞인 버전



인터넷에 보관된 Remote Git

만약 복사를 안하고, 인터넷에 있는 Remote와 바로 합하면... 원본파일이 손상된다

Local vs Remote

- 따라서, 하드디스크 내의 어떤 폴더를 Local 저장소로 만들 수 있다
 - Local 저장소는 Git의 버전관리 기능(Commit, Branch..)을 이용할 수 있다
- Local 저장소의 내용을 Github에 올리고 싶다면, Github에 Remote를 생성하고 연결해야한다.
- 방금의 과정을 이용하면,
 - Remote 저장소와 Local 저장소를 동시에 생성해준다.
 - 특정 명령을 내리지 않는 이상, **각 저장소는 서로 영향을 주지 않는다.**

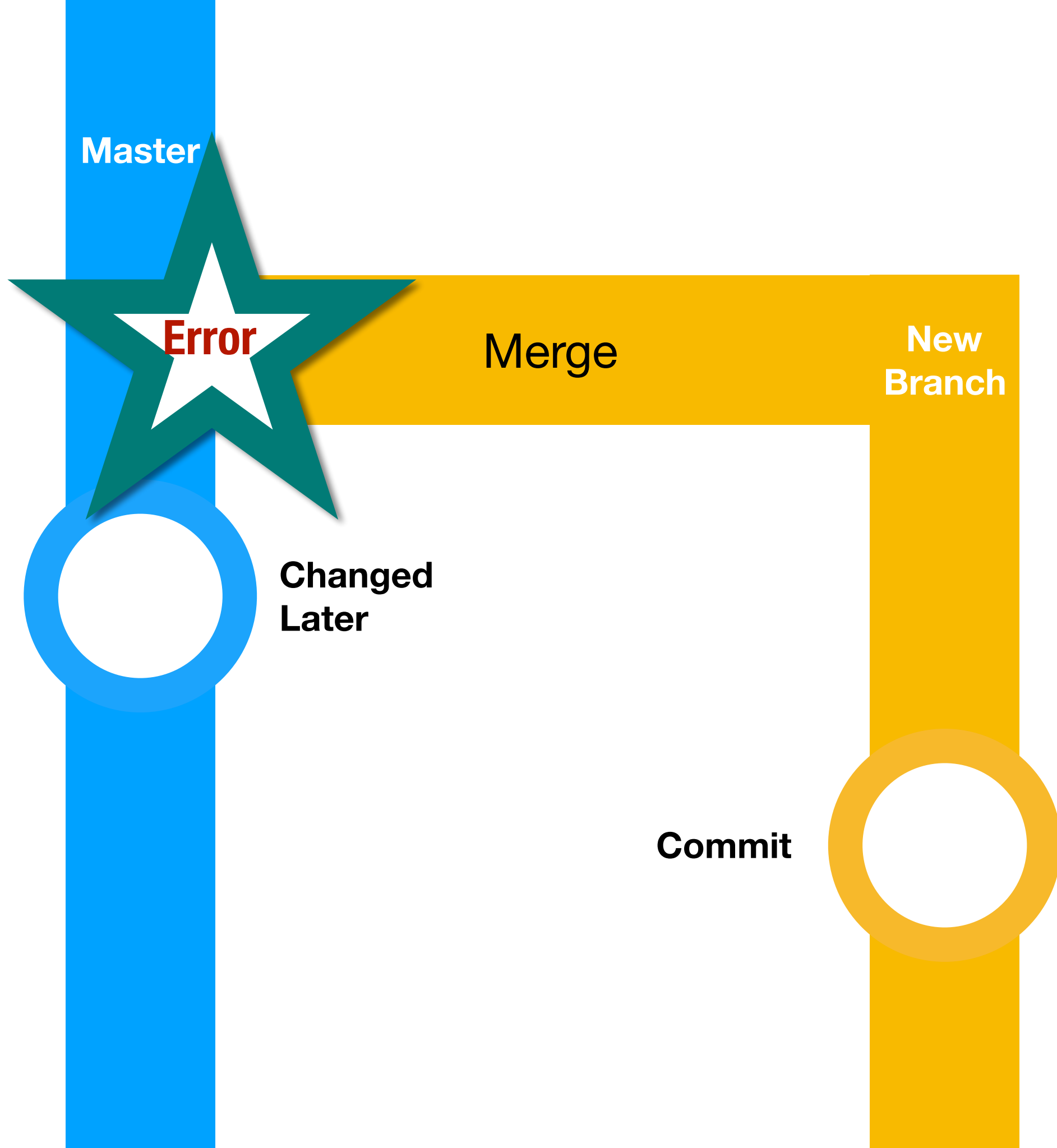
Local 저장소를 만들고
Remote 저장소를 연결하기

Lv2. Commit

- README.md 파일 생성
- Commit하여 Local Git 중간 저장
- Commit내역 모두 Local Git에 올리기
- 버전 한눈에 보기

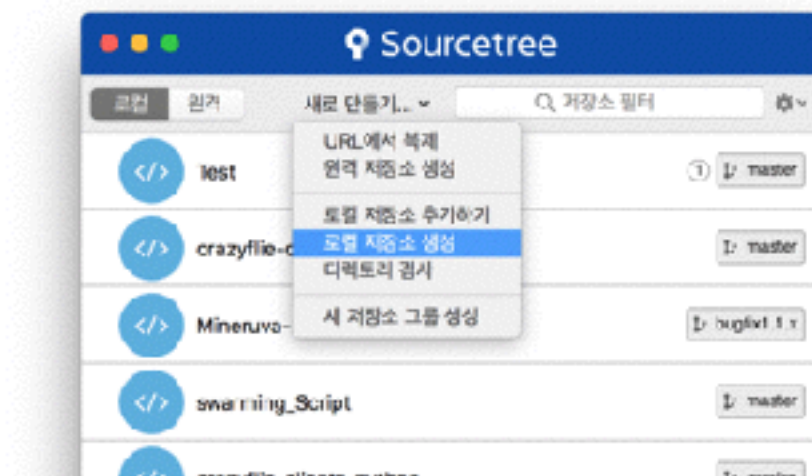
Lv3. Branch와 Merge

- Branch의 목적? 원본을 손상하지 않고 개선하기
- Merge와 Conflict
 - Conflict의 조건 (Branch change -> Master change)
 - Conflict 해결법
 - Example : Branch change -> Master change & Merge from branch to Master

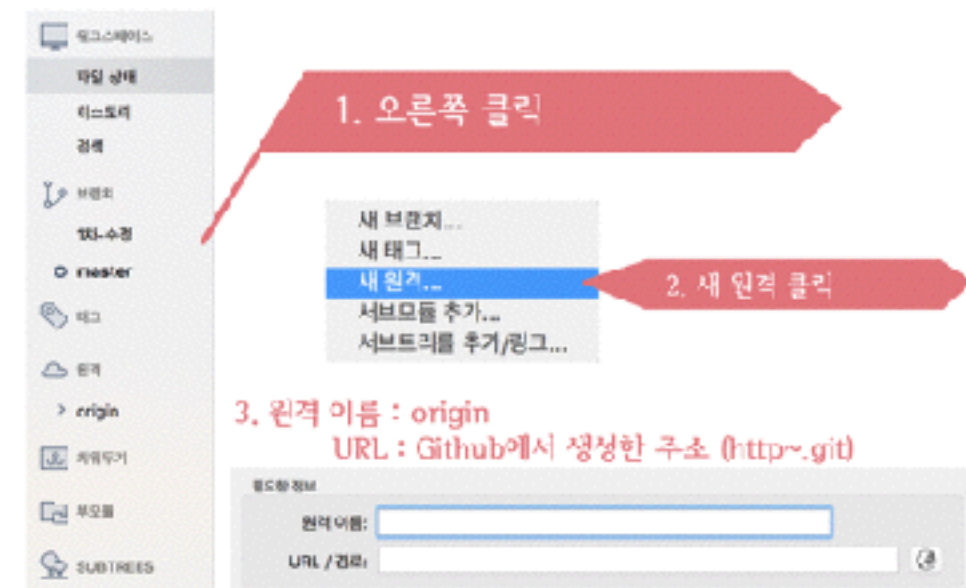


CheatSheet

- HDD내에 Git저장소 생성



- Github와 HDD내의 Git저장소 연결



- Pull : Web -> HDD



후에 Git을 이용하여 프로그램을 정리할때 참고!