

# Git / Github

오픈소스 프로젝트를 이해하고,  
Github을 통한 실제 오픈소스 프로젝  
트에 참여하는 워크샵

- 2017.08.08 -

# 발표자 - 김종광



- ▶ 이름 : 김종광
- ▶ 소속
  - 미래부/NIPA, 공개SW 개발자Lab - 연구원
  - 한국센차유저그룹 커뮤니티 - 운영자
- ▶ SNS
  - E-Mail : [kim@jongkwang.com](mailto:kim@jongkwang.com)
  - Blog : <http://jongkwang.com>
  - Facebook
    - > <https://www.facebook.com/kimjongkwang>

# Git - 그림설명

공개 SW

# Why? Open Source

- 공개된 Source Code
- 코드만 공개하면 공개SW 인가?
  - Android
  - Tizen
  - 공개SW 개발자 대회

# Why? Open Source

- ▶ 고액된 Source Code 가 있으면 사용 할 수 있는가?
- ▶ 모든것이 공개 되어야 Open Source 이다
  - 기획 / 의도
  - 의사결정
  - 메뉴얼
  - 소스코드
  - Patch

# Why? Open Source

## ▶ Why? Open Source?

- 1 vs. 70억
  - 사용자 환경
  - Test
  - 문서화 / 번역
- 안할수가 없다
- ‘오픈소스를 왜 해야 하는가?’라는 질문은 틀렸다.  
**혁신은 오픈을 통해서만 만들어진다.**  
‘우리는 어떻게 할 것인가?’를 고민할 때다. (국민대 이민석 교수님)
- 직원이 10만명인 회사를 운영하는 가이드북이 있다면 구매 하겠다  
없으니 소통으로 해법을 찾을 수 밖에 없다. - 코카콜라 회장

# Why? Open Source

## ▶ Open Source 를 하는 이유?

- 개인
- 재단
- 삼성
- 구글

# Why? Open Source

## ▶ Google

- white paper 작성
  - MapReduce → Hadoop
  - BigTable → HDFS → HBase
  - Google Code Search → Sourcegraph
  - Borg → Docker
  - TensorFlow??

분산관리 시스템

Download

# Download

- ▶ Git
  - <http://git-scm.com/download/win>

- ▶ SourceTree
  - <https://www.sourcetreeapp.com/>

- ▶ ATOM
  - <https://atom.io/>

- ▶ Java
  - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
  - JDK 다운로드 (가입/로그인 필수)

- ▶ Eclipse
  - <https://www.eclipse.org/downloads/>

분산관리 시스템

分散관리

시스템

# 버전관리 시스템

## ▶ 게임에서 Save 같은 존재

- RPG 게임에서 Save Point

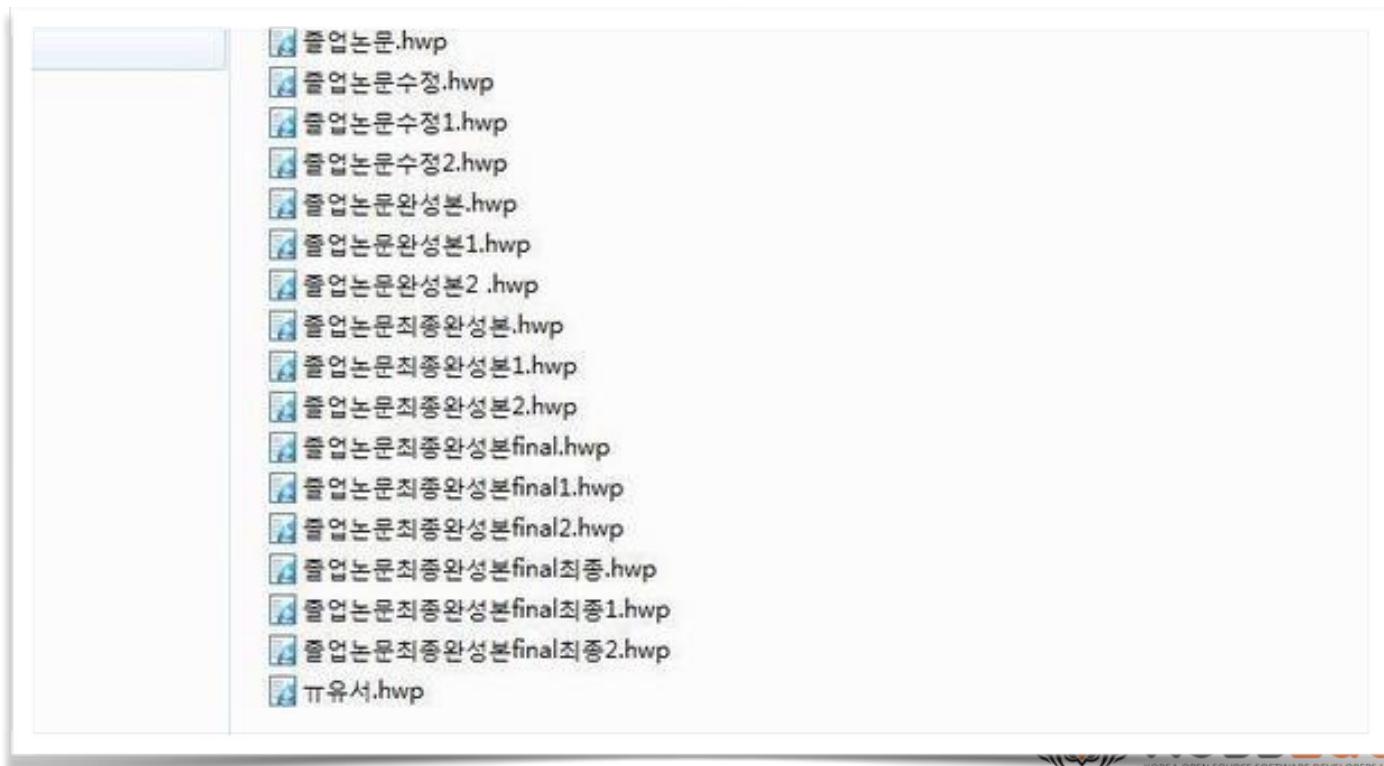
- 게임에서는 Save Point 를 만나야 Save 하지만
- 버전관리 시스템에서는 내가 원할 때 Save 가능



# 버전관리 시스템

## ▶ 우리는 이미 리포트 작성할 때 경험 했다

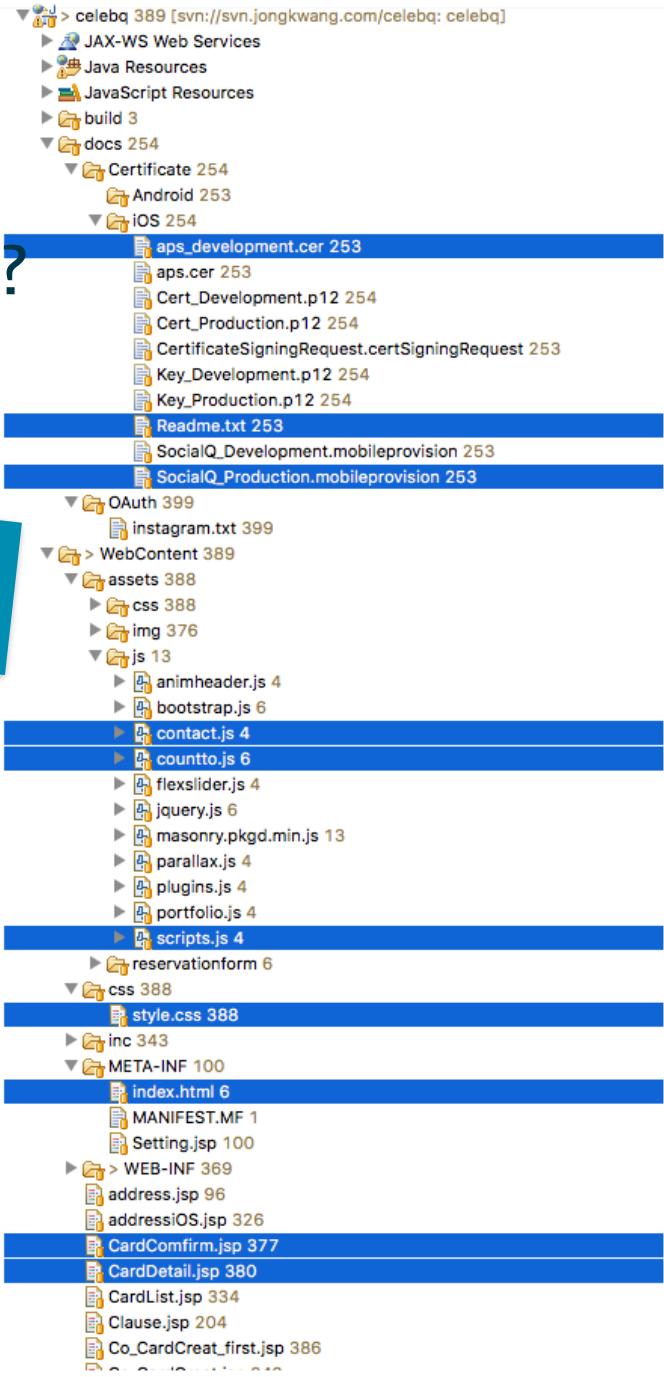
- 파일에 번호를 부여하여 History를 관리하는 방식은 상당히 훌륭하다
  - SVN에서도 같은 방식을 사용하며
  - SVN에서는 이 번호를 Revision Number 라고 부른다
  - 하지만, 파일이 1개 일 때만 가능하다



# 버전관리 시스템

- ▶ 하지만, 여러 파일이 변경되는 경우에는?
  - 파일 넘버로 관리 할 수 없다

클라우드 백업?



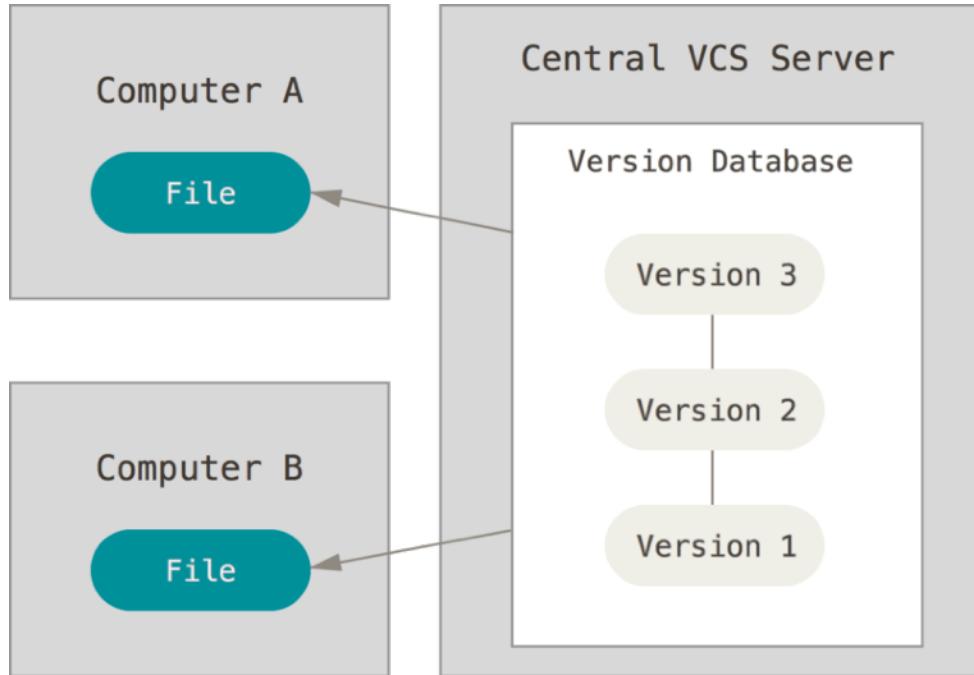
분산관리 시스템

分散관리 시스템  
도구로 이용하자

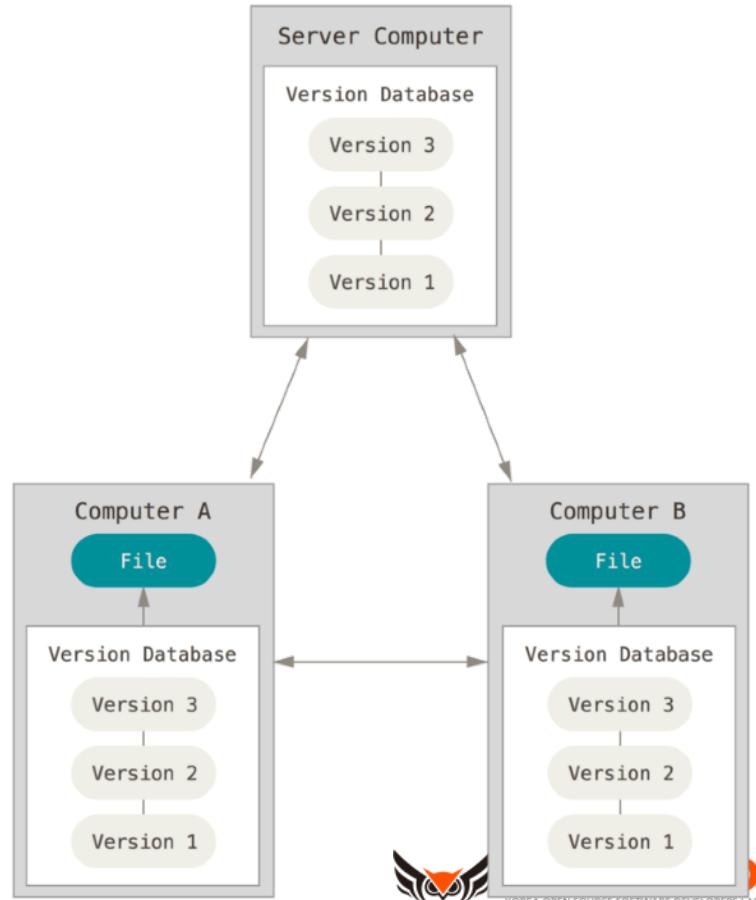
# 버전관리 시스템

- ▶ 버전관리 시스템의 2가지 분류

## 클라이언트-서버

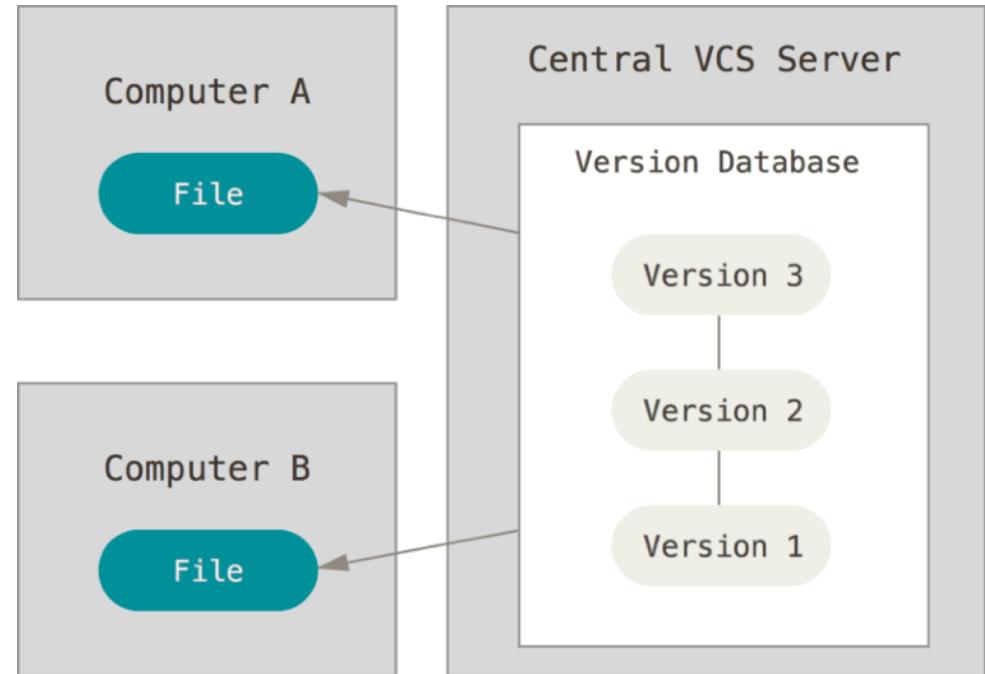


## 분산



# 클라이언트-서버

- › 서버와 클라이언트 구조
- › 서버가 최종 버전을 관리
- › SVN, CVS 등이 여기에 해당
- › 장점
  - 쉽다
- › 단점
  - 서버가 고장나면 끝이다
  - Online 상태가 아니면 작업불가
  - Branch 가 불편하다
  - 느리다 (Online)



# 분산

## 특징

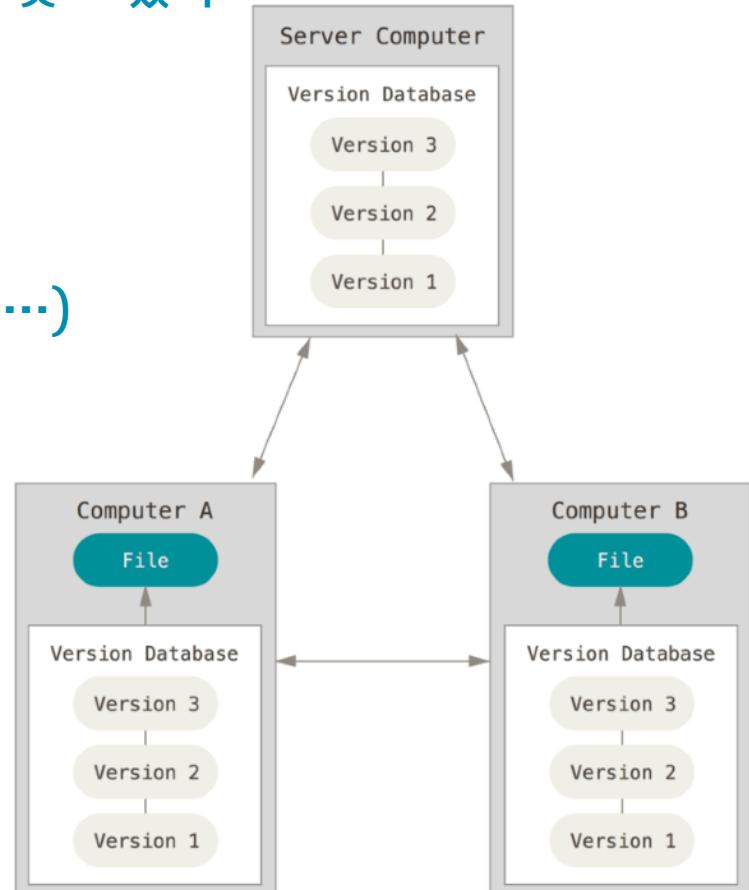
- 모든 클라이언트가(개발자)가 전체 저장소를 갖고 있다
- 전체 저장소의 사본을 모두 갖는다

## 장점

- Offline 작업 가능
- Commit 이 자유롭다 (중간버전 커밋 찍어도...)
- Branch 가 자유롭다
- 빠르다

## 단점

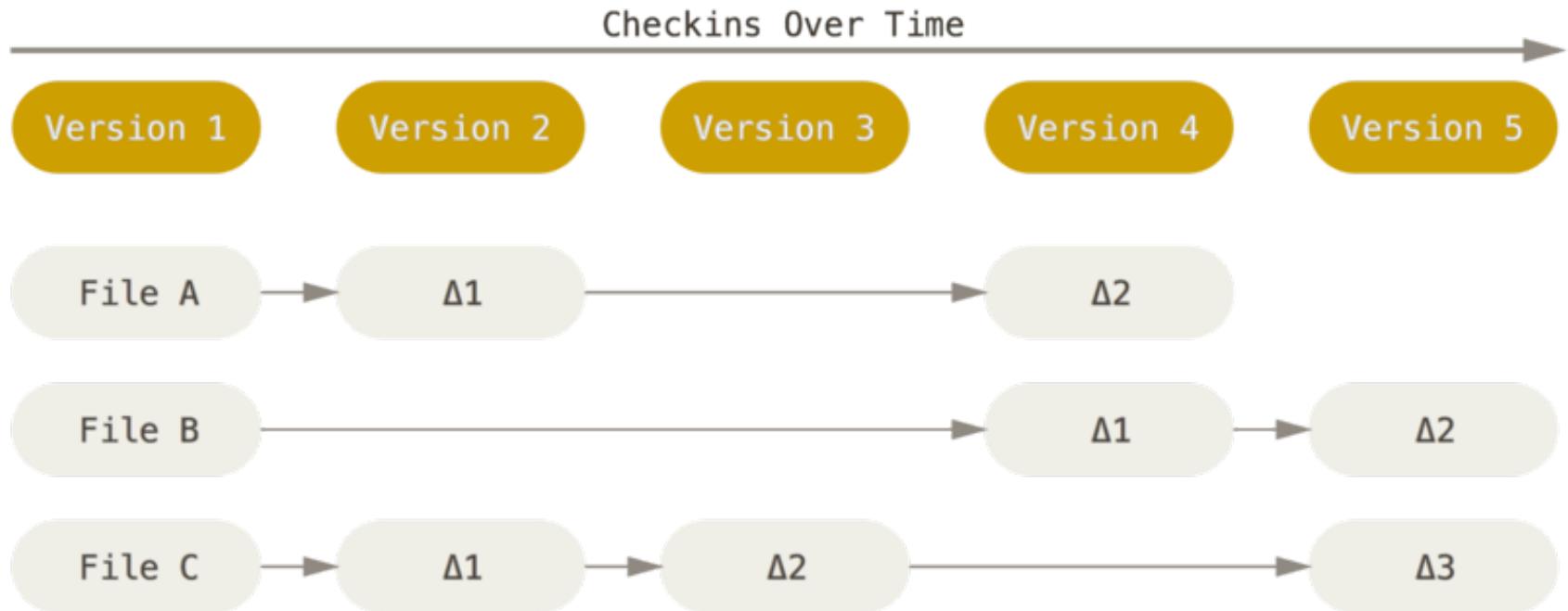
- 조금 어려운 정도
- Client에 저장소를 설치해야 한다



# 데이터를 다루는 차이

## ▶ SVN

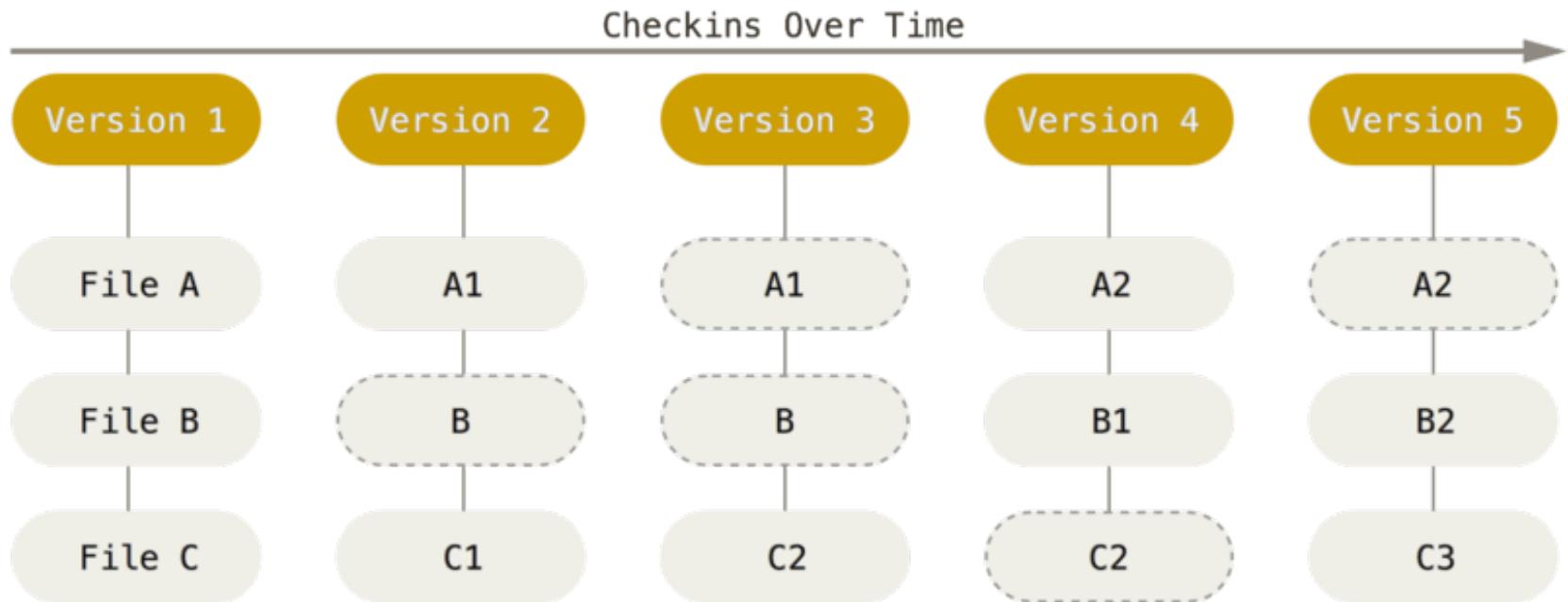
- 파일의 변화를 시간순으로 관리
- 파일들의 집합을 관리



# 데이터를 다루는 차이

## ▶ Git

- 순간의 Snapshot 관리
- 변경이 없으면 Link 만 저장



# 취업가기 - Git History

- › Git 을 만든 사람은?

# 취업가기 - Git History

- ▶ Git 을 만든 사람은?



# 취어가기 – Git History

- › 2002년 Linux Kernel 개발을 위해 BitKeeper 사용
- › 2005년 BitKeeper에서 무료 사용에 난색
- › 이에 분노한 리누스 토발즈



# 취미가기 – Git History

## ▶ 2주만에 Git을 만듬

- 처음 용도 : Linux Kernel 버전관리

## ▶ 목표

- 완벽한 분산 환경
- 빠르고 단순하게 수백/수천 개의 Branch 작업을 목표
- 실제로 리눅스 커널 같은 대형 프로젝트에 사용됨

## ▶ 개발기간

- 3일만에 Git 의 버전관리를 Git으로
- 2주만에 여러 Branch 를 병합하며 Git 완성
  - 현재까지 큰 변화 없이 사용됨



# Git의 특징

## ▶ 특징

- 로컬 및 원격 저장소 생성
- 로컬 저장소에 파일 생성 및 추가
- 수정 내역을 로컬 저장소에 커밋
- 파일 수정 내역 비교
- 원격 저장소에 커밋된 수정 내역을 로컬 저장소에 적용
- 로컬 커밋 내용을 원격 저장소로 Push
- Branch 생성
- 브랜치 병합(Merge)
- 브랜치 병합시 충돌 확인

# 개발자만 Git 을 사용해야 하는가?

## ▶ 누구나 사용 가능

- 디자이너 : 이미지 파일
- 기획자 : 발표자료

## ▶ 사례

- 서울 정보소통광장 행정정보 공개
  - <https://github.com/seoul-opengov/opengov>
- 백악관 각종 정보
  - 예산안 : <https://github.com/WhiteHouse/budgetdata>
  - 그외 여러가지
- 고위공직자 재산 공개
  - <https://github.com/codenamu/official-assets-explorer-2017>

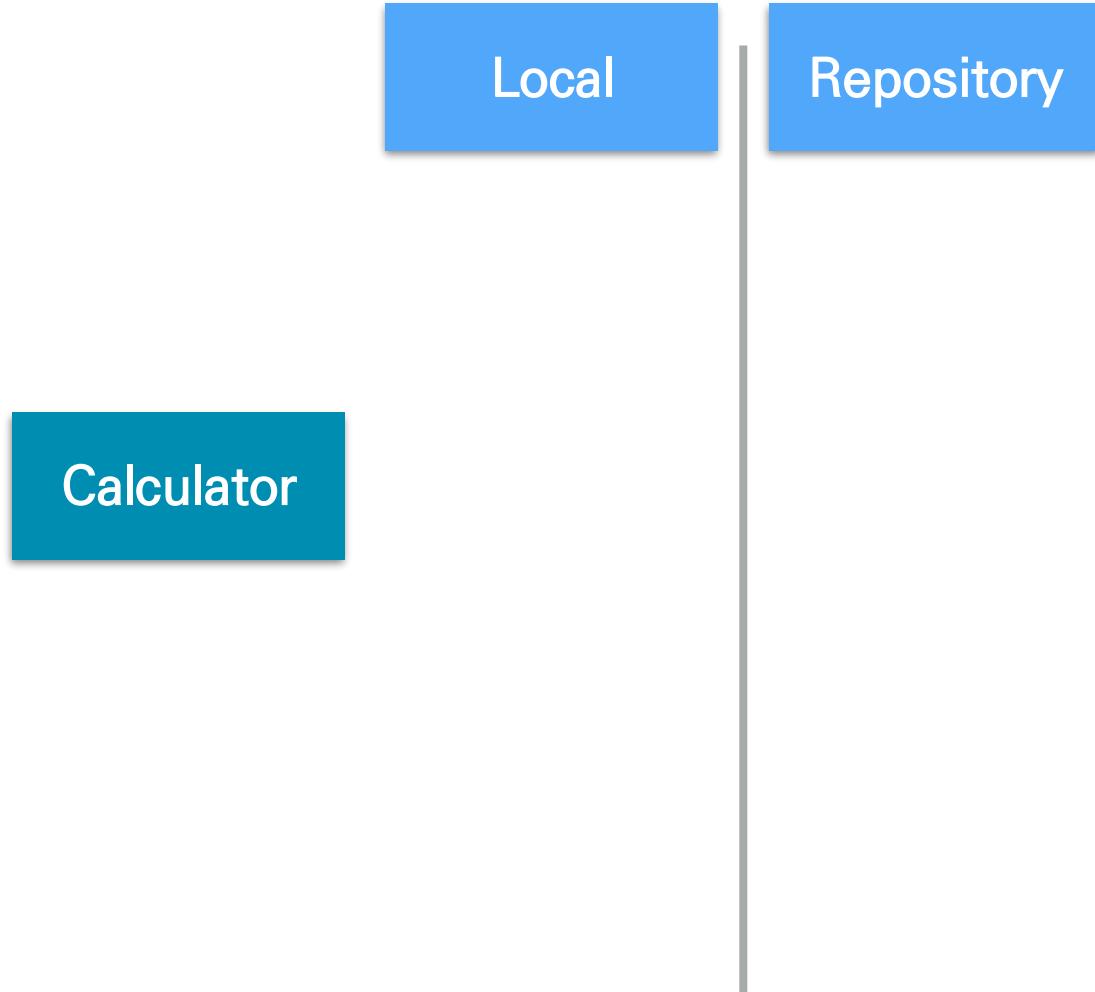
분산관리 시스템

Git

그림설명

# Git - 그림 설명

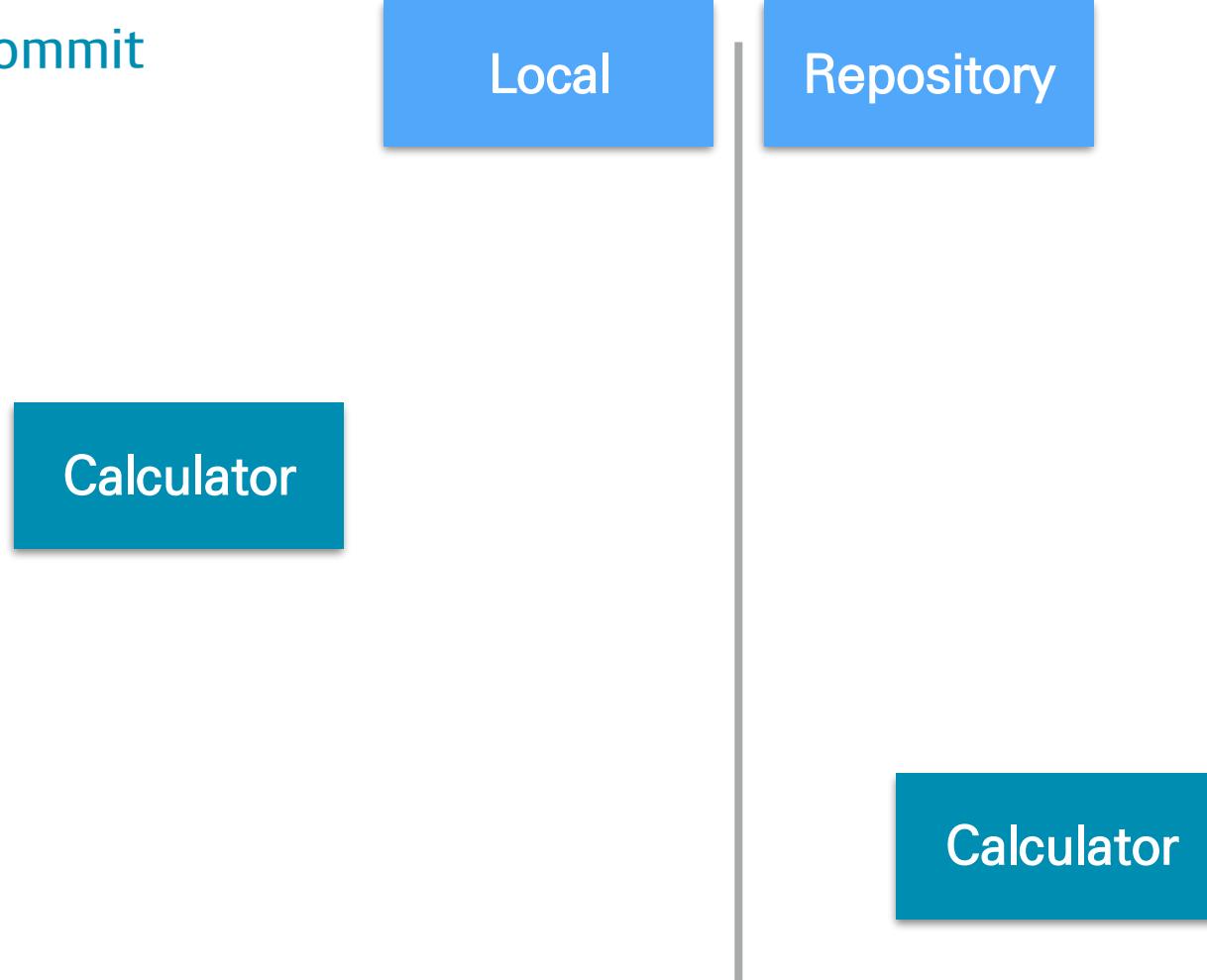
- › 계산기 클래스 생성



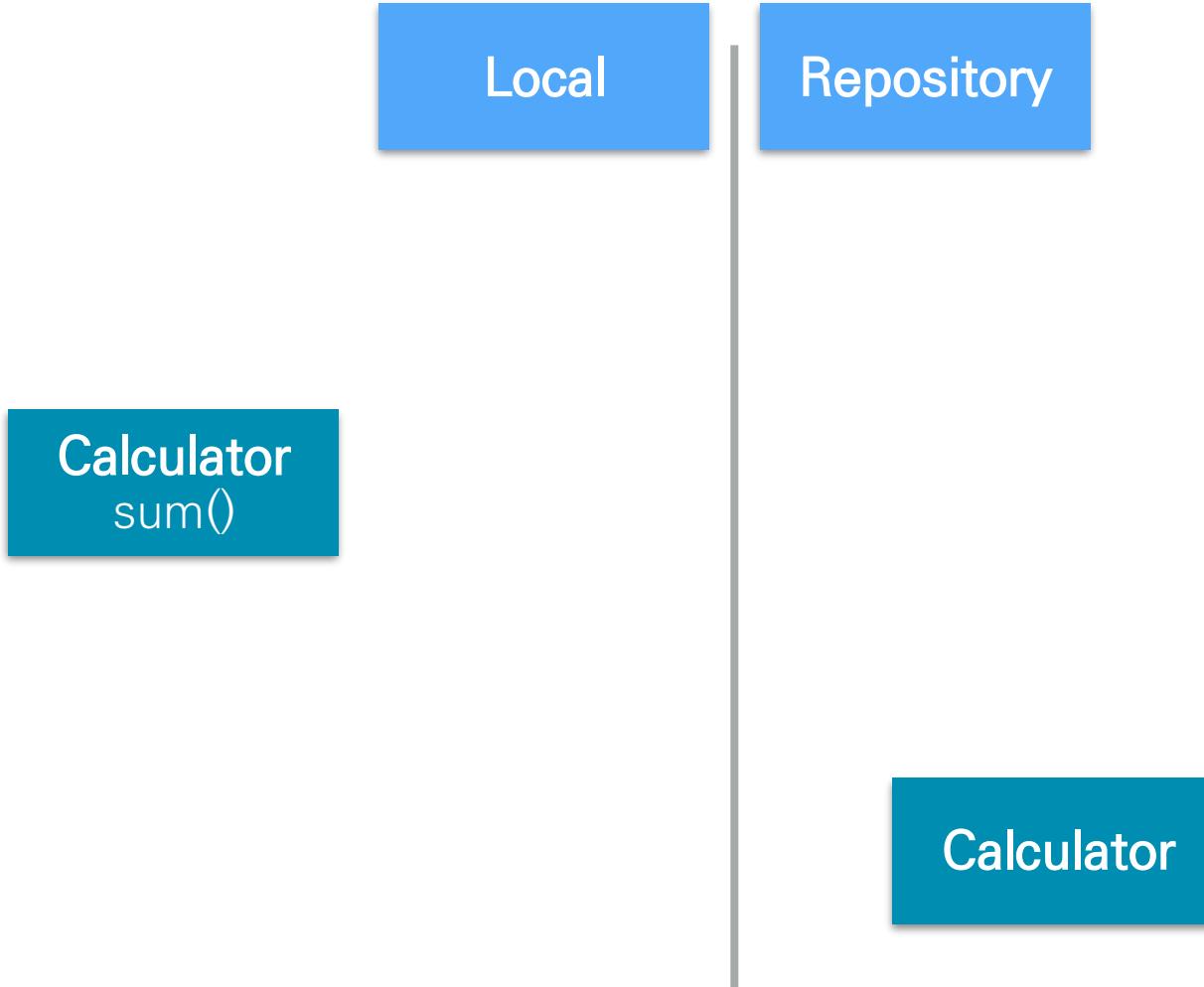
# Git – 그림설명

## ▶ Repository 에 저장

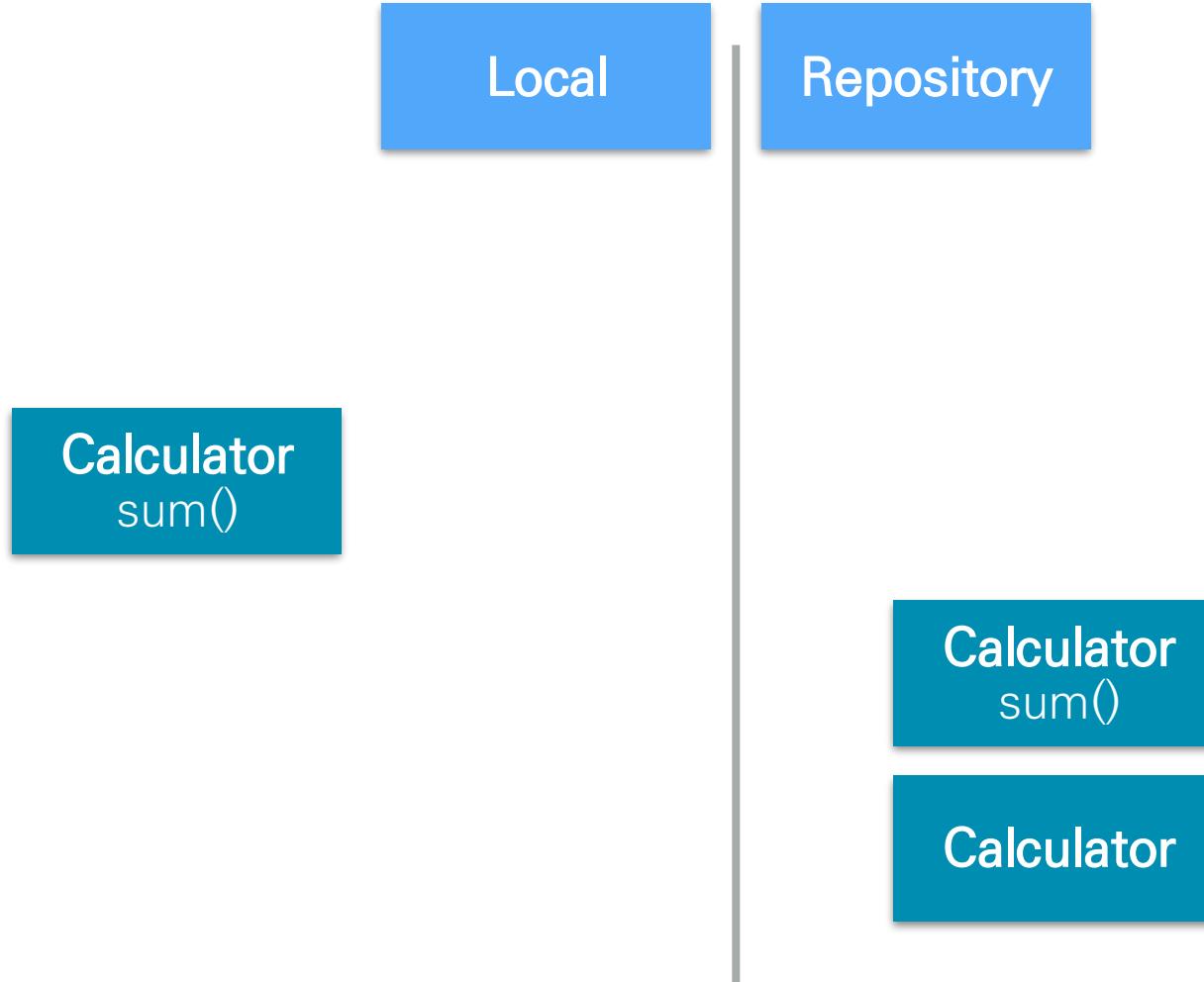
- Commit



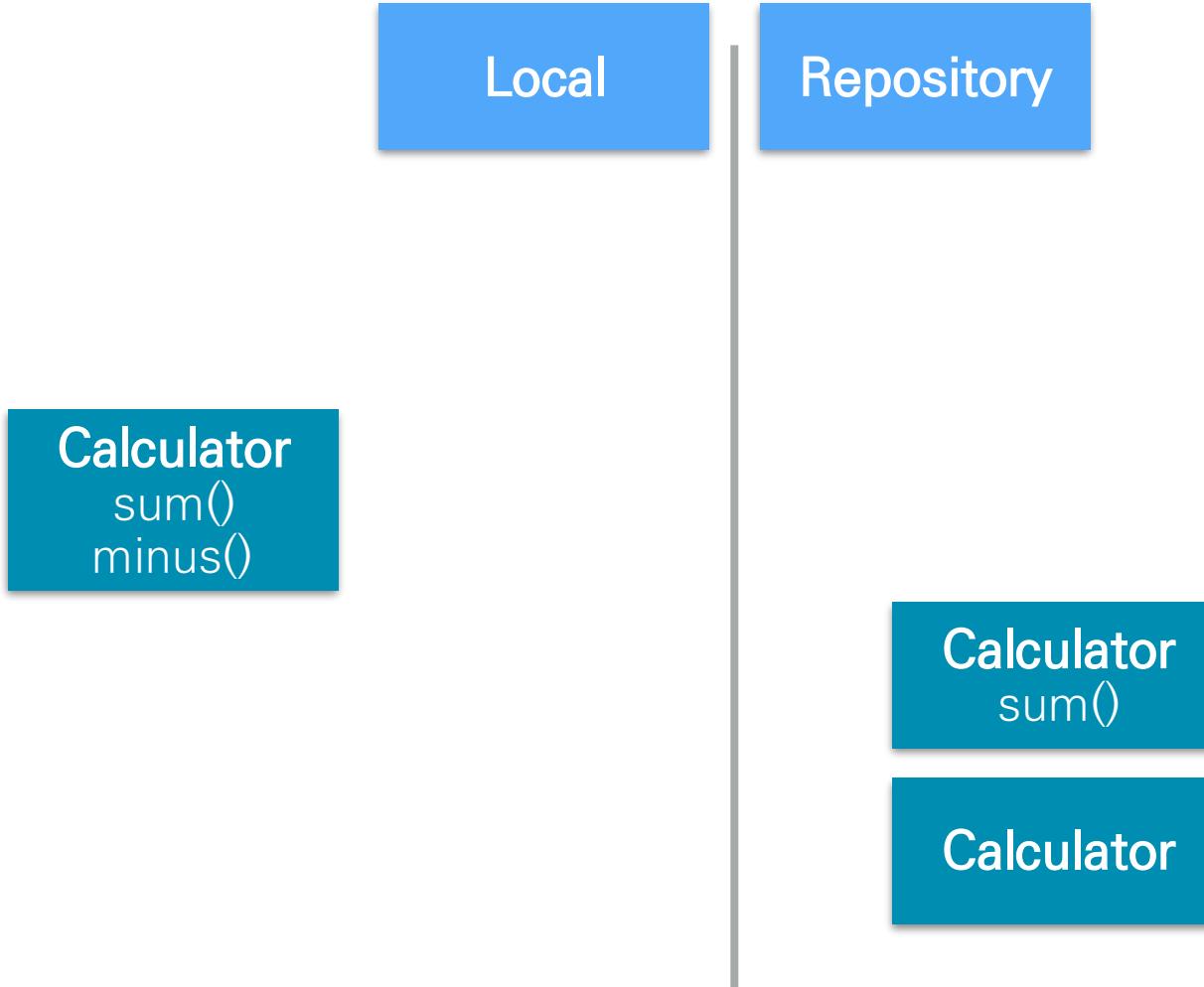
# Git - 그림 설명



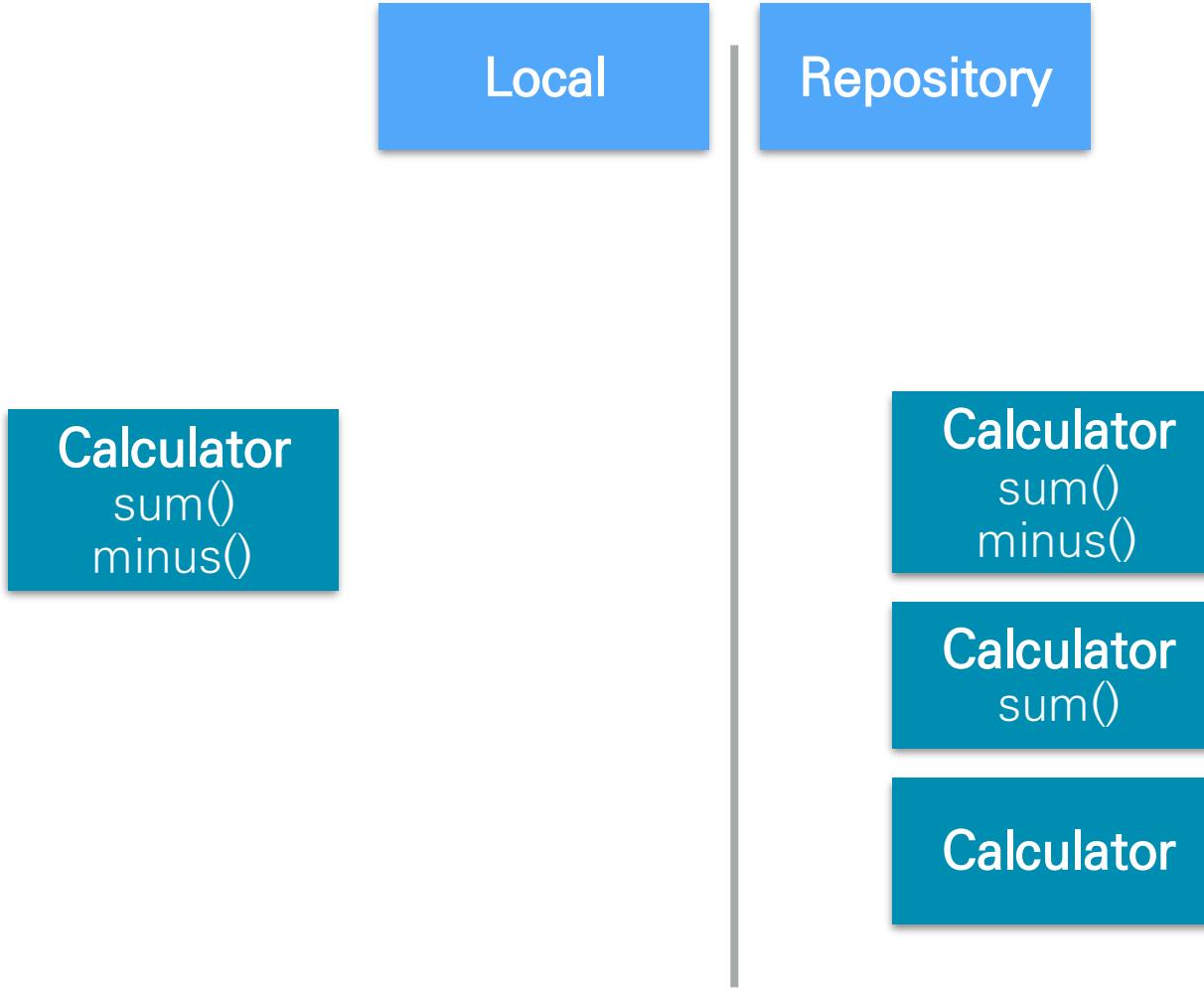
# Git - 그림 설명



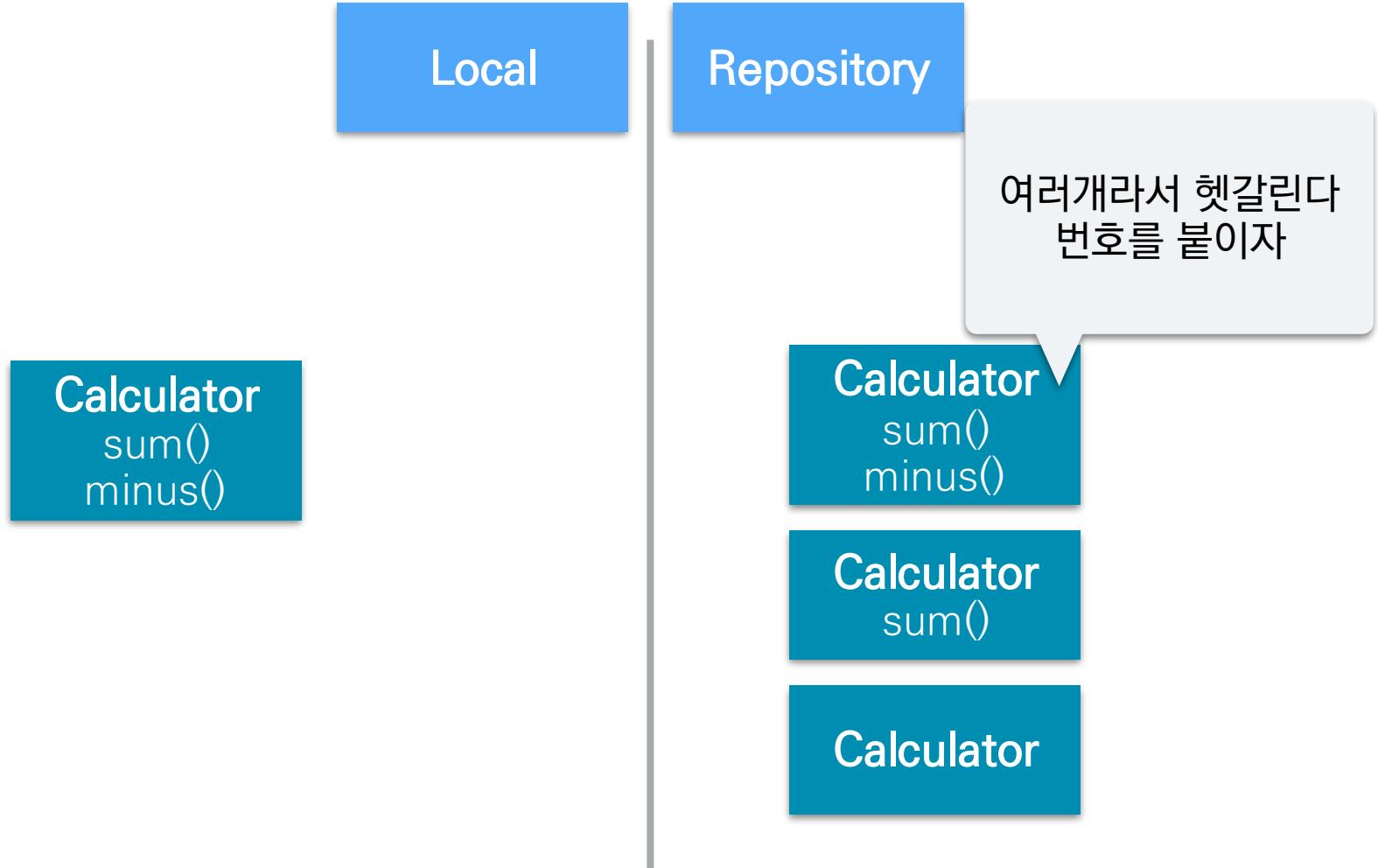
# Git – 그림 설명



# Git – 그림 설명



# Git – 그림 설명



# Git - 그림 설명

Local

Repo

..  
..

S U B V E R S I O N

Calculator

3

sum()  
minus()

Calculator

3

sum()  
minus()

Calculator

2

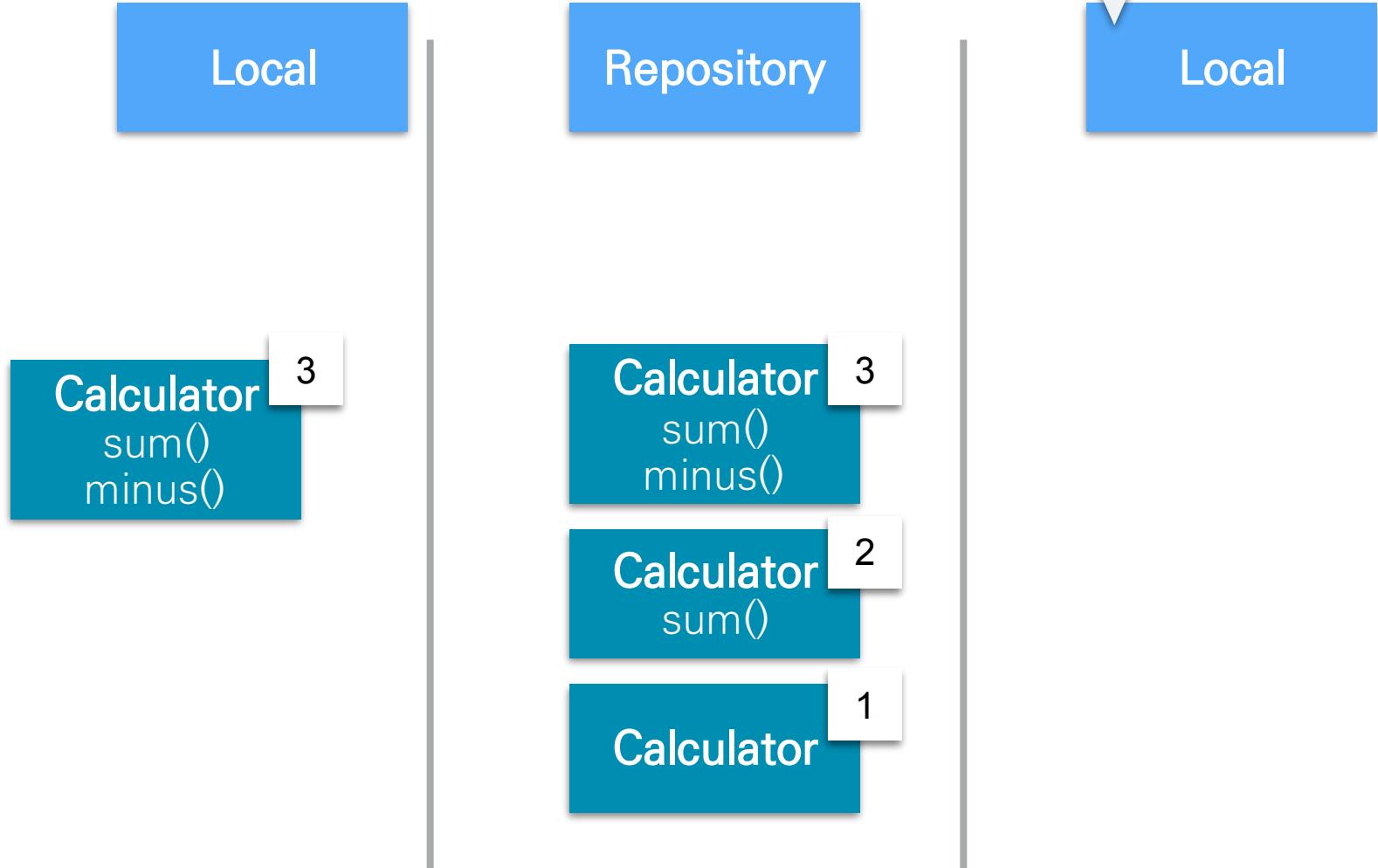
sum()

Calculator

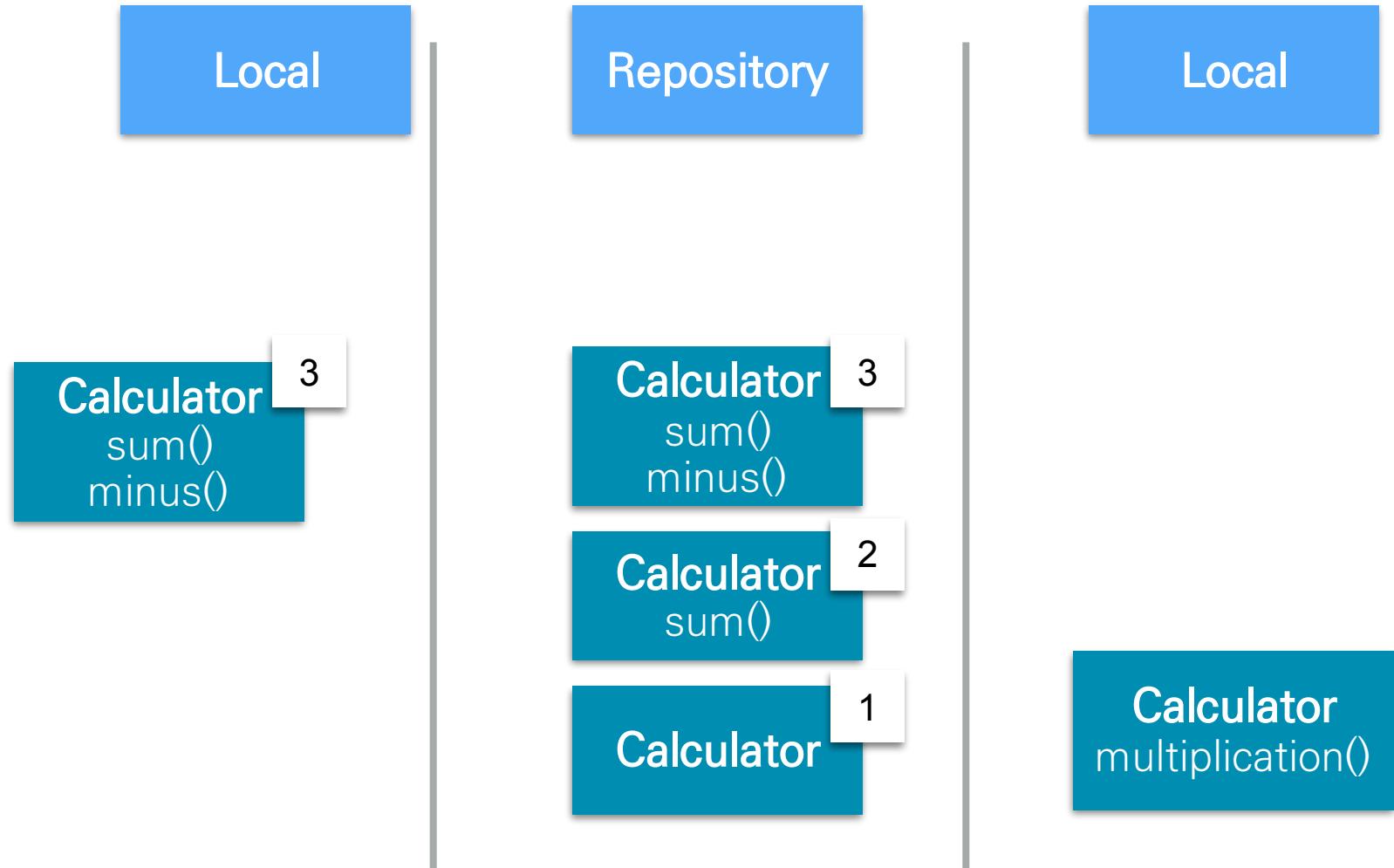
1

# Git – 그림 설명

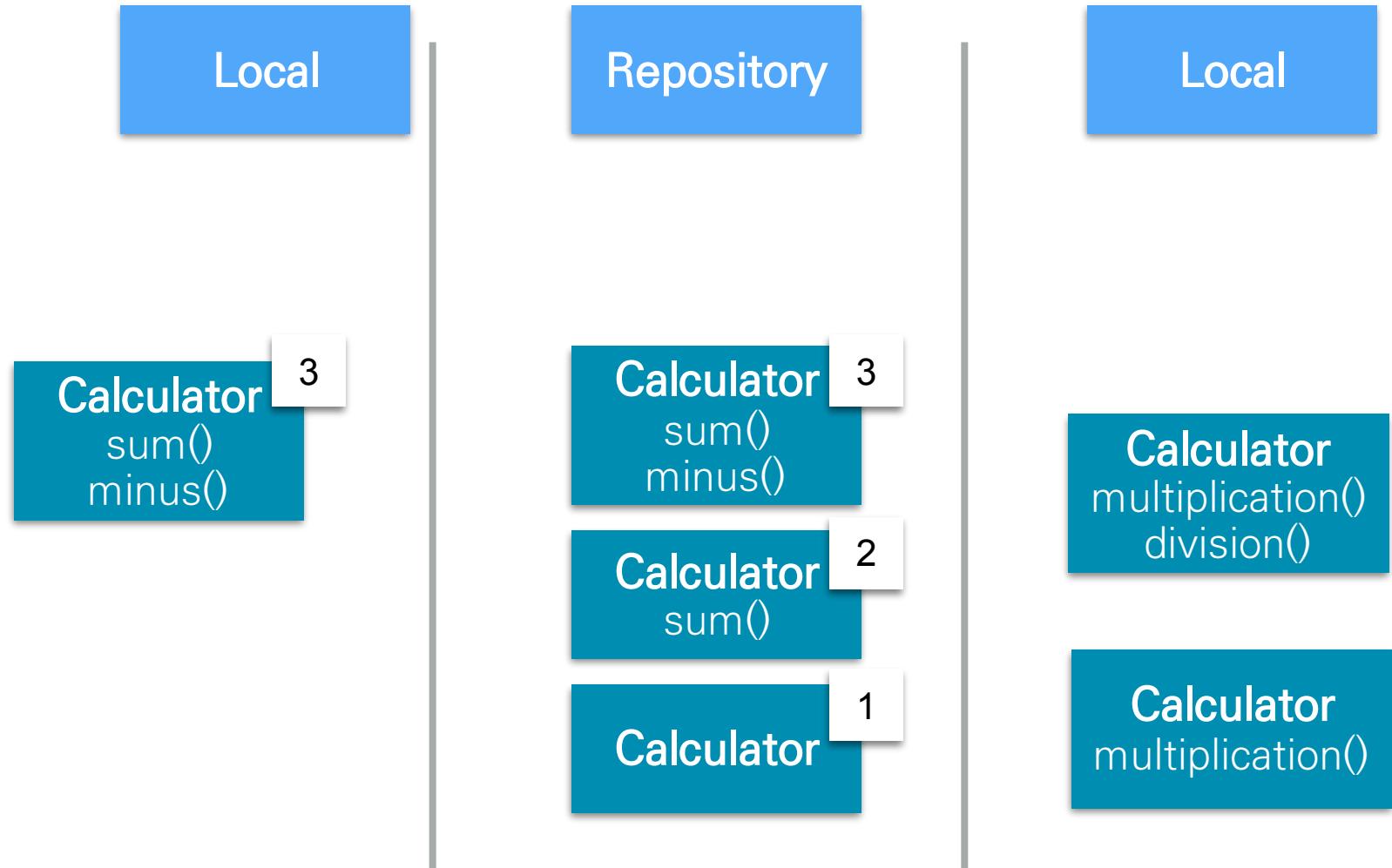
곱셈을 개발하는  
동료



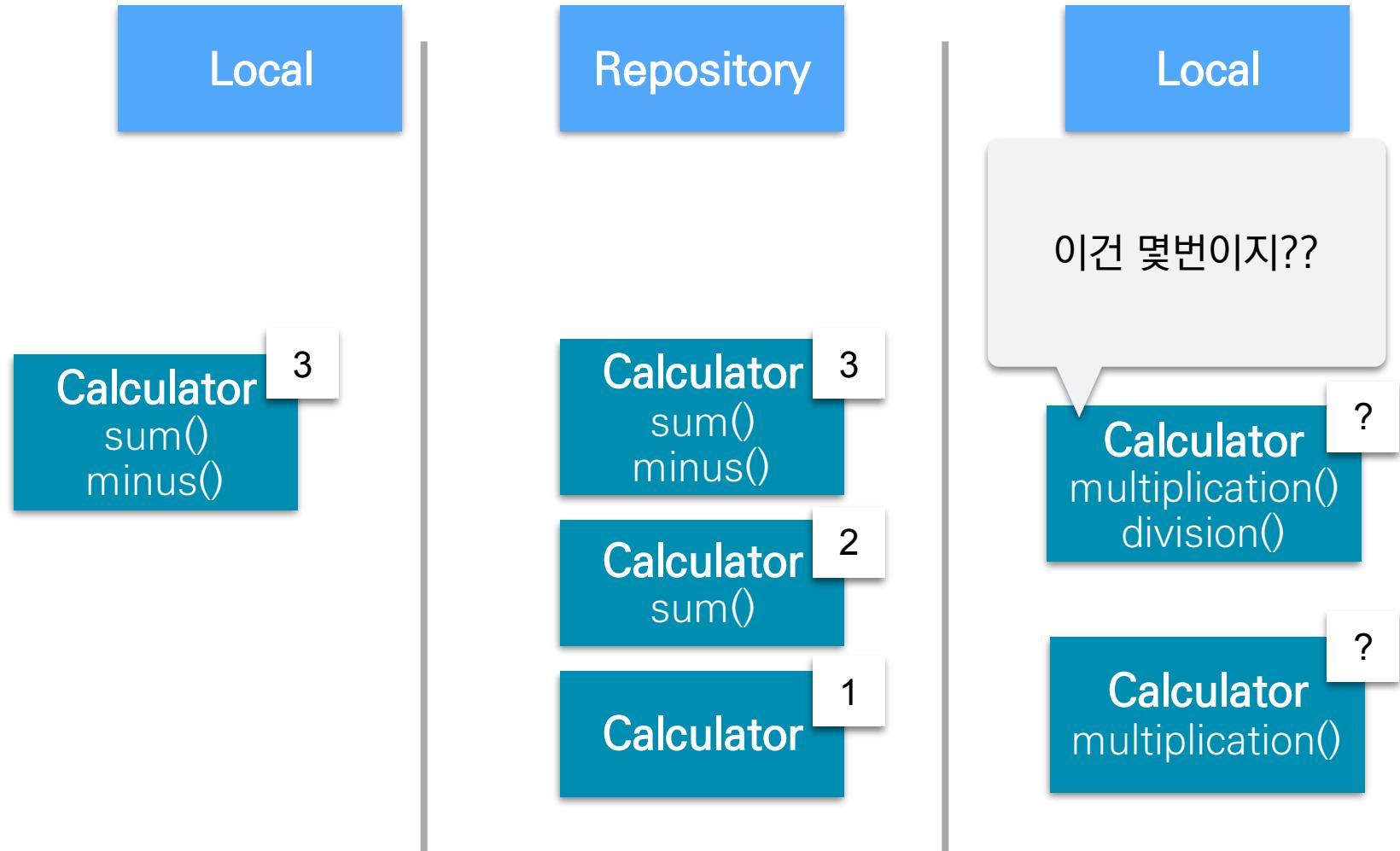
# Git – 그림 설명



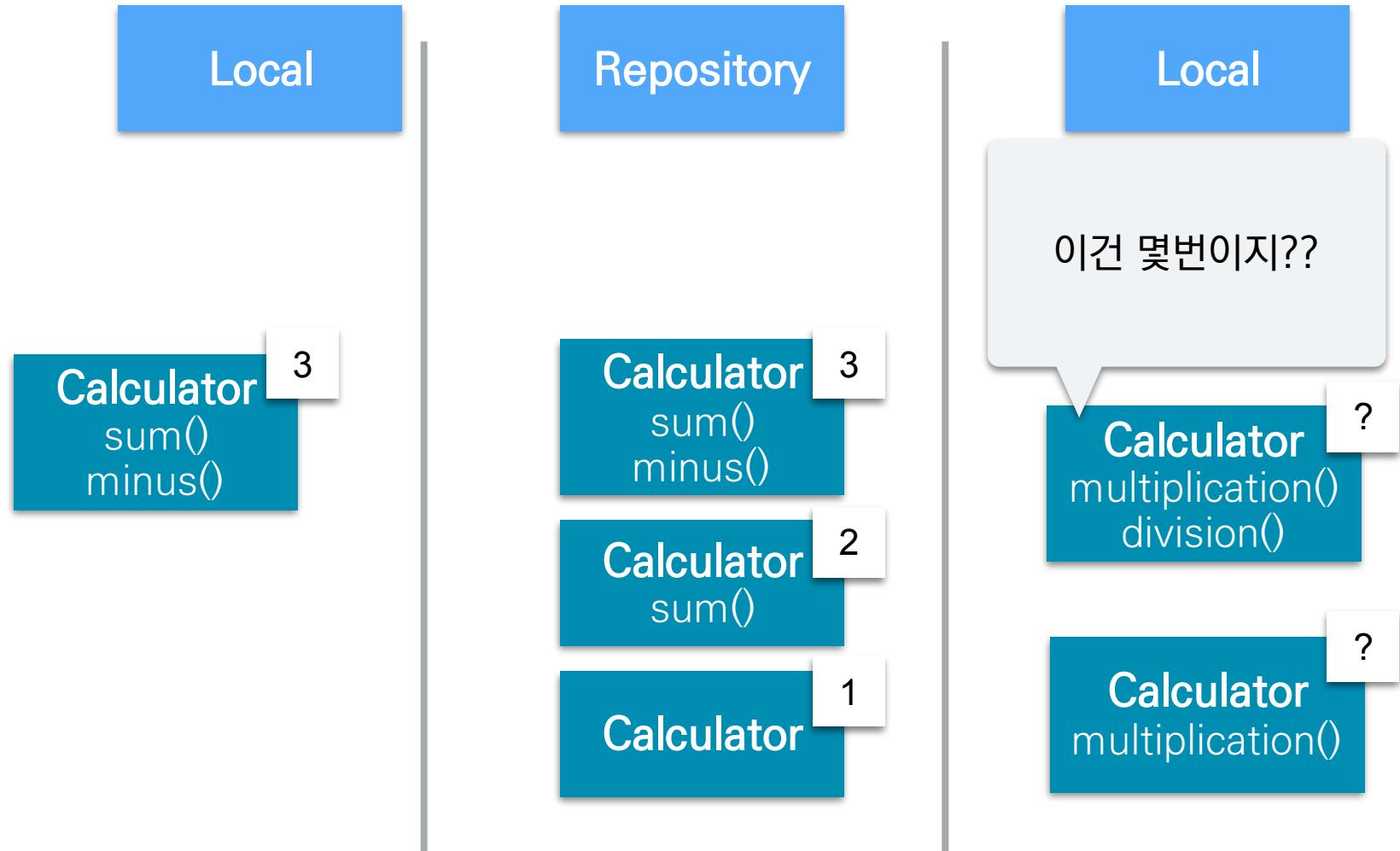
# Git – 그림 설명



# Git – 그림 설명



# Git – 그림 설명



# Git - 그림설명

변호로 암해!!  
Hash 씨자

# Git – Hash

## ▶ 모양

- “834a6d13967395216936419680ebd65ef8550566”
- “doca7b8294401dcfdaafa406c5ac039d8fb98c32”

# Git – Hash

- 서버에 커밋 순번을 불일 수 없다
  - 나의 로컬 Repository에 커밋
  - 다른 개발자도 로컬에 커밋
- 그냥 아무값이나 붙이자
  - Hash
  - SHA-1 값을 생성하여 붙인다 (랜덤값)
- 아무값이나 붙이면 순서는?
  - 부모를 기록하자

# Git – Hash

## ▶ 랜덤값이면 중복은?

- 가능
- 확률 :  $1/2^{80}$ 
  - 1.2경 : 12,000,000,000,000,000
  - 지구에 존재하는 모래알 수의 1,200배
- 확률 예시
  - 지구상의 6.5억명의 개발자가 있다고 가정하고
  - 매초, 리눅스 커널 히스토리 전체(100만개)를 Push 할 때
  - 5년이 지나면, 중복 확률이 50% 정도 된다 (from Pro Git 서적)

# Git – Hash

## ▶ 모양

- “834a6d13967395216936419680ebd65ef8550566”
- “doca7b8294401dcfdaafa406c5ac039d8fb98c32”

The screenshot shows a GitHub repository dashboard for 'iot-labs / dashboard'. The commit history is displayed, with one specific commit highlighted by a red box. The commit details are as follows:

- Author:** jongkwang
- Date:** committed on GitHub on Jun 29
- Parents:** 1 parent 6fab19b
- Commit Hash:** commit 9872e70f9fe5ff3dc387521fd74705c4201100c8

Below the commit, it says "Showing 1 changed file with 35 additions and 0 deletions." The file shown is README.md, which has 35 additions and 0 deletions.

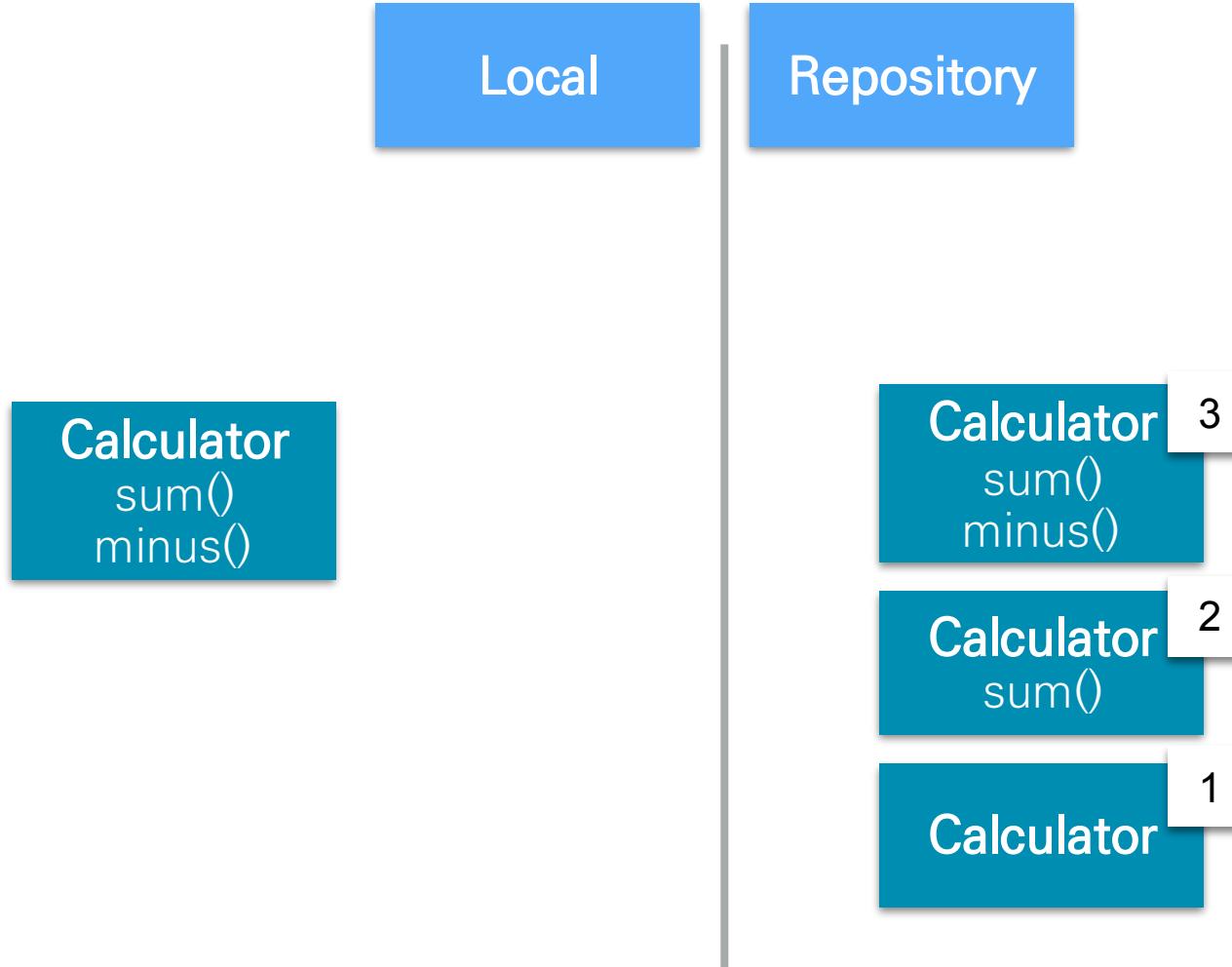
# Git – Hash

## ▶ 모양

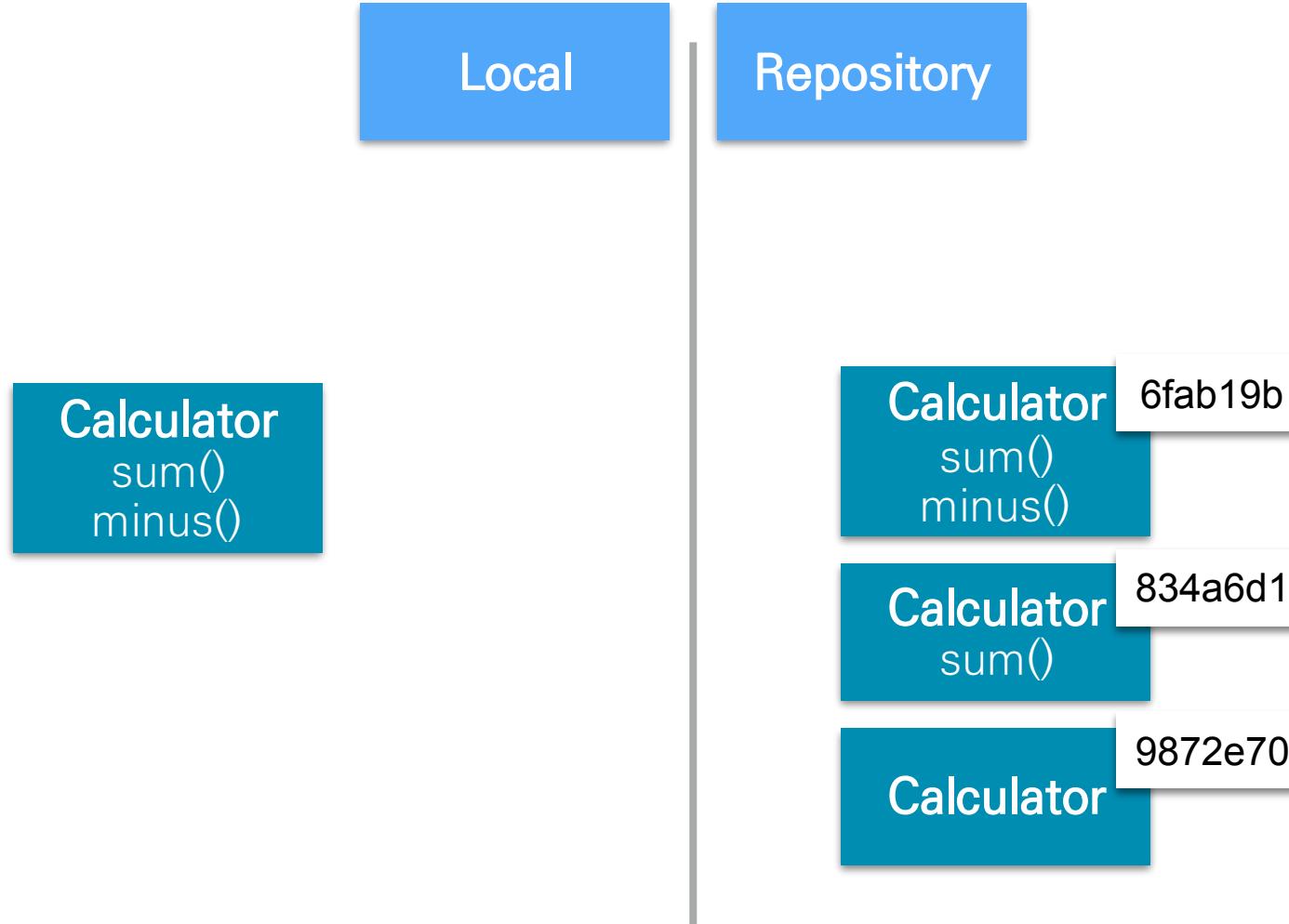
- “**834a6d13967395216936419680ebd65ef8550566**”
- “**doca7b8294401dcfdaafa406c5ac039d8fb98c32**”

The screenshot shows a GitHub repository dashboard for the 'iot-labs / dashboard' repository. The 'Code' tab is selected. At the top right, there are buttons for Unwatch (3), Star (6), and Fork (3). Below the tabs, there are links for Issues (1), Pull requests (0), Projects (0), Wiki, Settings, and Insights. The branch is set to 'master'. The main content area shows a commit history for the file 'dashboard / README.md'. The first commit is by 'jongkwang' with the message 'Add a license policy.' and a timestamp of '9872e70 on Jun 29'. This commit is highlighted with a red rectangular box. Below the commit, it says '1 contributor'. At the bottom of the commit card, there are buttons for Find file and Copy path. At the very bottom of the page, there are buttons for Raw, Blame, History, and file operations (Raw, Blame, History, copy, edit, delete).

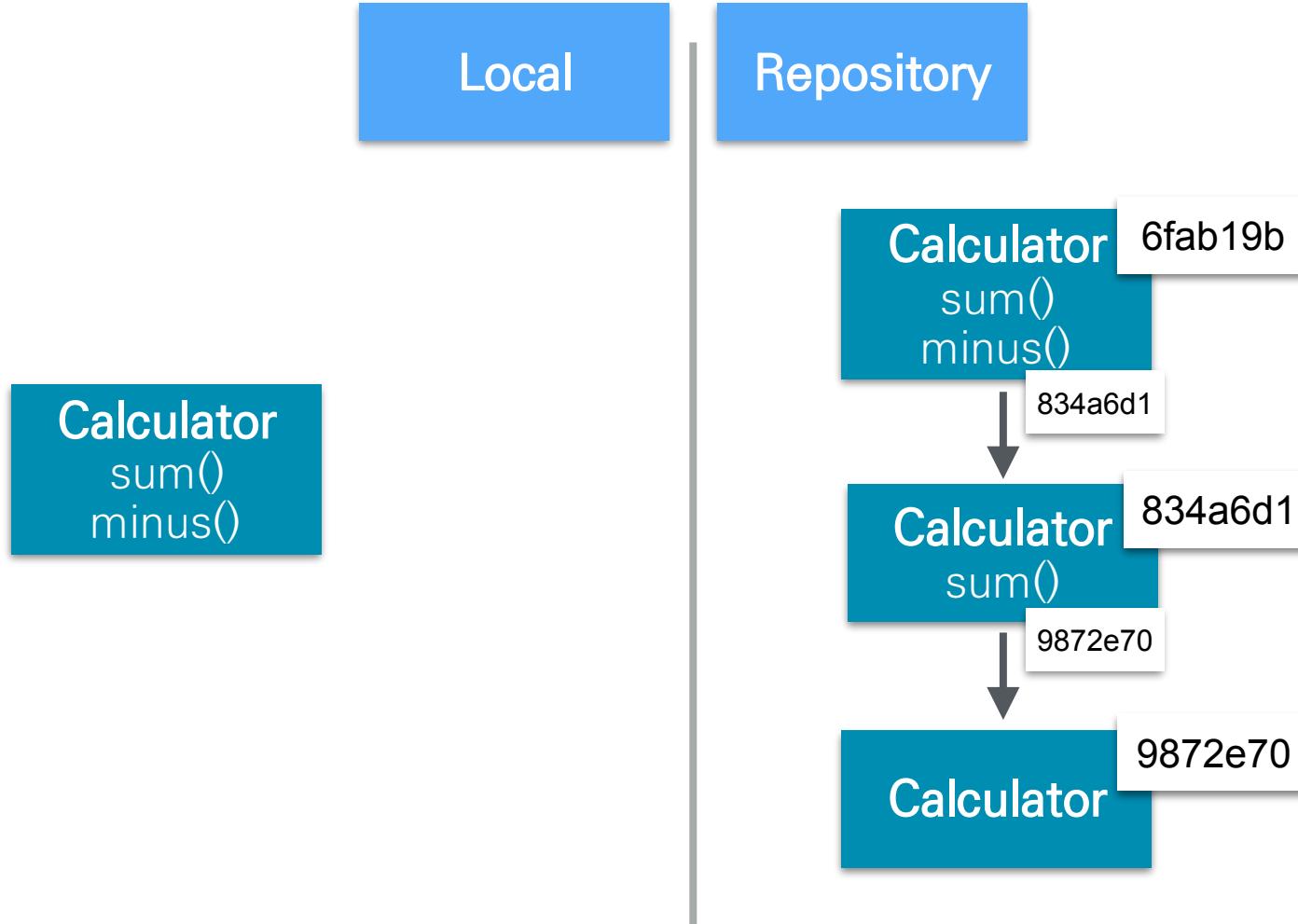
# Git – 그림 설명



# Git – 그림 설명



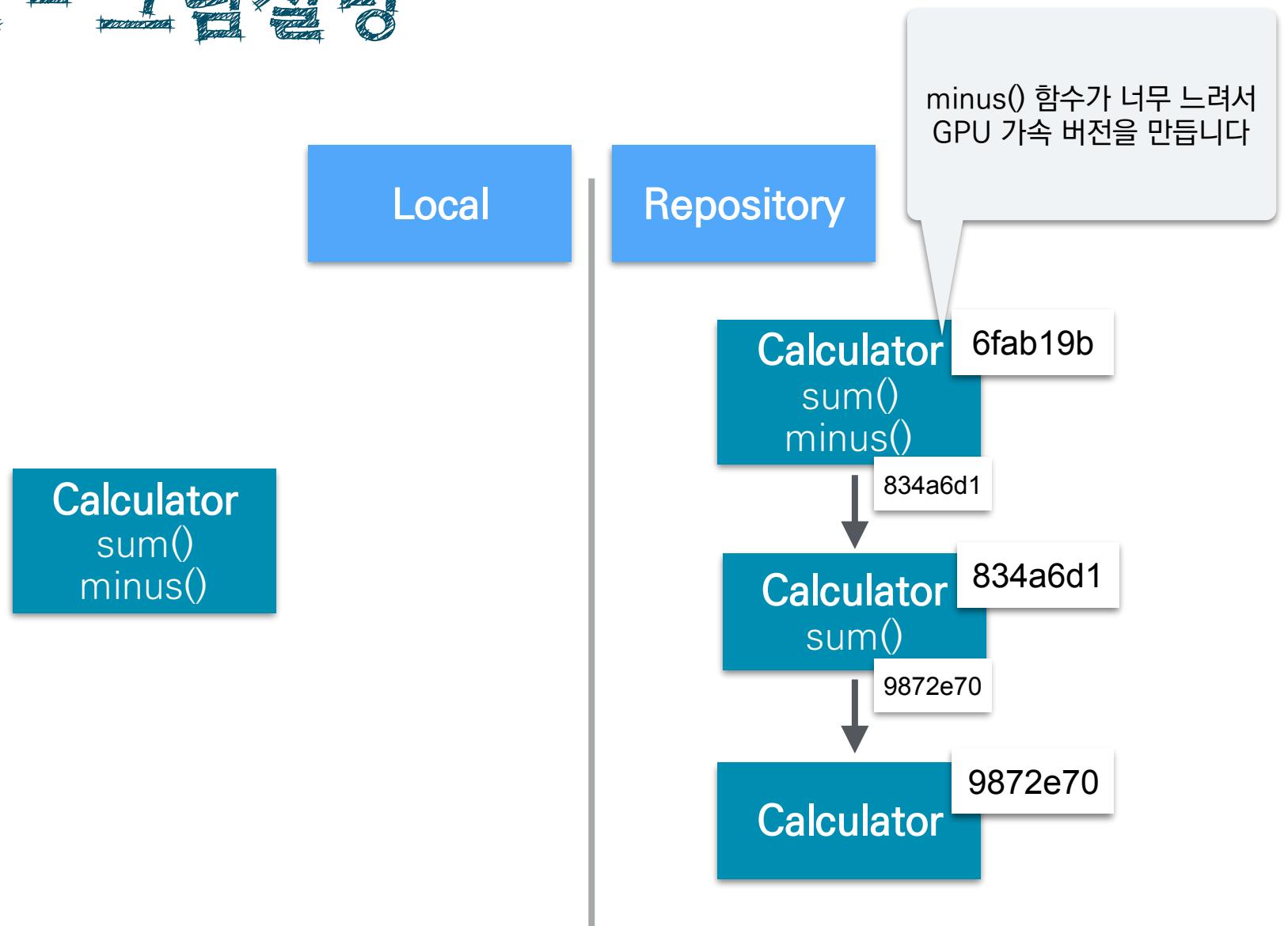
# Git – 그림 설명



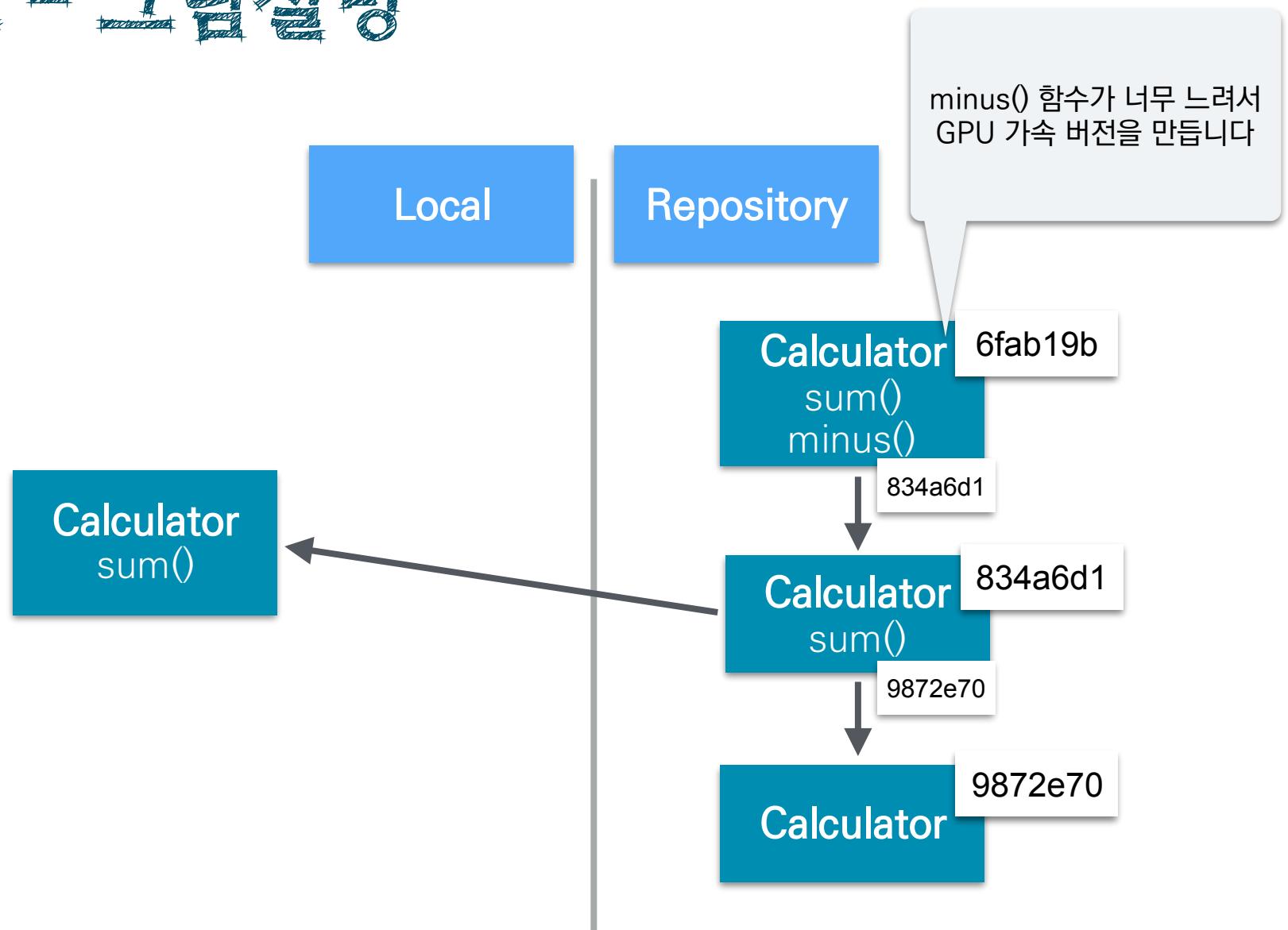
Git - 그림설명

minus0 재개발  
with GPU

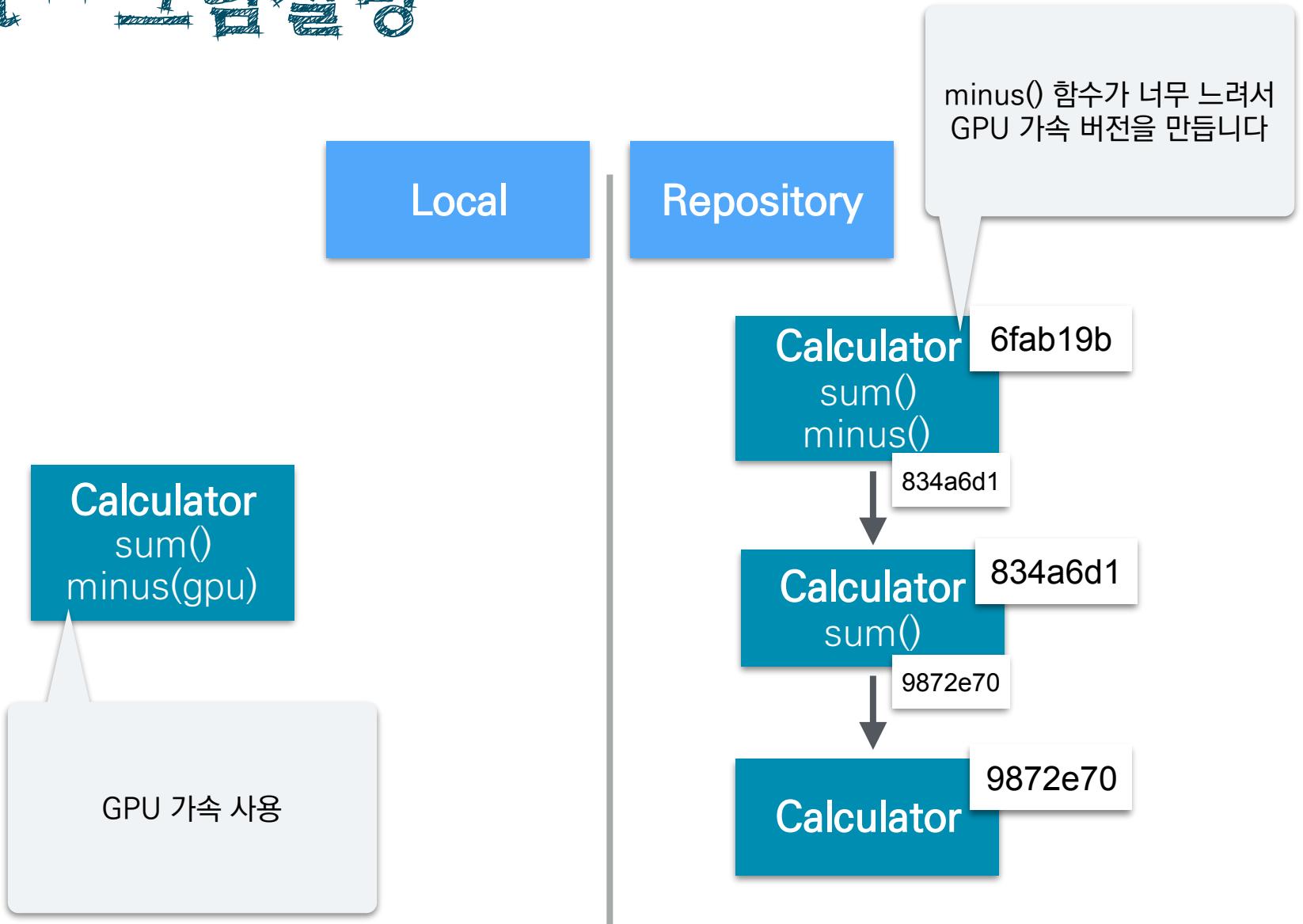
# Git – 그림 설명



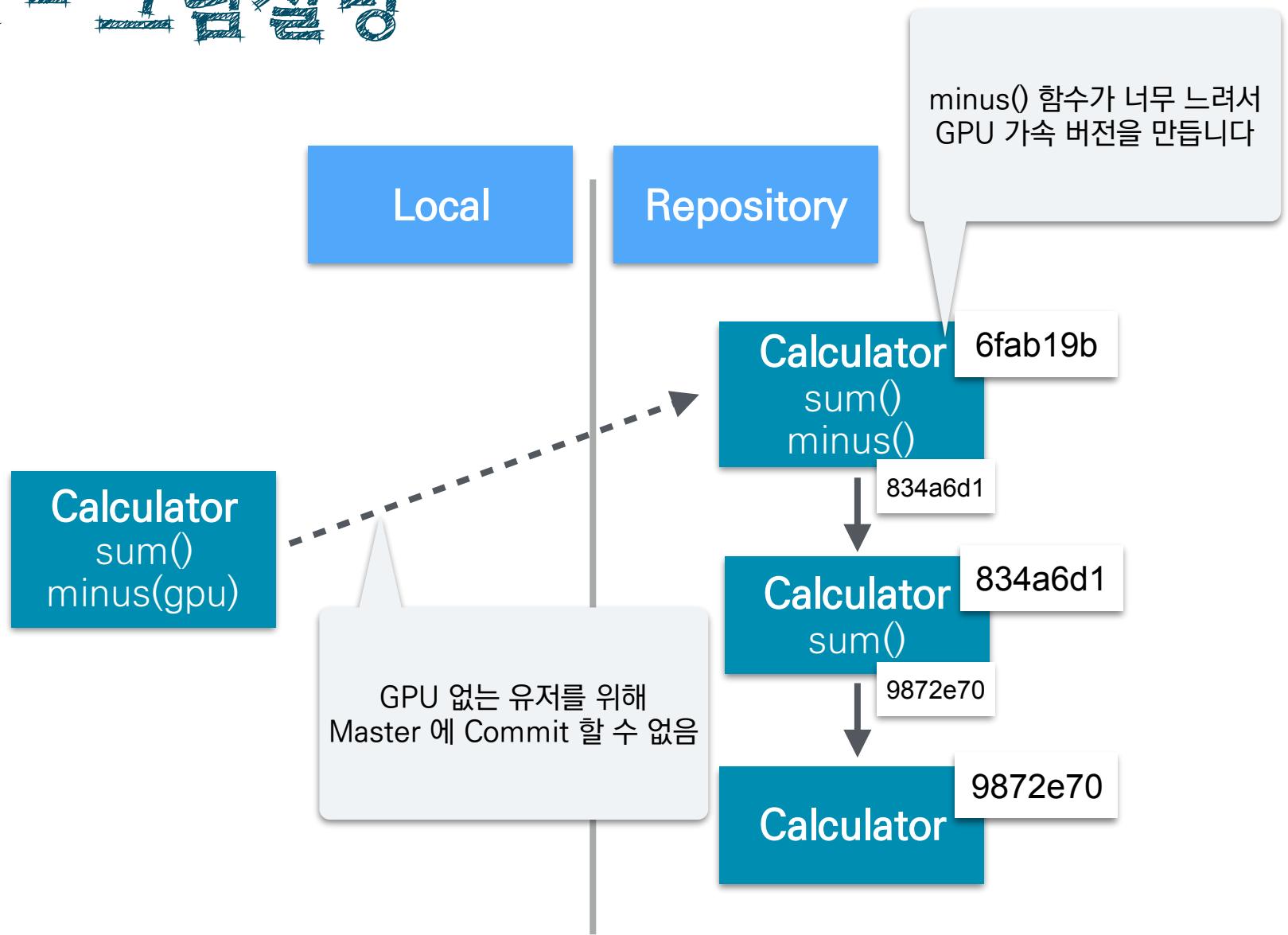
# Git – 그림 설명



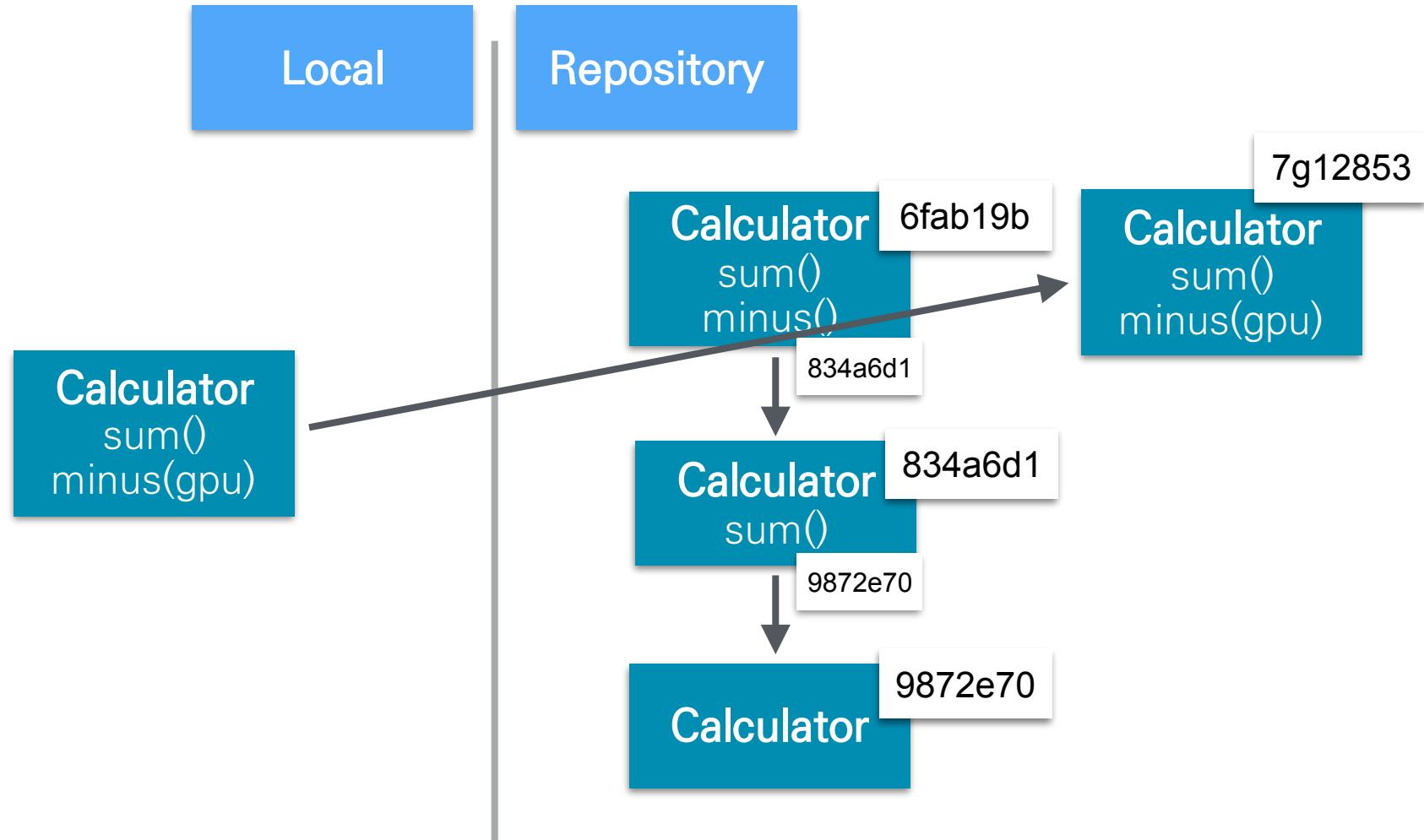
# Git – 그림 설명



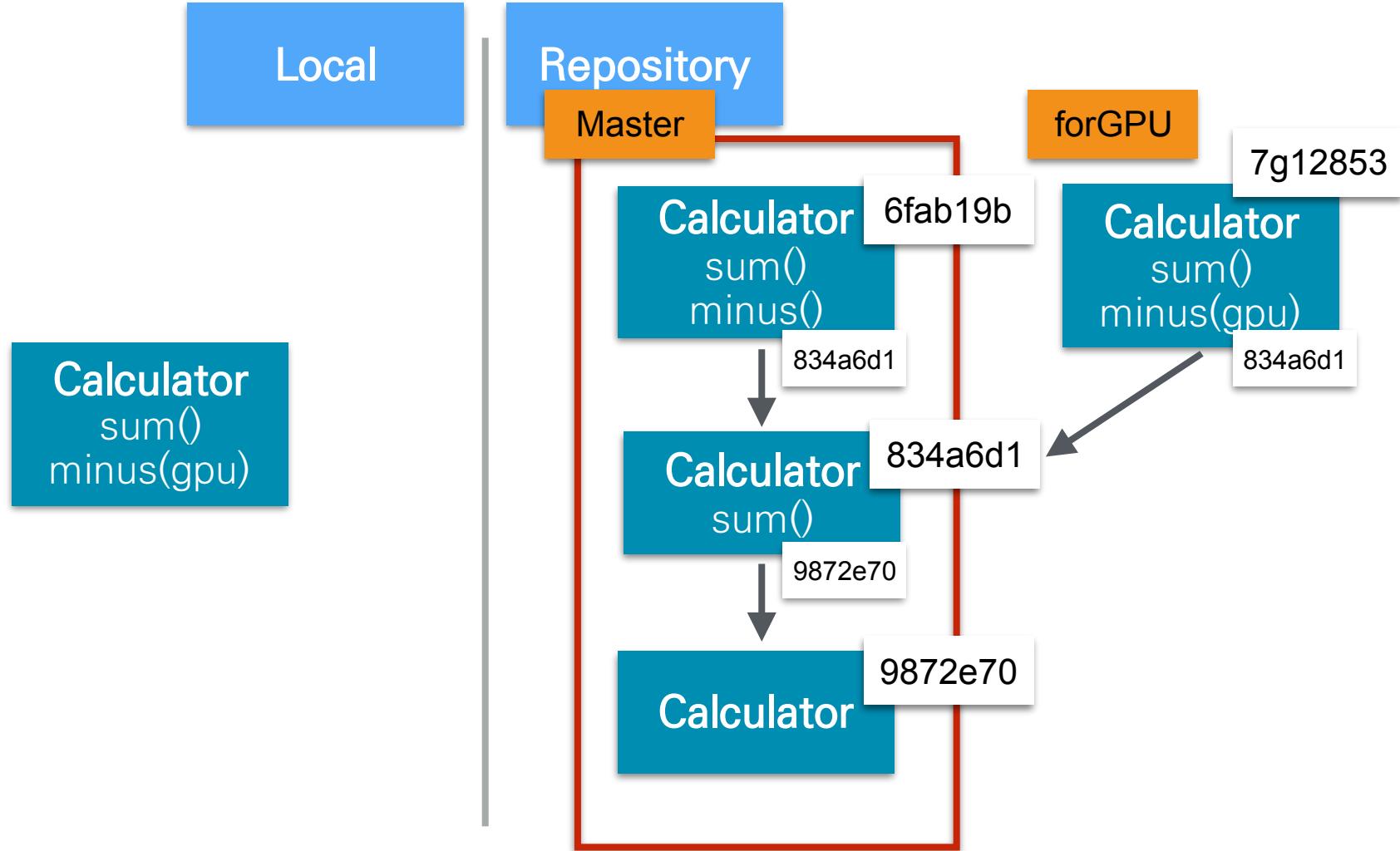
# Git – 그림 설명



# Git – 그림 설명



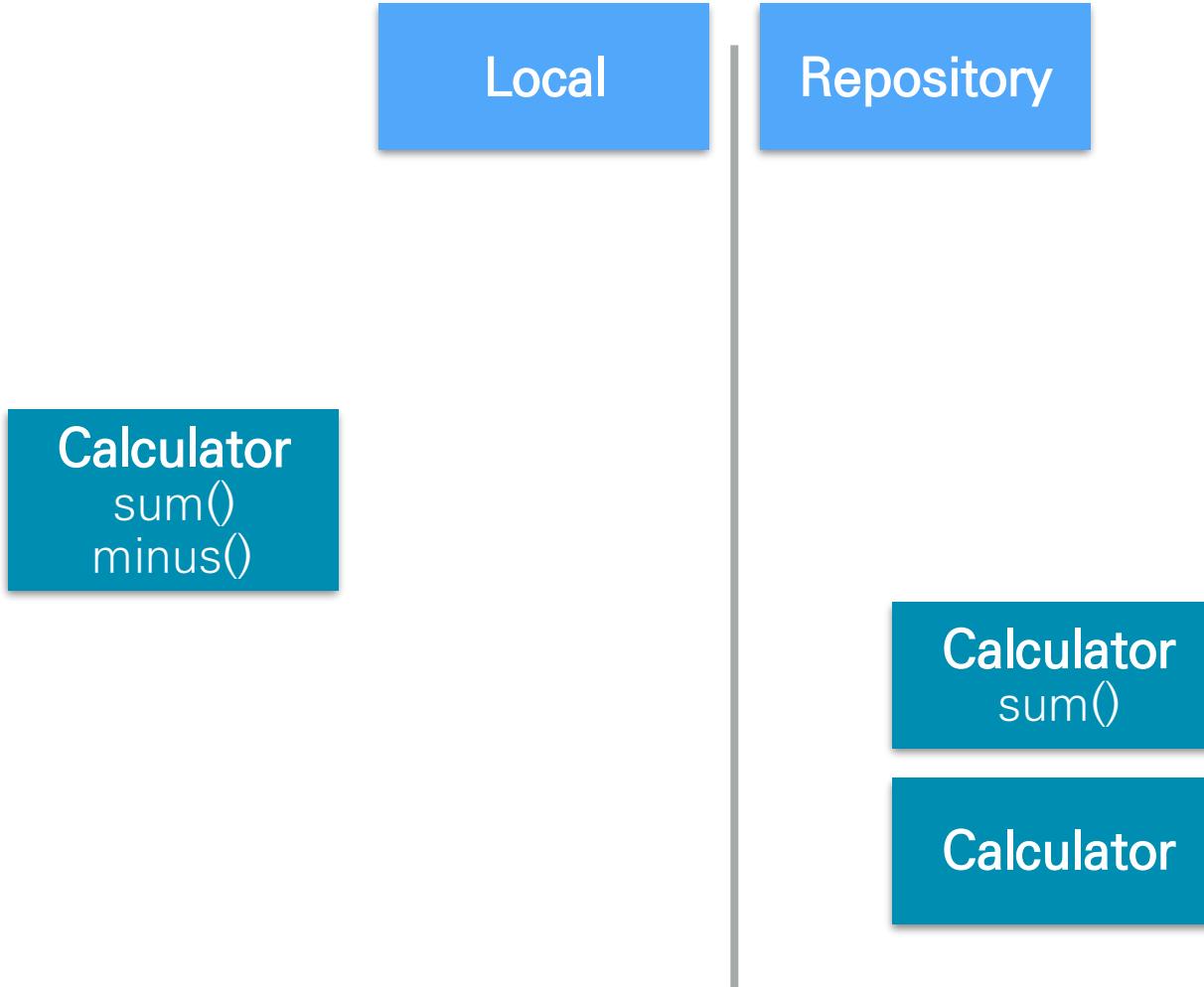
# Git – 그림 설명



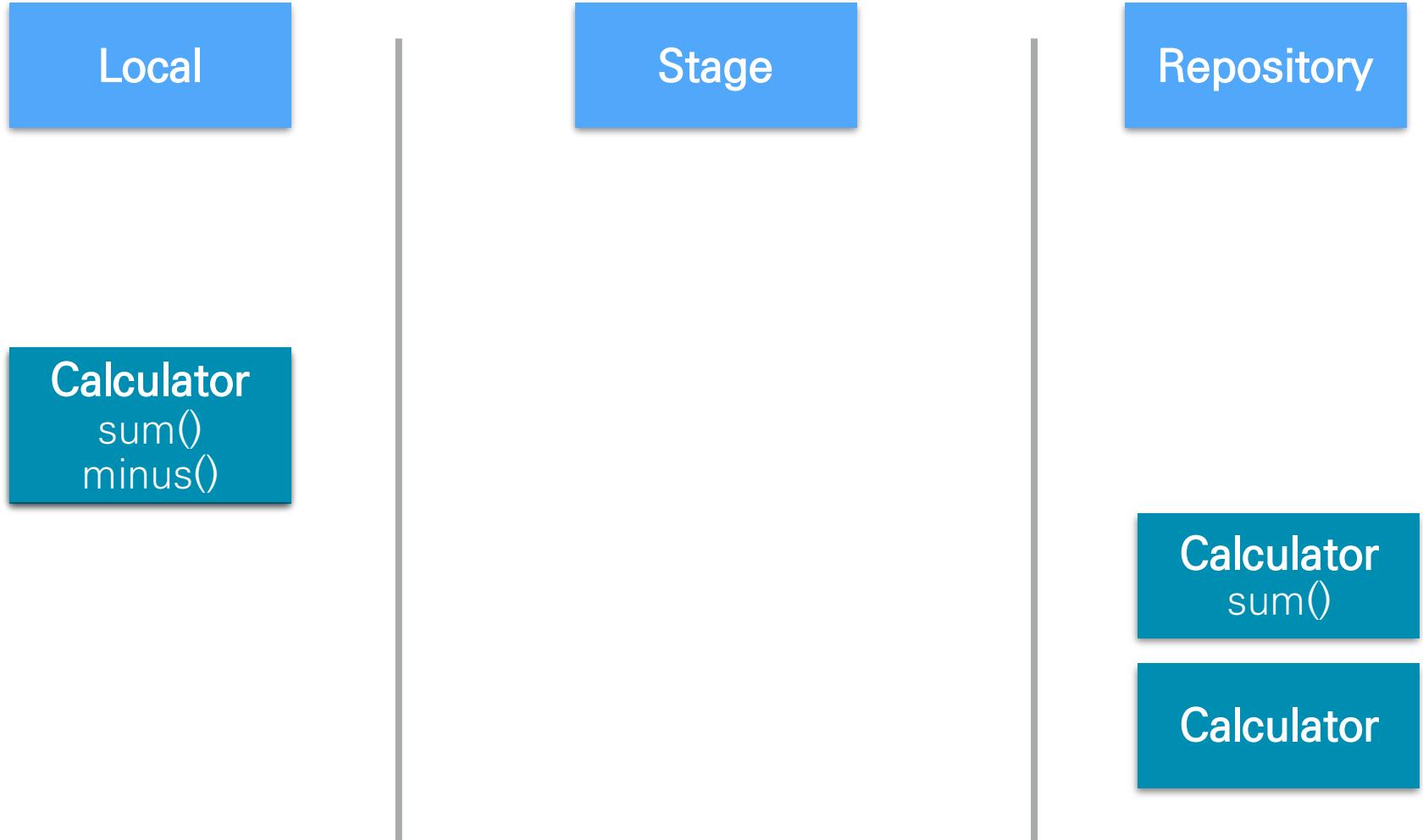
Git - 그림설명

Stage ← → 푸대

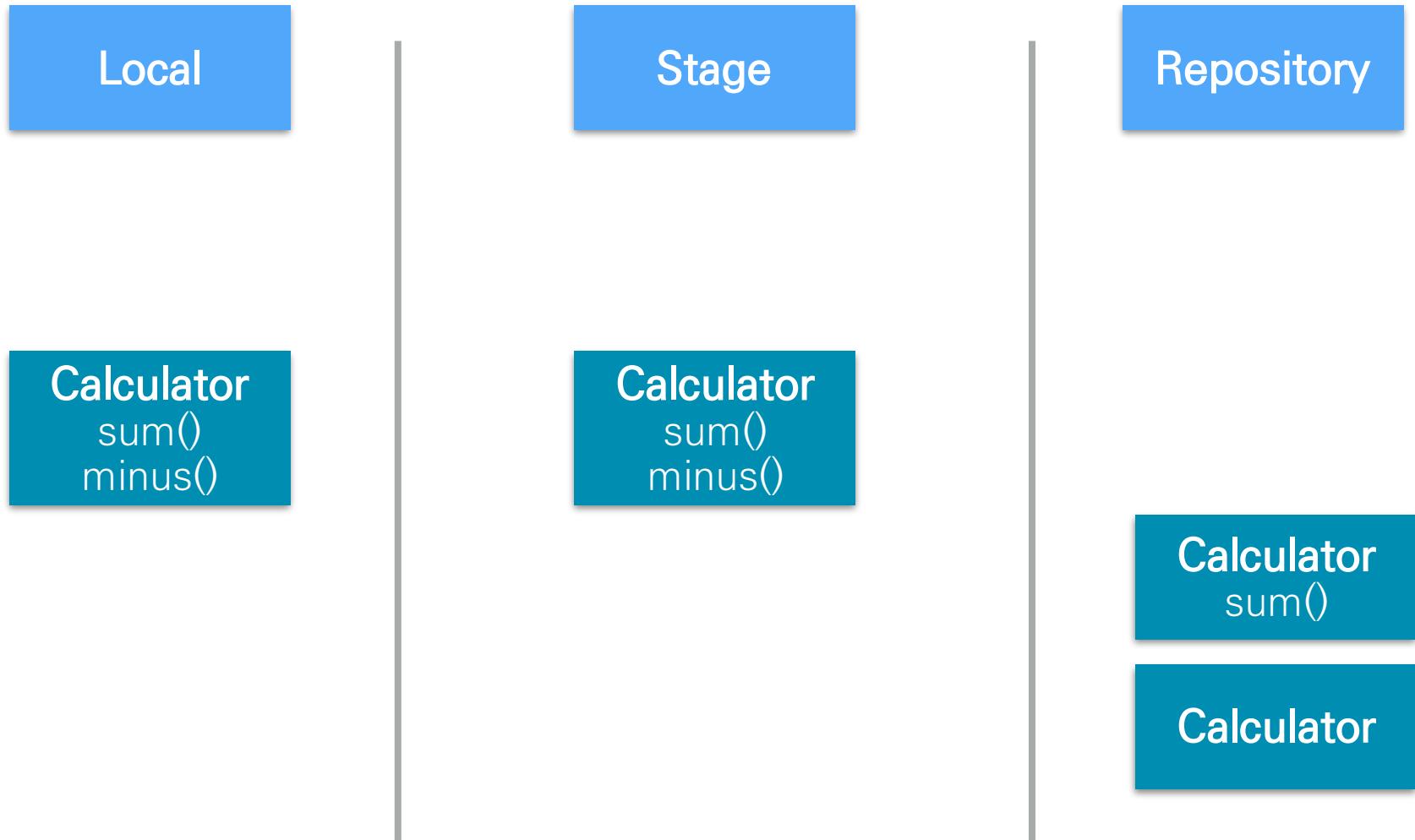
# Git – 그림 설명



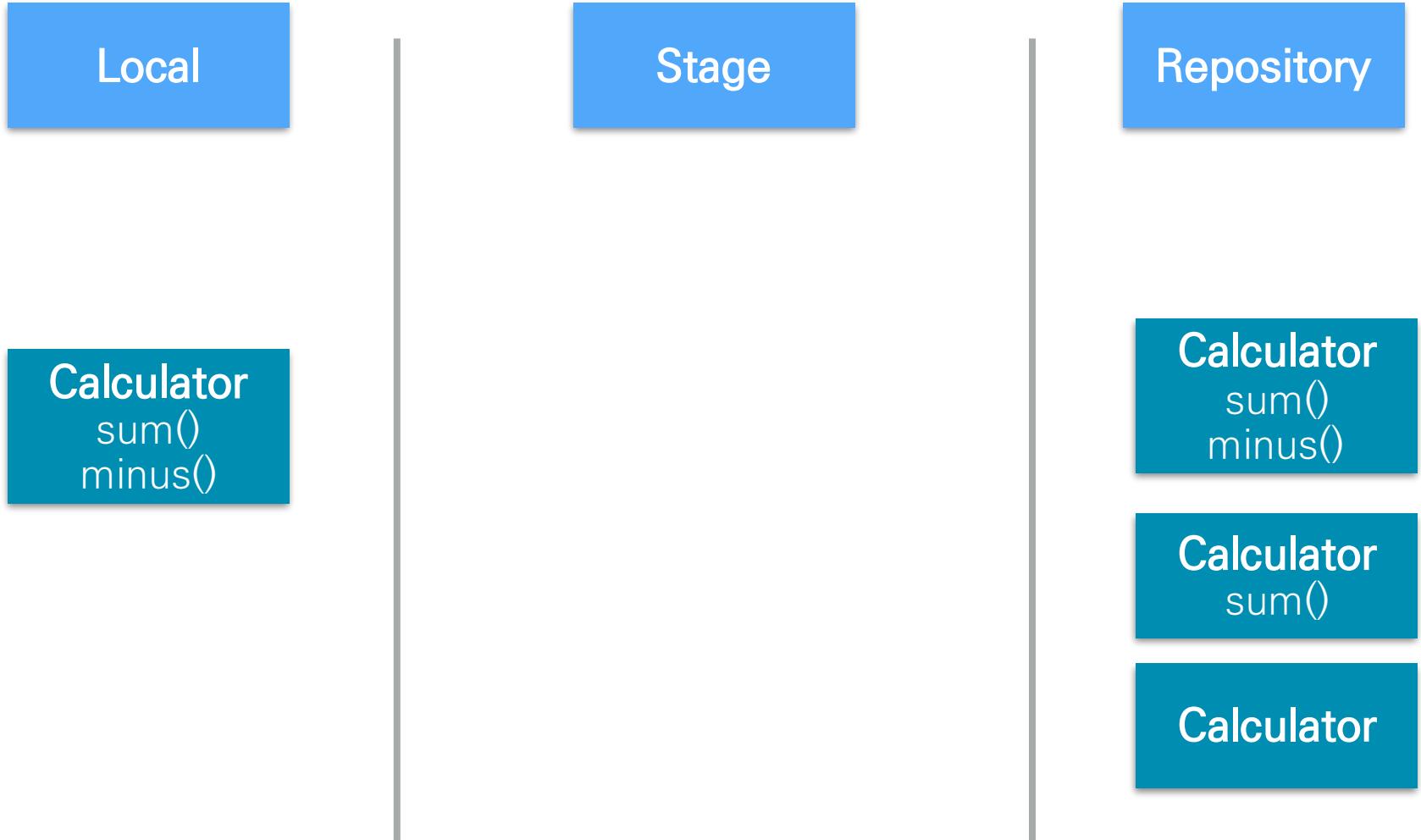
# Git - 그림 설명



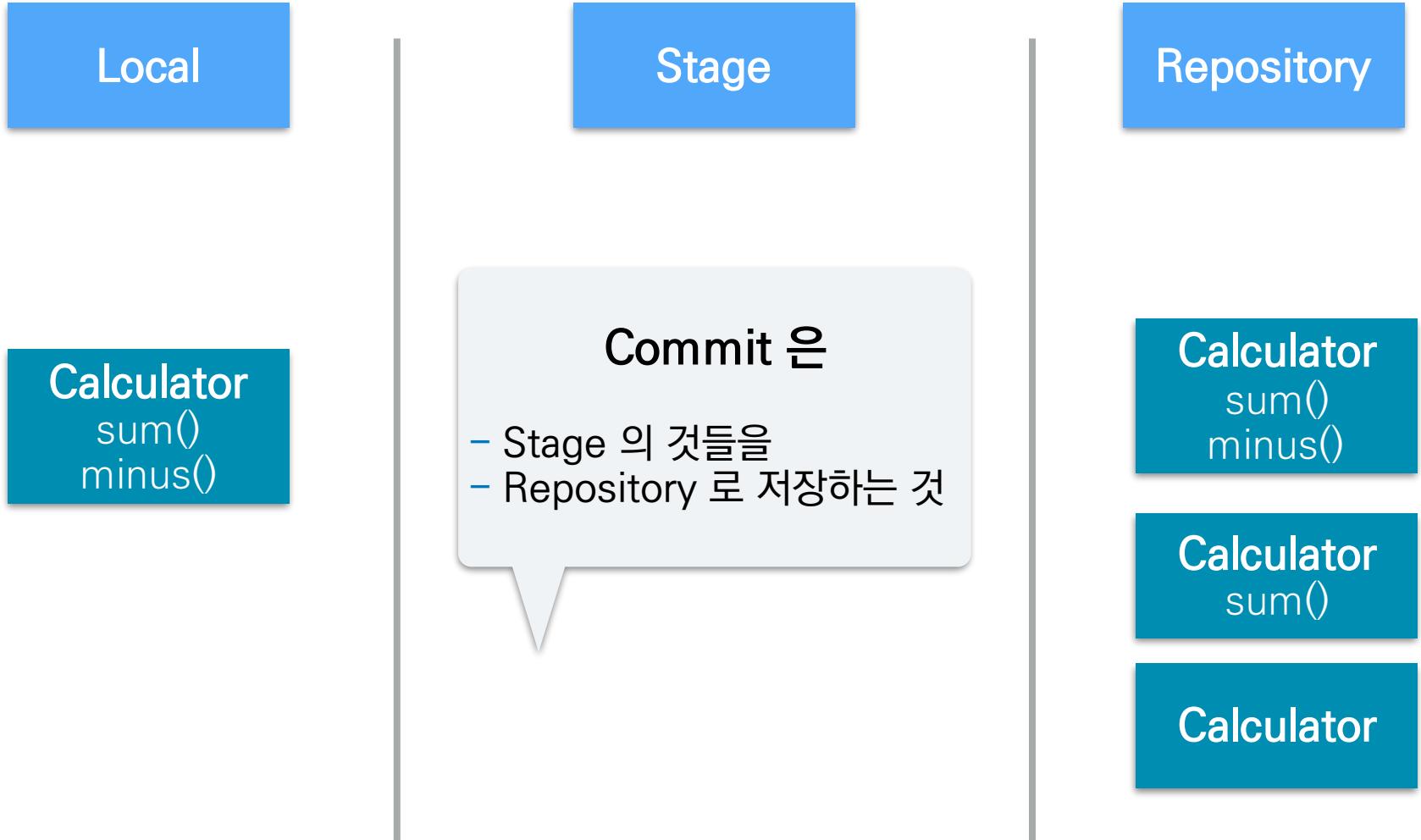
# Git - 그림 설명



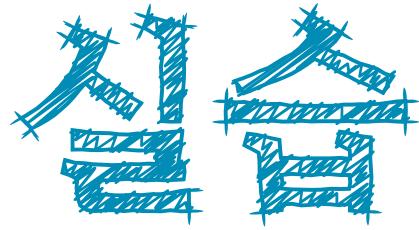
# Git - 그림 설명



# Git – 그림 설명



# Git - 그림설명



Git

Join Github

# Join Github

## ▶ Github

- <https://github.com>

The screenshot shows the GitHub 'Join GitHub' page. At the top, there's a navigation bar with links for Features, Business, Explore, Marketplace, Pricing, a search bar, and a 'Sign In' button. The main title 'Join GitHub' is displayed prominently, followed by the subtitle 'The best way to design, build, and ship software.' Below this, there are three steps: 'Step 1: Create personal account', 'Step 2: Choose your plan', and 'Step 3: Tailor your experience'. The first step is currently active. The 'Create your personal account' form includes fields for 'Username' (placeholder: 'YourUsername'), 'Email Address' (placeholder: 'youremail@example.com'), and 'Password' (placeholder: 'password'). Below the form, a note states: 'By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).'. A large green 'Create an account' button is at the bottom of the form. To the right, a sidebar titled 'You'll love GitHub' lists benefits: 'Unlimited collaborators', 'Unlimited public repositories', 'Great communication', 'Frictionless development', and 'Open source community', each preceded by a green checkmark.

# Join Github

- ▶ Unlimited public repositories for free
  - 공개SW 무제한 무료

The screenshot shows the GitHub welcome screen after account creation. At the top, there's a navigation bar with icons for user profile, search, pull requests, issues, marketplace, and gist. Below the bar, the main heading is "Welcome to GitHub" with the subtext "You've taken your first step into a larger world, @jongkwangmail." Three steps are outlined: "Completed Set up a personal account" (green checkmark), "Step 2: Choose your plan" (blue icon), and "Step 3: Tailor your experience" (grey gear icon). The "Choose your personal plan" section contains two options: a selected radio button for "Unlimited public repositories for free." and an unselected radio button for "Unlimited private repositories for \$7/month.". A note below says "Don't worry, you can cancel or upgrade at any time." There's also a checkbox for "Help me set up an organization next" with explanatory text about organizations. A green "Continue" button is at the bottom. To the right, a box lists "Both plans include:" with several checked items: collaborative code review, issue tracking, open source community, unlimited public repositories, and joining any organization.

Welcome to GitHub  
You've taken your first step into a larger world, @jongkwangmail.

Completed  
Set up a personal account

Step 2:  
Choose your plan

Step 3:  
Tailor your experience

Choose your personal plan

Unlimited public repositories for free.

Unlimited private repositories for \$7/month.

Don't worry, you can cancel or upgrade at any time.

Help me set up an organization next  
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations.](#)

[Continue](#)

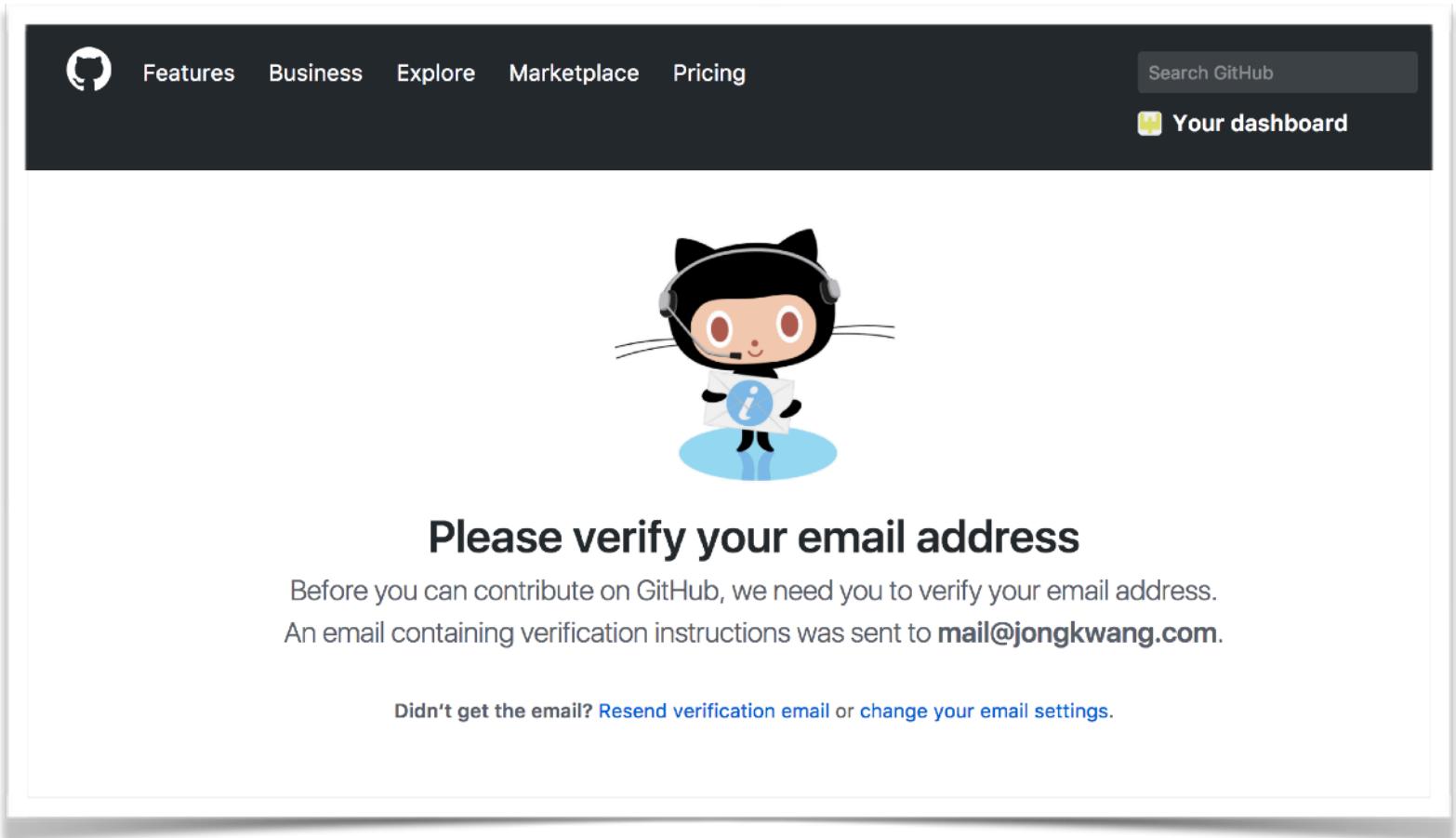
Both plans include:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

# Join Github

## ▶ Verify email address

- 확인 전까지 Repository 생성이 안된다



The screenshot shows the GitHub login page. At the top, there is a dark header with the GitHub logo, navigation links for Features, Business, Explore, Marketplace, and Pricing, a search bar labeled "Search GitHub", and a link to "Your dashboard". The main content area features the GitHub cat character wearing a headset and holding a blue envelope with an "i" on it. Below the character, the text "Please verify your email address" is displayed in bold. A message follows: "Before you can contribute on GitHub, we need you to verify your email address. An email containing verification instructions was sent to [mail@jongkwang.com](mailto:mail@jongkwang.com)". At the bottom, there is a link for users who didn't receive the email: "Didn't get the email? [Resend verification email](#) or [change your email settings](#)".

Git

Profile

# Profile

## ▶ Profile

- Repository
- Contributions
- Activity

The screenshot shows a GitHub user profile for 'jongkwang'. At the top, there's a navigation bar with links for Overview, Repositories (11), Stars (22), Followers (28), and Following (18). Below the navigation is a section titled 'Popular repositories' featuring six repository cards:

- Korea-Sencha-User-Group**: Forked from 'Korea Sencha User Group'. 1 star.
- MyGitHubTest**: Forked from 'HTML'.
- 12**: Forked from '1step6thswmaestro/12'. 12 forks. Author: 김종광. Languages: Visual Basic.
- tensorflow**: Forked from 'tensorflow/tensorflow'. Computation using data flow graphs for scalable machine learning. Languages: C++, Python.
- cf-sample\_spring-music**: Forked from 'bleix/cf-sample\_spring-music'. A sample application for using database services on Cloud Foundry with Spring Framework. Languages: Java.
- test-travis1**: Languages: Java.

On the left side of the profile page, there's a sidebar with links for 'Follow', 'Block or report user', and 'Organizations'. It also lists the user's website (<http://jongkwang.com>), location (Korea), email (kim@jongkwang.com), and another website (<http://jongkwang.com>). Below the sidebar is a section titled 'Contribution activity' with a heatmap showing contributions per month and day for the last year. The heatmap uses a color scale from light green ('Less') to dark green ('More'). The heatmap shows a high density of contributions in November and December, with a smaller cluster in July.

# Profile

## ▶ Profile

- Repository
- Contributions
- Activity



TensorFlow 개발자의  
Profile 을 찾아보세요  
(5분)

**jongkwang (JongKwang)**

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

Search GitHub Pull requests Issues Marketplace Gist

Overview Repositories 11 Stars 22 Followers 28 Following 18

Popular repositories

- MyGitHubTest**  
HTML
- tensorflow**  
Forked from tensorflow/tensorflow  
Computation using data flow graphs for scalable machine learning  
C++ 1
- test-travis1**  
Java

1,196 contributions in the last year

Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul

Mon Wed Fri

Learn how we count contributions.

Contribution activity

August 2017

Jump to ▾ 2017

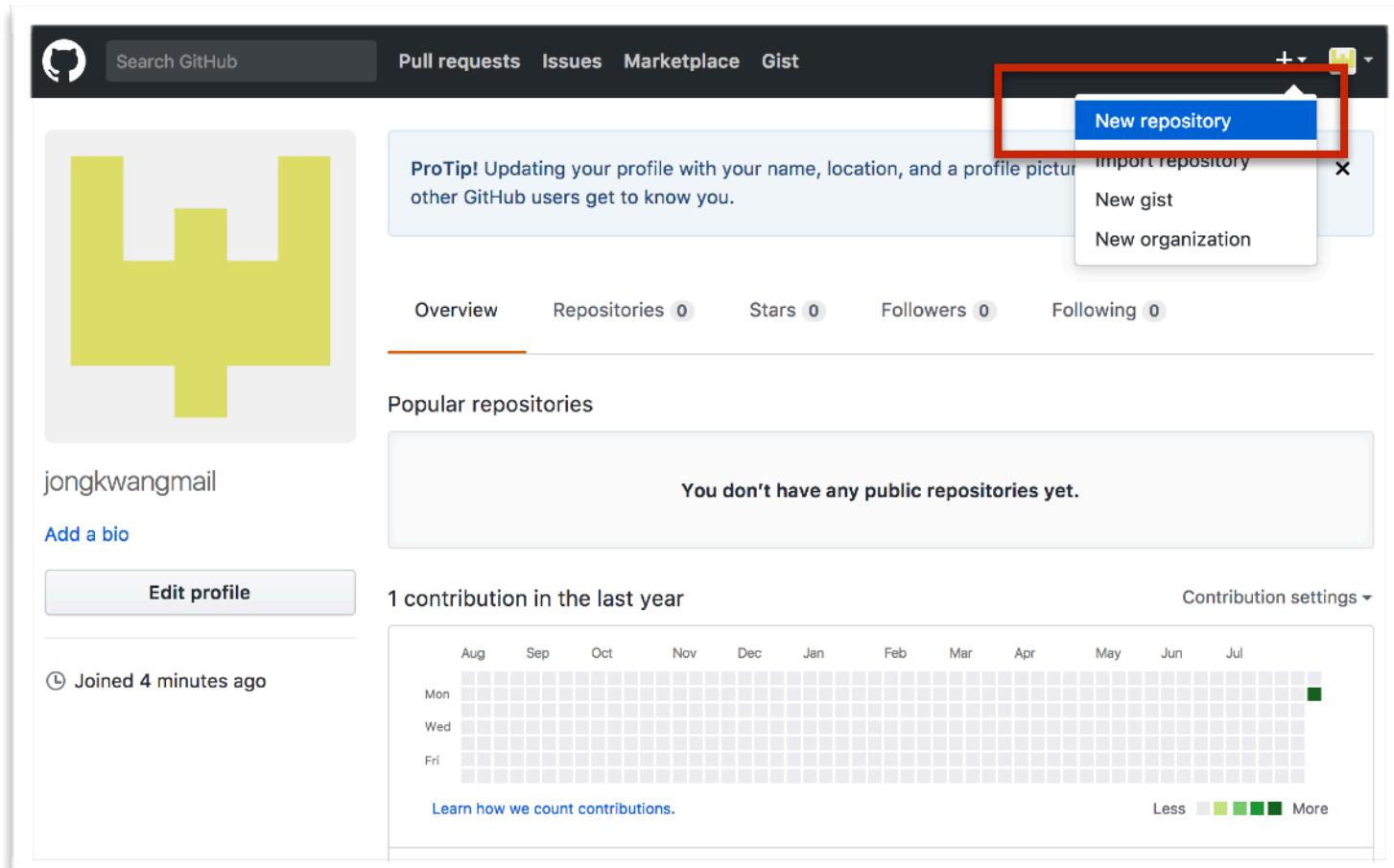
KOREA OPEN SOURCE SOFTWARE DEVELOPERS LAB

Git

New Repository

# New Repository

- ▶ 새로운 Repository 생성
  - 확인 전까지 Repository 생성이 안된다



# New Repository

Create a new repository

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

Owner: jongkwang

Repository name: test-github

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

Description (optional):

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

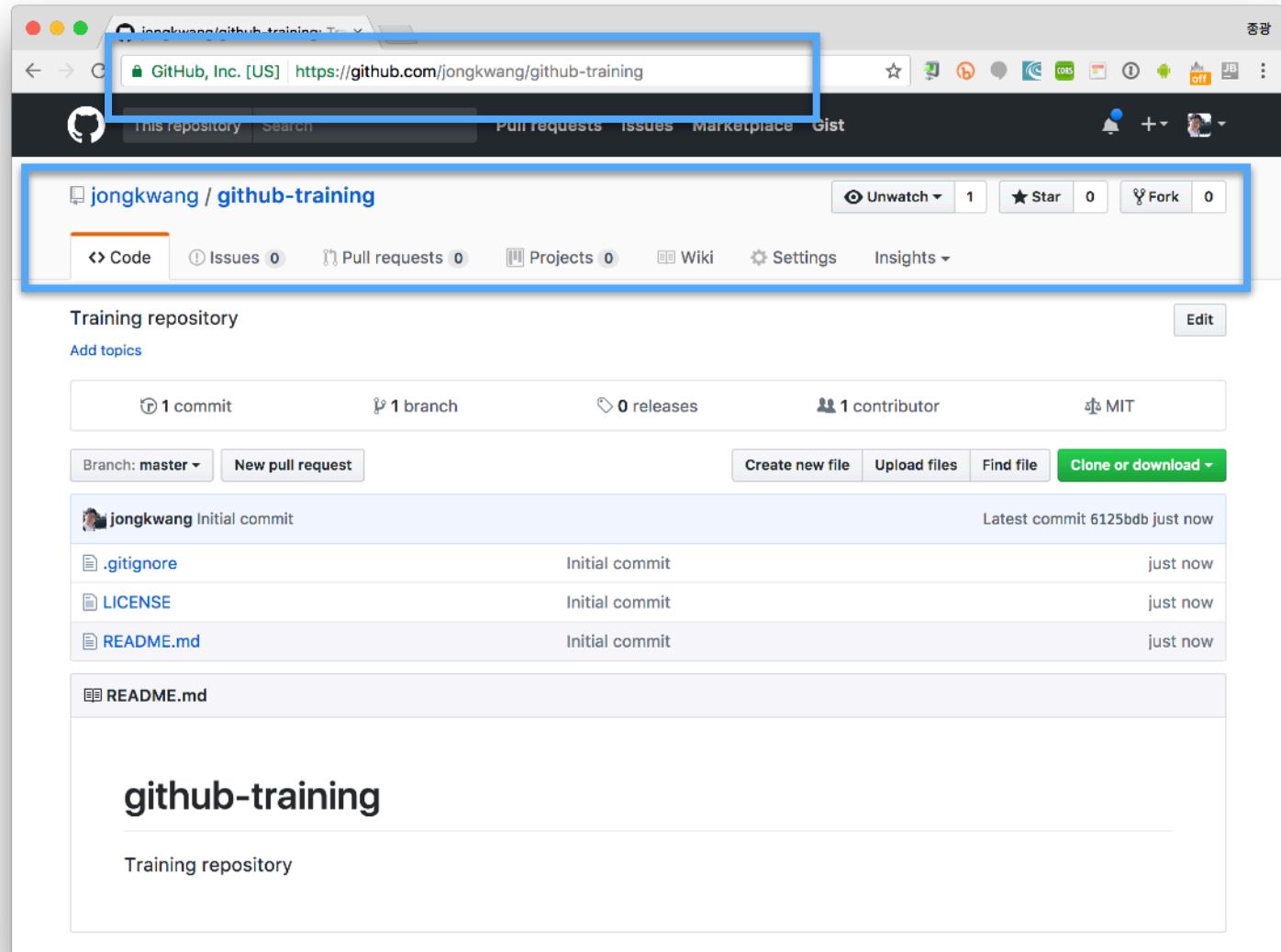
Private: You choose who can see and commit to this repository.

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**

# New Repository



The screenshot shows a GitHub repository page for 'jongkwang / github-training'. A blue box highlights the browser's address bar showing the URL 'GitHub, Inc. [US] https://github.com/jongkwang/github-training'. Another blue box highlights the main header area, including the repository name 'jongkwang / github-training', the navigation tabs ('Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', 'Insights'), and the statistics ('1 commit', '1 branch', '0 releases', '1 contributor', 'MIT').

**Repository Overview:**

- Code**: 1 commit
- Issues**: 0
- Pull requests**: 0
- Projects**: 0
- Wiki**
- Settings**
- Insights**

**Commit History:**

- jongkwang Initial commit (Latest commit 6125bdb just now)
- .gitignore (Initial commit, just now)
- LICENSE (Initial commit, just now)
- README.md (Initial commit, just now)

**File Content Preview:**

```
github-training
```

**Description:** Training repository

Git - 그림설명

Git Install

# Git Install

## ▶ Windows

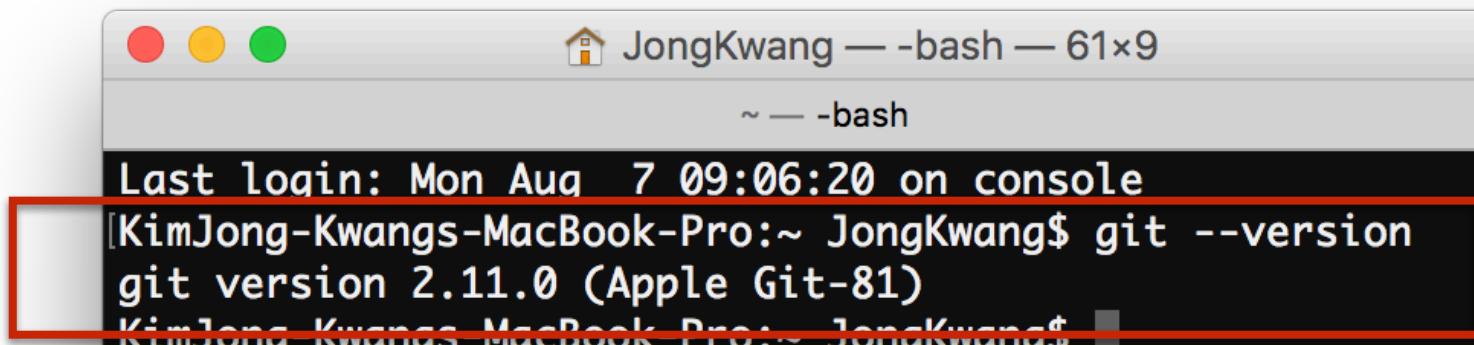
- <http://git-scm.com/download/win>

## ▶ macOS

- Terminal에서 “git” 실행
  - Mavericks(10.9) 부터 없으면 설치된다.
- 직접 설치 : <http://git-scm.com/download/mac>

## ▶ 설치 확인

- Terminal : git --version



The screenshot shows a macOS terminal window titled "JongKwang — -bash — 61x9". The command "git --version" was run, and the output is displayed in white text on a black background. A red box highlights the command and its output. The output shows the last login information and the git version installed on the system.

```
Last login: Mon Aug  7 09:06:20 on console
[KimJong-Kwangs-MacBook-Pro:~ JongKwang$ git --version
git version 2.11.0 (Apple Git-81)
KimJong Kwangs MacBook Pro:~ JongKwang$ ]
```

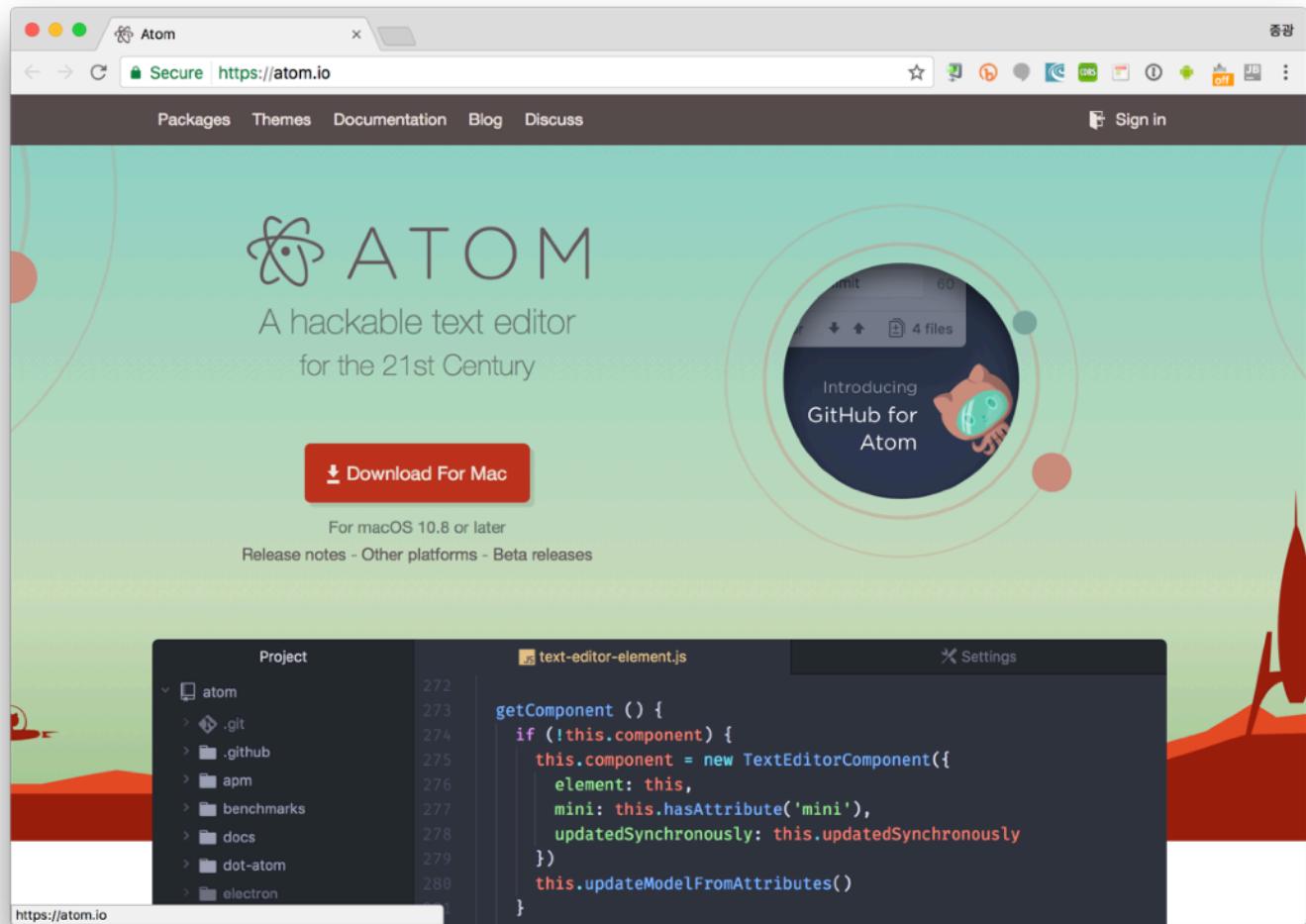
Git - 그림설명

ATOM

Text editor

# ATOM Install

› <https://atom.io>



Git - 그림설명

SourceTree

# SourceTree Install

› <https://www.sourcetreeapp.com>

The screenshot shows the SourceTree website on the left and its Mac OS X application interface on the right.

**SourceTree Website (Left):**

- Header: SourceTree | Free Git GUI for Mac
- Address bar: Secure | https://www.sourcetreeapp.com
- Logo: SourceTree
- Text: Simplicity and power in a beautiful Git GUI
- Buttons: Download for Mac OS X, Also available for Windows

**SourceTree Application (Right):**

- Title bar: sourcetree-website (Git)
- Toolbar: Commit, Pull, Push, Branch, Merge, Shelf
- Menu bar: Show in Finder, Terminal, Settings
- Central area: A Git commit history graph showing branches and merges. The commits listed are:

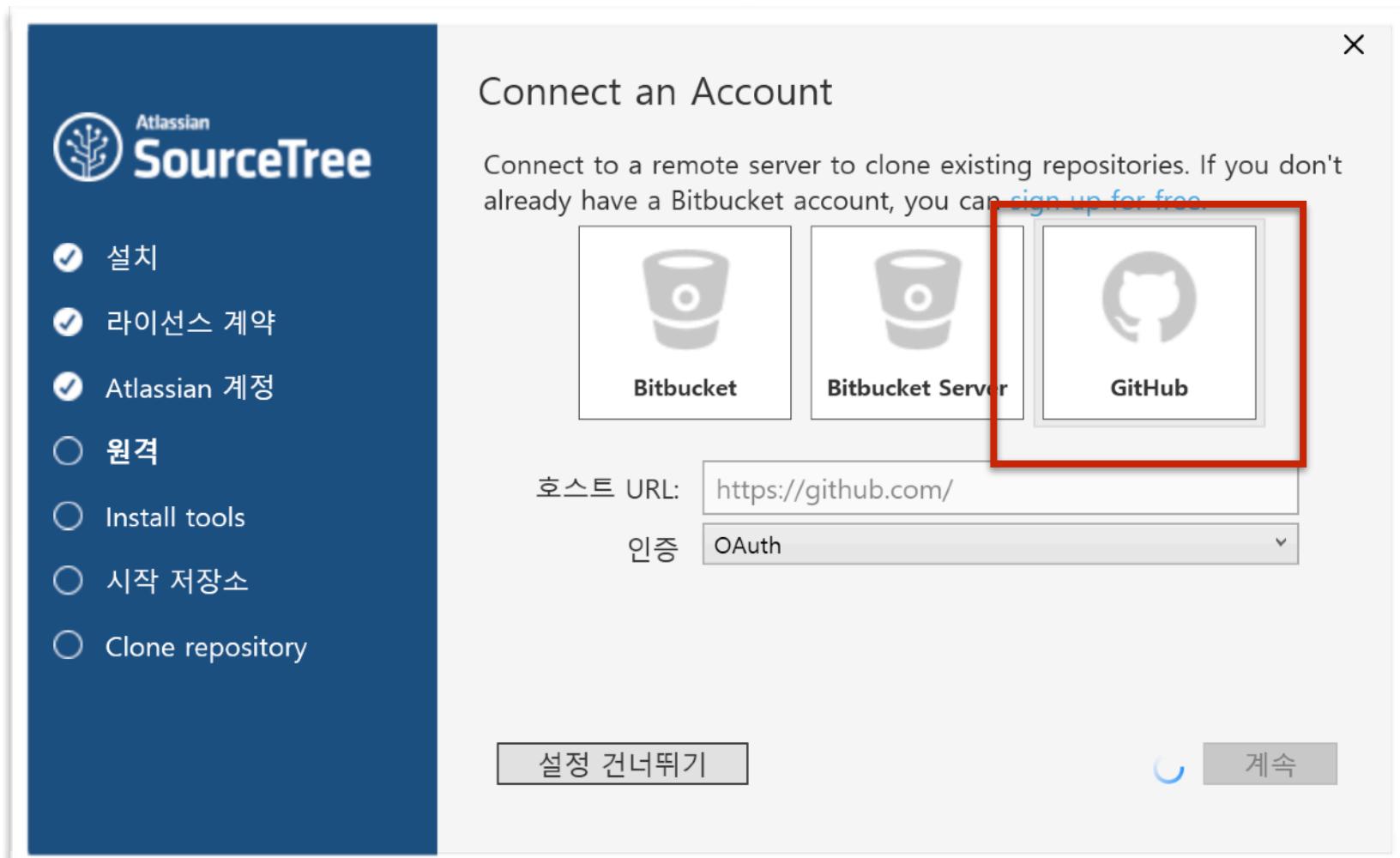
Commit	Author	Description	Date
b7358c7	Rahul Chhab...	[r] master [r] origin/master [r] origin/HEAD Removing ol...	Mar 3, 2016, 11...
bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:2...
dfe975d	Tyler Tadej...	[r] origin/update-google-verification Update google verificati...	Feb 11, 2016, 2:2...
3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 15, 2016, 10:...
dba47f9	Tyler Tadej...	Add gittignore	Feb 11, 2016, 1:3...
ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
246c4ff	Joel Unger...	[r] origin/hero_images [r] hero_images Used TinyPng to c...	Feb 11, 2016, 3:3...
9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
85367bb	Patrick Tho...	[r] origin/bug/date-https fixed date and https errors	Jan 7, 2016, 12:2...
49fb557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
384e6d5	Rahul Chhab...	[r] origin/search-console-access search console google ver...	Feb 3, 2016, 2:09...
6fa47e9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
0cdfe96	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
a723bc2	Mike Minns...	Merge branch 'au2ge*	Nov 23, 2015, 1:5...
65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
500a892	Michael Min...	Merged in au2ge* (pull request #11)	Nov 23, 2015, 1:0...

**A free Git client for Windows and Mac**

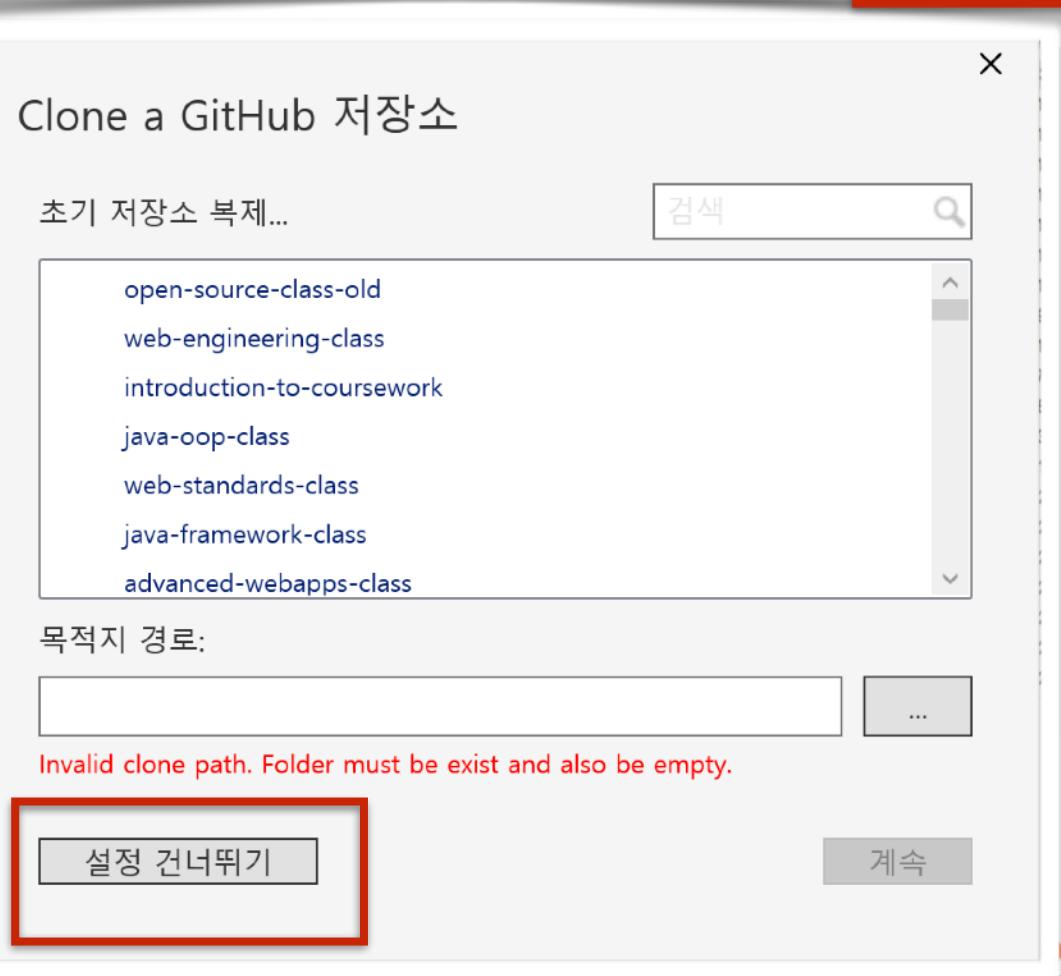
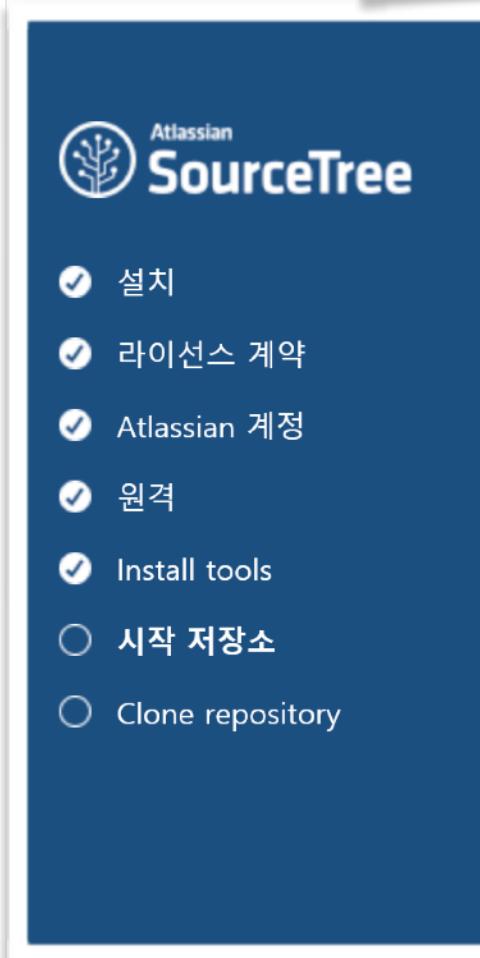
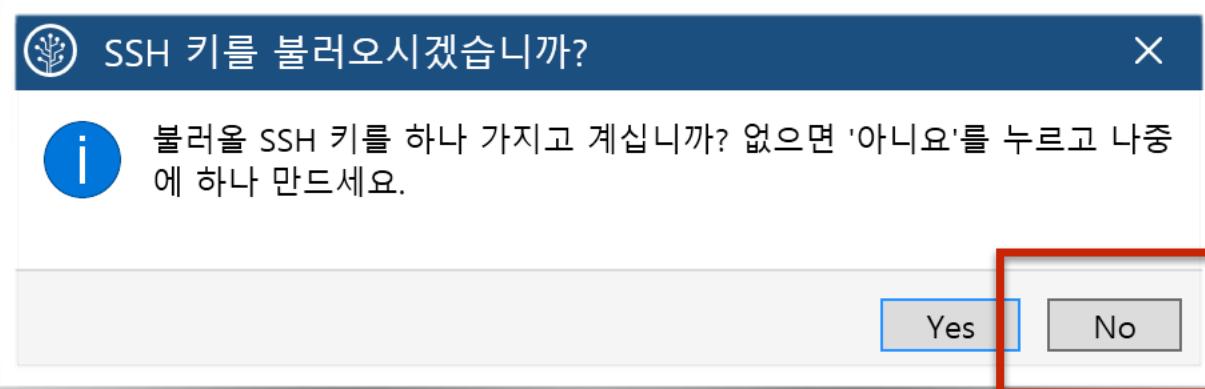
SourceTree simplifies how you interact with your Git repositories so you can focus on coding. Visualize and manage your repositories through SourceTree's simple Git GUI.

# SourceTree Install

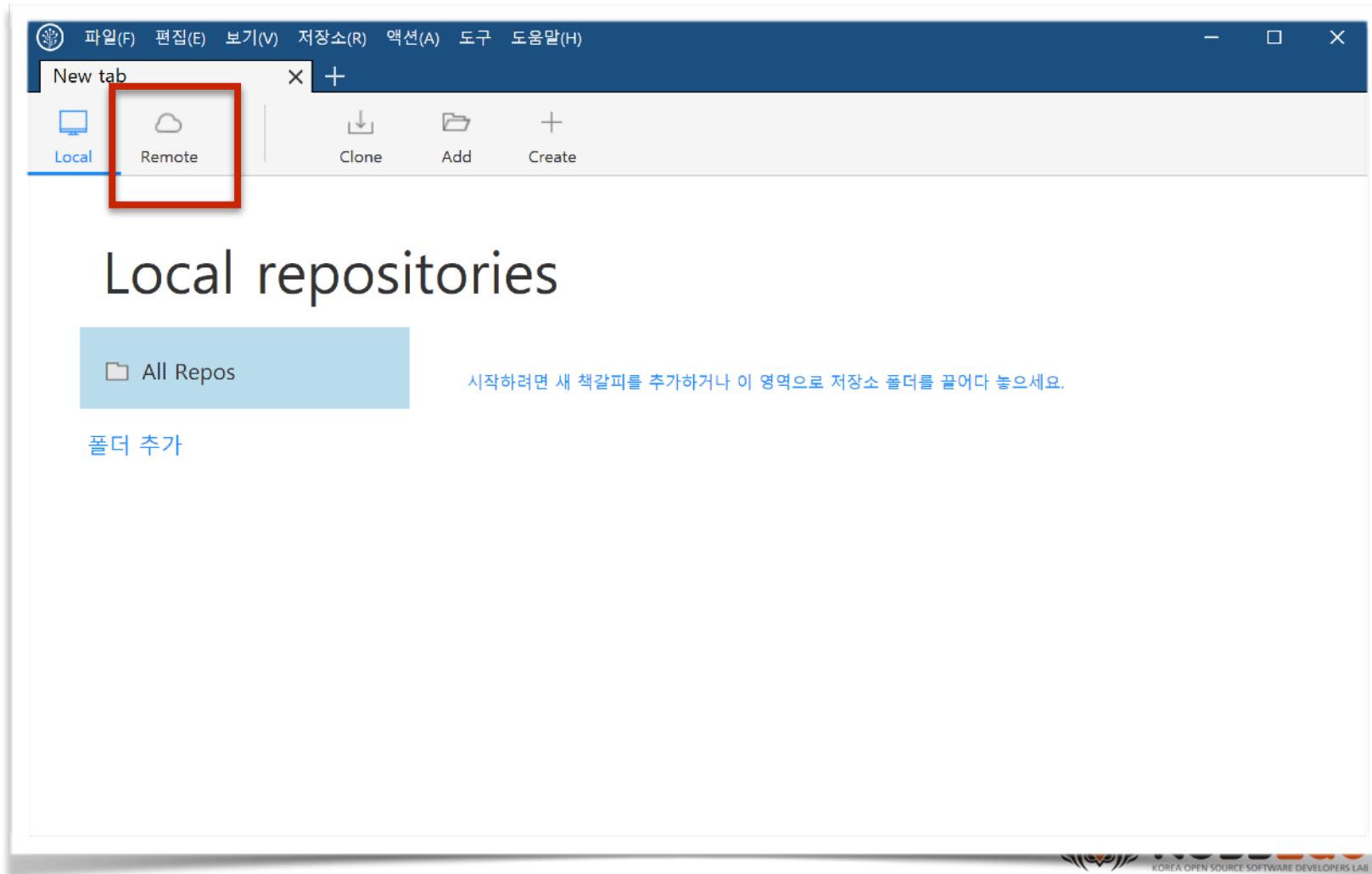
## ▶ Github 선택



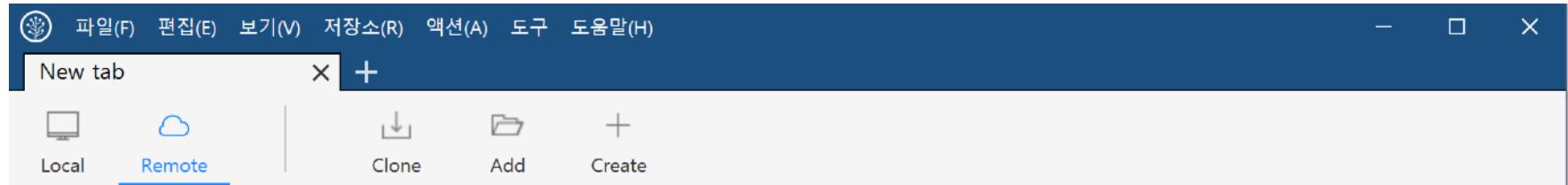
# SourceTree



# SourceTree Install



# SourceTree Install



## Remote repositories

A screenshot of the 'Remote repositories' screen in SourceTree. On the left, there's a sidebar with a user profile for 'jongkwang' (GitHub) and a '+ 계정 추가...' (Add account...) button. In the center, a search bar contains the text 'training'. Below the search bar, a list of repositories is shown. The first repository is 'github-training' (GitHub), which has a blue circular icon with '&lt;/&gt;'. To the right of this repository is a 'Clone' button, which is highlighted with a red rectangular box. Above the 'Clone' button is a '새로고침' (Refresh) button with a circular arrow icon. At the bottom left of the main area, there's a '계정 변경' (Change account) link. The background is white.

# SourceTree Install

The screenshot shows the SourceTree application interface. At the top, there's a dark blue header bar with a tree icon and Korean menu items: 파일(F), 편집(E), 보기(V), 저장소(R), 액션(A), 도구, and 도움말(H). Below the header is a toolbar with a 'New tab' button, a close button (X), and a plus sign (+) for creating new tabs. Underneath the toolbar are three buttons: 'Local' (local disk icon), 'Remote' (cloud icon), and 'Clone' (down arrow icon). The 'Clone' button is underlined, indicating it is selected. To its right are 'Add' (folder icon) and 'Create' (plus sign icon) buttons.

**Clone**

Cloning is even easier if you set up a [remote account](#)

Browse

저장소 종류: ? Git 저장소입니다

Browse

Local Folder:

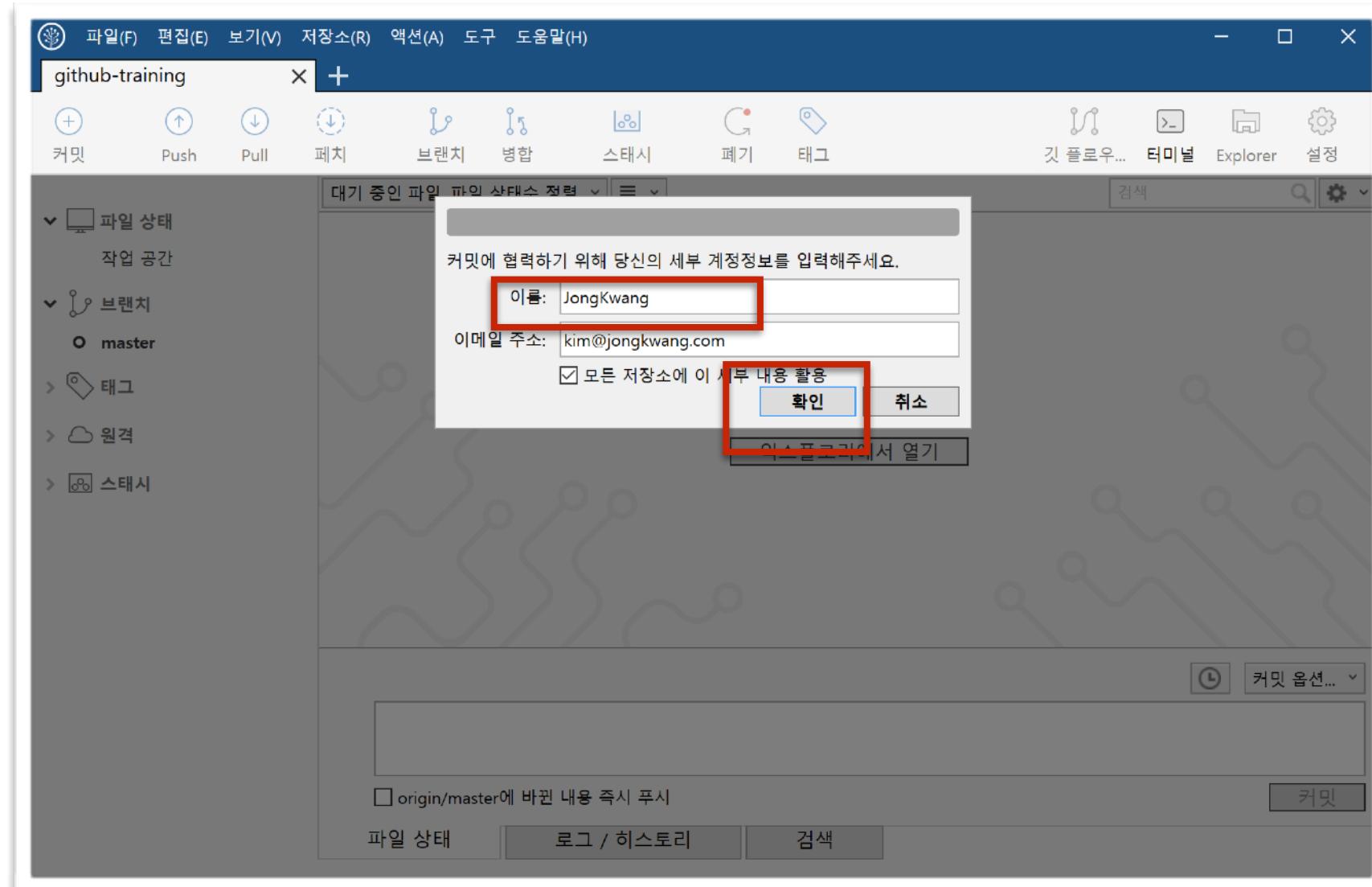
▼

(▼) 고급 옵션

클론

KOREA OPEN SOURCE SOFTWARE DEVELOPERS LAB

# SourceTree Install

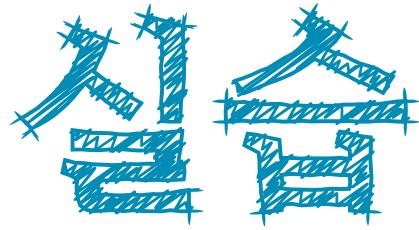


# SourceTree Install

## › 여기까지 오신분은

- 탐색기에서 해당 폴더를 살펴 봅니다
- SourceTree 를 살펴봅니다

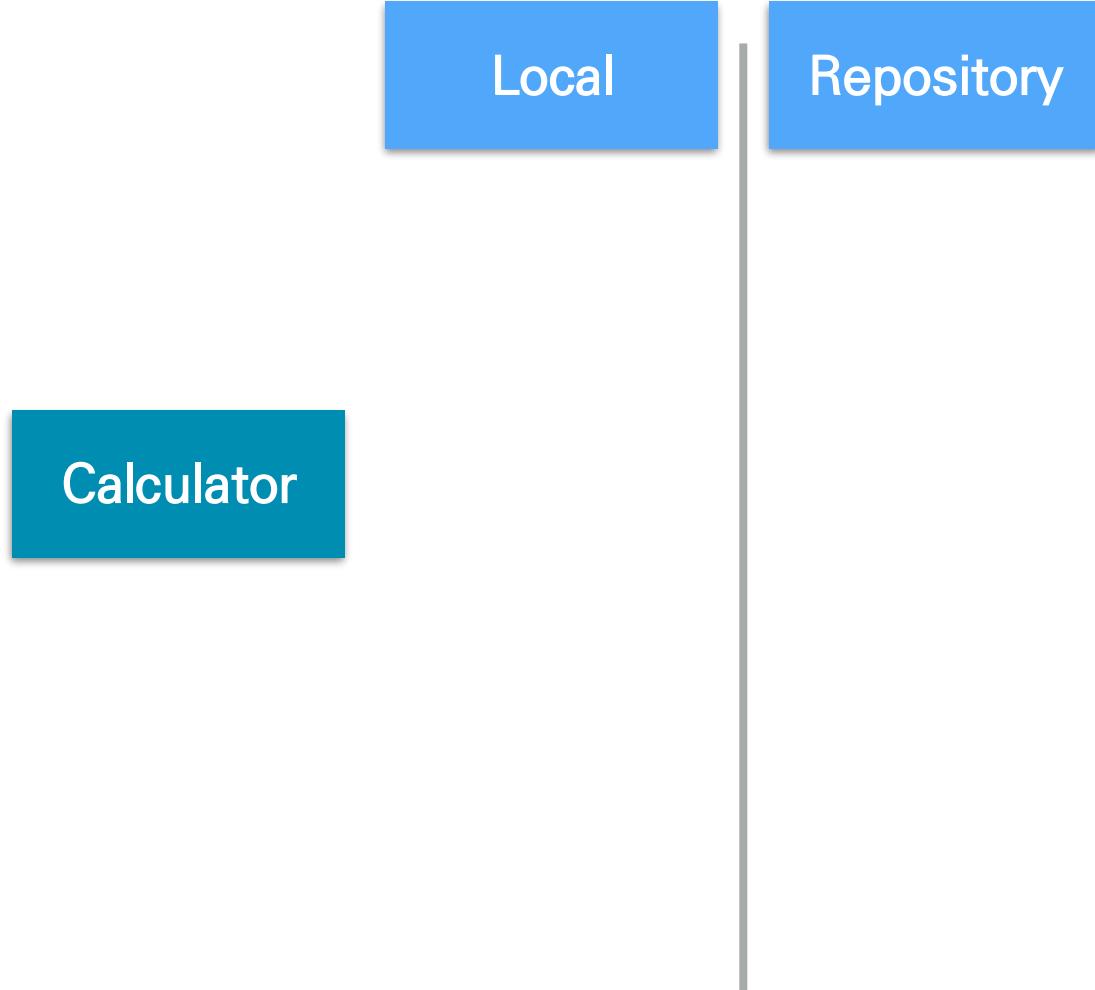
# Git - 그림설명



# Preview

# Git - 그림 설명

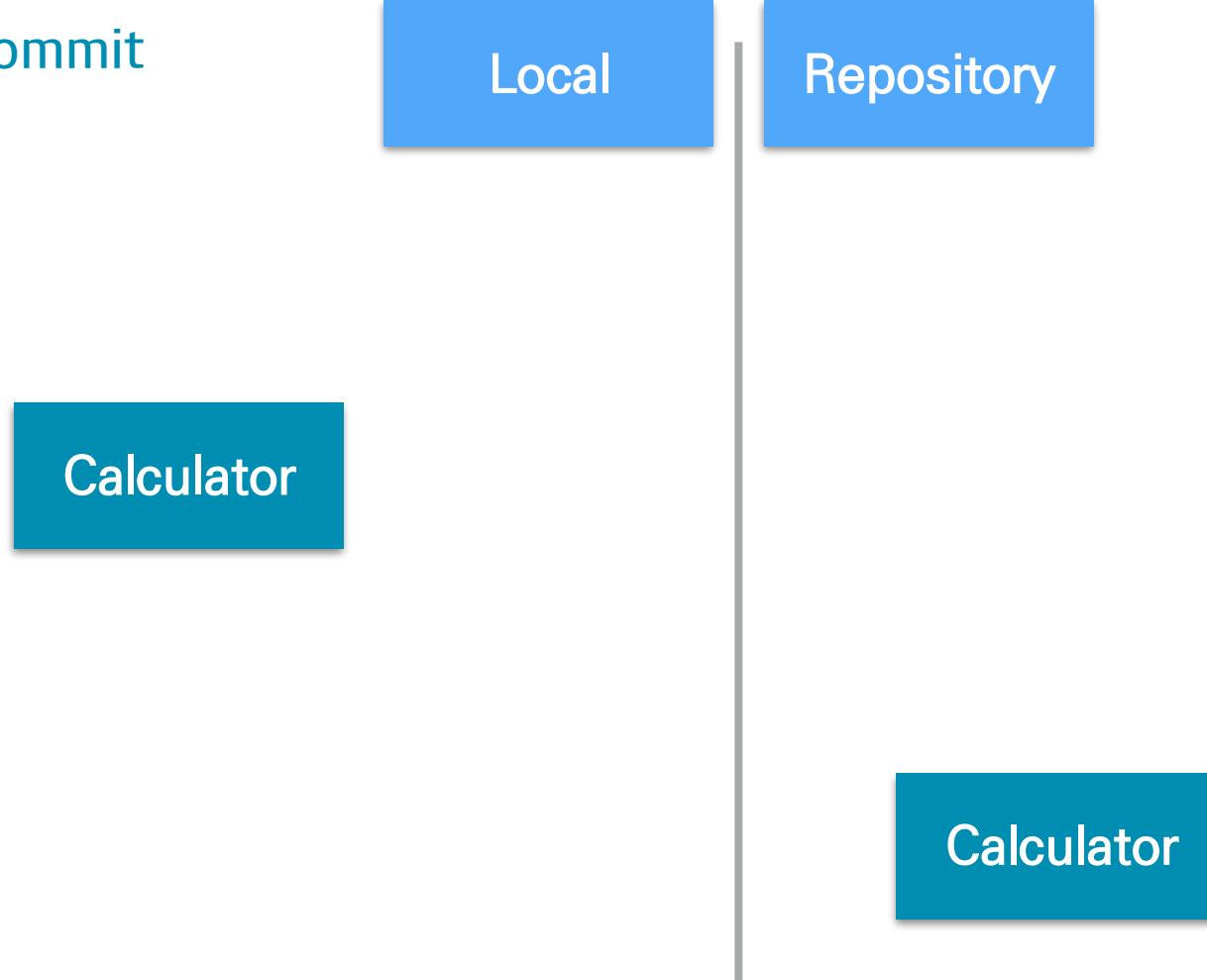
- ▶ 계산기 클래스 생성



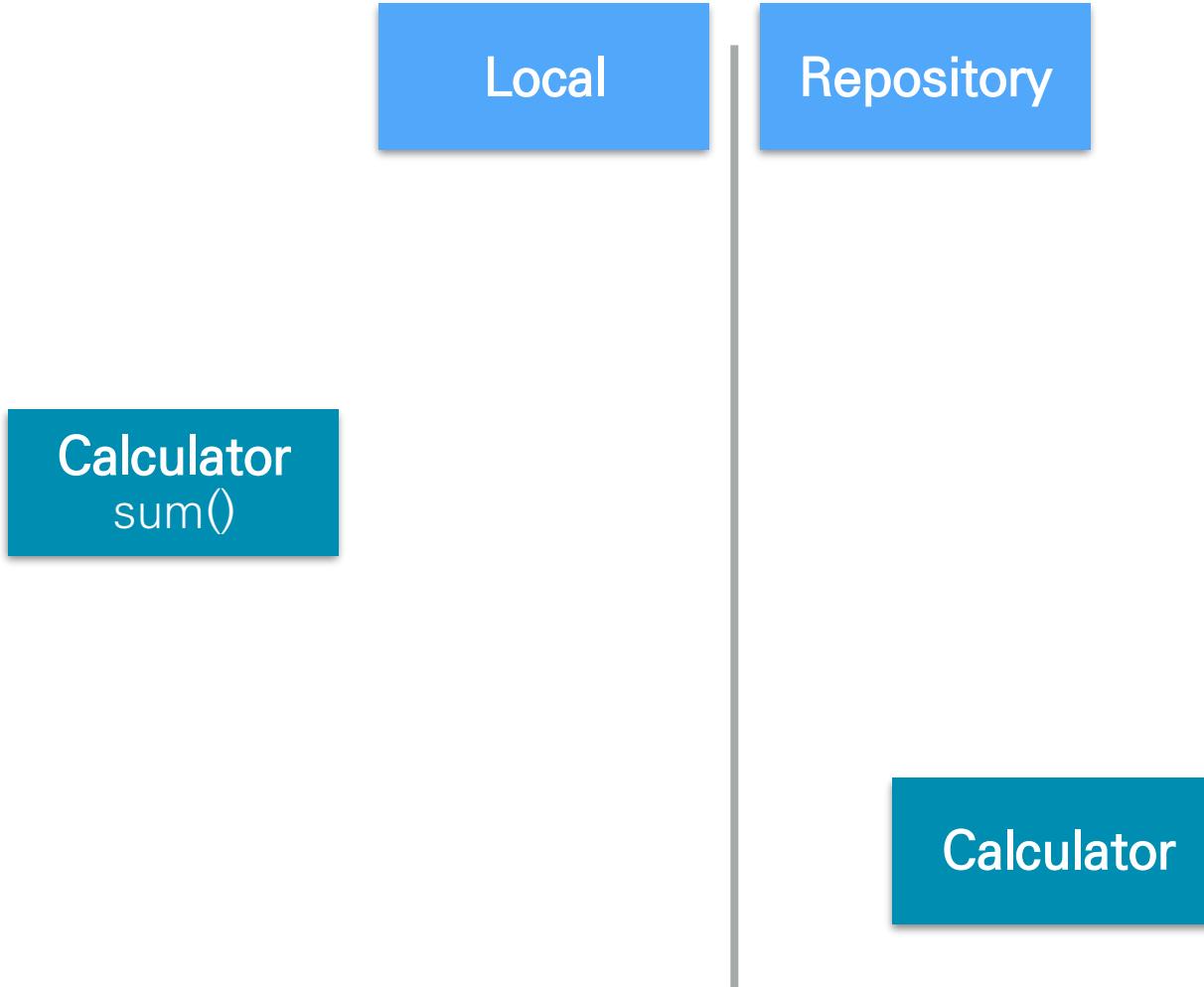
# Git – 그림설명

## ▶ Repository 에 저장

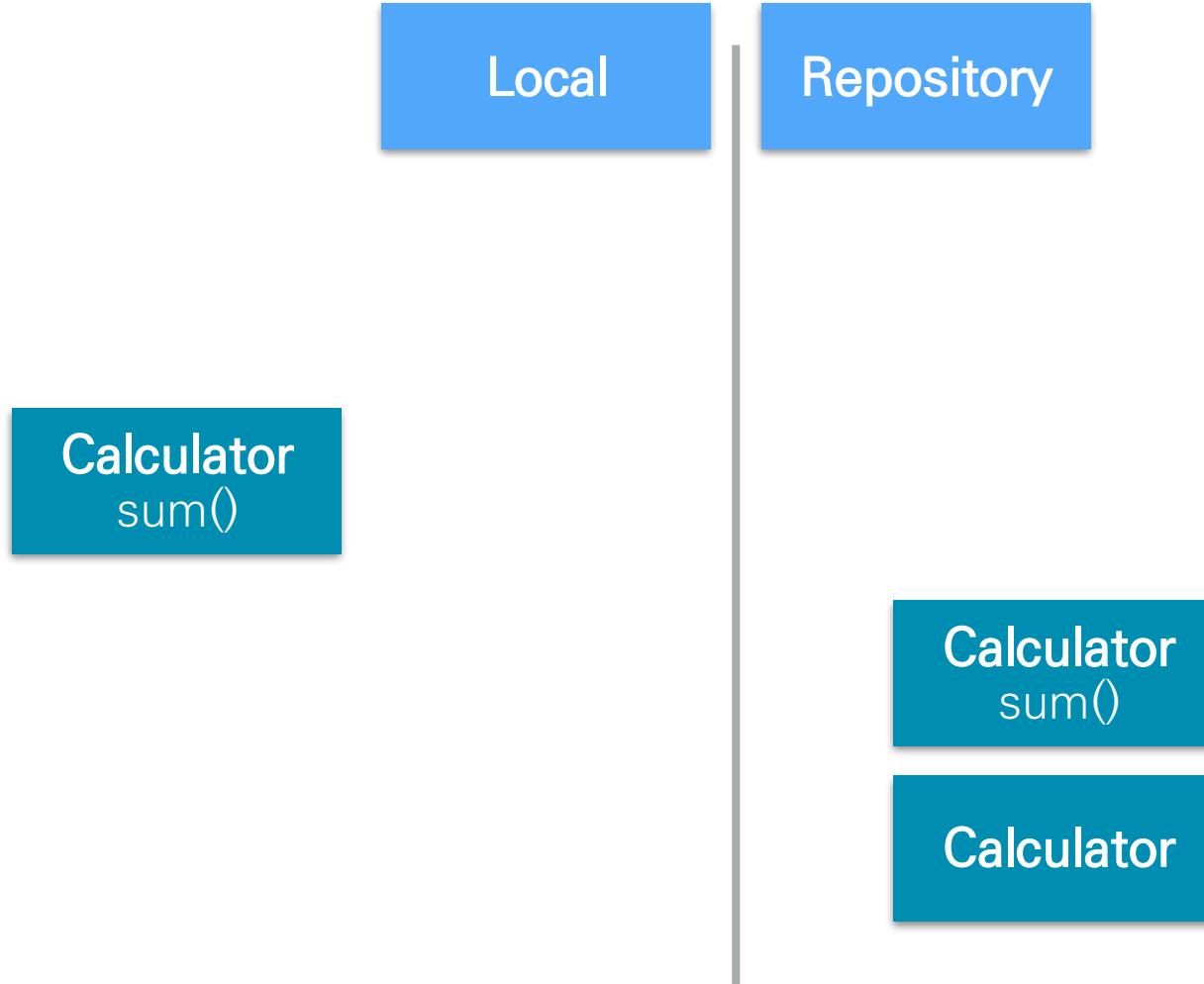
- Commit



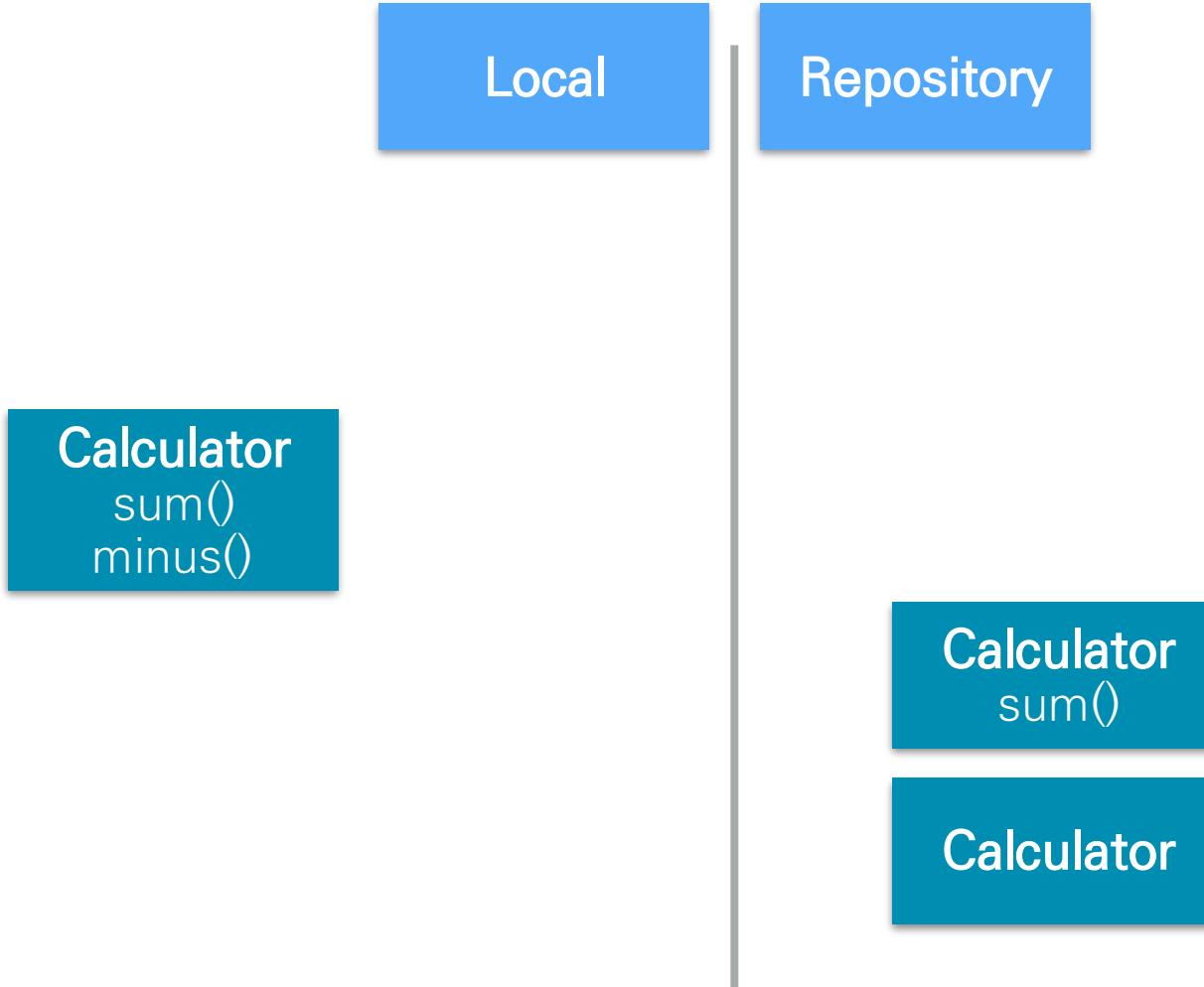
# Git - 그림 설명



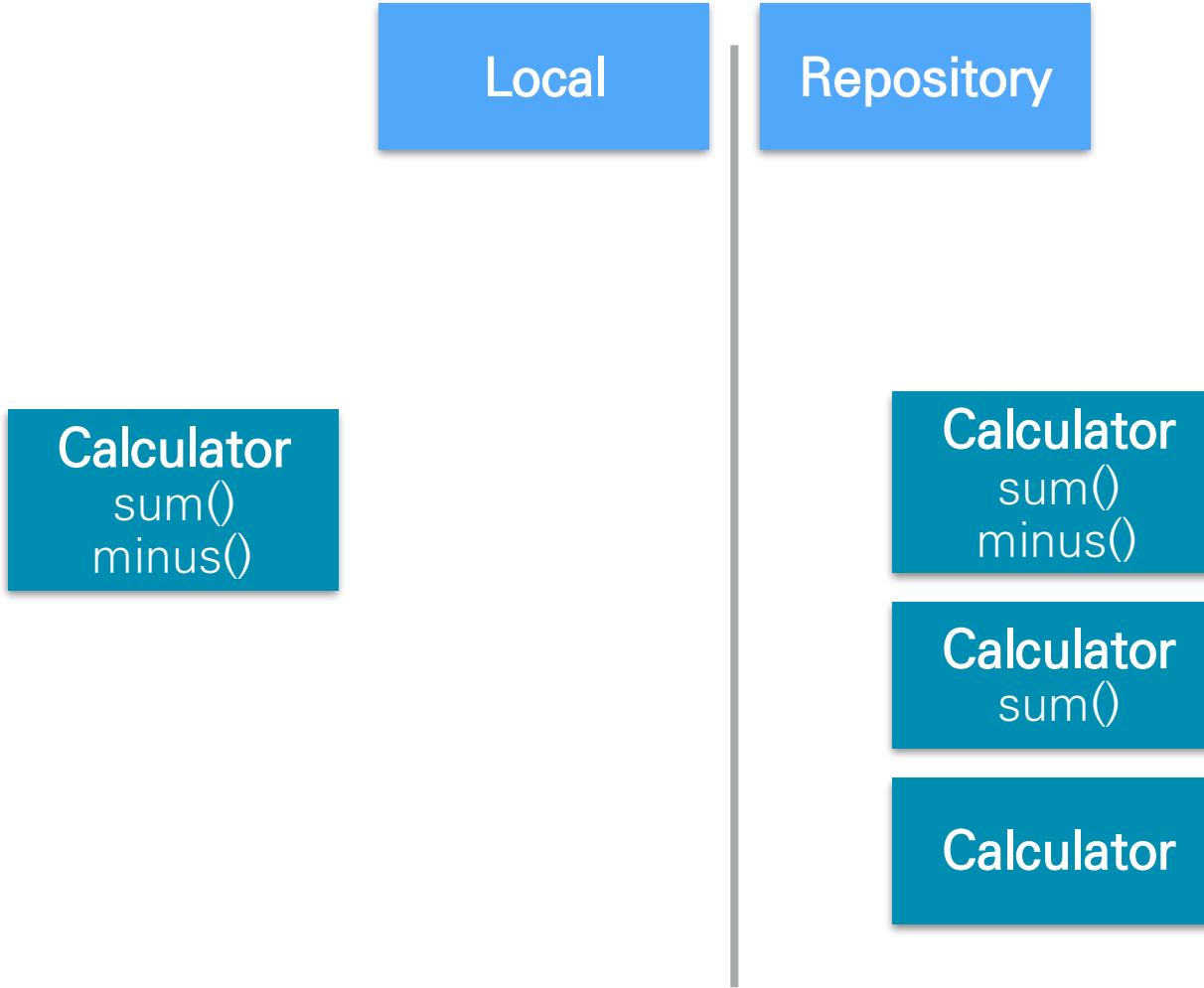
# Git - 그림 설명



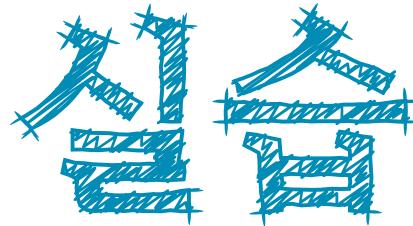
# Git – 그림 설명



# Git – 그림 설명



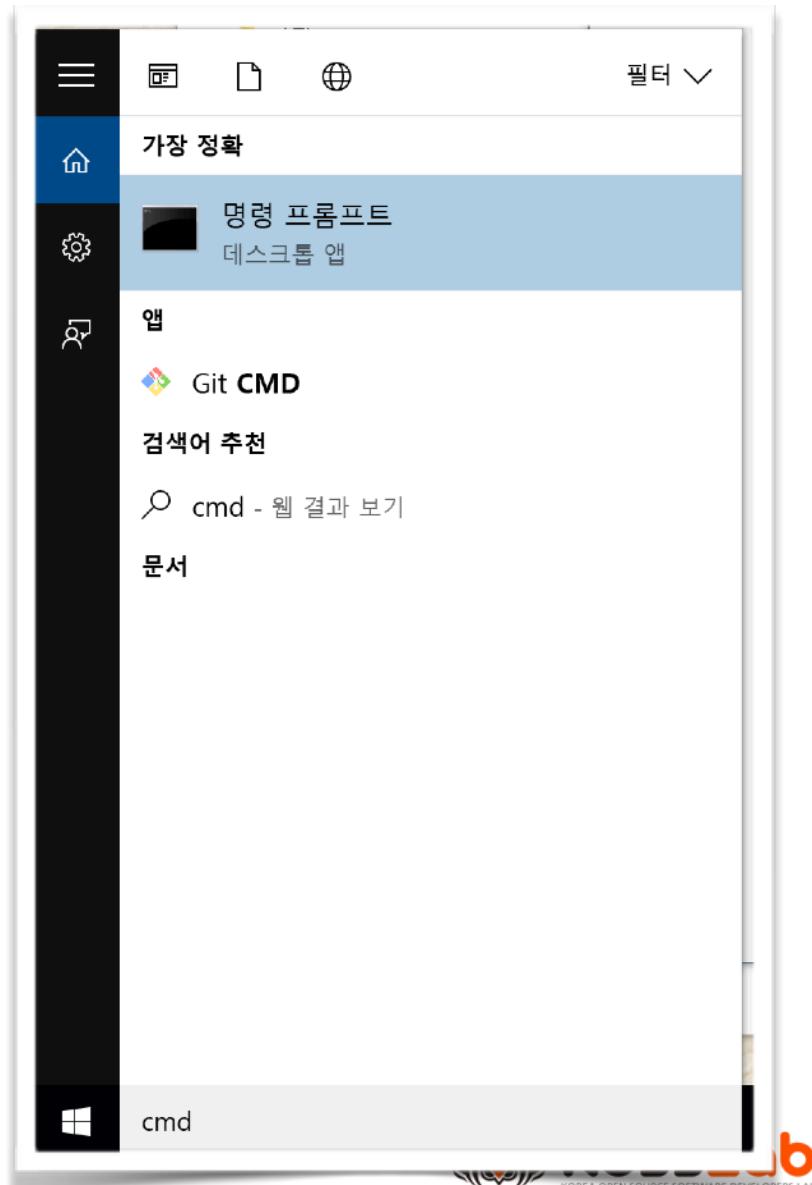
설습 – with SourceTree



with SourceTree

# 실습 – with SourceTree

## ▶ cmd 실행





## 명령 프롬프트

Microsoft Windows [Version 10.0.10240]  
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\JongKwang>cd C:\Users\JongKwang\Documents\test-github

C:\Users\JongKwang\Documents\test-github>dir

C 드라이브의 볼륨: BOOTCAMP  
볼륨 일련 번호: FCDE-4CF5

C:\Users\JongKwang\Documents\test-github 디렉터리

2017-08-07	오후 11:31	<DIR>	.
2017-08-07	오후 11:31	<DIR>	..
2017-08-07	오후 11:31	294 .gitignore	
2017-08-07	오후 11:31	1,087 LICENSE	
2017-08-07	오후 11:31	13 README.md	
	3개 파일	1,394 바이트	
	2개 디렉터리	32,227,774,464 바이트	남음

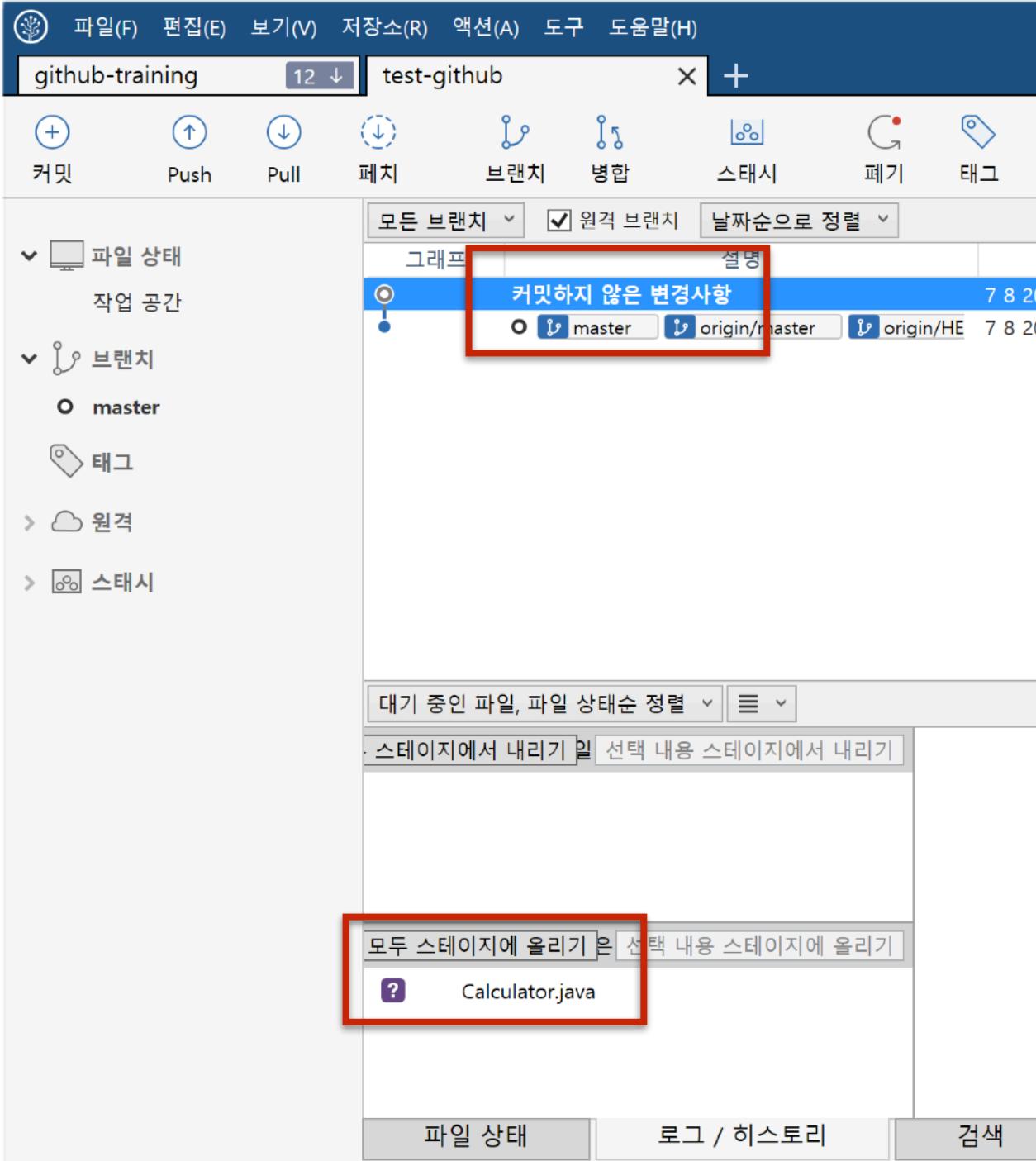
C:\Users\JongKwang\Documents\test-github>atom .

# 실습 - with SourceTree

- ▶ Calculator.java 생성
- ▶ Calculator 코딩

The screenshot shows the Atom IDE interface. The title bar reads "Calculator.java — C:\Users\JongKwang\Documents\test-github — Atom". The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. On the left, the Project pane shows a file tree with a red box around the "Calculator.java" file under the "test-github" directory. The main editor pane displays the following Java code:

```
1 package com.mycompany.test;  
2  
3 public class Calculator {  
4  
5     public static void main(String[] args) {  
6         }  
7     }  
8 }
```

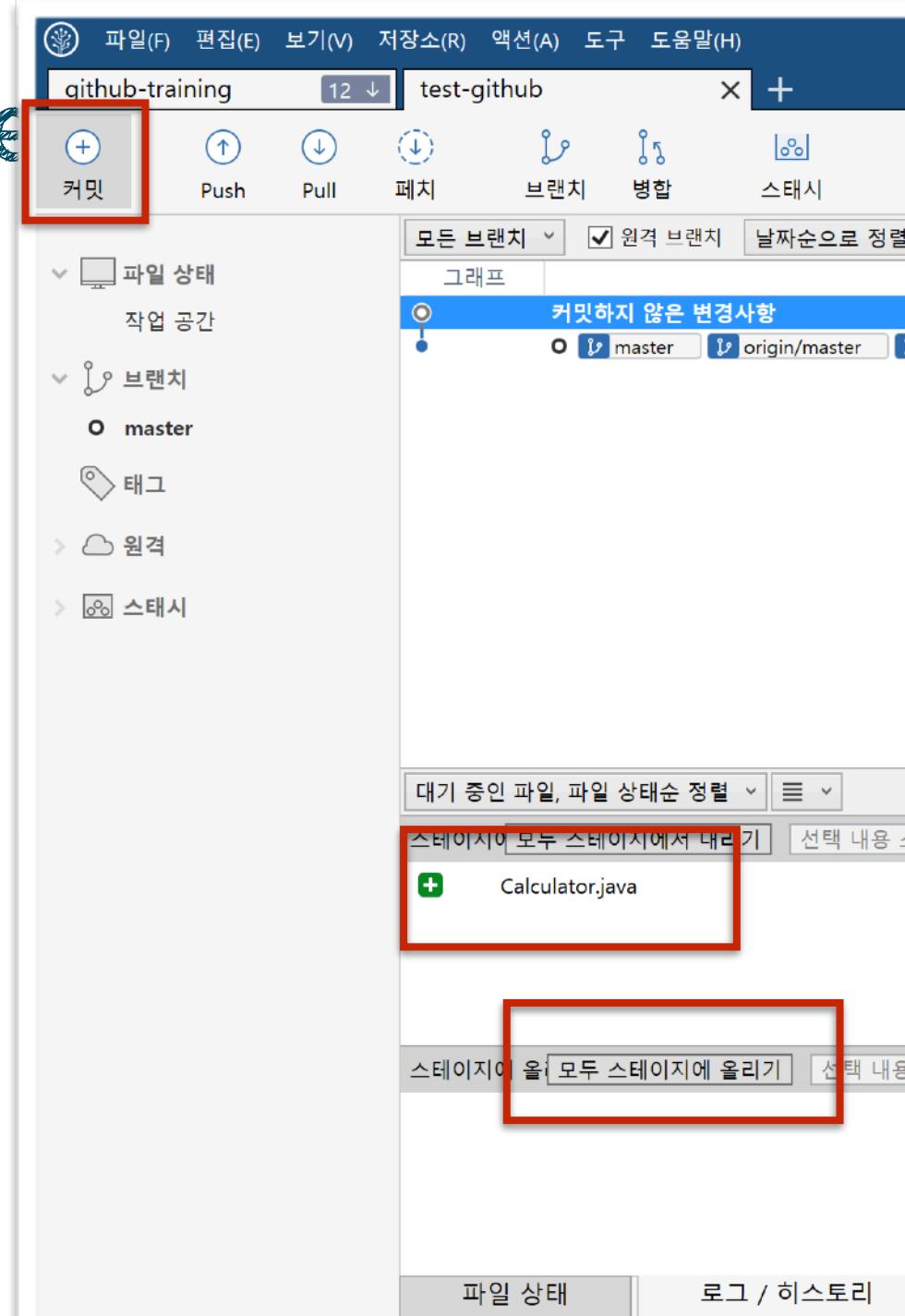


# 실습 - with

- ▶ Unstage에 파일이
- ▶ 커밋하지 않은 변경

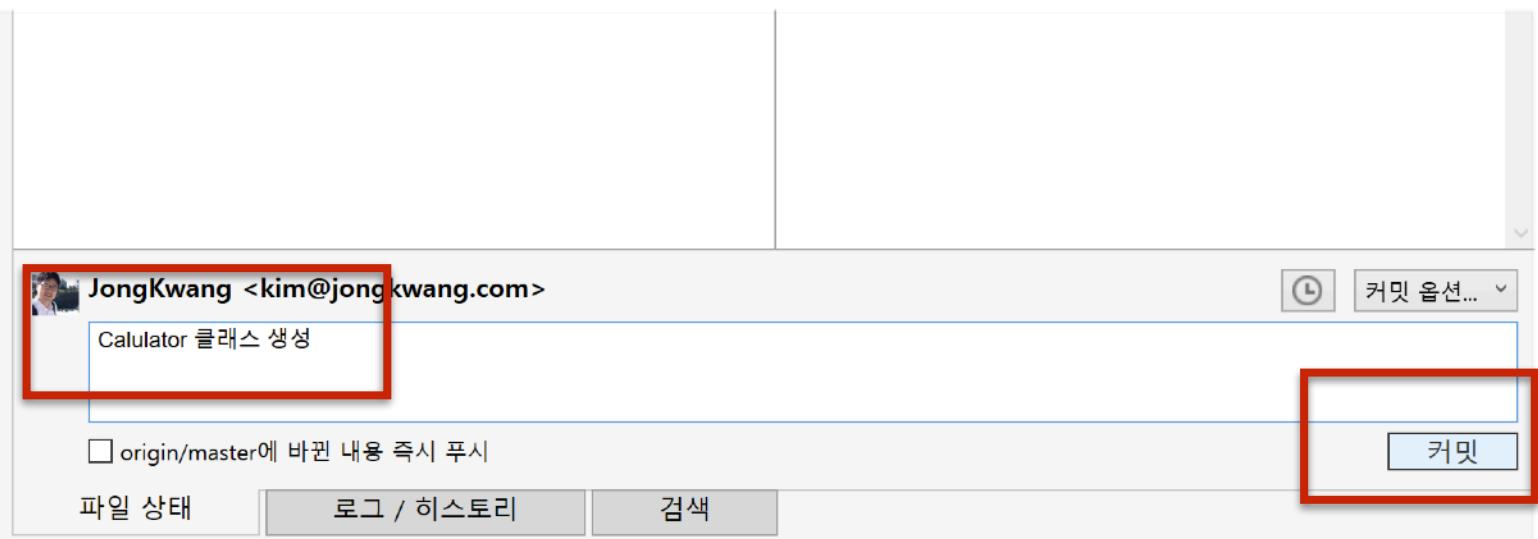
# 실습 - with Source

- ▶ 모두 스테이지에 올리고
- ▶ 커밋



# 실습 - with SourceTree

- ▶ 커밋로그 작성
- ▶ 커밋



파일(F) 편집(E) 보기(V) 저장소(R) 액션(A) 도구 도움말(H)

github-training 12 test-github X +

커밋 Push Pull 폐지 브랜치 병합 스태시 폐기 태그 깃 플로우... 터미널 Explorer 설정

파일 상태 작업 공간 브랜치 master 태그 원격 스태시

모든 브랜치 원격 브랜치 날짜순으로 정렬

그래프 설명 날짜 작성자 커밋

	master	Calulator 클래스 생성	날짜	작성자	커밋
origin/master	origin/HEAD	Initial commit	7 8 2017 23:30	JongKwang <kim@...>	bdc66b8

파일 상태순 정렬 김색

커밋: 09dd07fa6f36d1294ffb4fe86df0a4d6d0161908 [09dd07f]  
상위 할목: bdc66b83e6  
작성자: JongKwang <kim@jongkwang.com>  
날짜: 2017년 8월 7일 월요일 오후 11:38:26  
커밋한 사람: JongKwang

Calulator 클래스 생성

Calculator.java

파일 내용 코드 봉지 되돌리기

```
+ package com.mycompany.test;
+
+ public class Calculator {
+
+     public static void main(String[] args)
+
+ }
```

파일 상태 로그 / 히스토리 검색

KOREA OPEN SOURCE SOFTWARE DEVELOPERS LAB

# 실습 – with SourceTree

▶ 하지만  
Github 에는 없다

The screenshot shows a GitHub repository page for 'test-github'. At the top, it says 'jonkwang / test-github Private'. Below that is a navigation bar with 'Code' (highlighted in orange), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', and 'Settings'. A message 'No description, website, or topics provided.' is displayed, along with a link 'Add topics'. Below this are summary statistics: '1 commit', '1 branch', and '0 releases'. A dropdown menu 'Branch: master ▾' is shown, along with 'New pull request' and 'Create new'. The main content area shows a commit by 'jongkwang' with the message 'Initial commit'. It lists files: '.gitignore', 'LICENSE', 'README.md', and another 'README.md'. The file 'README.md' is expanded, showing its content: 'test-github'.

# 실습 - wi

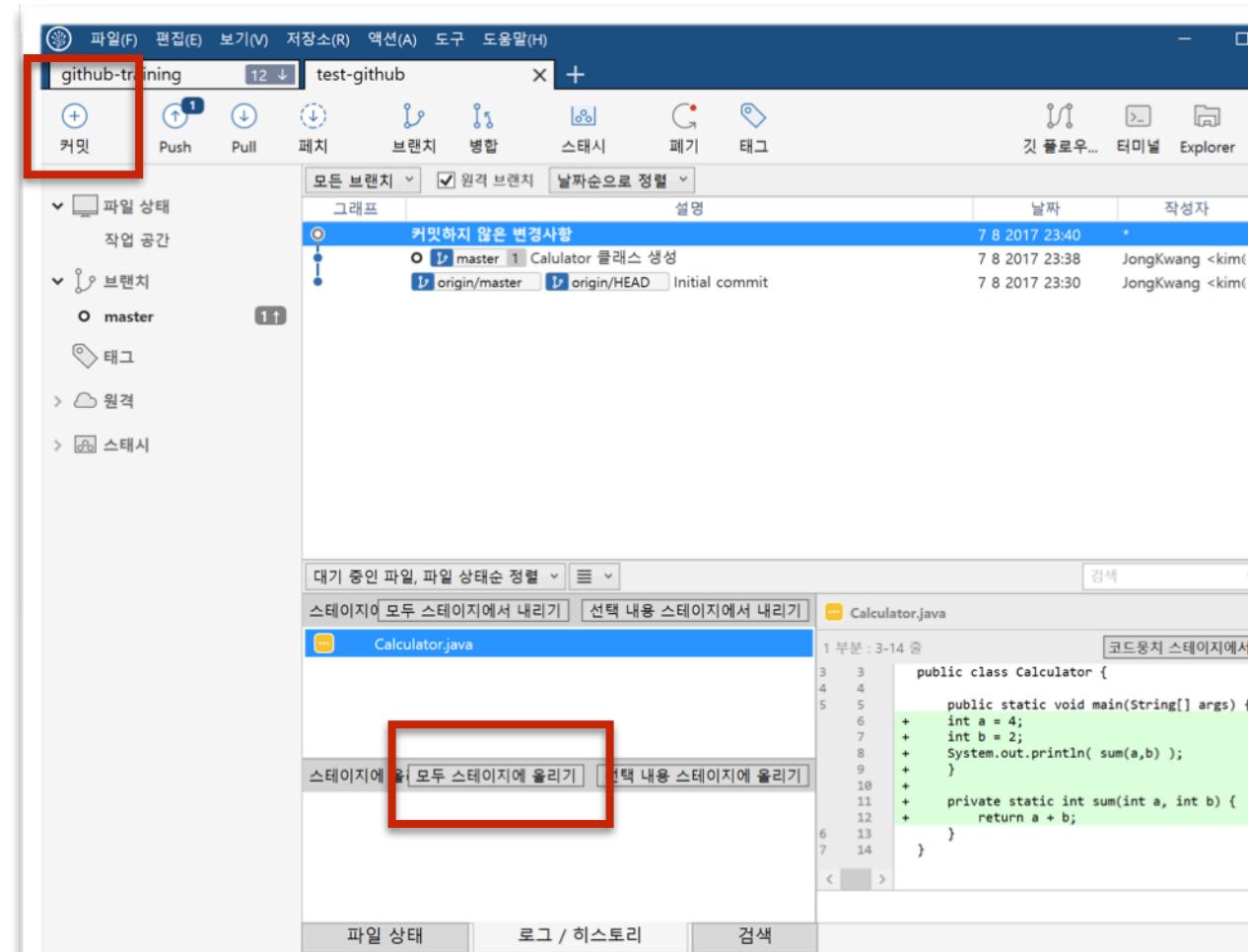
## ▶ sum() 코딩

```
Calculator.java  
package com.mycompany.test;  
  
public class Calculator {  
  
    public static void main(String[] args) {  
        int a = 4;  
        int b = 2;  
        System.out.println( sum(a,b) );  
    }  
  
    private static int sum(int a, int b) {  
        return a + b;  
    }  
}
```

# 실습 - with SourceTree

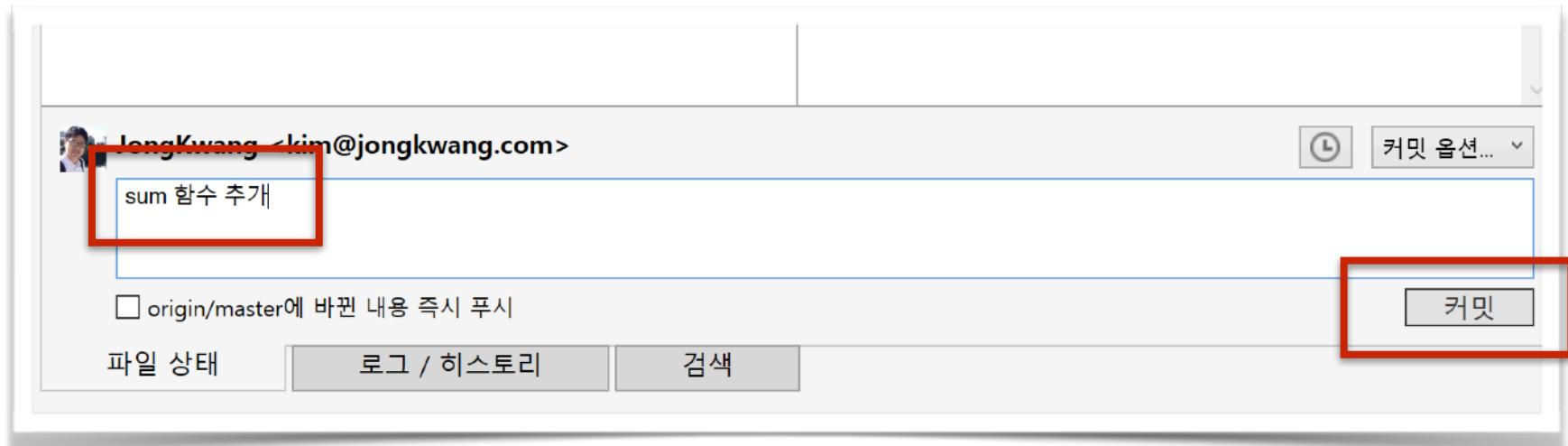
▶ Stage에 올리고

▶ 커밋



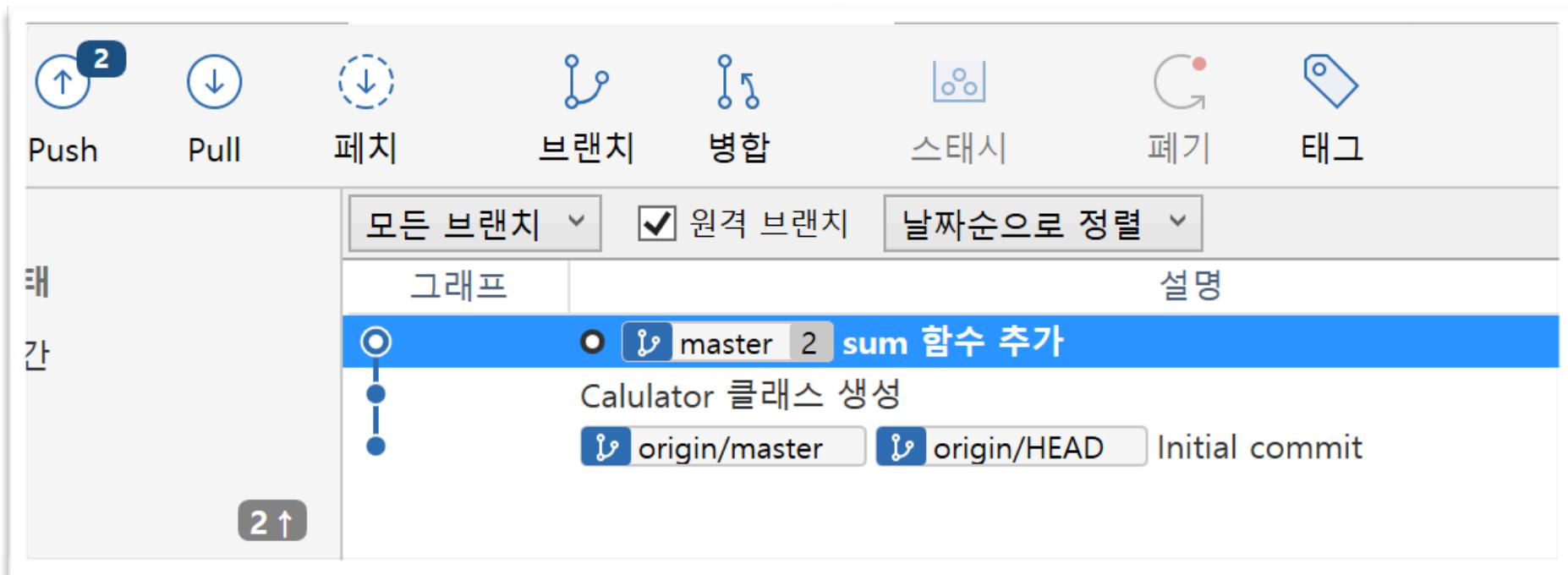
# 실습 - with SourceTree

- ▶ 커밋 로그 작성 후
- ▶ 커밋



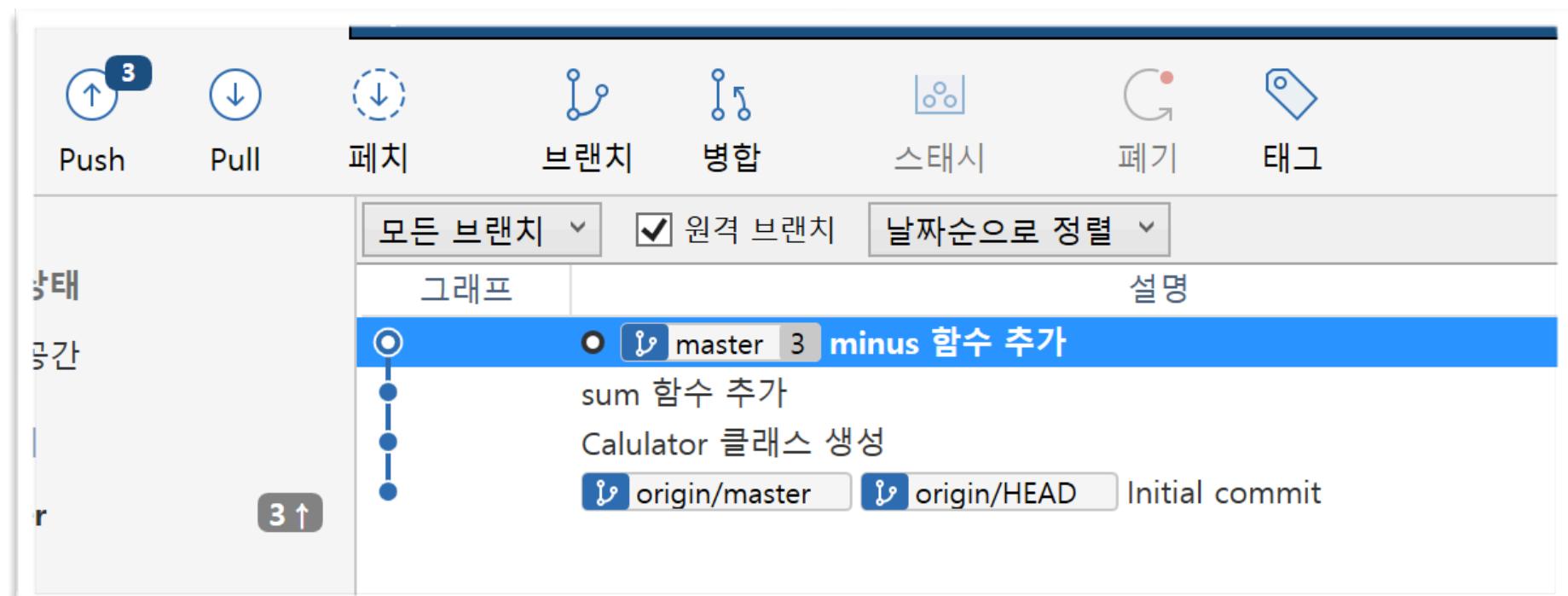
# 실습 – with SourceTree

- ▶ “sum 함수 추가” 확인



# 실습 – with SourceTree

- ▶ 같은 방식으로
- ▶ minus() 함수 추가



실습 – with SourceTree

Calculator2.java

파일로  
입력이

문서에  
제작

해주세요

# Git - 그림설명

Git → CLI

# 실습 - with CLI

## ▶ Clone

- git clone <https://github.com/jongkwang/test-github.git>

## ▶ 중복 방지를 위해 “cli” 폴더 생성

The screenshot shows a Windows Command Prompt window titled "명령 프롬프트". The command history is as follows:

```
C:\#Users\JongKwang>cd Documents
C:\#Users\JongKwang\Documents>mkdir cli
C:\#Users\JongKwang\Documents>cd cli
C:\#Users\JongKwang\Documents\cli>git clone https://github.com/jongkwang/test-github.git
Cloning into 'test-github'...
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.

C:\#Users\JongKwang\Documents\cli>
```

The command `git clone https://github.com/jongkwang/test-github.git` is highlighted with a red rectangle.

# 실습 - with CLI

## ▶ 해당 Repository 로 이동

```
C:\ 명령 프롬프트
C:\Users\JongKwang\Documents\cli>dir
C 드라이브의 볼륨: BOOTCAMB
볼륨 일련 번호: FCDE-4CF5

C:\Users\JongKwang\Documents\cli 디렉터리

2017-08-08 오전 12:32 <DIR> .
2017-08-08 오전 12:32 <DIR> ..
2017-08-08 오전 12:32 <DIR> test-github
          0개 파일           0 바이트
          3개 디렉터리  31,707,774,976 바이트 남음

C:\Users\JongKwang\Documents\cli>cd test-github
C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 – with CLI

- ▶ “atom .” ATOM 실행

```
명령 프롬프트

::#Users#JongKwang#Documents#cli#test-github>dir
C 드라이브의 볼륨: BOOTCAMP
볼륨 일련 번호: FCDE-4CF5

C:#Users#JongKwang#Documents#cli#test-github 디렉터리

017-08-08 오전 12:32 <DIR> .
017-08-08 오전 12:32 <DIR> ..
017-08-08 오전 12:32 294 .gitignore
017-08-08 오전 12:32 1,087 LICENSE
017-08-08 오전 12:32 13 README.md
017-08-08 오전 12:32 3개 파일 1,394 바이트
017-08-08 오전 12:32 2개 디렉터리 31,707,774,976 바이트 남음

::#Users#JongKwang#Documents#cli#test-github>atom .
```

# 실습 - with CLI

The screenshot shows a code editor interface with two main panes. On the left is a 'Project' pane showing a file tree for a project named 'test-github'. The file 'Calculator.java' is selected and highlighted with a red box. The right pane is a 'Calculator.java' code editor with the following content:

```
1 package com.mycompany.test;  
2  
3 public class Calculator {  
4  
5     public static void main(String[] args) {  
6         }  
7     }
```

The code editor pane also has a red border around its content area.

# 실습 – with CLI

## ▶ git status

- 현재 상황을 표시한다
- Calculator.java 파일이 Unstage 상태로 보인다

명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git status
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Calculator.java

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

C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

## ▶ git add Calculator.java

- Calculator.java 파일을 Stage에 올린다
- add 명령어는 신규 파일을 추가하기도 하지만 기존파일을 Stage에 올리기도 한다

```
C:\Users\JongKwang\Documents\cli\test-github>git add Calculator.java
```

```
C:\Users\JongKwang\Documents\cli\test-github>git status
```

```
On branch master
```

```
Your branch is up-to-date with 'origin/master'.
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    new file:   Calculator.java
```

```
C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

## ▶ git commit

- Stage 파일을 Commit 한다

명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git commit
```

# 실습 - with CLI

## ▶ commit message 를 작성한다.

- 첫째줄이 비워져 있고, 이곳에 작성한다
- 하단 주석들은 시스템에서 git status 값을 보여준다



A screenshot of a terminal window titled "명령 프롬프트 - git commit". The window contains the following text:

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
#   new file:   Calculator.java
#
# ~
# ~
# ~
# ~
# ~
# ~
```

The terminal window has a dark background and light-colored text. The title bar says "명령 프롬프트 - git commit". The bottom of the window shows the command line path and some statistics: "/Documents/cli/test-github/.git/COMMIT\_EDITMSG [unix] (00:37 08/08/2017)" and "1,0-1 All Kers/JongKwang/Documents/cli/test-github/.git/COMMIT\_EDITMSG~ [converted][unix] 10L, 268C".

# 실습 - with CLI

## ▶ 저장은

- ESC + wq + 엔터

The screenshot shows a terminal window with the following content:

```
명령 프롬프트 - git commit
Init Calculator class
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
#   new file:   Calculator.java
#
~
```

The bottom status bar indicates the path: /Documents/cli/test-github/.git/COMMIT\_EDITMSG[+] [unix] (00:37 08/08/2017) and the line number: 1,21 All. A red box highlights the command ':wq' at the bottom of the terminal window.

# 실습 – with CLI

## ▶ 커밋 성공

- Commit message 를 남기지 않으면 커밋이 되지 않는다 (Default)
- 커밋이 성공하여 “1be5fcf” 라는 Hash 값이 생겼다

명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git commit  
[master 1be5fcf] Init Calculator class  
 1 file changed, 7 insertions(+)  
 create mode 100644 Calculator.java  
  
C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 – with CLI

## ▶ git status

- 모두 반영되어 깨끗하다

```
C:\Users\JongKwang\Documents\cli\test-github>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

## ▶ git log

- 커밋 로그를 표시한다

C:\ 명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git log
commit 1be5cfcb61a7c4d8fbe7e090c38d9cbe1ebe505 (HEAD -> master)
Author: JongKwang <kim@jongkwang.com>
Date:   Tue Aug 8 00:37:00 2017 +0900

    Init Calculator class

commit bdc66b83e692c400250faec145377e2d2fe5357f (origin/master, origin/HEAD)
Author: JongKwang <kim@jongkwang.com>
Date:   Mon Aug 7 23:30:00 2017 +0900

    Initial commit

C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

- ▶ sum() 메소드 작성

```
package com.mycompany.test;

public class Calculator {

    public static void main(String[] args) {
        int a = 4;
        int b = 2;
        System.out.println( sum(a,b) );
    }

    private static int sum(int a, int b) {
        return a + b;
    }
}
```

# 실습 - with CLI

- ▶ git status
- ▶ git add Calculator.java

c:\ 명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Calculator.java

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\JongKwang\Documents\cli\test-github>git add Calculator.java

C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 – with CLI

## ▶ git reset Calculator.java

- Stage 의 파일을 Unstage 로 옮긴다

C:\ 명령 프롬프트

```
C:\#Users\JongKwang\Documents\cli\test-github>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
```

```
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   Calculator.java
```

```
C:\#Users\JongKwang\Documents\cli\test-github>git reset Calculator.java
Unstaged changes after reset:
M       Calculator.java
```

```
C:\#Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

- ▶ git reset 되어 Unstage 된 자바 파일

## 명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   Calculator.java

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 – with CLI

- ▶ commit message 작성하지 않으면
  - 커밋이 되지 않는다

## 명령 프롬프트

```
C:\Users\JongKwang\Documents\cli\test-github>git add Calculator.java
```

```
C:\Users\JongKwang\Documents\cli\test-github>git commit  
Aborting commit due to empty commit message.
```

```
C:\Users\JongKwang\Documents\cli\test-github>
```

# 실습 - with CLI

## ▶ 커밋 메세지 작성

```
명령 프롬프트 - git commit

Add sum() method
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:   Calculator.java
#
# ~
# ~
# ~
# ~
# ~
# ~
# ~

~/Documents/cli/test-github/.git/COMMIT_EDITMSG[+] [unix] (01:11 08/
:wq
```

실습 – with SourceTree

Calculator2.java

파일로  
입력이

문서에  
제작

해주세요

Git - 그림설명

Pointer

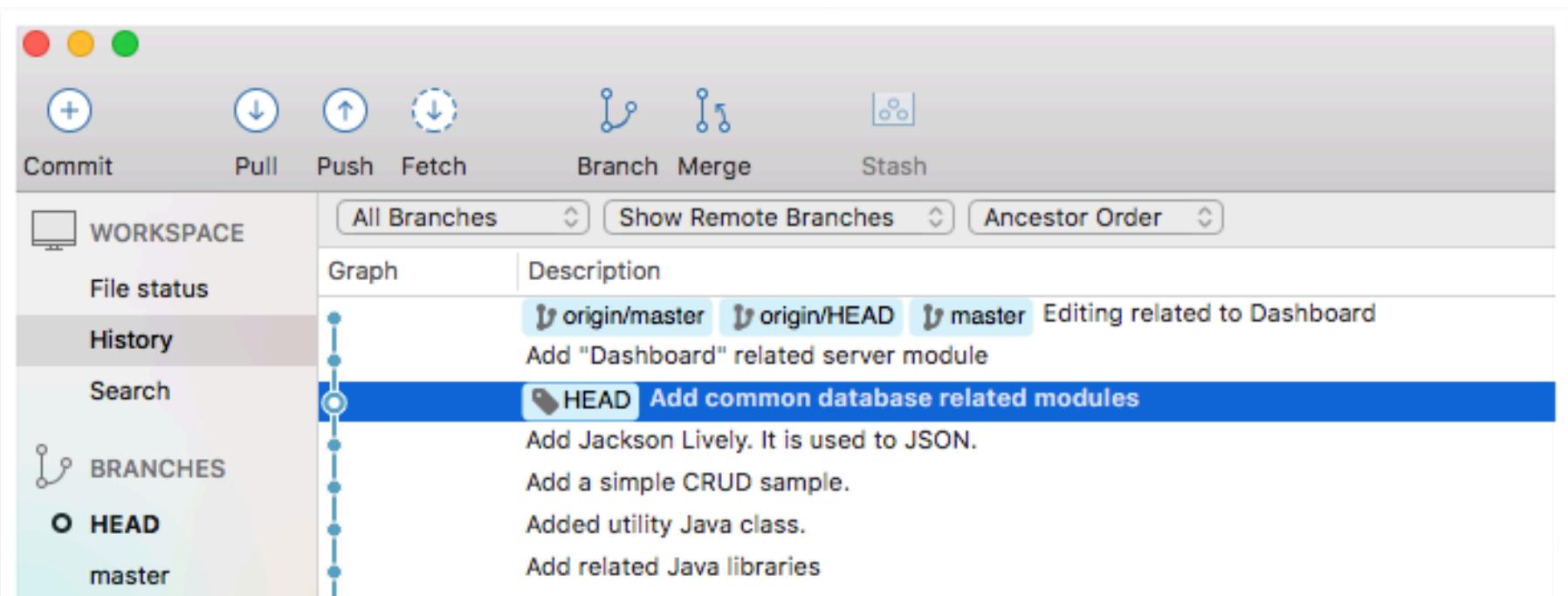
# Git – Pointer

## ▶ HEAD tag

- 작업폴더가 바라보고 있는 버전을 표시

## ▶ Master branch

- 가장 기본이 되는 Brach
- Master branch 는 Git 에서 기본으로 생성해준다



# Git – Pointer

## ▶ origin

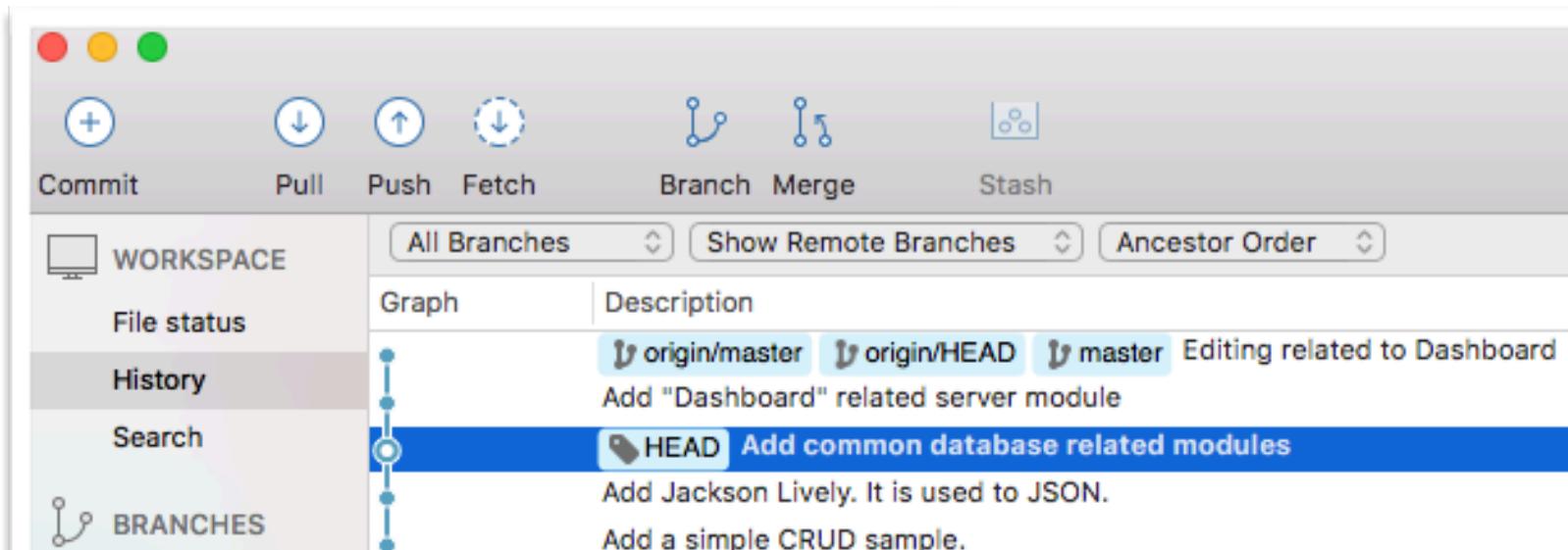
- 원격 Repository 를 뜻하는 Alias
- Alias 이름을 변경 할 수 있다

## ▶ origin/master

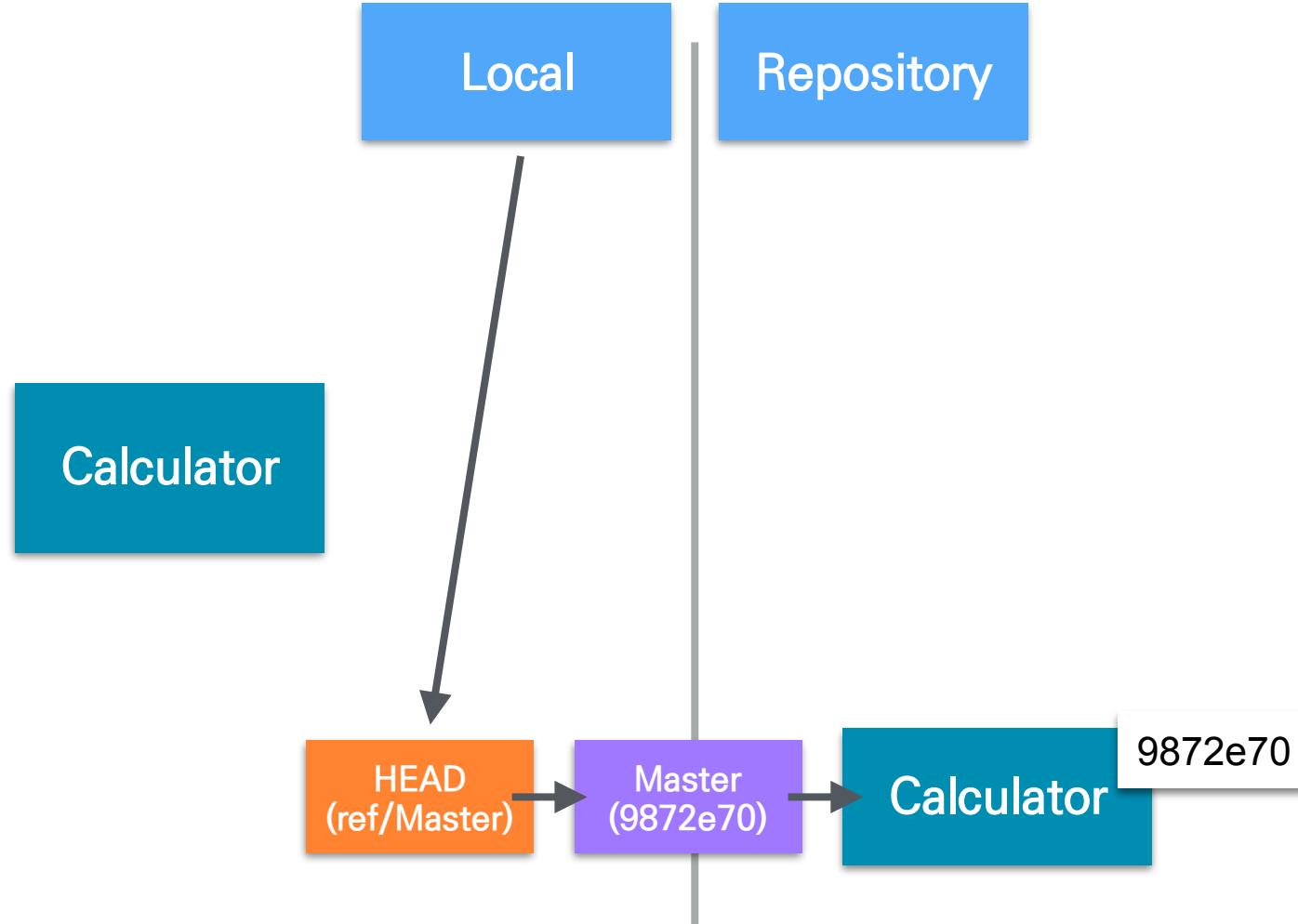
- 원격 Repository 의 master branch

## ▶ origin/HEAD

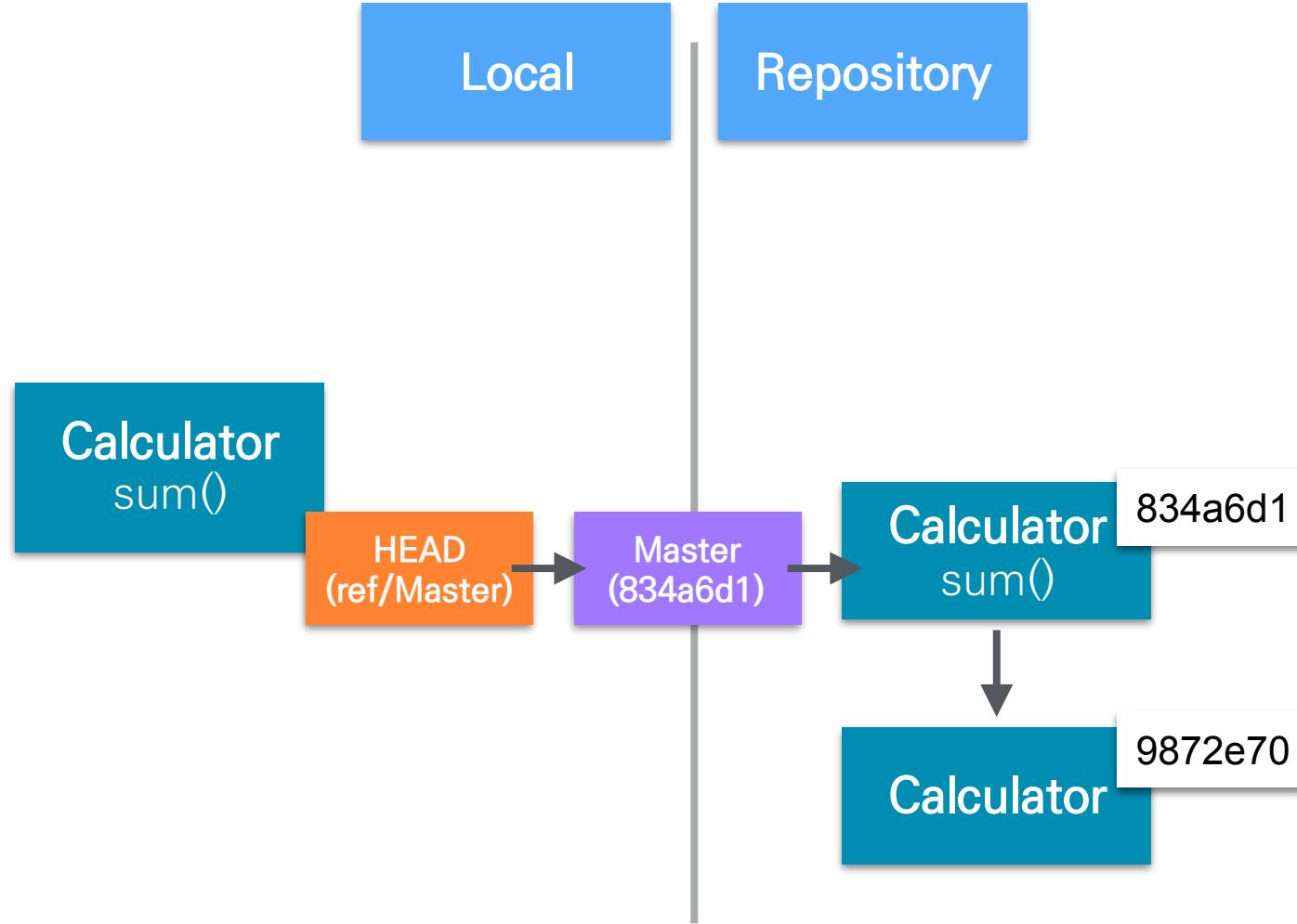
- 원격 Repository 가 바라보는 버전



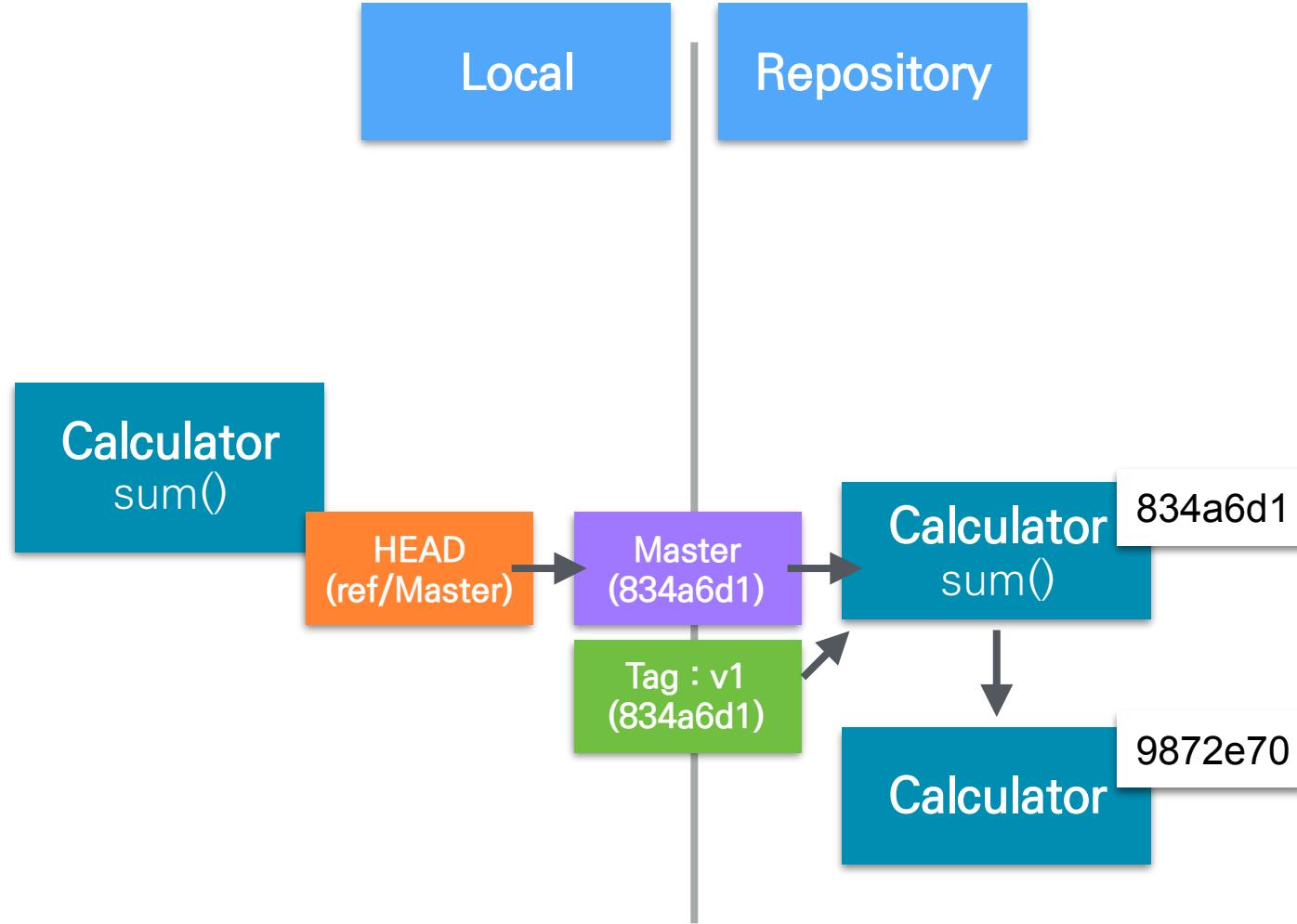
# Git - 그림 설명



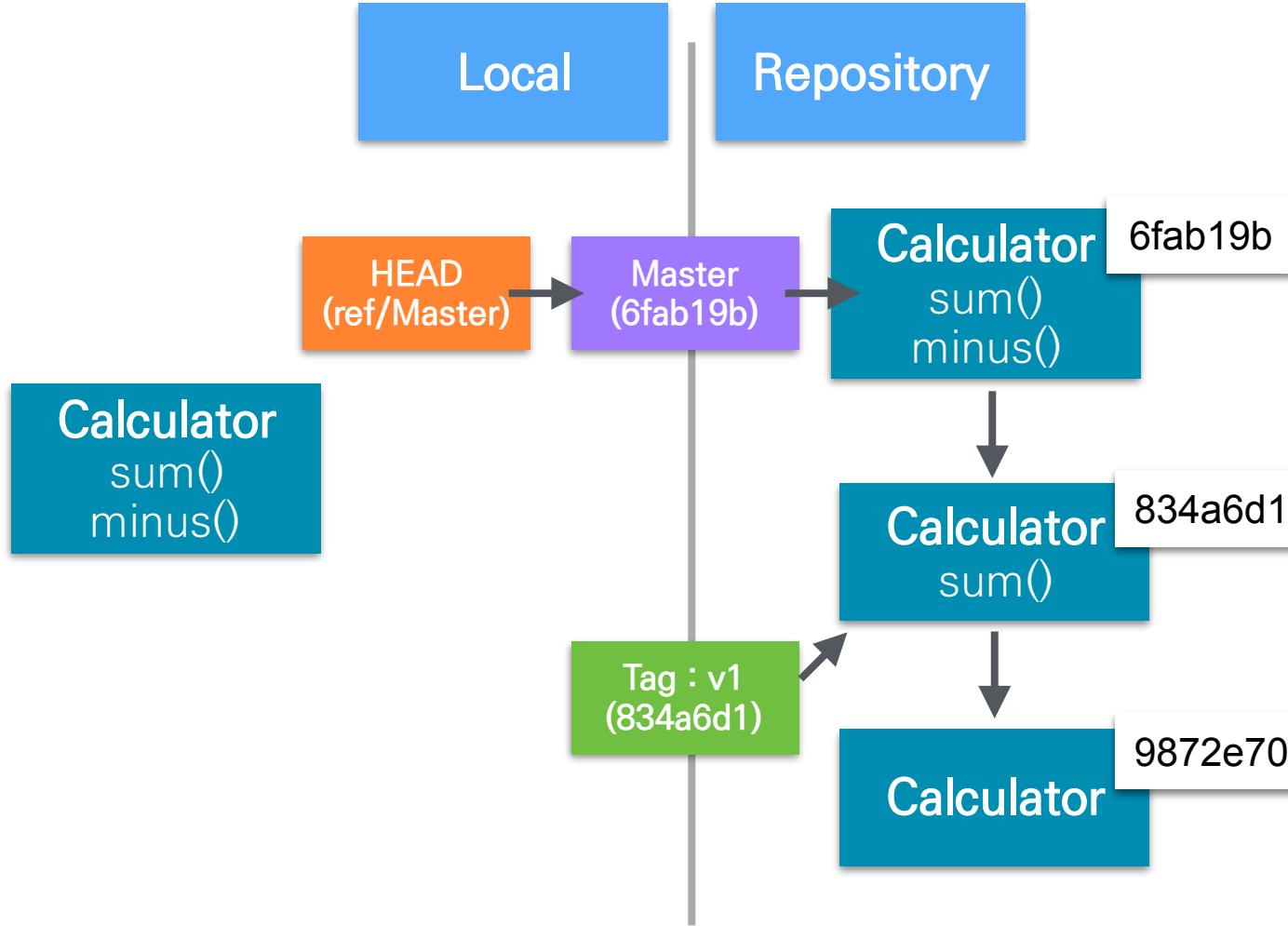
# Git – 그림 설명



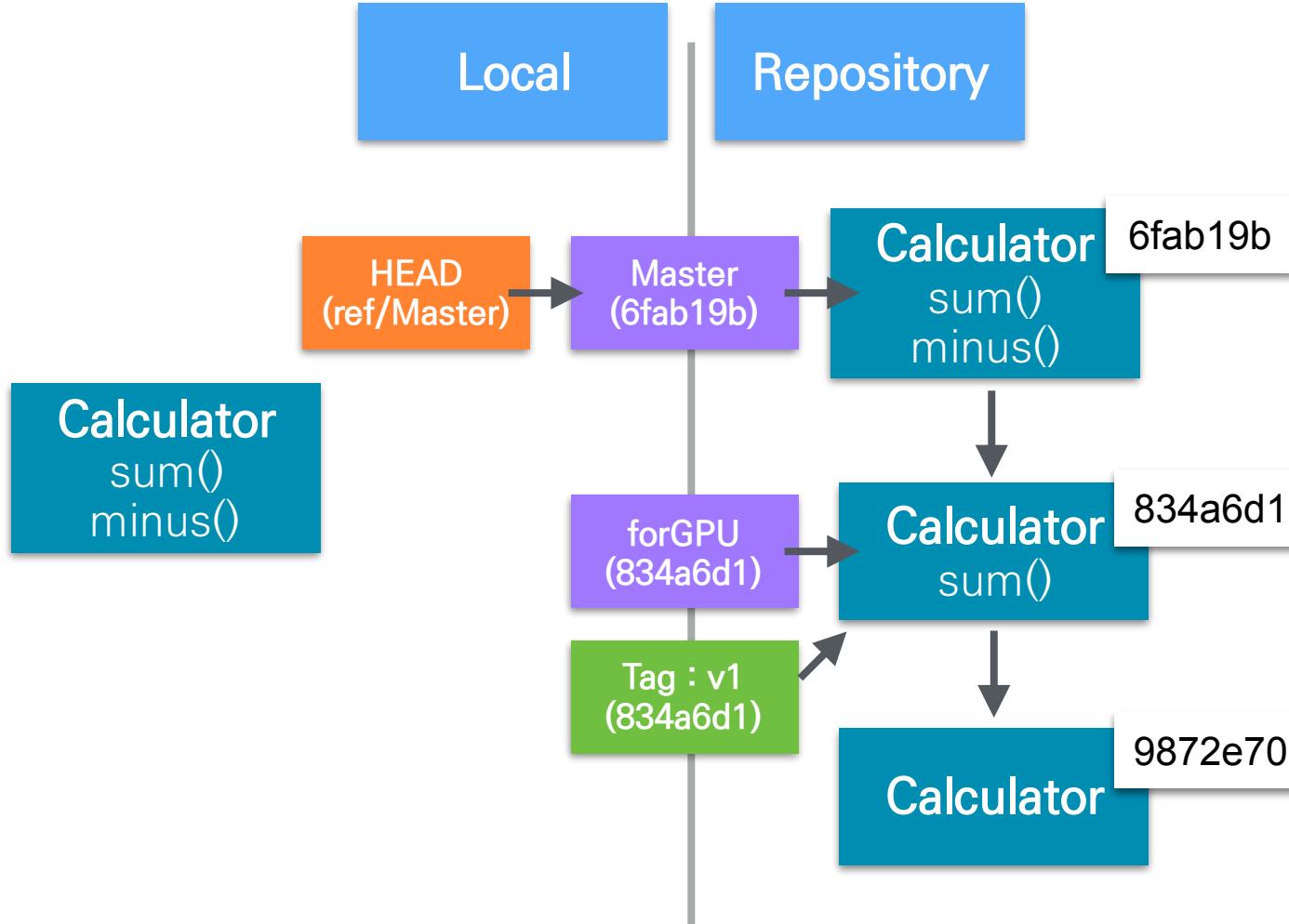
# Git – 그림 설명



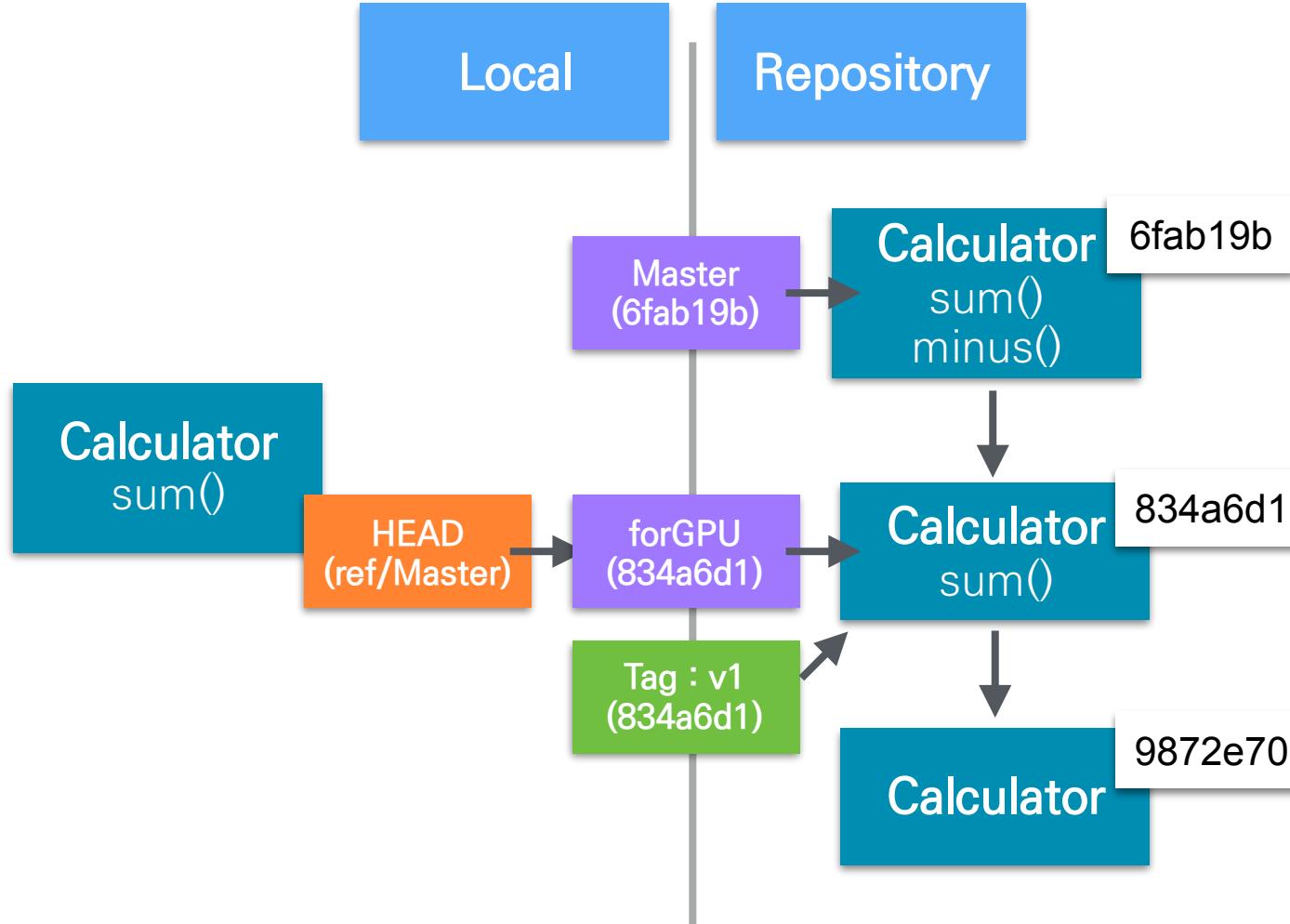
# Git – 그림 설명



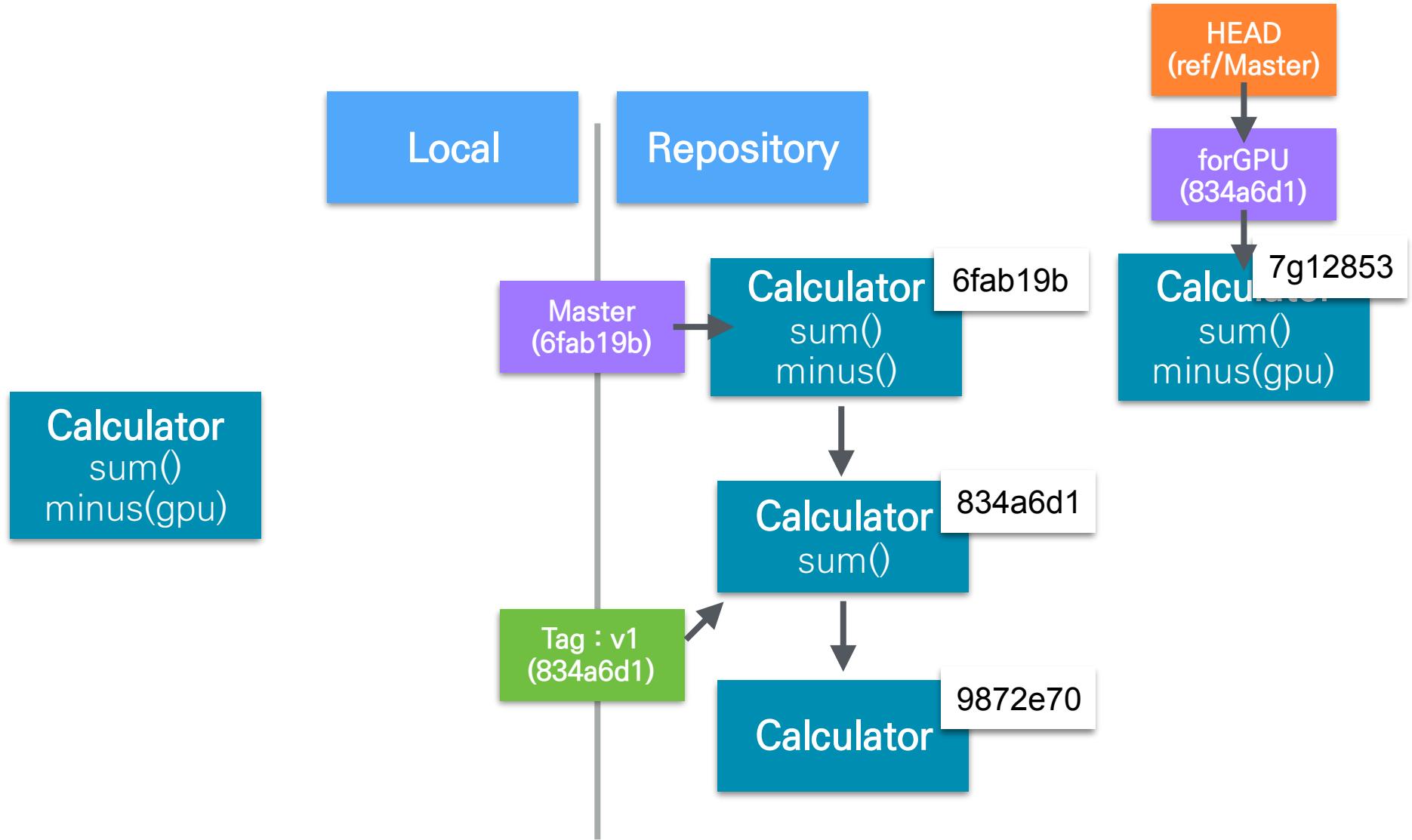
# Git – 그림 설명



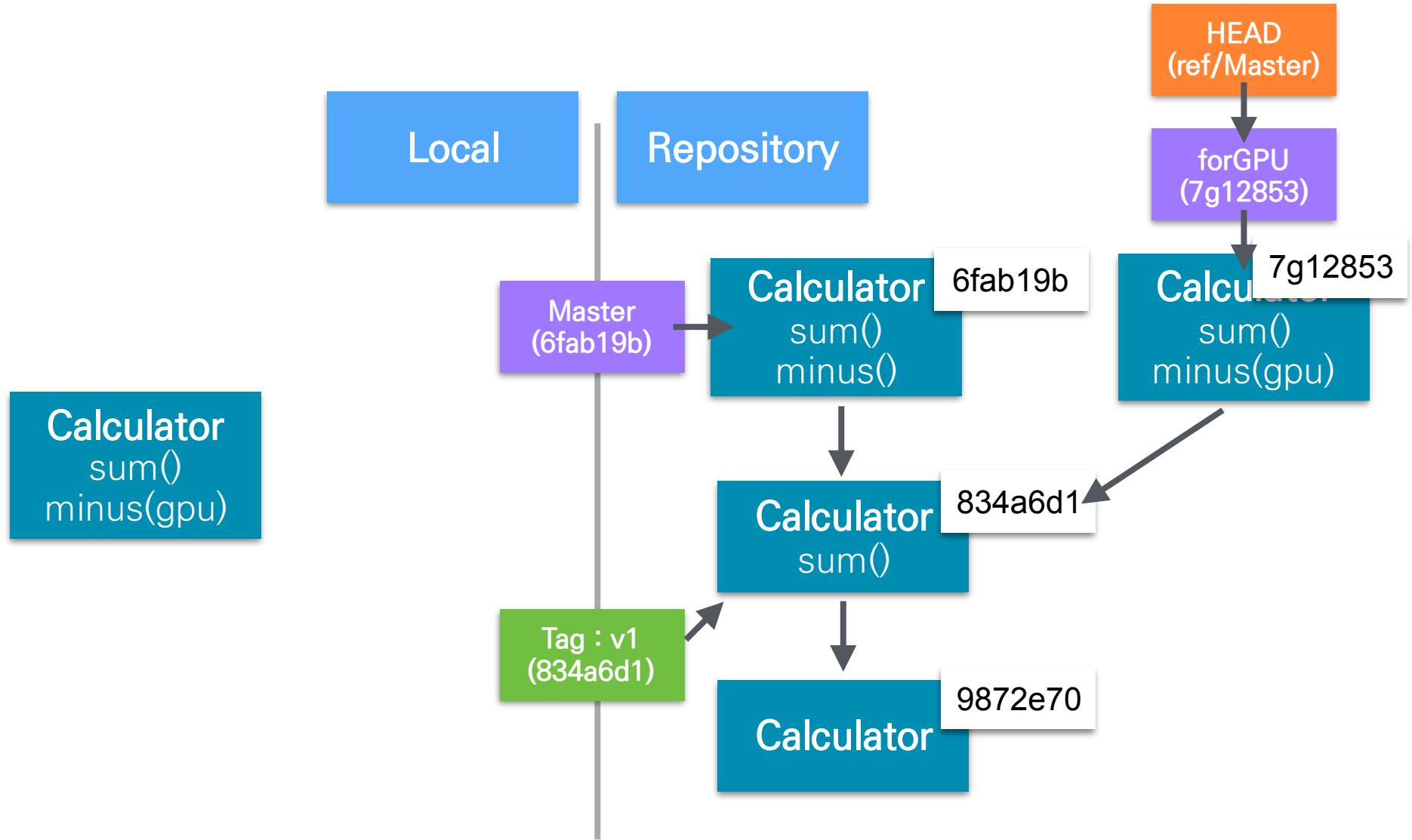
# Git – 그림 설명



# Git – 그림 설명



# Git – 그림 설명



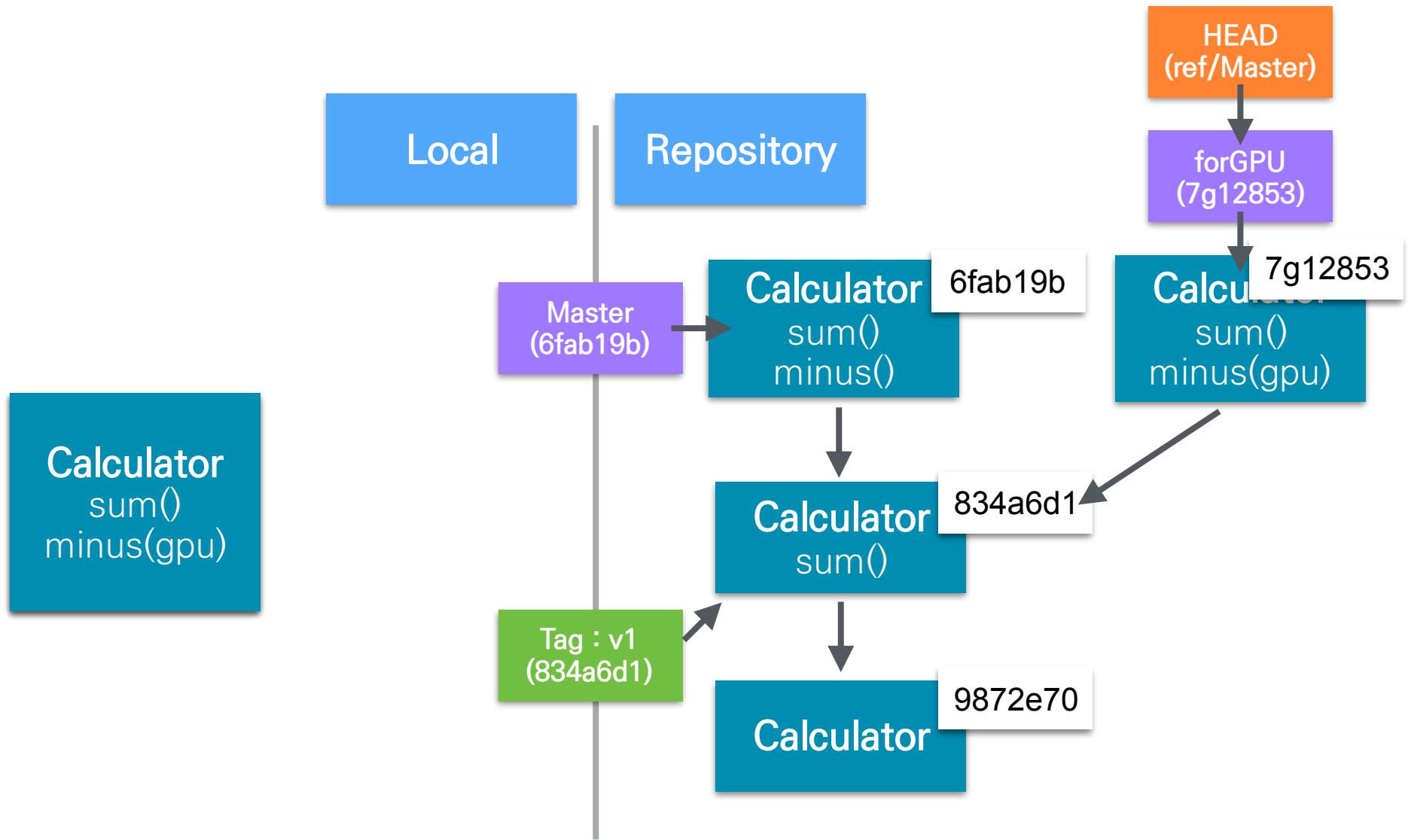
# Git - 그림설명

병합

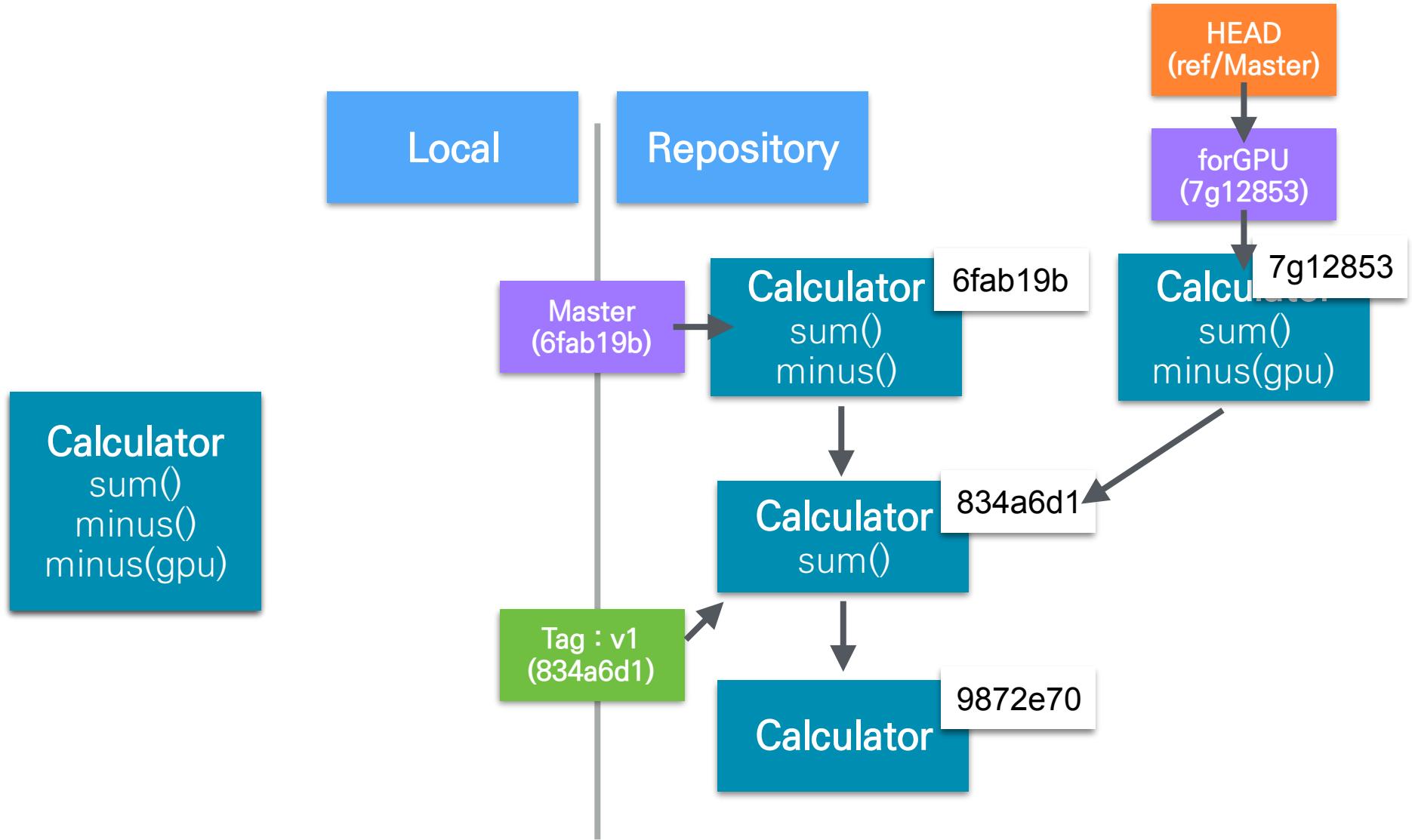


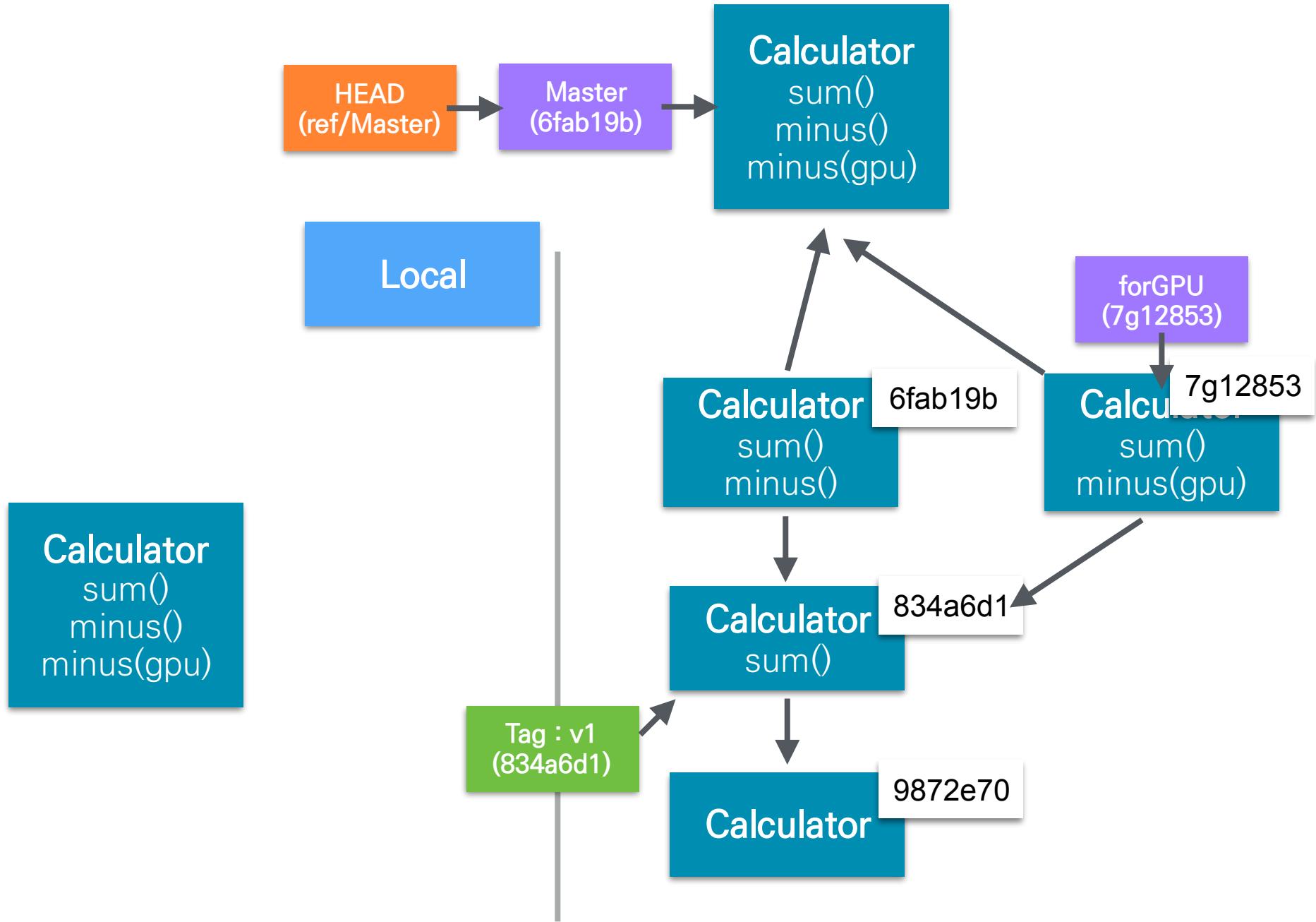
Merge

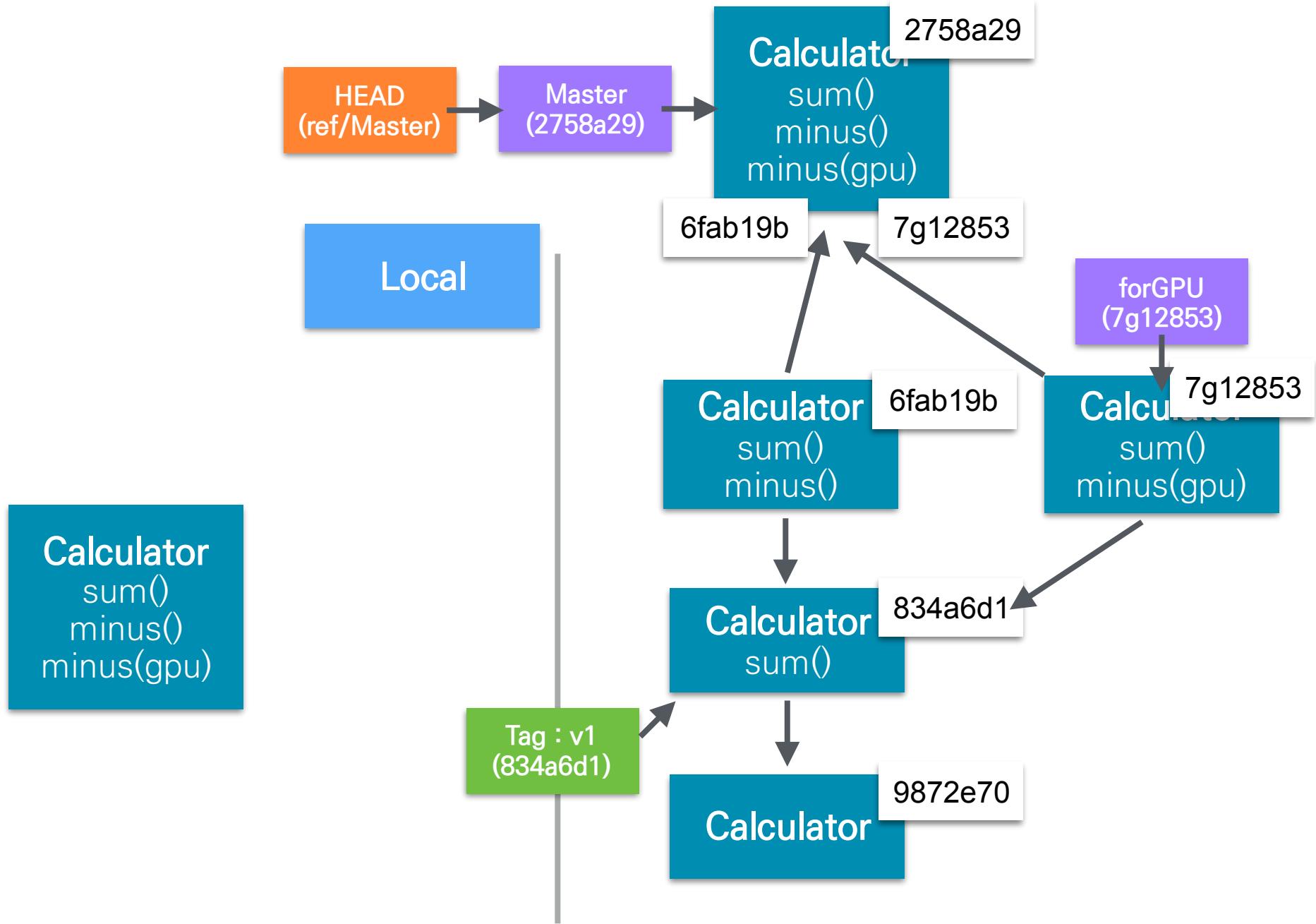
# Git – 그림 설명



# Git – 그림 설명

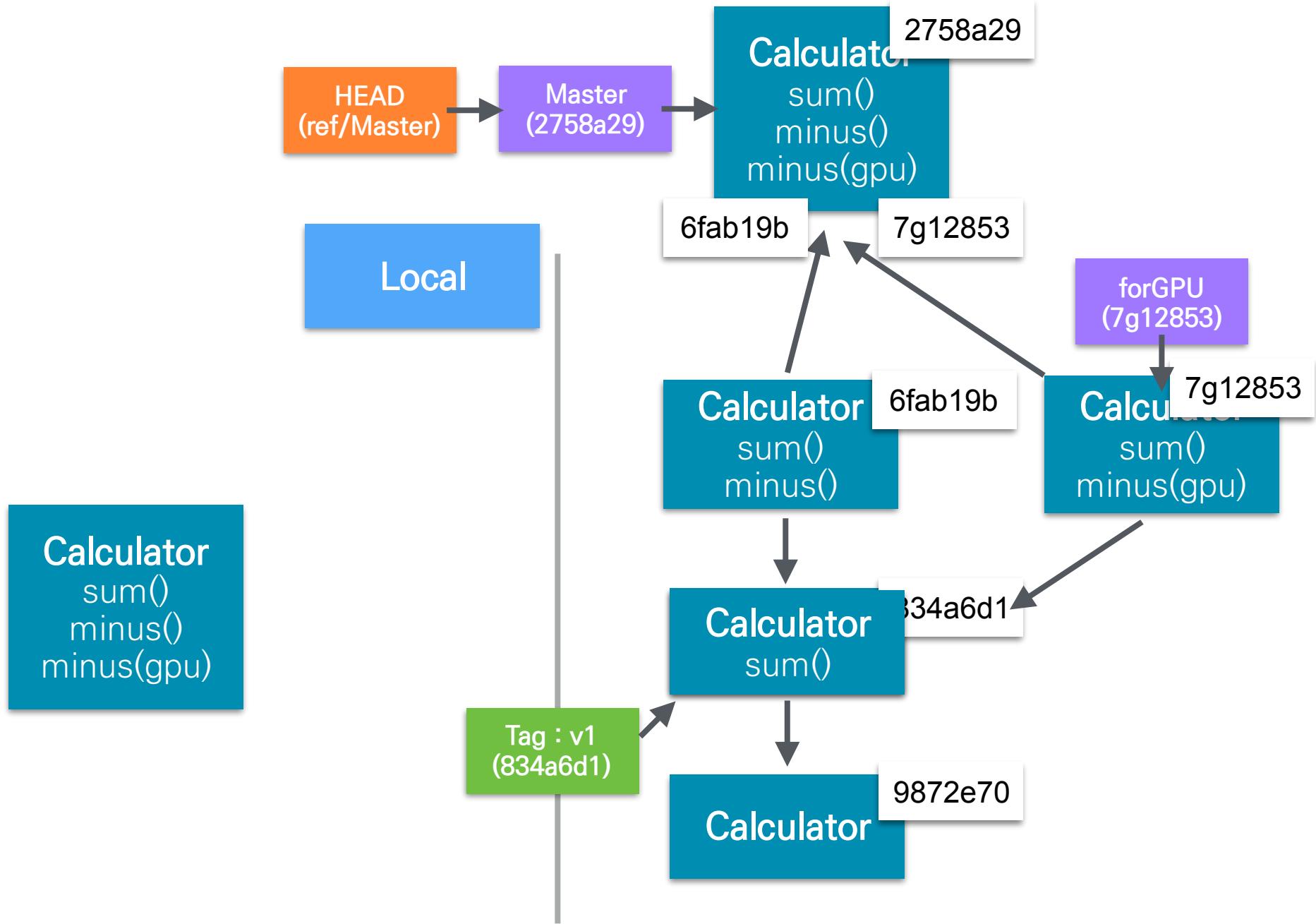


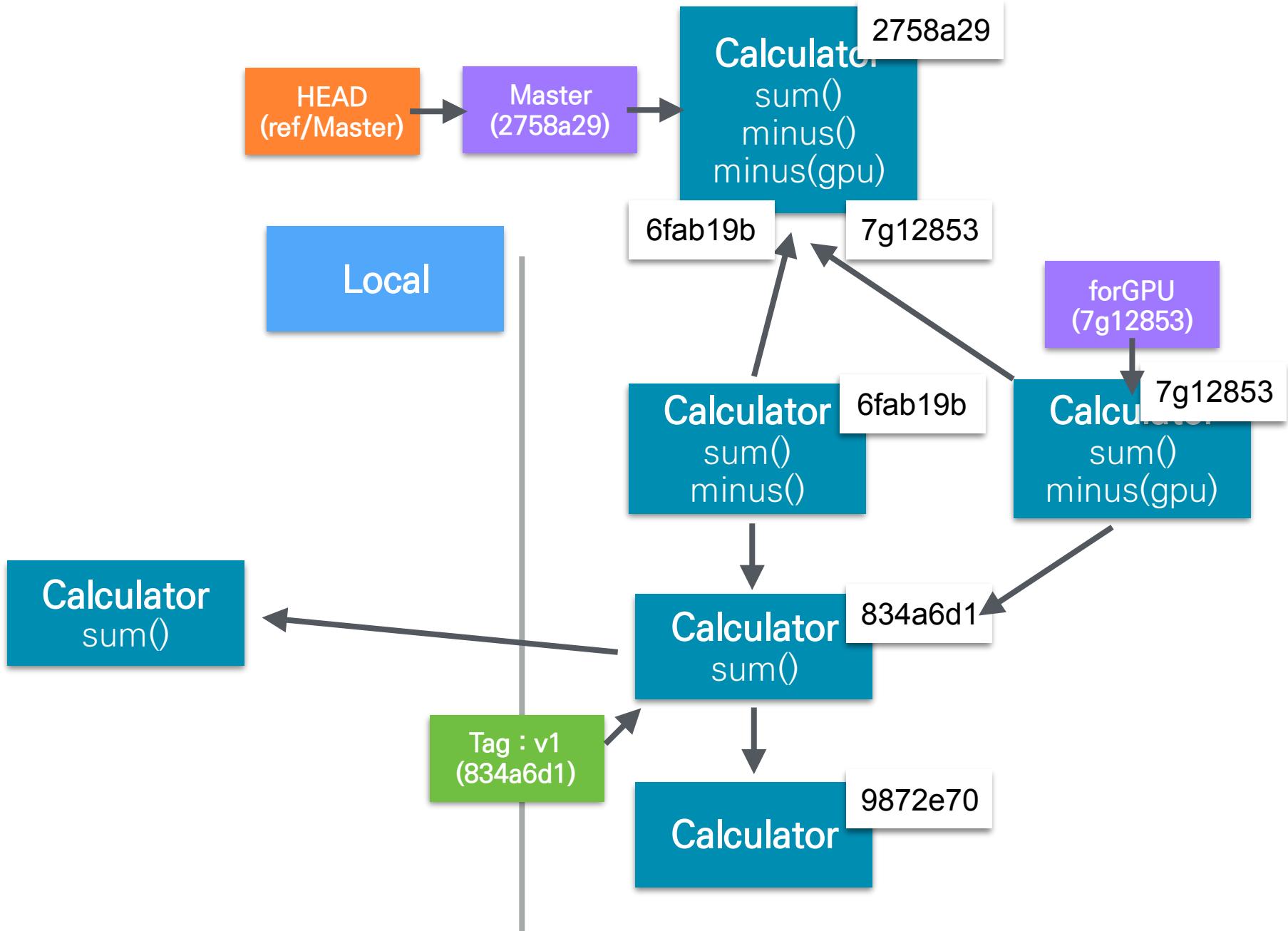


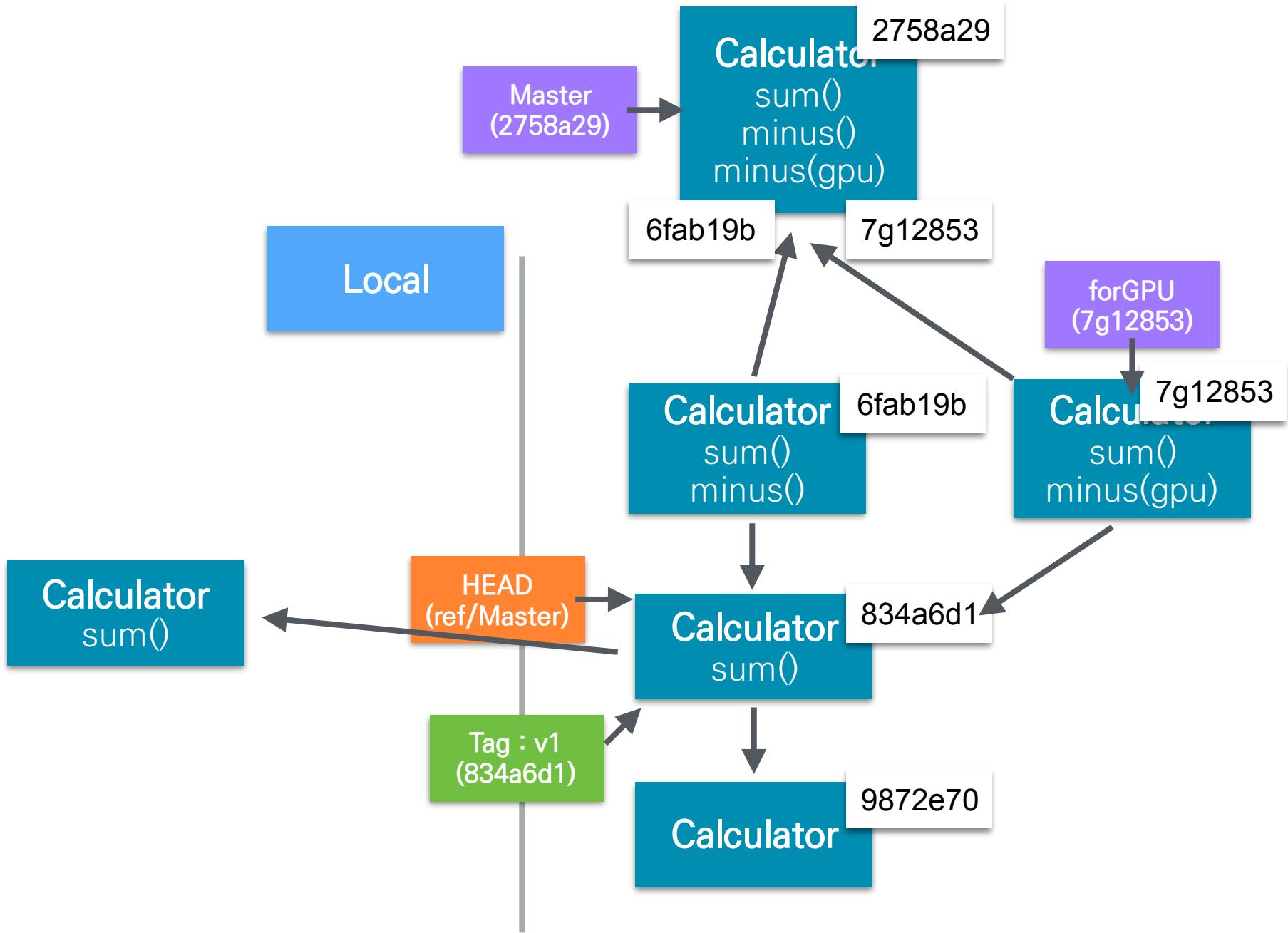


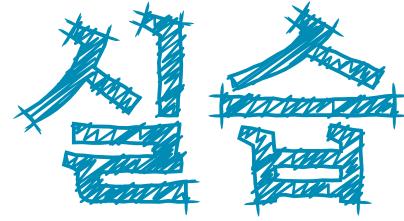
Git - 그림설명

Checkout









Git - 그림설명

# Reference

# Reference

## ▶ Git

- <https://git-scm.com/>
- <https://git-scm.com/book/ko/v2>

## ▶ 이종은님 Github

- [jsdev.kr](http://jsdev.kr) 운영진
- Titanium 커뮤니티
- <https://github.com/yomybaby/gitlecture>

# Git / Github

깃허브