

Git / Github 입문

190323
@한빛미디어

한홍근

- enuri91@gmail.com
- (주)엑셈 개발자
- 취미 : IT 개발서적 베타리딩

순서

1. 형상 관리와 버전 관리
2. 버전 관리 시스템
3. Git과 Github
4. Git Hosting
5. Git Client
6. Git을 이용한 버전 관리
 - a. init / status / add / commit / diff / branch / checkout / merge / conflict /
 - b. fork / clone / pull request / fetch pull
7. GitHub 둘러보기
8. 알아두면 좋은 것

소프트웨어 형상 관리와 버전 관리

소프트웨어 형상 관리

- 소프트웨어의 **변경사항을 체계적으로 추적하고 관리, 통제하는 것**
- 형상 관리 대상
 - 요구사항 명세서
 - 설계서
 - 소스코드
 - 이미지
 - 라이브러리 파일
 - 프로젝트 회의록
 - 그 외 많은 관리 대상들

면류 / 밥류

옛날짜장 ₩5,500 옛날짜장밥 ₩6,000

돼지고기를 넣어 볶아낸 아이들부터 어른들까지 누구나 즐겨 찾는 짜장면.



볶음짜장 ₩7,000 볶음짜장밥 ₩7,500

돼지고기, 해물, 신선한 야채를 푸짐하게 넣어 춤장과 셀룰로 볶은 후 면과 함께 한번 더 볶아 더욱 더 진한 맛이 일품인 볶음짜장면.



불짜장 ₩8,000 불짜장밥 ₩8,500

청양고추와 상하이반점만의 매콤한 소스에 맛있게 볶아진 춤장이 어우러져 화끈하게 맛있는 매운맛의 짜장면.



물짜장 ₩7,000

전주의 명물!

돼지고기, 해산물이 가득하고 깊은 소스 맛이 일품인 물짜장.



상하이짬뽕 ₩6,000 상하이짬뽕밥 ₩6,500

푸짐한 돼지고기, 해물, 신선한 야채를 셀룰로 볶아내 불맛이 일품인 상하이 반점의 짬뽕.

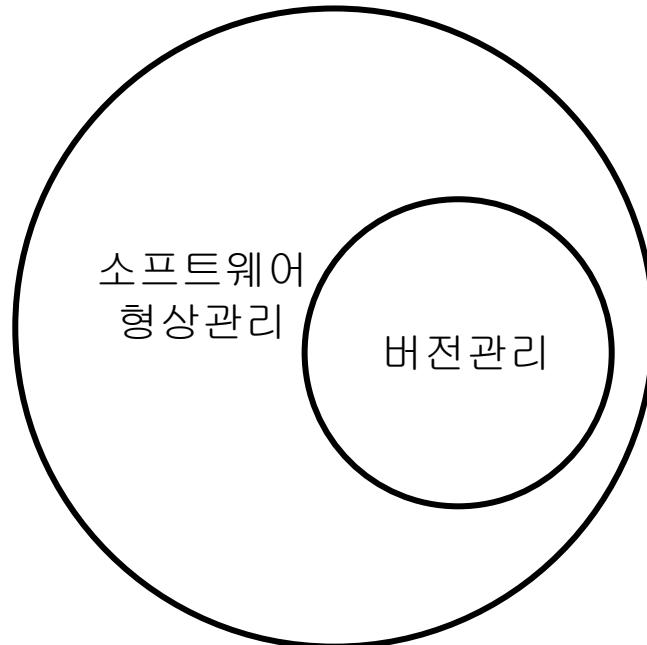


볶음짜장면 ₩7,500



소프트웨어 형상 관리

- 소프트웨어의 **변경사항을 체계적으로 추적하고 관리, 통제하는 것**
- 형상 관리 대상
 - 요구사항 명세서
 - 설계서
 - 소스코드
 - 이미지
 - 라이브러리 파일
 - 프로젝트 회의록
 - 그 외 많은 관리 대상들



면류 / 밥류

옛날짜장 ₩5,500 옛날짜장밥 ₩6,000

돼지고기를 넣어 볶아낸 아이들부터 어른들까지 누구나 즐겨 찾는 짜장면.



볶음짜장 ₩7,000 볶음짜장밥 ₩7,500

돼지고기, 해물, 신선한 야채를 푸짐하게 넣어 춤장과 셀룰로 볶은 후 면과 함께 한번 더 볶아 더욱 더 진한 맛이 일품인 볶음짜장면.



불짜장 ₩8,000 불짜장밥 ₩8,500

청양고추와 상하이반점만의 매콤한 소스에 맛있게 볶아진 춤장이 어우러져 화끈하게 맛있는 매운맛의 짜장면.



물짜장 ₩7,000

전주의 명물!

돼지고기, 해산물이 가득하고 깊은 소스 맛이 일품인 물짜장.



상하이짬뽕 ₩6,000 상하이짬뽕밥 ₩6,500

푸짐한 돼지고기, 해물, 신선한 야채를 셀룰로 볶아내 불맛이 일품인 상하이 반점의 짬뽕.



보리짜장 ₩7,500



형상관리를 왜 해야 할까?

- 소프트웨어 개발 생산성을 향상시켜 준다
- (요구사항을 바탕으로 개발을 진행하므로) 소프트웨어의 품질을 향상시켜 준다
- (문제가 있는 경우를 쉽게 발견할 수 있으므로) 유지보수 비용을 절감시켜 준다

A screenshot of the Visual Studio Code interface. The title bar reads "[Extension Development Host] - app.js - vscode-annotator". The left sidebar shows icons for settings.json, app.js (which is the active tab), and other files. The main editor area displays the following JavaScript code:

```
1 'use strict';
2
3
4 class App {
5   constructor(params) {
6     this._commands = params.commands;
7     this._logger = params.logger;
8     this._uriService = params.uriService;
9   }
10
11   annotate(editor) {
12     return Promise.resolve().then(() => {
13       const uri = this._uriService.convertToAnnotateFileAction(editor.document.uri);
14       if (!uri) return;
15
16       const title = this._uriService.getTitle(uri, 'annotation: ');
17       return this._commands.executeCommand('vscode.previewHTML', uri, undefined, title);
18     }).catch(e => this._handleError(e));
19   }
20
21   takeDiff(params) {
22     return Promise.resolve().then(() => {
23       const paramsBefore = {
24         commitHash: params.previousCommitHash,
25         path: params.previousPath,
26         repositoryRoot: params.repositoryRoot
27       };
28       const uriBefore = this._getUri(paramsBefore);
29     });
30   }
31 }
32
33 module.exports = App;
```

The code uses the 'vscode' extension API, specifically the `convertToAnnotateFileAction` and `executeCommand` methods. There are several annotations in the code, indicated by small blue circles with numbers (1, 2, 3, 4) and arrows pointing to specific lines or words. The status bar at the bottom shows the file is on master*, line 15, column 30, with 4 spaces, encoding UTF-8, and LF line endings.

<https://code.visualstudio.com/blogs/2016/07/29/extensions-roundup-git>

[Extension Development Host] - annotation: /Users/ryuichi/repos/vscode-annotator/lib/app.js - vscode-annotator

The screenshot shows the Visual Studio Code interface with an extension named "vscode-annotator" open. The left sidebar displays a list of commits from a GitHub repository. A tooltip is displayed over the commit at line 16, which has the hash 928ced08dc884af2718b0e126ccee5ce6f25f55b. The tooltip contains the commit details: "Commit: 928ced08dc884af2718b0e126ccee5ce6f25f55b", "Author: Ryuichi Inagaki", and "Date: 2016/7/18 22:54:00". The main editor area shows the source code of the app.js file, which includes annotations for the commit at line 16.

```
'use strict';

class App {

    constructor(params) {
        this._commands = params.commands;
        this._logger = params.logger;
        this._uriService = params.uriService;
    }

    annotate(editor) {
        return Promise.resolve().then(() => {
            const uri = this._uriService.convertToAnnotateFileAction(editor.document.uri);
            if (!uri) return;
            title(uri, 'annotation: ');
            id('vscode.previewHTML', uri, undefined, title);
        });
    }

    takeDiff(params) {
        return Promise.resolve().then(() => {
            const paramsBefore = {
                commitHash: params.previousCommitHash,
                path: params.previousPath,
                repositoryRoot: params.repositoryRoot
            };
            const uriBefore = this._uriService.getUri(paramsBefore);
        });
    }
}
```

<https://code.visualstudio.com/blogs/2016/07/29/extensions-roundup-git>

[Extension Development Host] - lib/app.js@920ced0 - vscode-annotator

```
settings.json      app.js      annotation: /Users/yuchi/repos/vscode-annotator/lib/app.js      lib/app.js@920ced0
```

```
1 'use strict';
2
3
4 class App {
5
6   constructor(params) {
7     this._commands = params.commands;
8     this._logger = params.logger;
9     this._uriService = params.uriService;
10  }
11
12  annotate(editor) {
13    return Promise.resolve().then(() => {
14      const uri = this._uriService.convertToAnnotateUri(editor);
15
16      const title = this._uriService.getTitle(uri);
17      return this._commands.executeCommand('vscode.annotate', {uri}).catch(e => this._handleError(e));
18    });
19
20  takeDiff(params) {
21    return Promise.resolve().then(() => {
22      const paramsBefore = {
23        commitHash: params.previousCommitHash,
24        path: params.previousPath,
25        repositoryRoot: params.repositoryRoot,
26      };
27      const uriBefore = this._getUri(paramsBefore);
28
29    });
30  }
31
32  _getUri(params) {
33    const uri = this._uriService.getUri(params);
34    return this._logger.info(`Annotation URI: ${uri}`);
35  }
36
37  _handleError(error) {
38    this._logger.error(`An error occurred while annotating: ${error.message}`);
39  }
40}
```

<https://code.visualstudio.com/blogs/2016/07/29/extensions-roundup-git>

소스코드 버전 관리

- 소스코드의 변경사항을 체계적으로 추적하고 관리, 통제하는 것
- 소스코드로 인한 혼동이 최소화 된다

버전 관리 시스템

현금 100억이 있다면...?



버전 관리 시스템

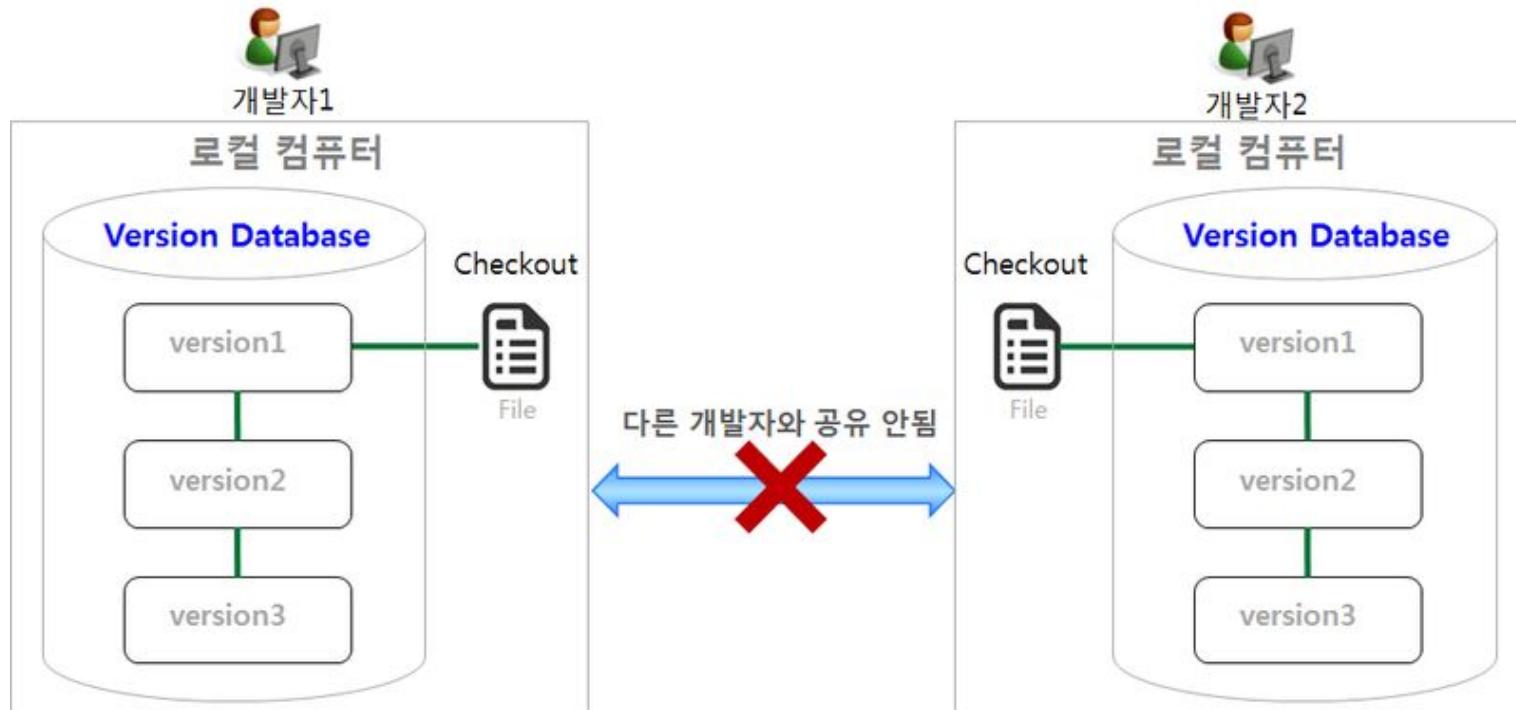
- 소스코드의 변경사항을 체계적으로 추적하고 관리, 통제하는 시스템

버전 관리 시스템

- 버전 관리 시스템 - VCS(Version Control System)
- 집중형 버전 관리 시스템 - CVCS(**Centralized** VCS)
- 분산형 버전 관리 시스템 - DVCS(**Distributed** VCS)

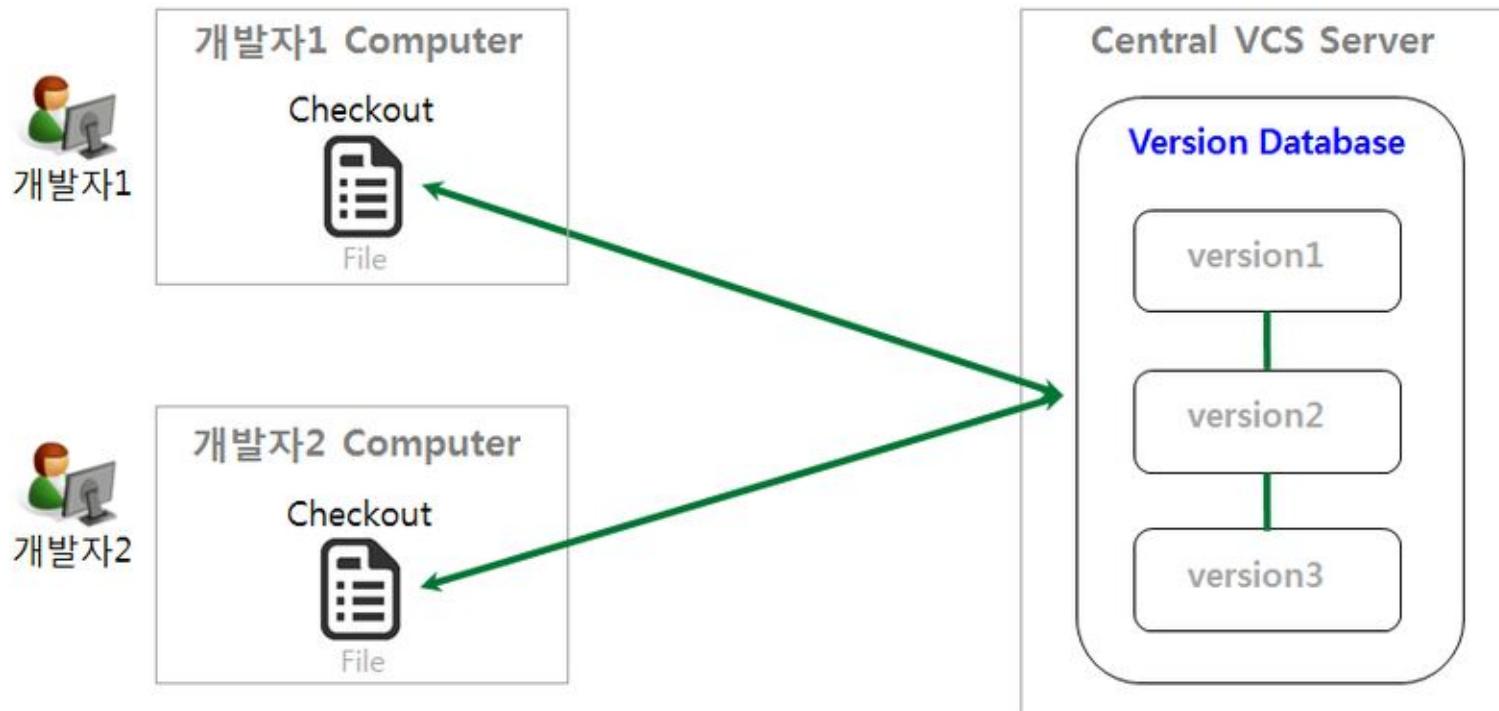
버전 관리 시스템 - VCS(Version Control System)

- 로컬에서 직접 모든 자원을 관리
- USB, 메일 첨부파일



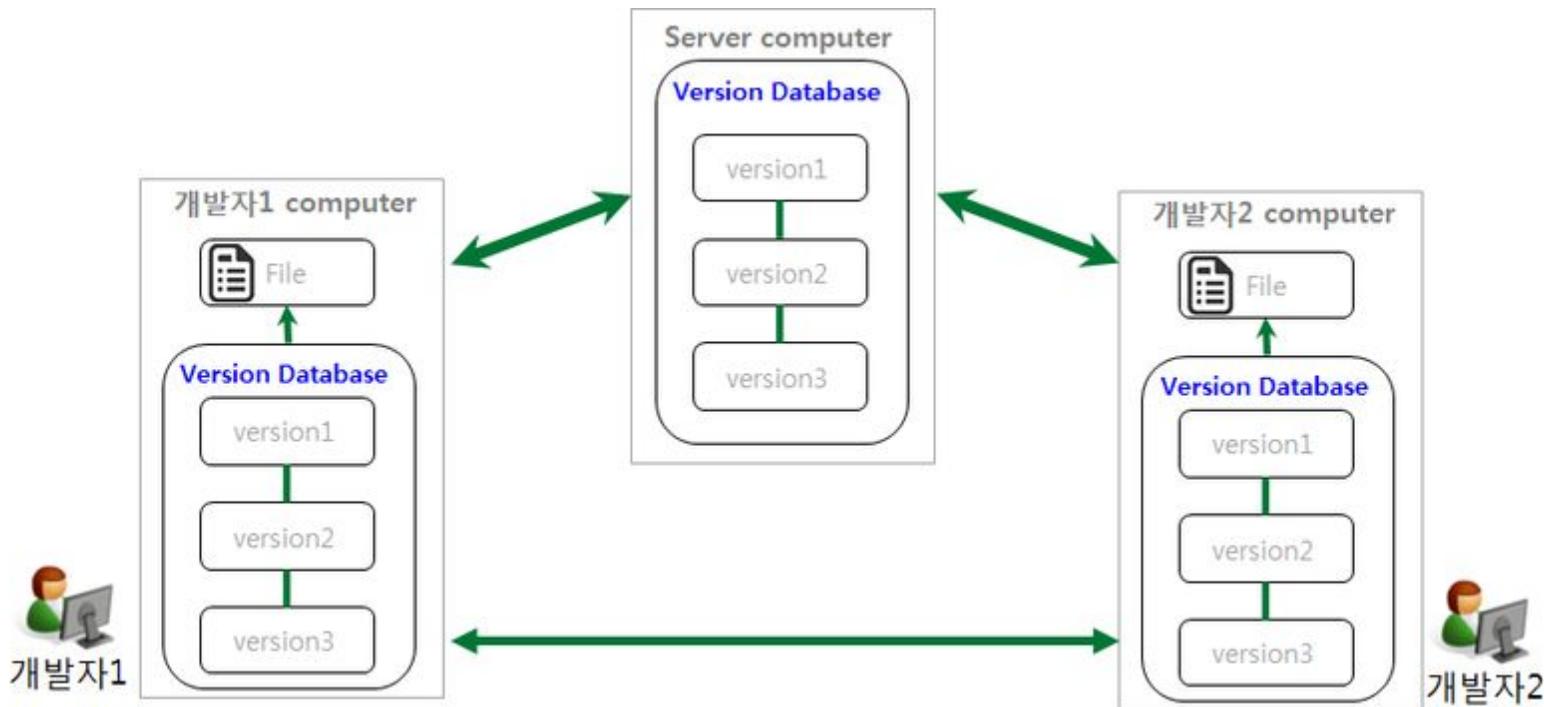
집중형 버전 관리 시스템 - CVCS(Centralized VCS)

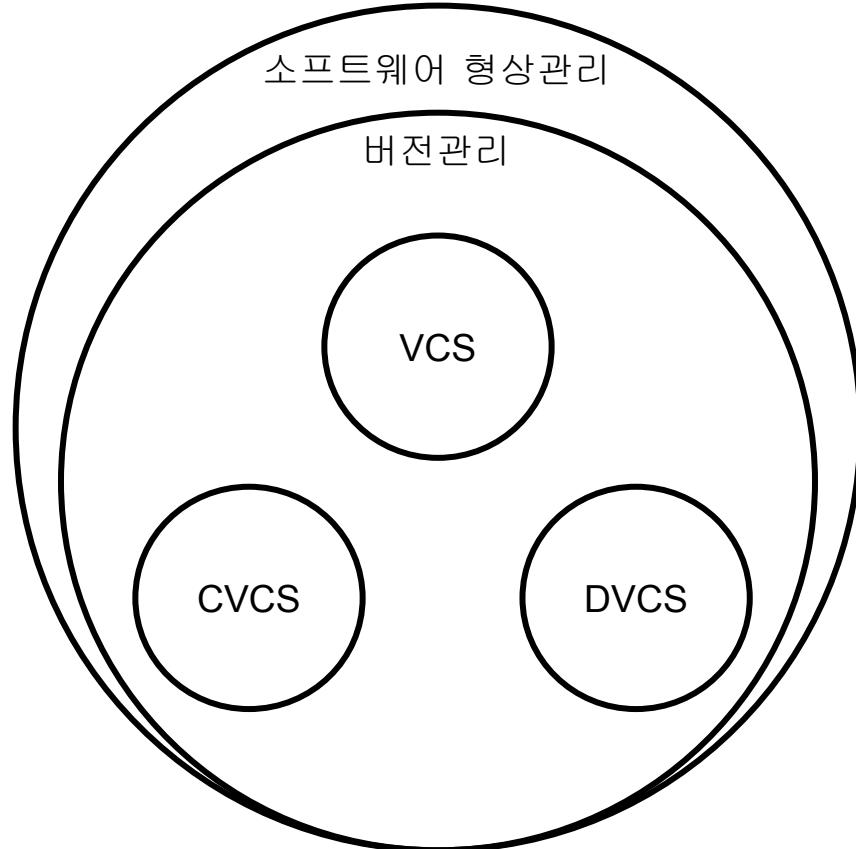
- 하나의 서버를 두고 서버를 중심으로 모든 자원을 관리
- SVN - <https://subversion.apache.org/>

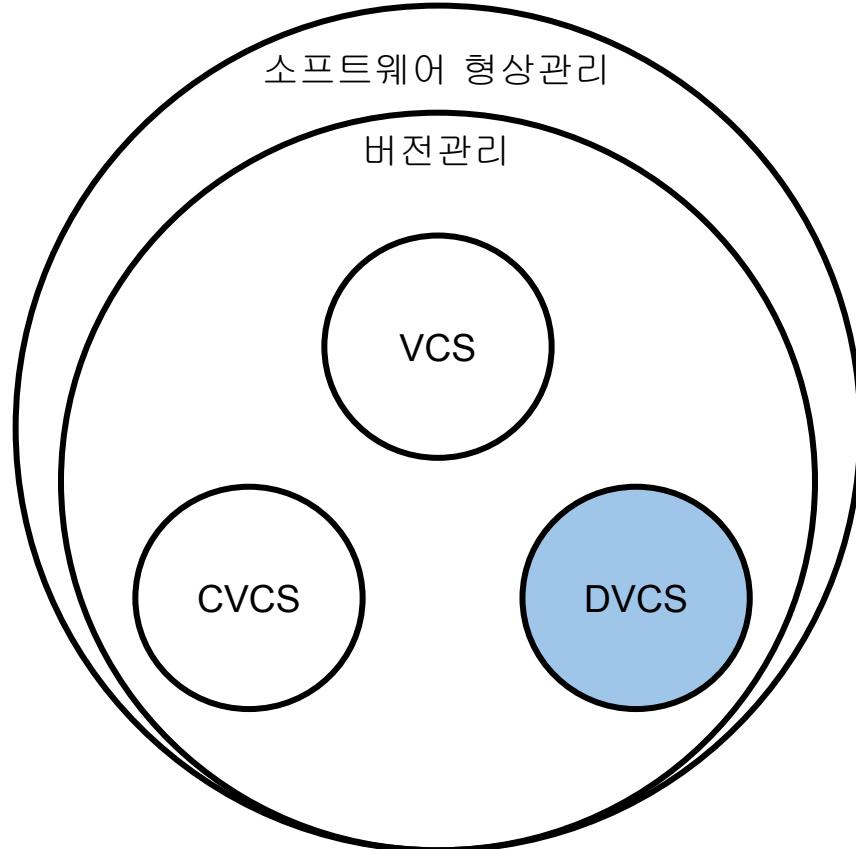


분산형 버전 관리 시스템 - DVCS(Distributed VCS)

- CVCS처럼 서버를 두고 서버에서 자원 관리를 하지만 로컬에서도 자원 관리를 함
- Git - <https://github.com/git/git>







Git과 GitHub

자바 /
자바스크립트

노트 / 노트북

바나나 우유
/ 바나나향 우유

자바 /
자바스크립트

노트 / 노트북

바나나 우유
/ 바나나향 우유

비슷하지만 다르다

Git과 Github

Git : 버전 관리 시스템

Github : Git 호스팅 사이트

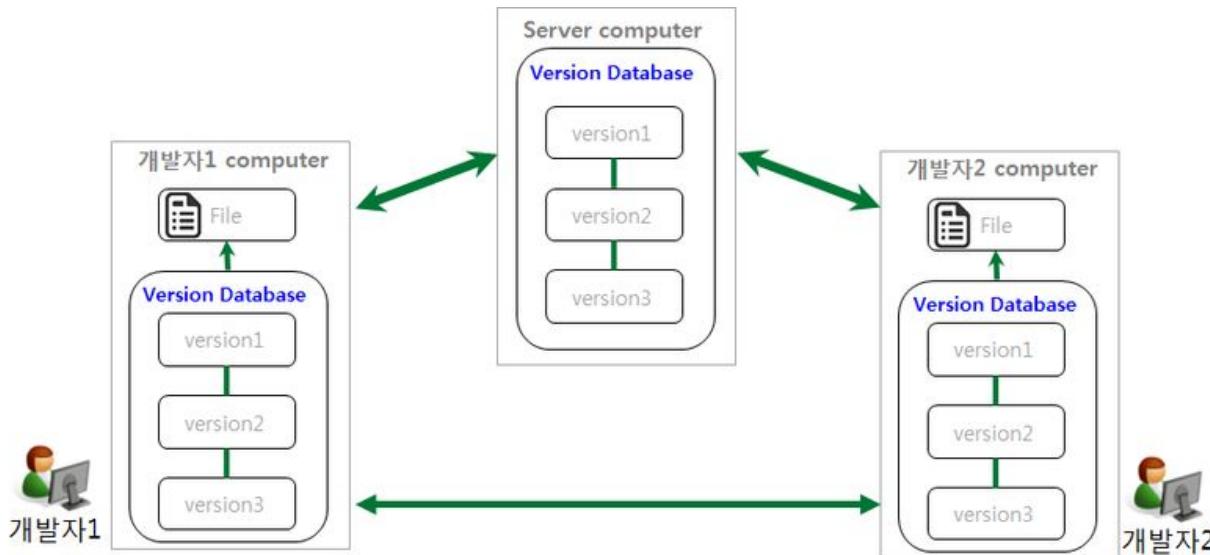
- 호스팅 : 웹의 공간을 빌려주거나 빌리는 것

Git과 Github

Git : 버전 관리 시스템

Github : Git 호스팅 사이트

- 호스팅 : 웹의 공간을 빌려주거나 빌리는 것



Git hosting

GitHub (<https://github.com/>)

GitLab (<https://gitlab.com>)

BitBucket (<https://bitbucket.org>)

Git Client

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

Git bash

- Git을 설치하면 기본제공되는 [CLI](#) 기반 Git Client
- Windows / Mac / Linux

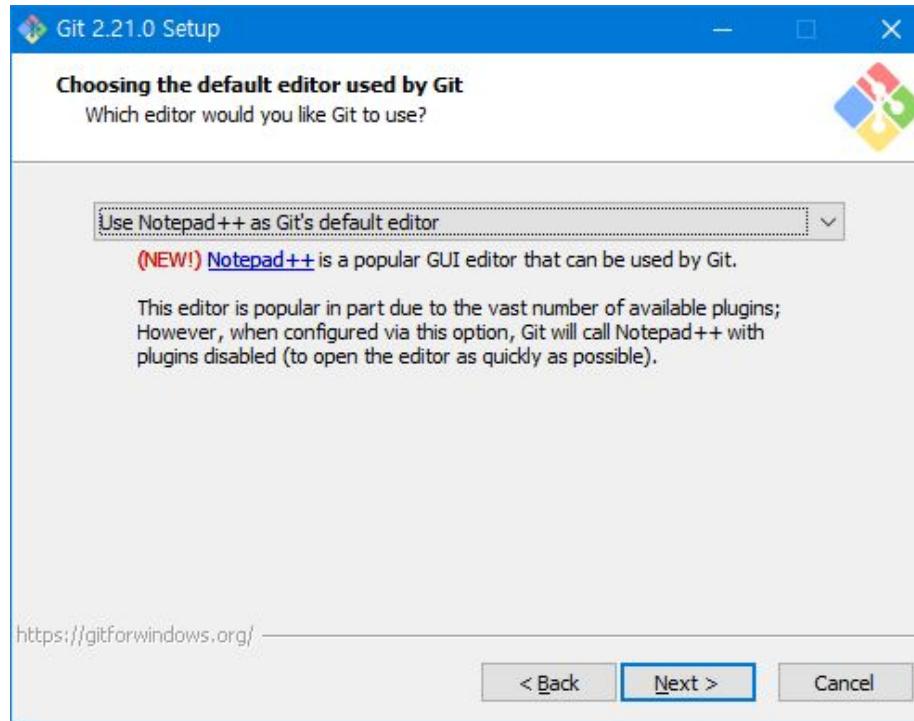
SourceTree

- <https://www.sourcetreeapp.com/>
- Windows / Mac

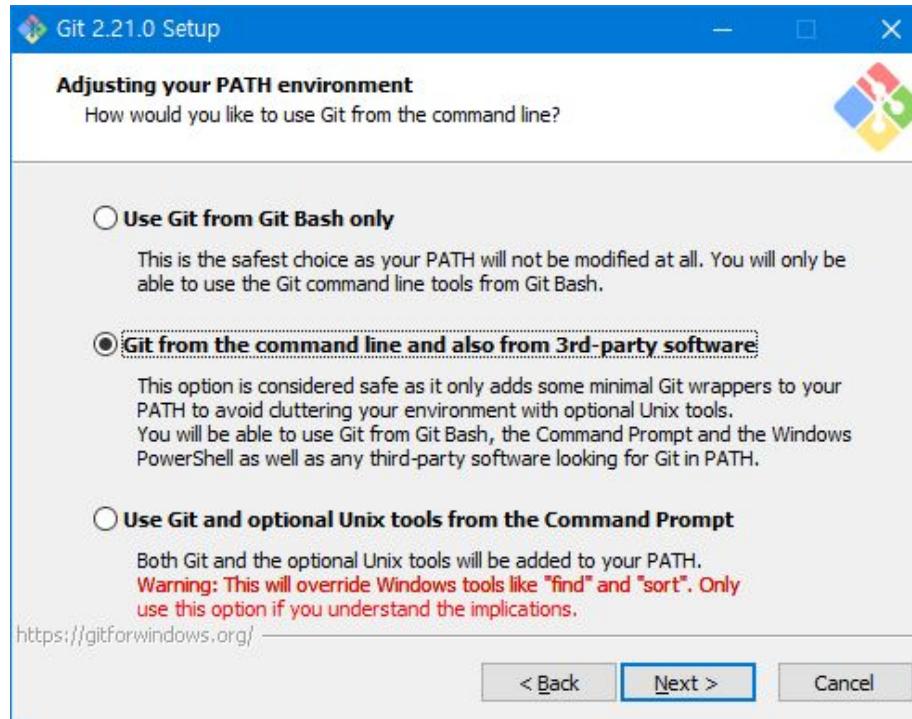
GitKraken

- <https://www.gitkraken.com/>
- Windows / Mac / Linux

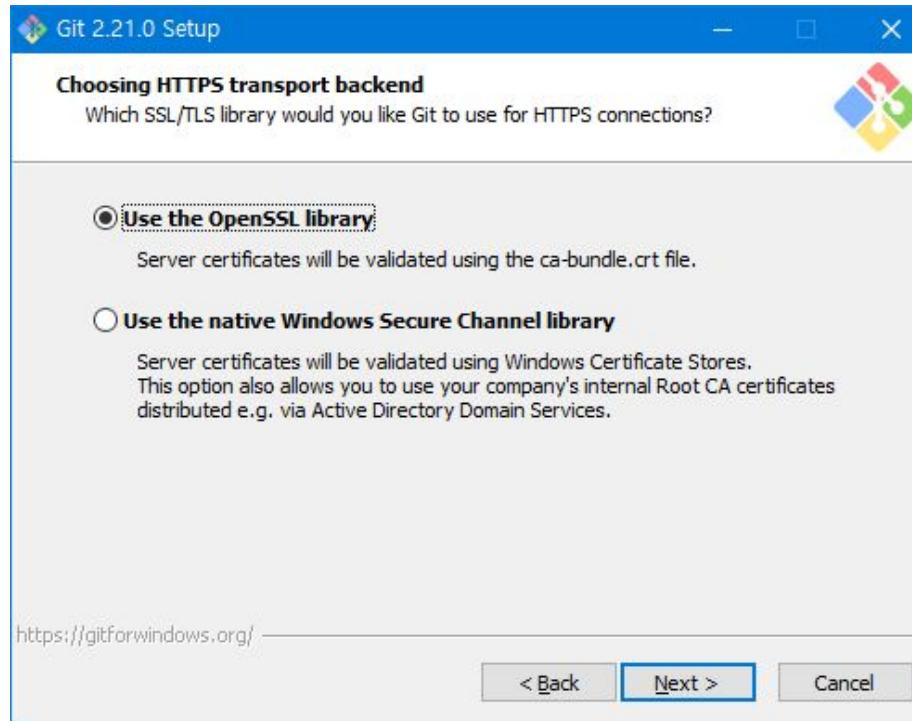
Git 설치



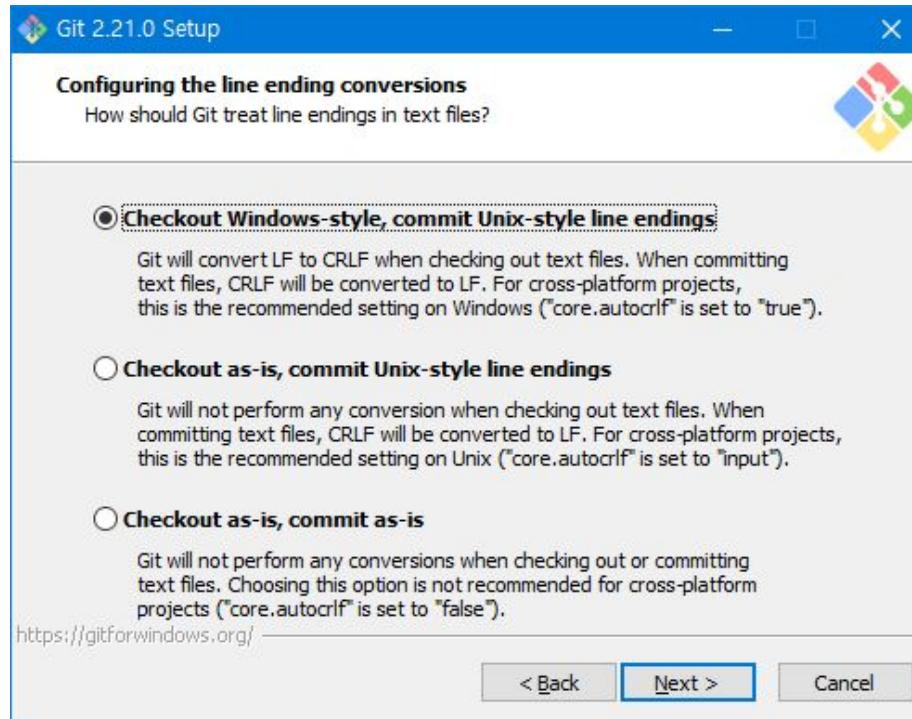
Git 설치



Git 설치

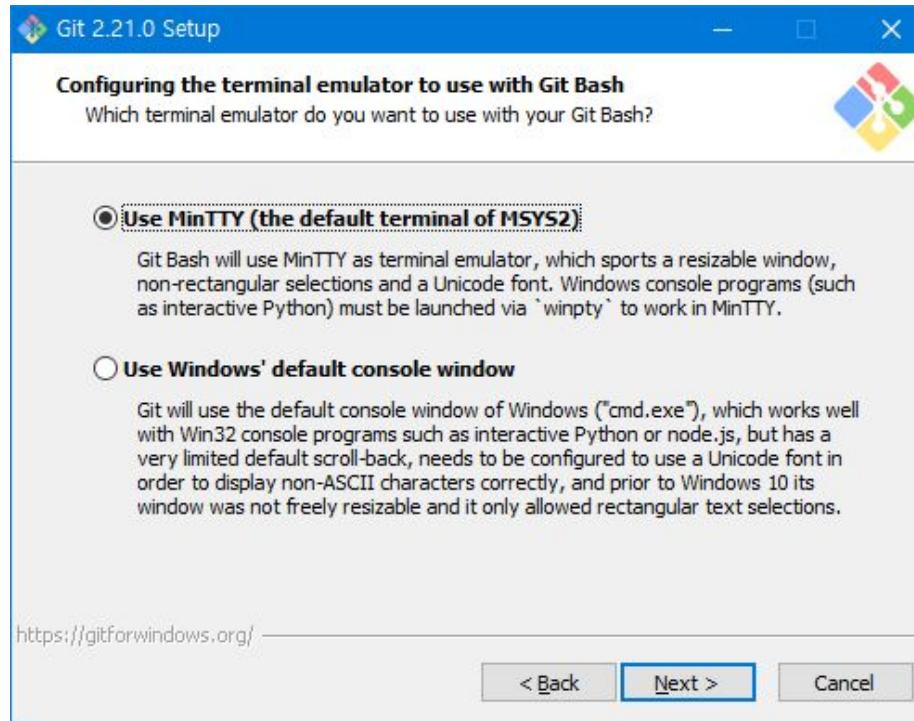


Git 설치

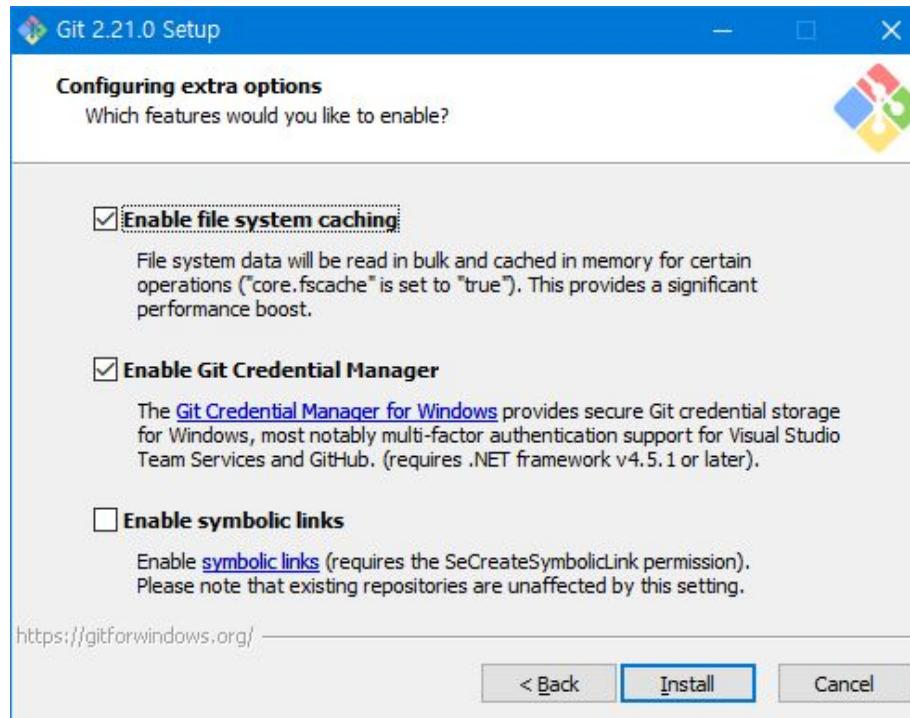


CRLF와 LF의 차이 : [링크](#)

Git 설치

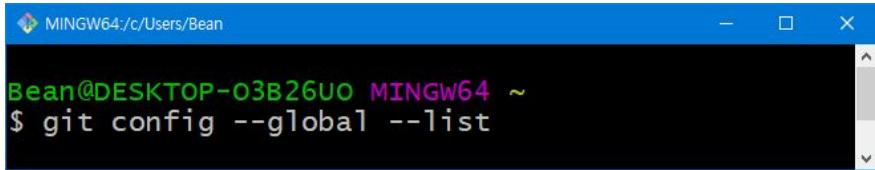


Git 설치



Git 설치

```
git config --global --list
```



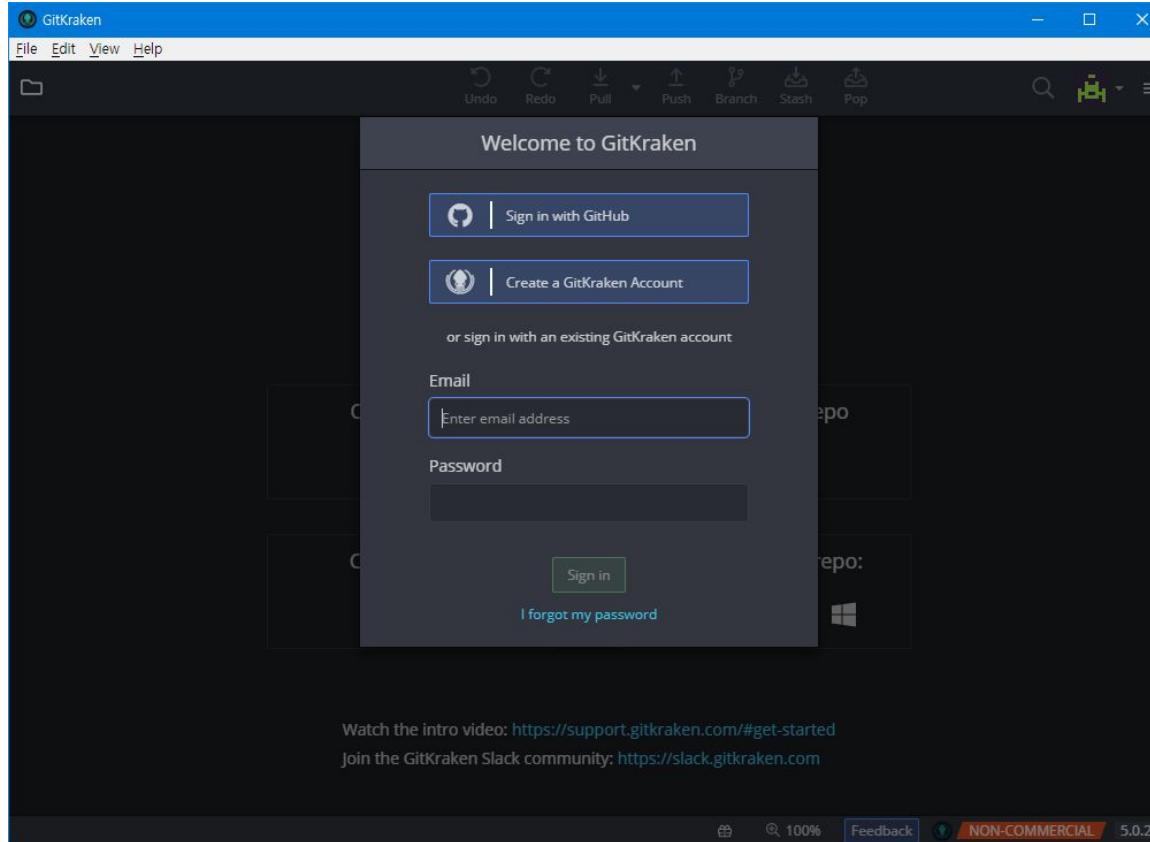
A screenshot of a terminal window titled "MINGW64:/c/Users/Bean". The window shows the command \$ git config --global --list being run. The output is empty, indicating no global configuration files were found.

이전에 git을 사용한 적이 있다면 사용자 정보를 지워줍니다

```
git config --global --unset-all user.name
```

```
git config --global --unset-all user.email
```

Git Kraken 설치



정리

버전관리 ⊂ 형상관리

Git은 분산형 버전 관리 시스템(DVCS)

Git ≠ GitHub

- GitHub는 Git Hosting 웹 서비스

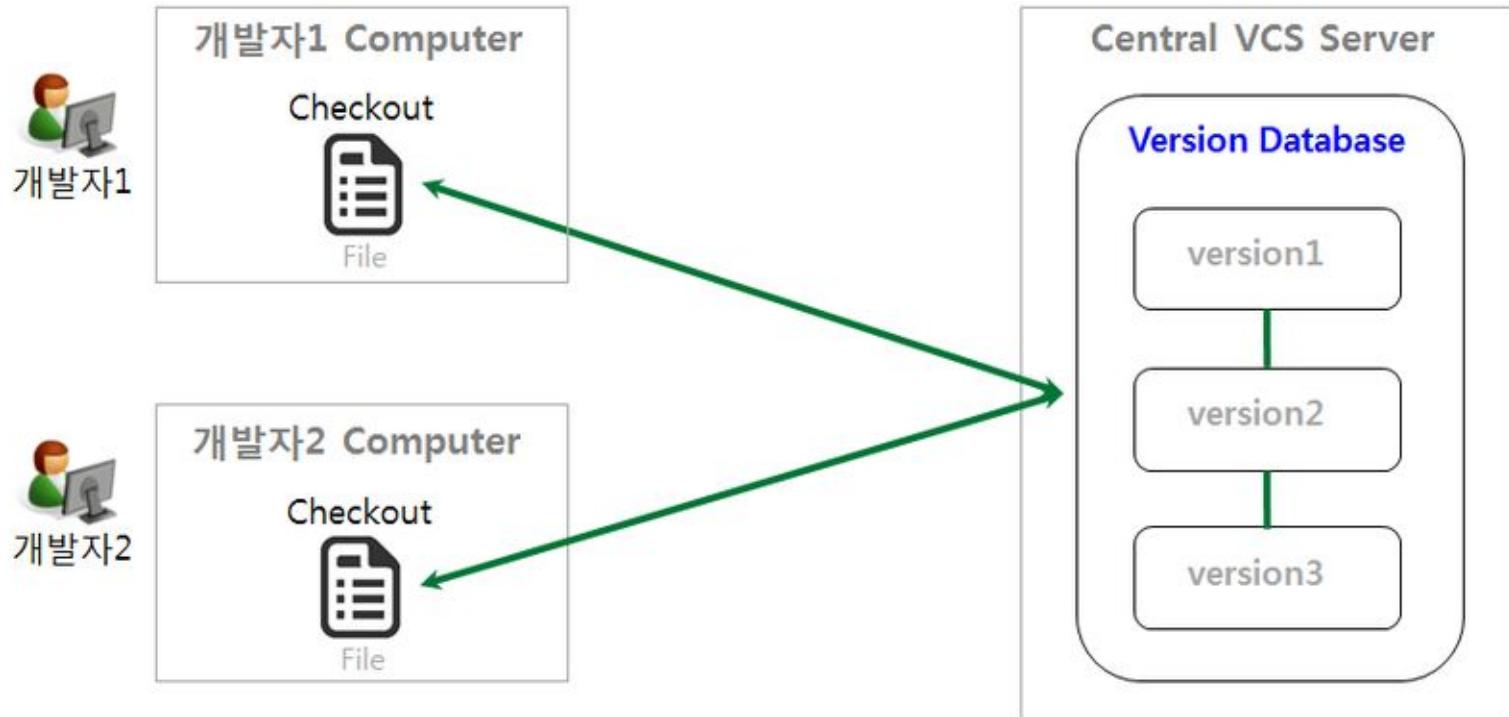
버전관리를 어떻게 할까 - CVCS

나	기준	다른사람	2 way merge
A	A		
B	B	B	
1	C	2	
	D	D	

버전관리를 어떻게 할까 - CVCS

나	기준	다른사람	2 way merge
A	A		?
B	B	B	B
1	C	2	?
	D	D	?

버전관리를 어떻게 할까 - CVCS



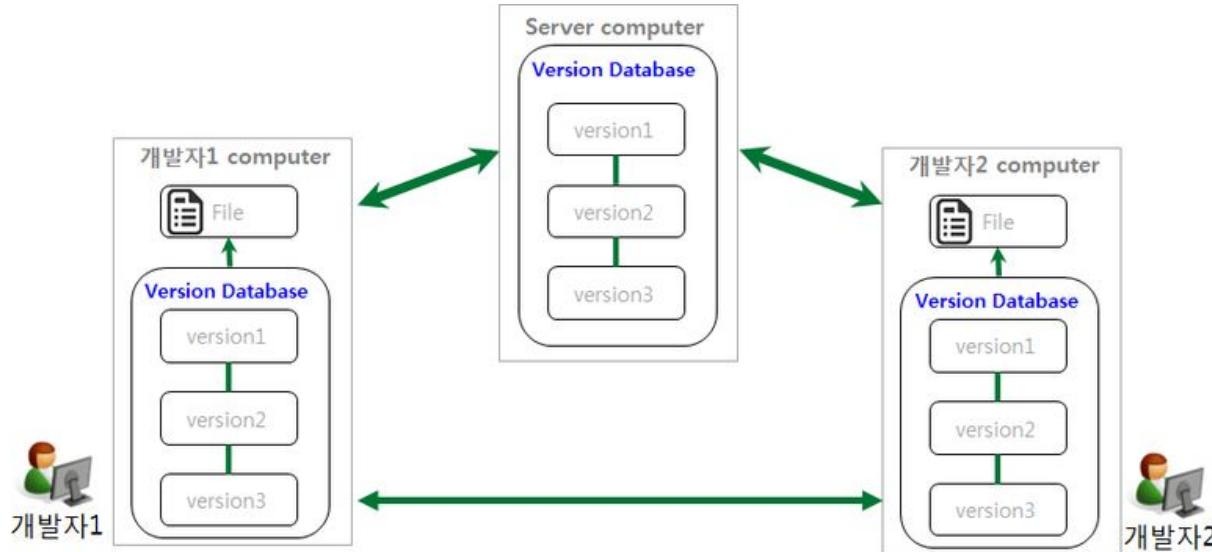
버전관리를 어떻게 할까 - DVCS

나	기준	다른사람	3 way merge
A	A		
B	B	B	
1	C	2	
	D	D	

버전관리를 어떻게 할까 - DVCS

나	기준	다른사람	3 way merge
A	A		
B	B	B	B
1	C	2	?
	D	D	

버전관리를 어떻게 할까 - DVCS

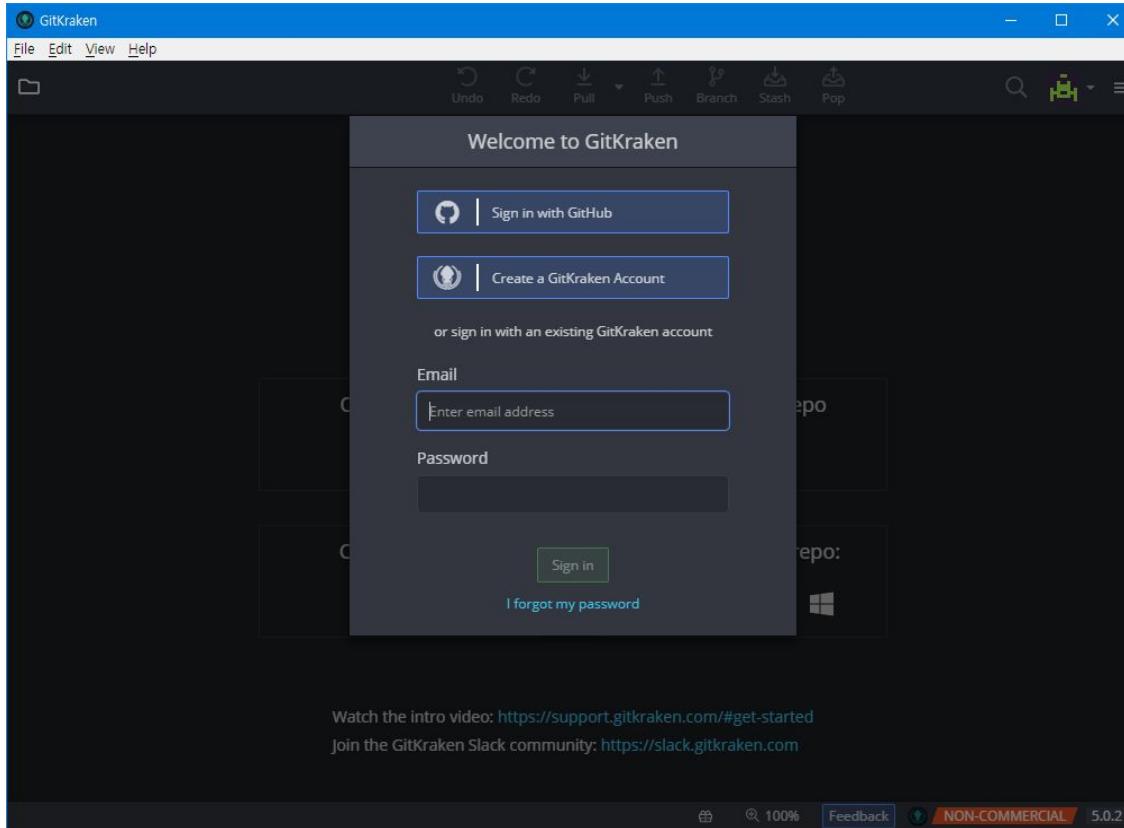


Git으로 버전관리를 어떻게 할까

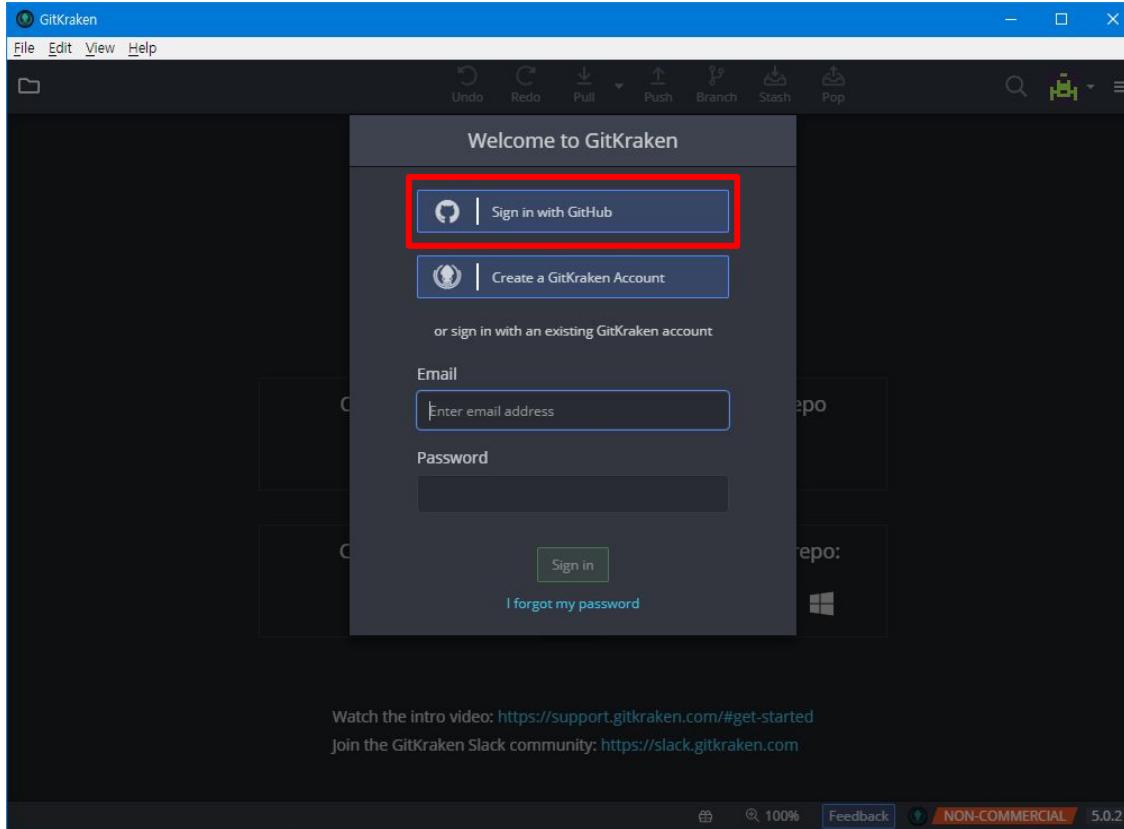
나	기준	다른사람	2 way merge	3 way merge
A	A		?	
B	B	B	B	B
1	C	2	?	?
	D	D	?	

Git을 이용한 버전 관리

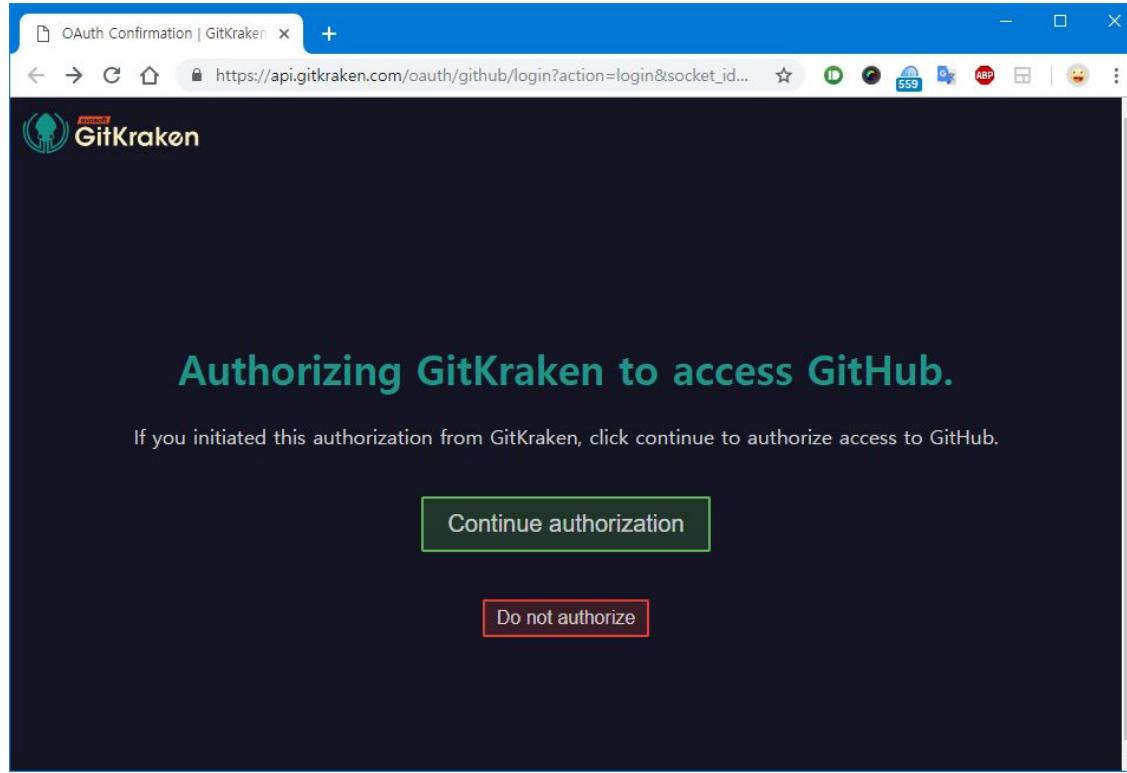
Git Kraken 설치



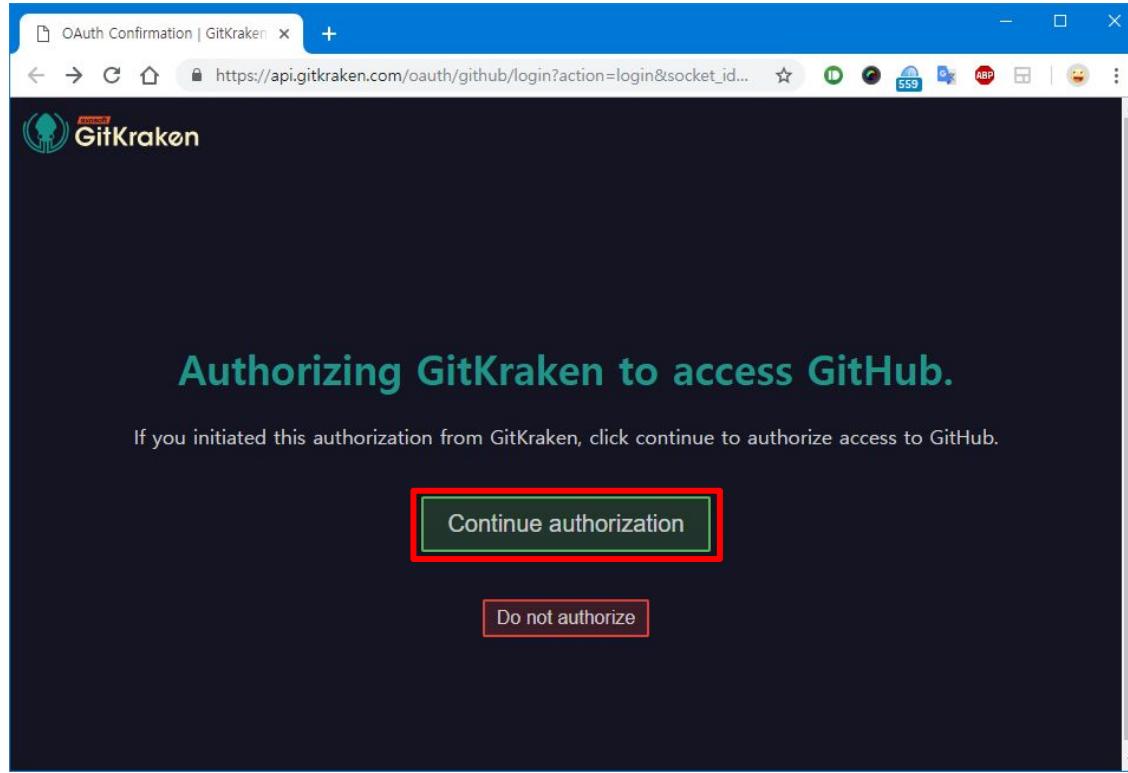
Git Kraken 설치

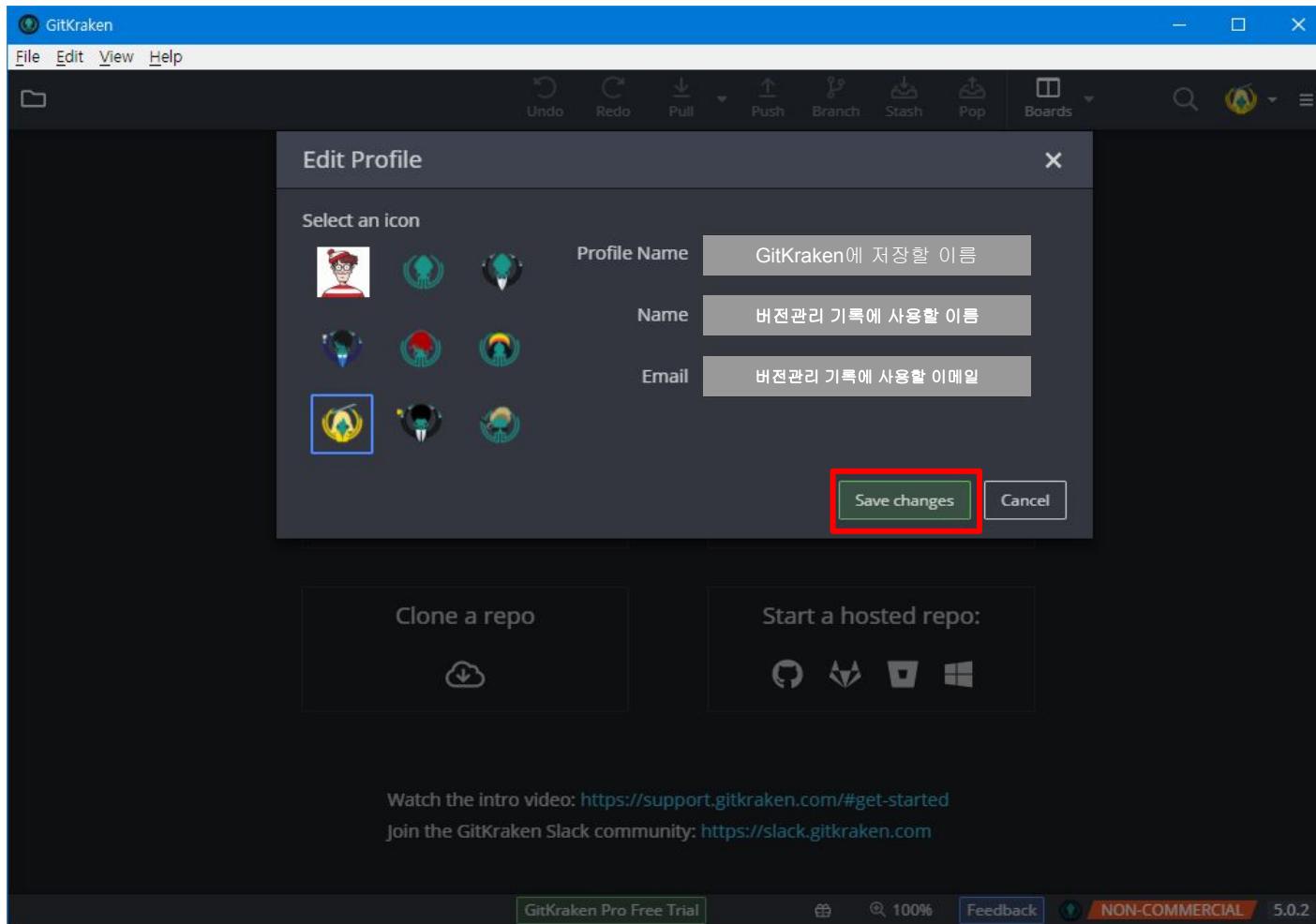


Git Kraken 설치

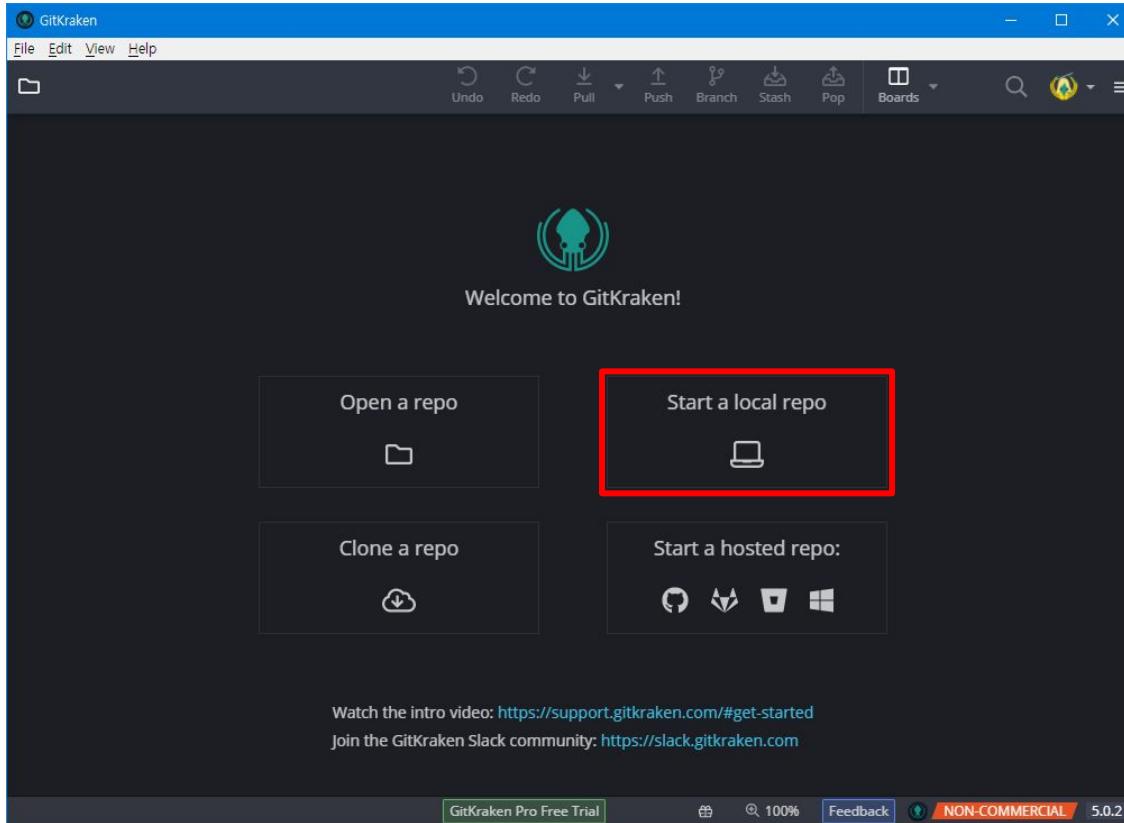


Git Kraken 설치

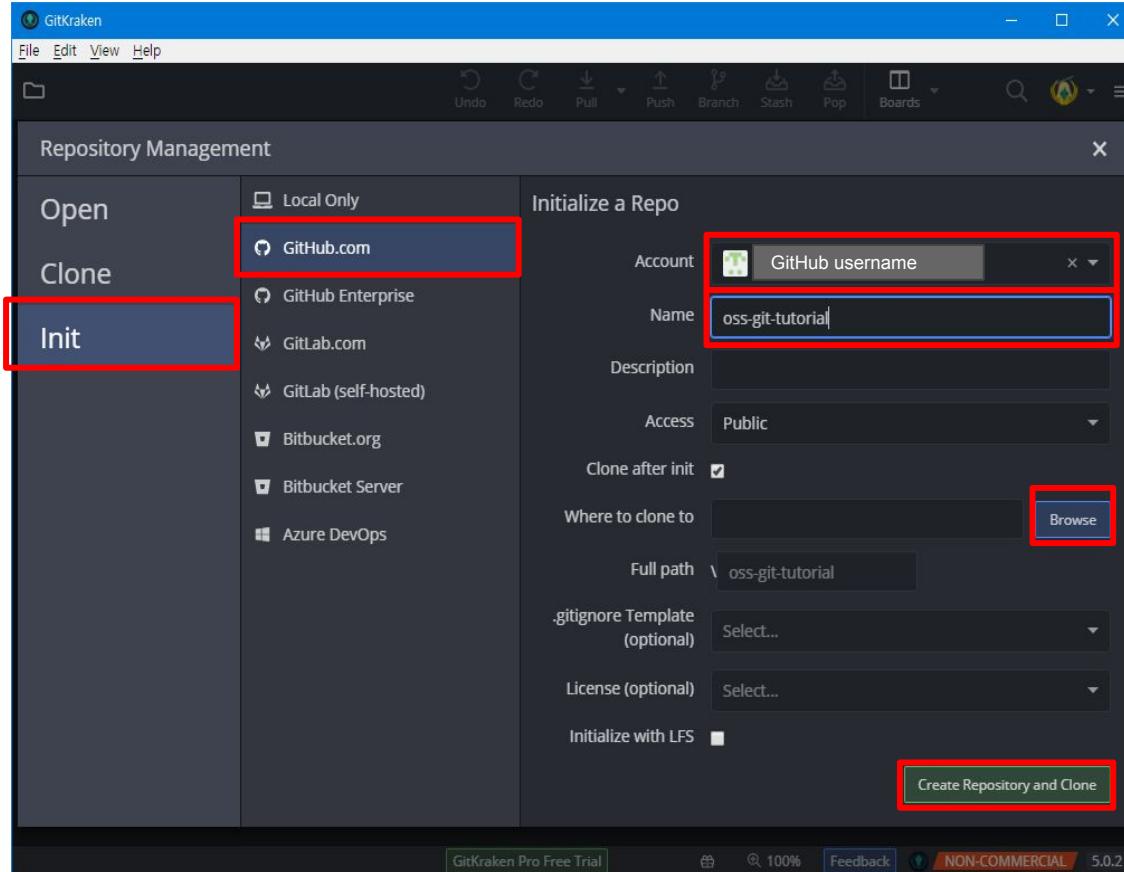




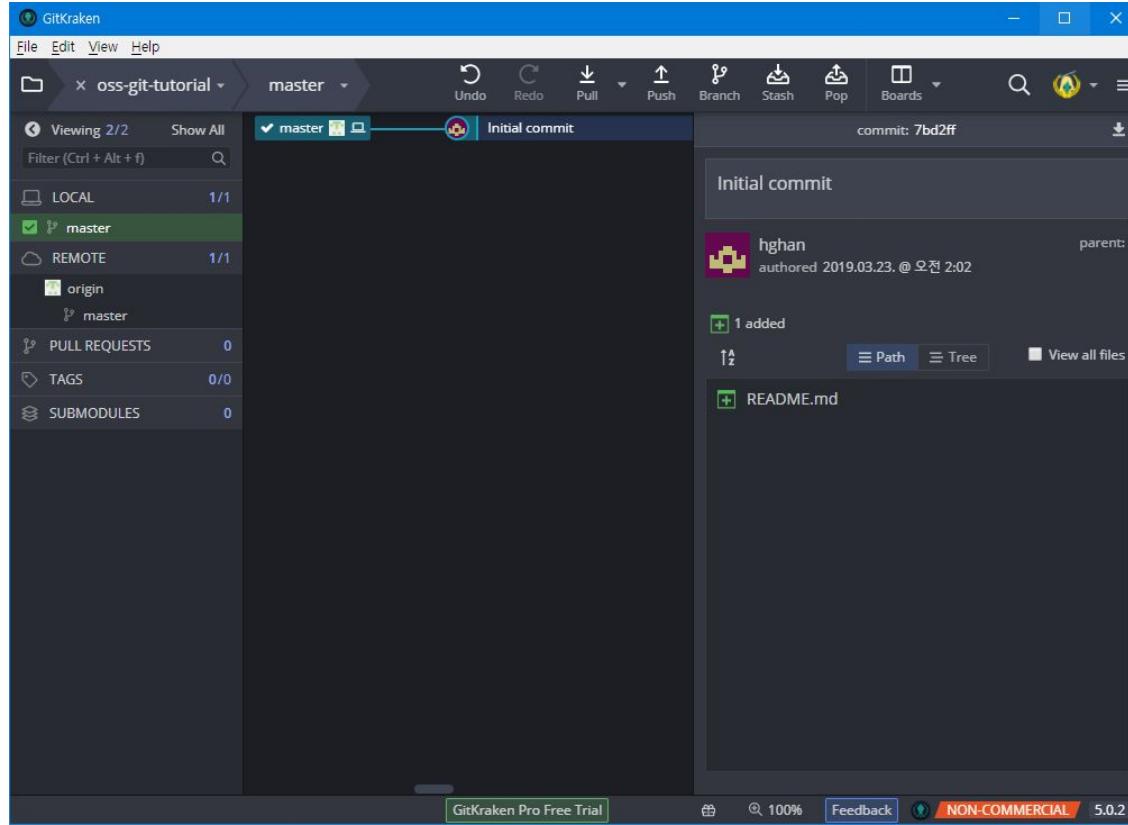
Git Kraken 설치



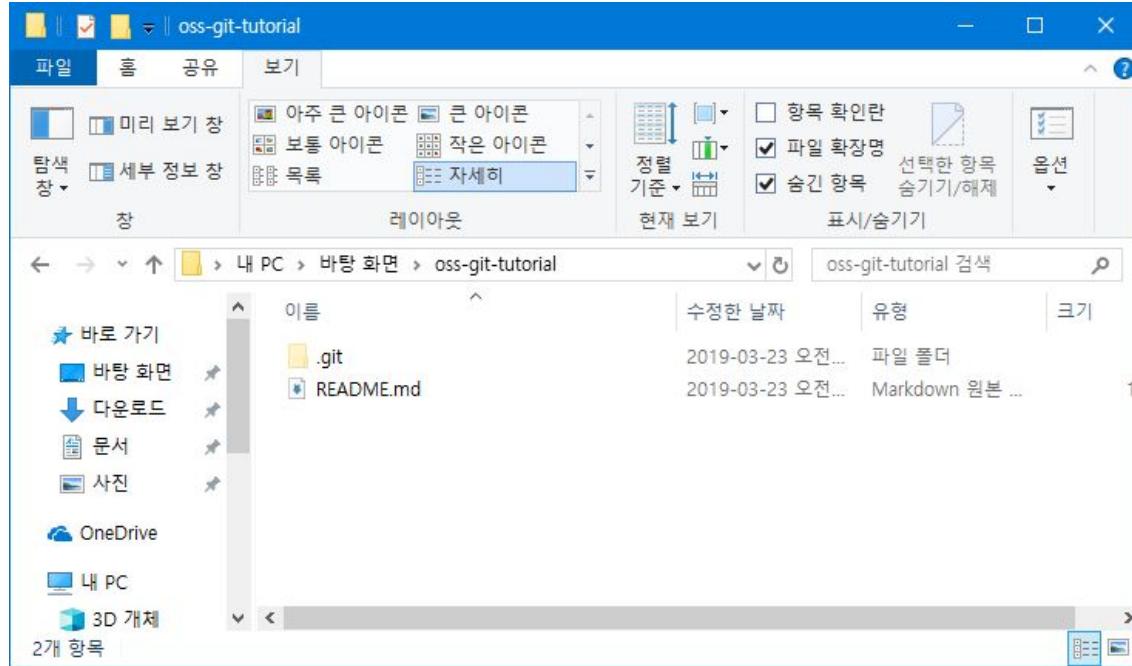
git init - 작업 경로내의 변화를 추적하기 위한 시작 명령



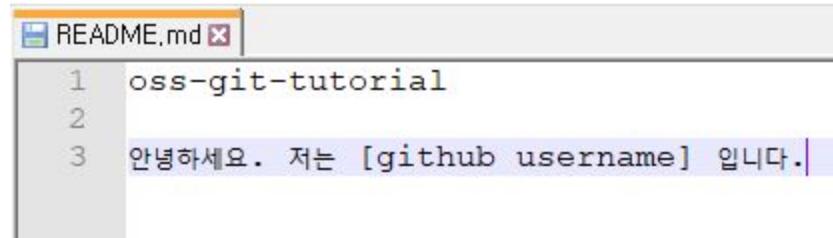
git init - 작업 경로내의 변화를 추적하기 위한 시작 명령



git init - 작업 경로내의 변화를 추적하기 위한 시작 명령



첫번째 commit 준비

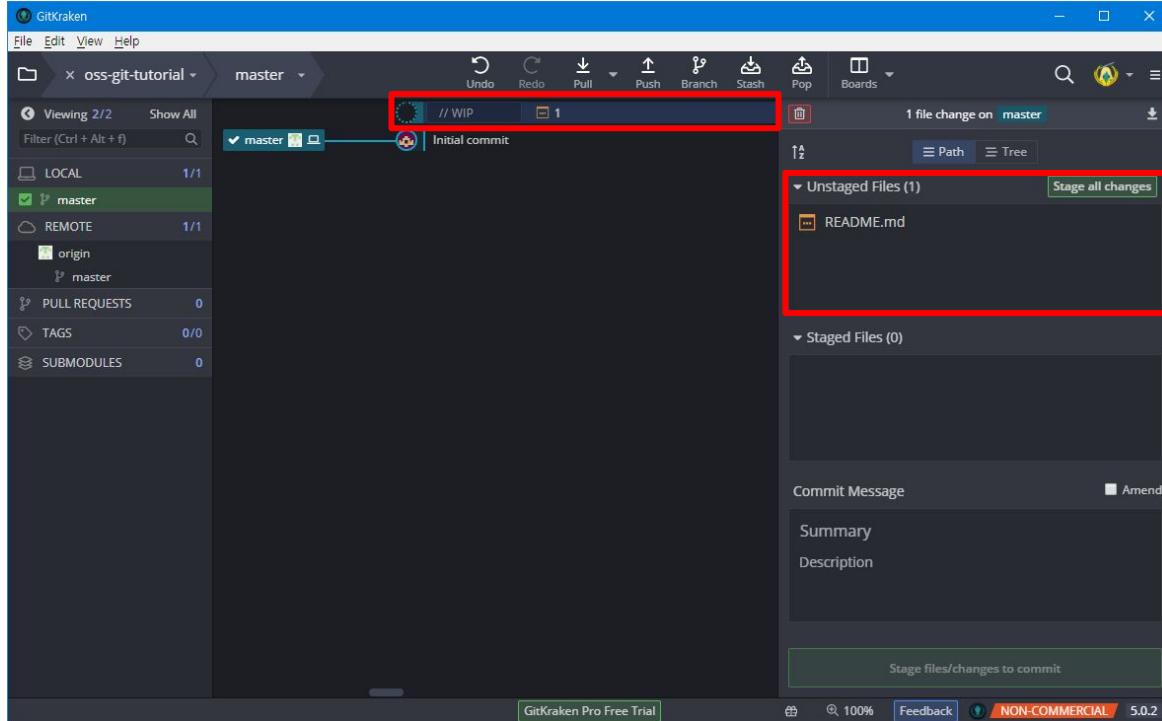


A screenshot of a code editor window titled "README.md". The content of the file is as follows:

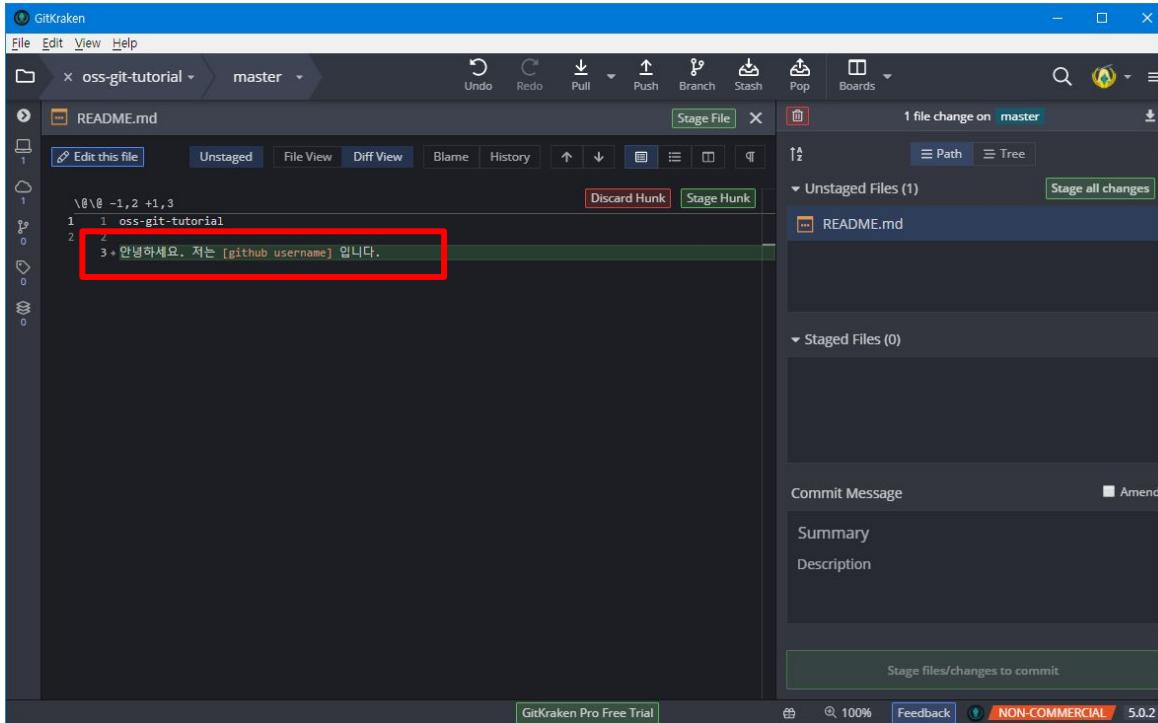
```
1 oss-git-tutorial
2
3 안녕하세요. 저는 [github username] 입니다.
```

The third line, "안녕하세요. 저는 [github username] 입니다.", is highlighted with a light blue background and a purple selection bar on the right side of the text area.

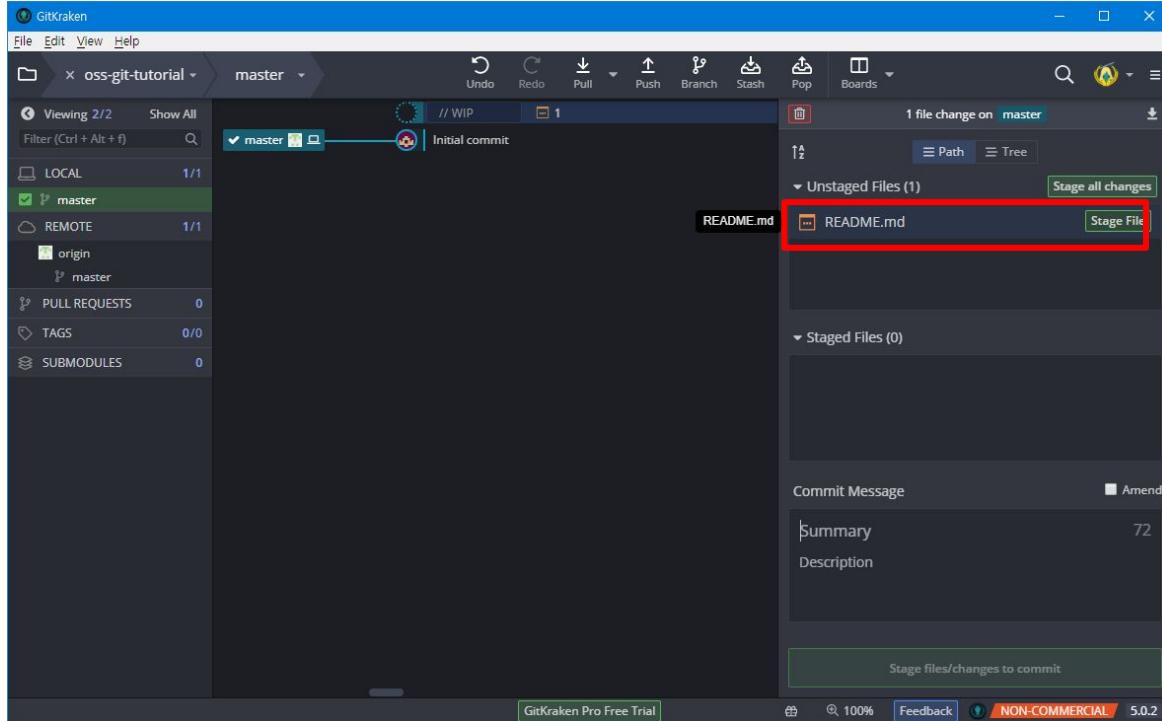
git add - staging area으로 파일 등록



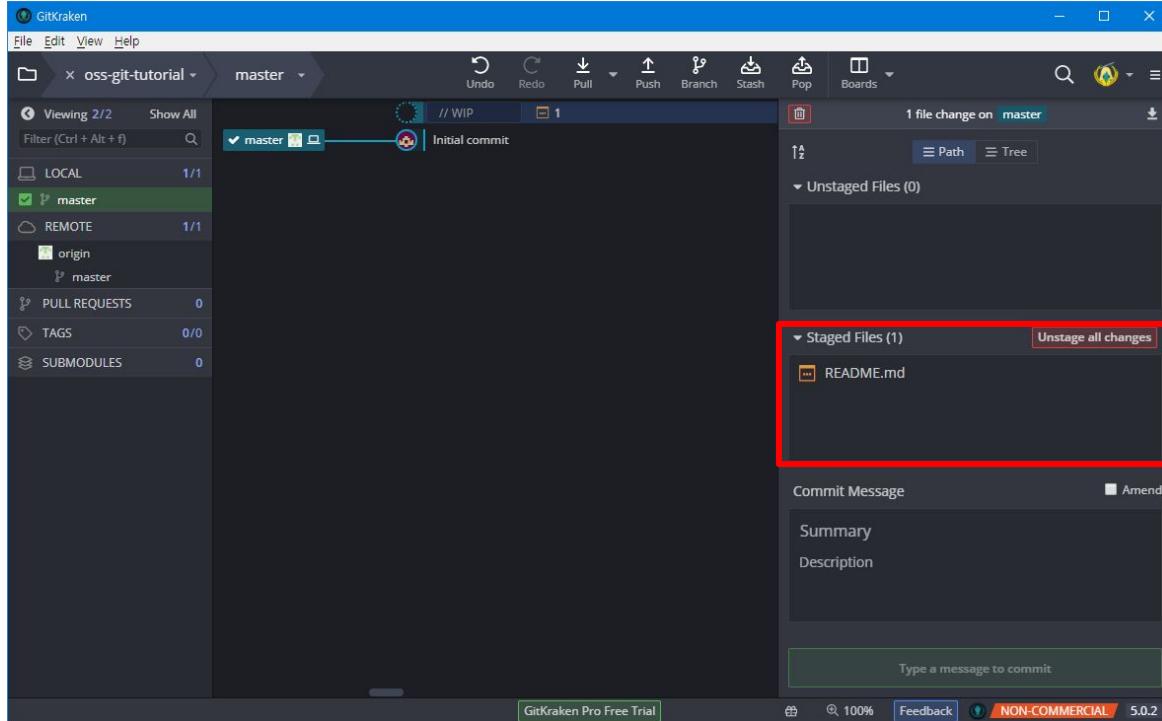
git add - staging area으로 파일 등록



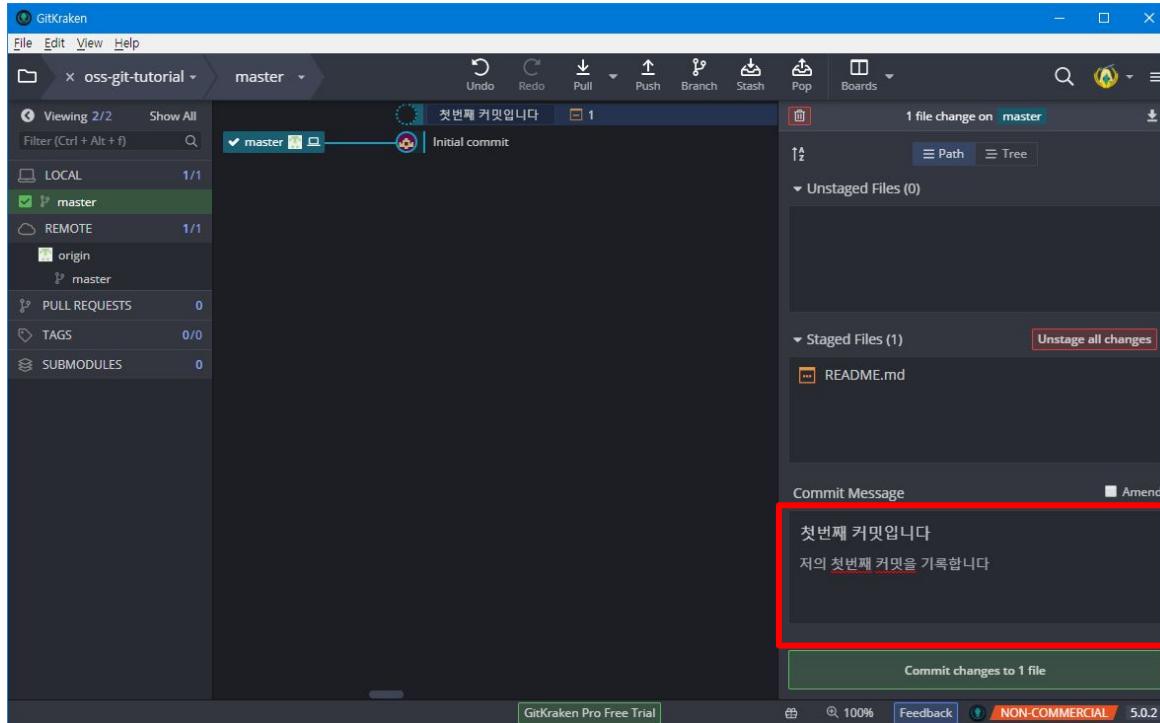
git add - staging area으로 파일 등록



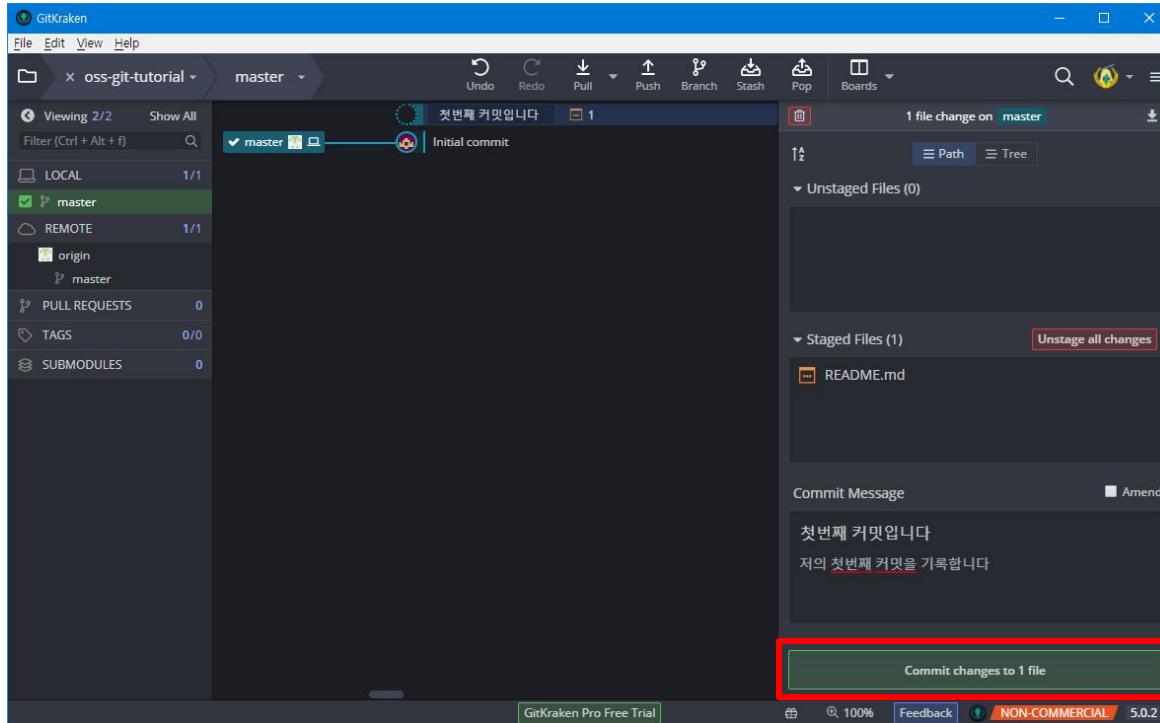
git add - staging area으로 파일 등록



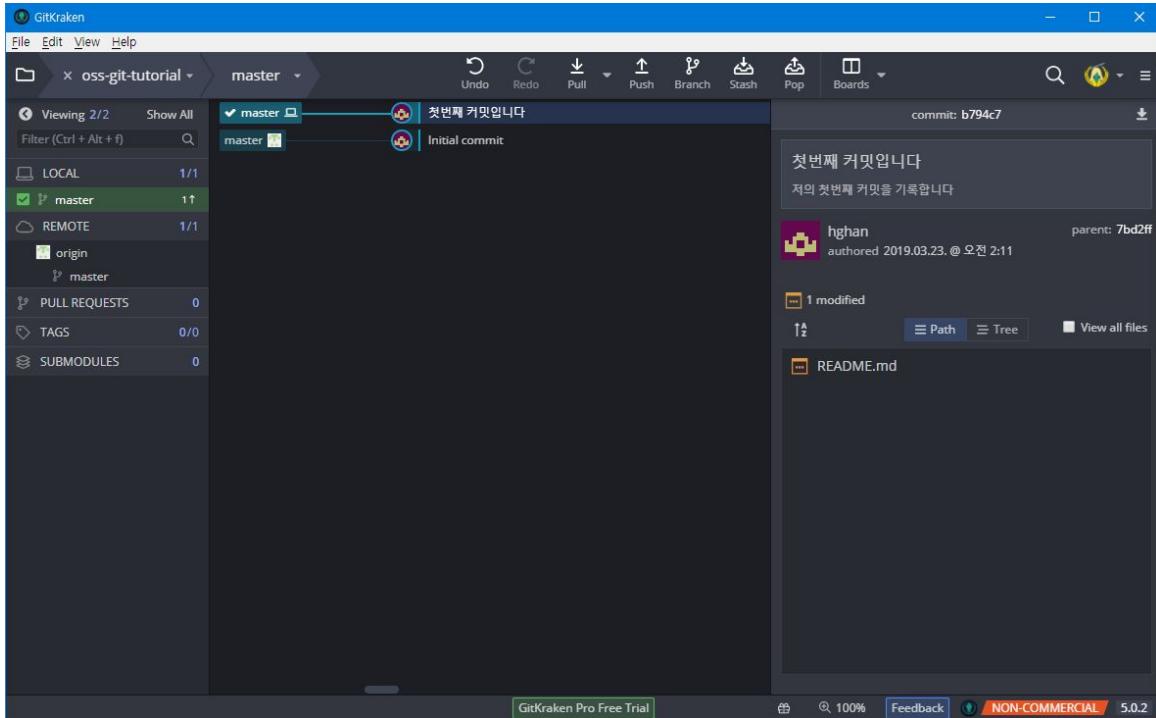
git commit - 확정된 기록을 남기기



git commit - 확정된 기록을 남기기



git commit - 확정된 기록을 남기기



0|션

README.md 파일에 아래 내용을 추가 하세요.

단, [이메일] 부분에는 본인의 이메일을 입력하고 저장하세요.

‘제 이메일 주소는 [이메일] 입니다.’

예) 제 이메일 주소는 hanbit@gmail.com 입니다.

그 후 commit을 해보세요. 커밋 제목, 커밋 내용은 자유롭게 써보세요.

git의 흐름(workflow)

working directory

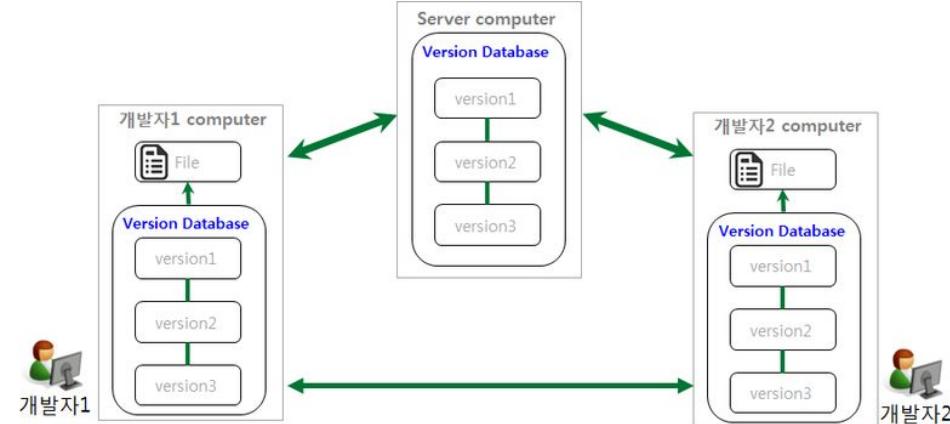
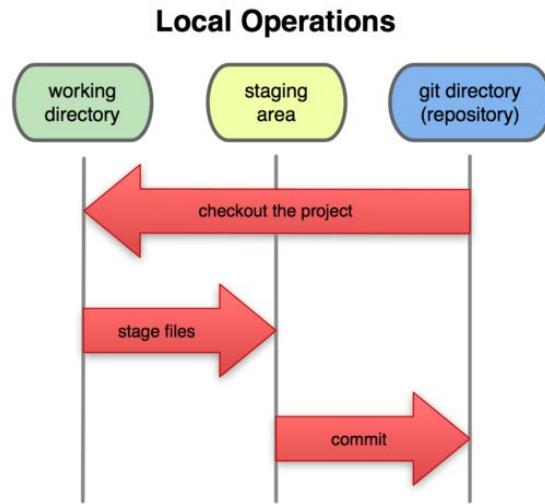
- 현재 작업 디렉토리

staging area

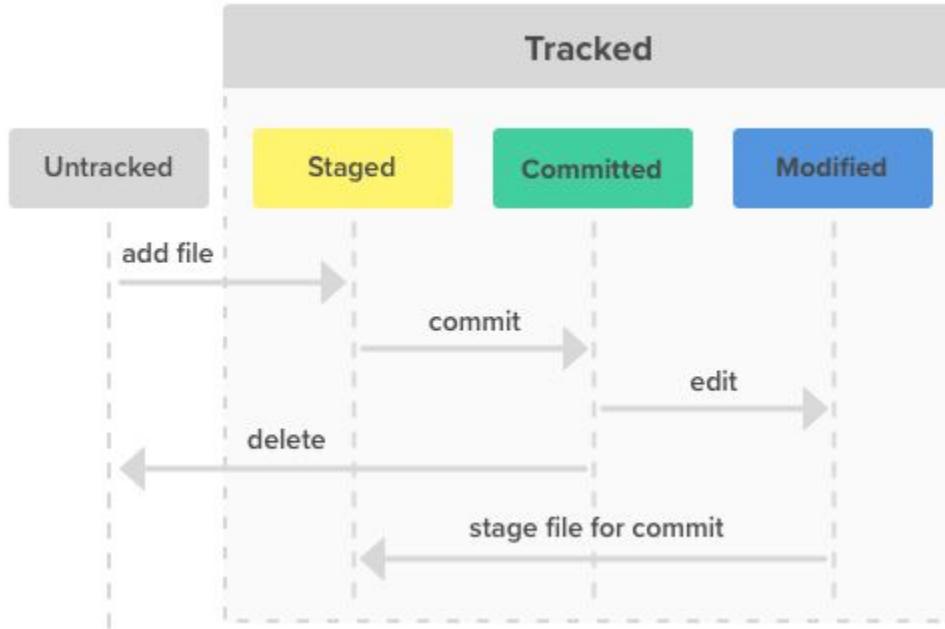
- 커밋시 반영되는 파일이 보관되어 있는 임시 공간

git directory

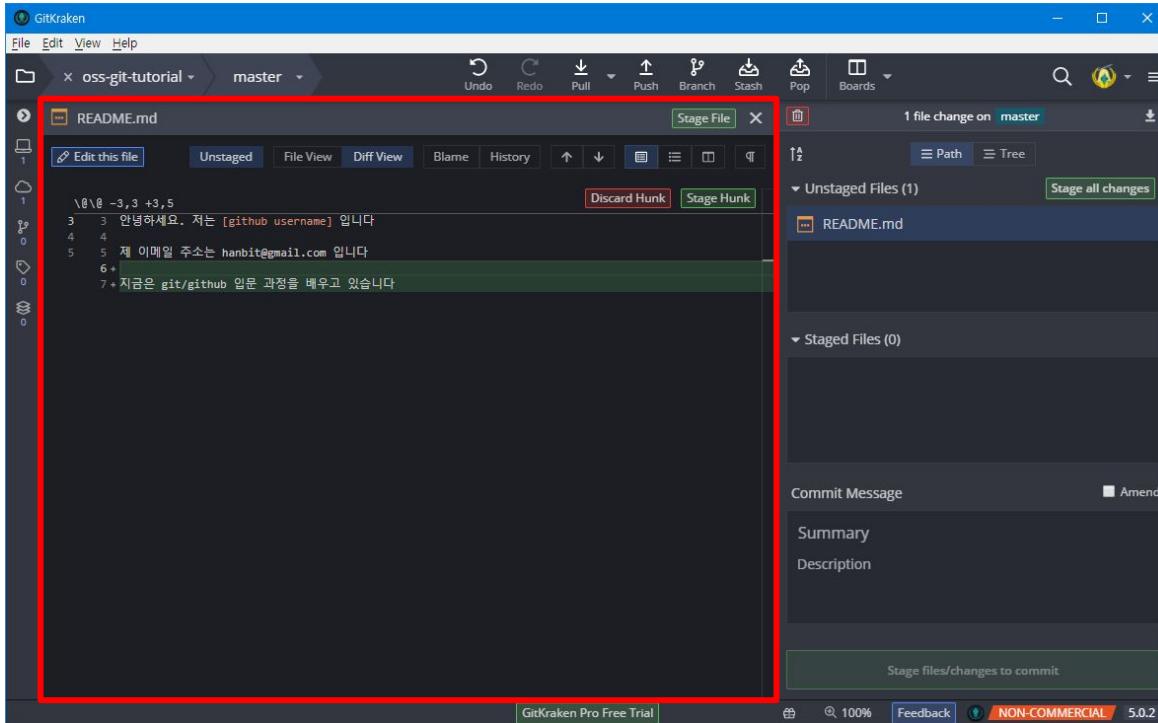
- 커밋 기록들이 모여있는 저장 공간



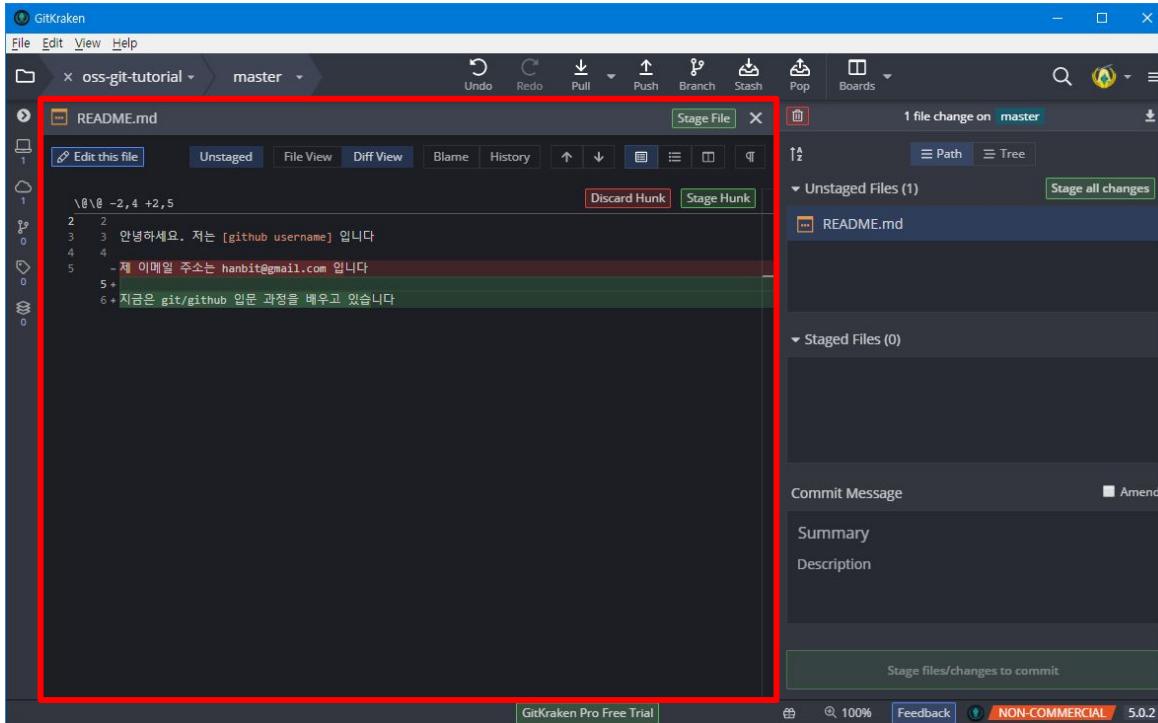
git의 흐름(workflow)



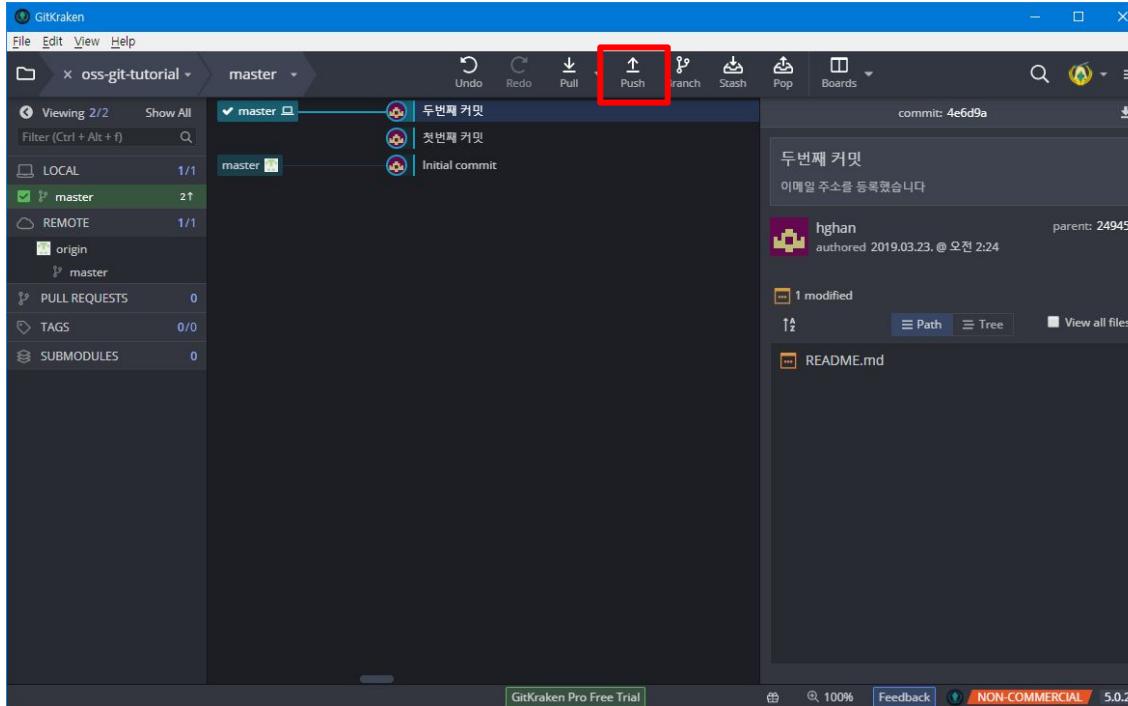
git diff - 파일의 커밋 기록을 비교해서 달라진 내용을 보여준다



git diff - 파일의 커밋 기록을 비교해서 달라진 내용을 보여준다



git push - remote repository로 commit들을 전달



No description, website, or topics provided.

Edit

Manage topics

3 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾



두번째 커밋 ...

Latest commit 4e6d9ac 22 minutes ago

README.md

두번째 커밋

22 minutes ago

README.md



oss-git-tutorial

안녕하세요. 저는 [github username] 입니다

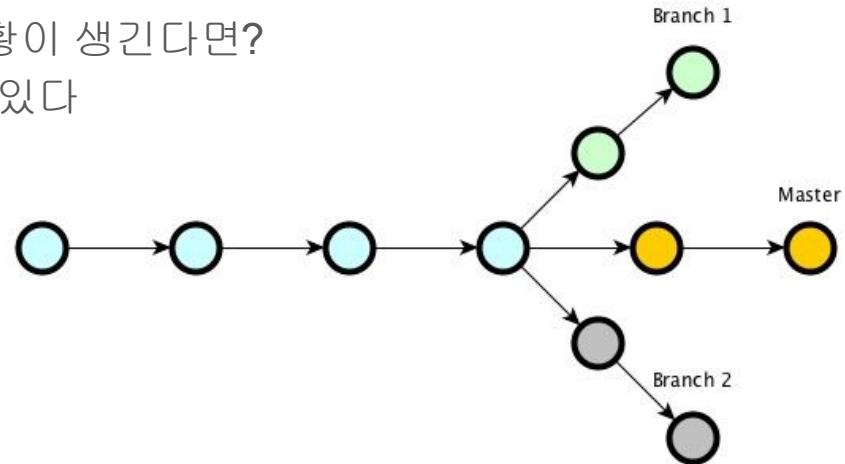
제 이메일 주소는 hanbit@gmail.com 입니다

git branch

- 특정 기준에 따라 commit들을 모아두는 공간
 - 추가 기능 개발, 버그 수정 등
- 하나의 파일에서 두가지 일을 해야 하는 상황이 생긴다면?
→ 두 작업을 하다가 작업내용이 섞일 수 있다

git branch

- 특정 기준에 따라 commit들을 모아두는 공간
 - 추가 기능 개발, 버그 수정 등
- 하나의 파일에서 두가지 일을 해야 하는 상황이 생긴다면?
→ 두 작업을 하다가 작업내용이 섞일 수 있다

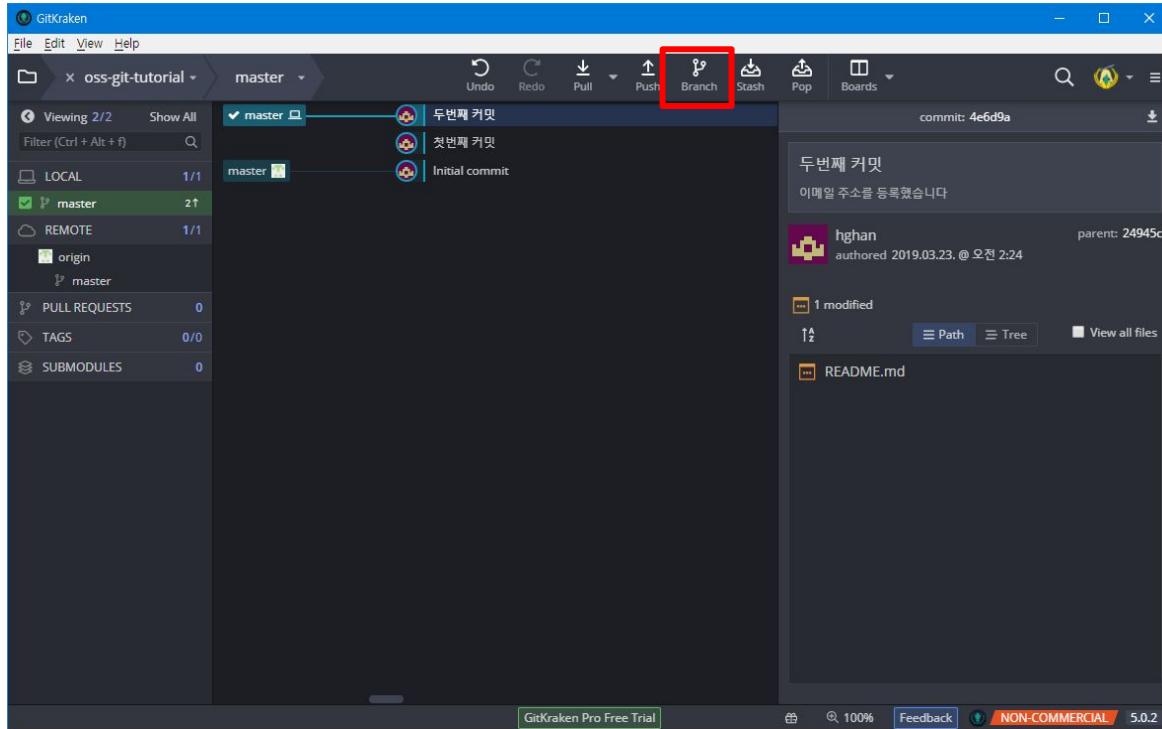


git checkout

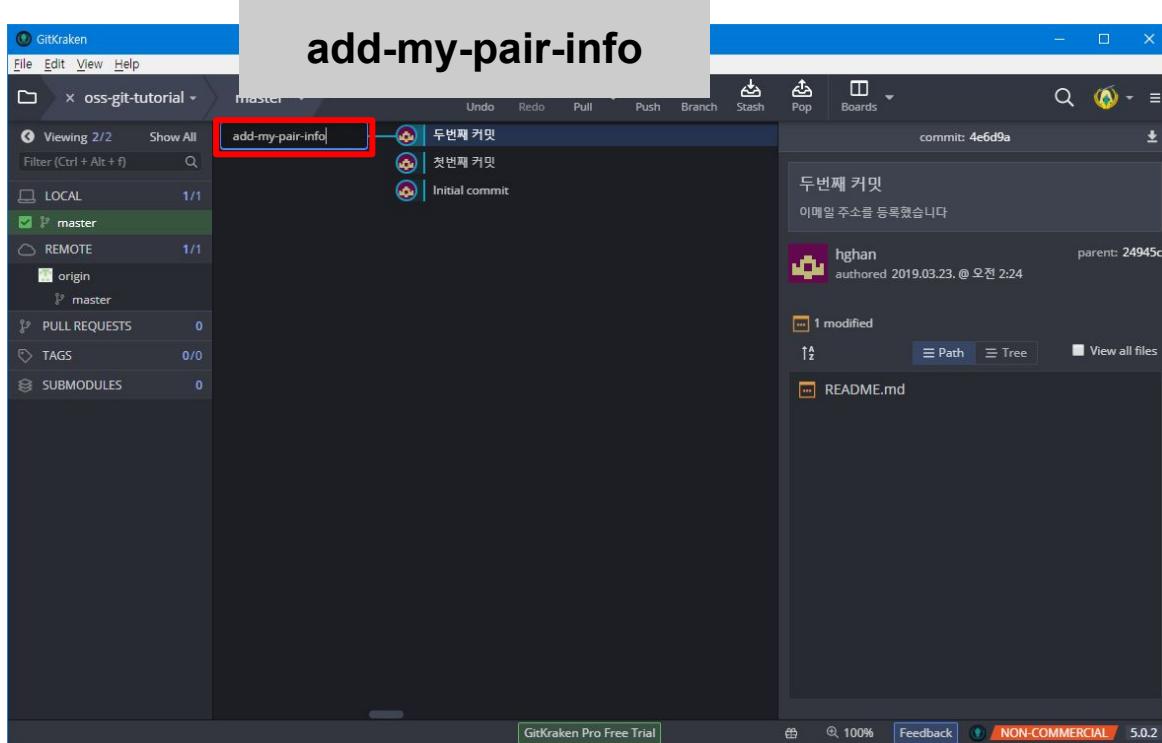
다른 branch로 이동하는 명령어

branch의 commit 이력에 따라 코드(메모)가 변경됨

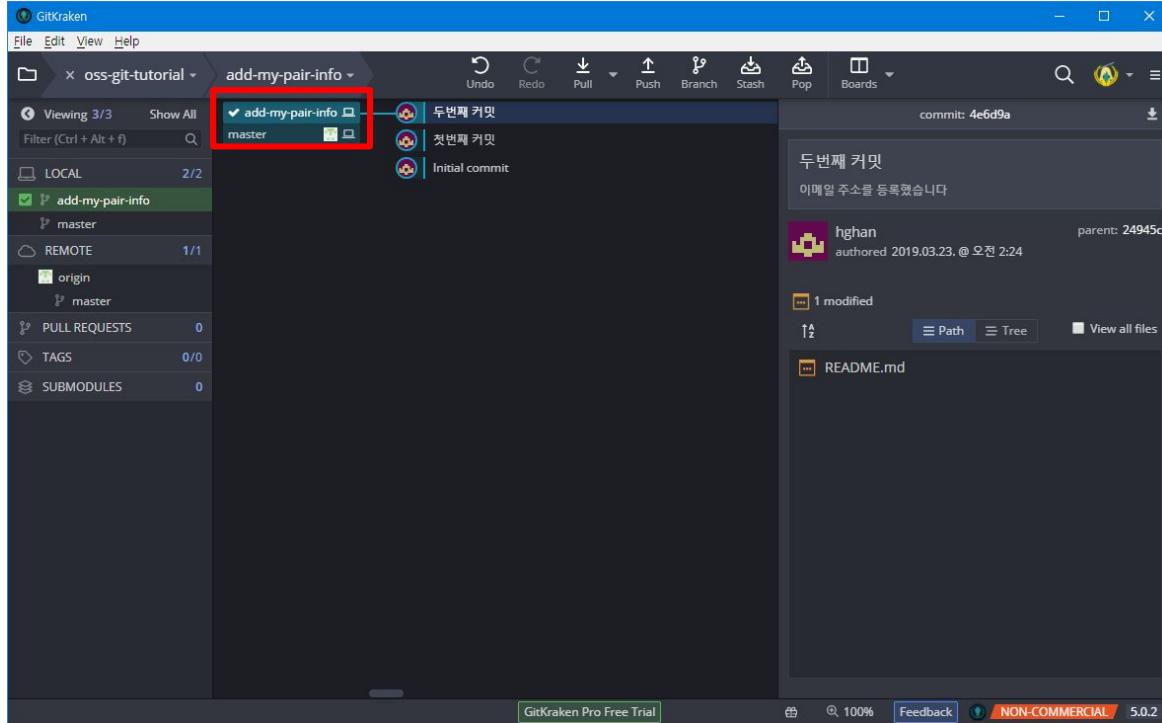
git branch



git branch



git branch



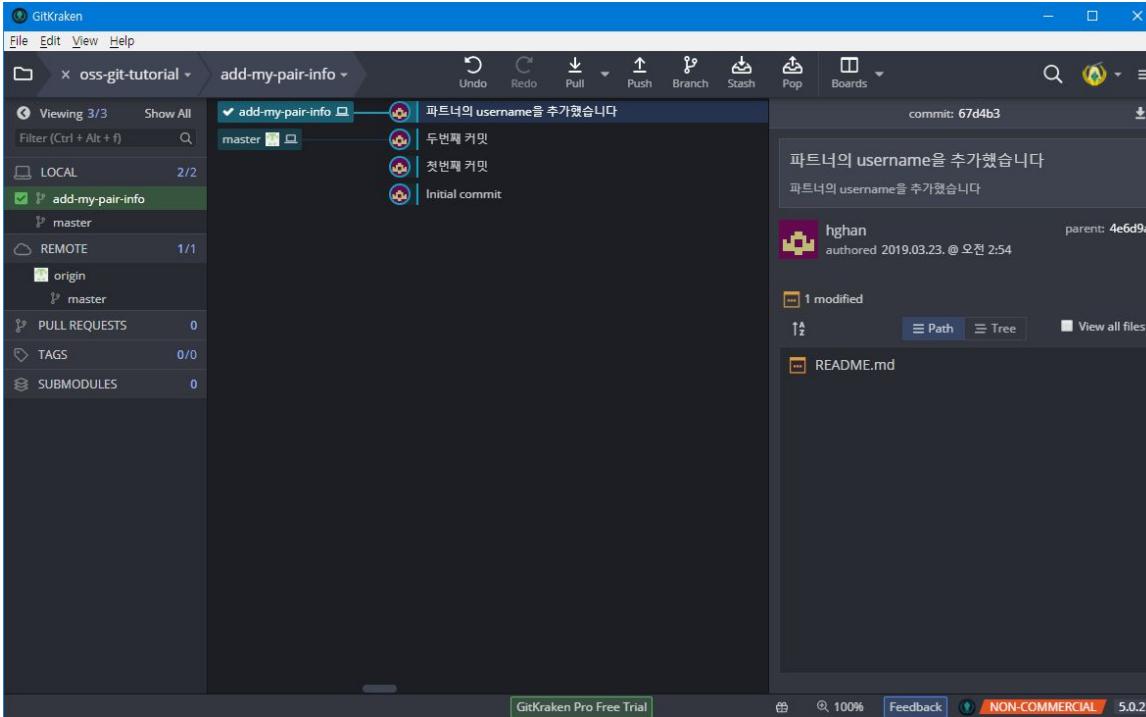
git branch - 새 브랜치(add-my-pair-info)에서 commit



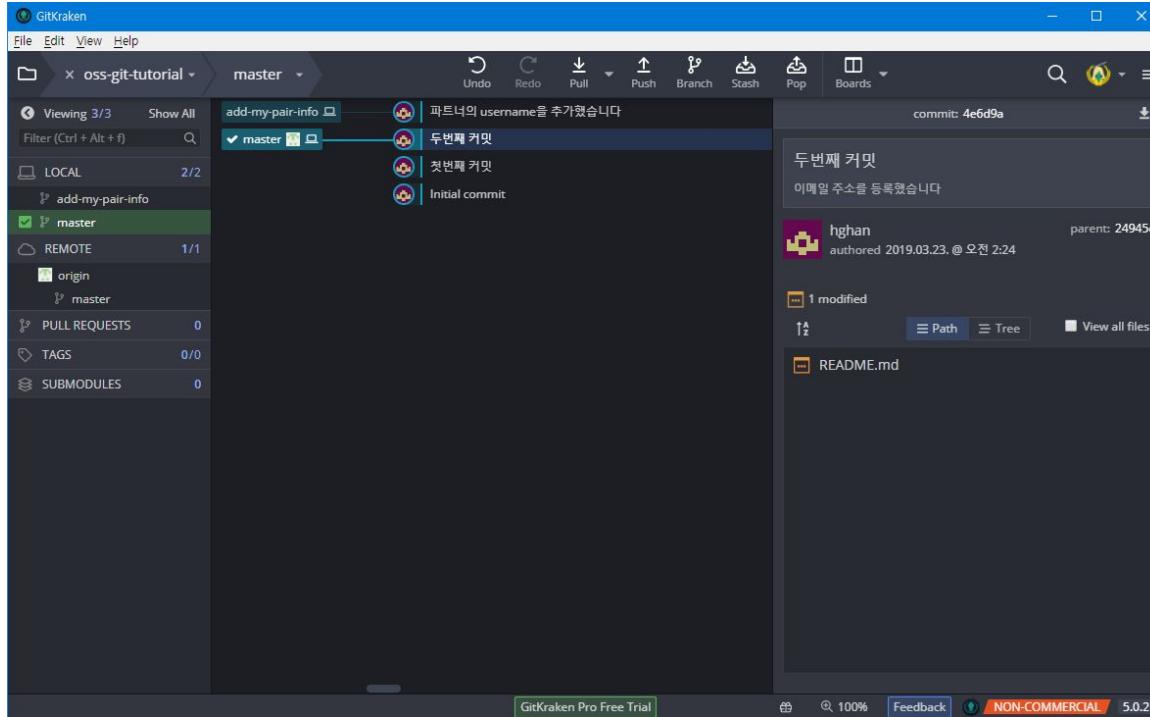
The screenshot shows a code editor window with the title "README.md". The content of the file is as follows:

```
1 oss-git-tutorial
2
3 안녕하세요. 저는 [github username] 입니다
4
5 제 이메일 주소는 hanbit@gmail.com 입니다
6
7 -----
8 저와 같이 실습을 진행하는 분은 [github username] 입니다
```

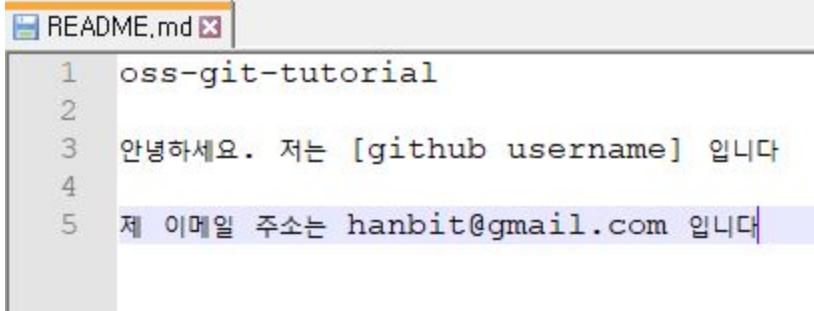
git branch - 새 브랜치(add-my-pair-info)에서 commit



git branch - 마스터 브랜치(master)에서 commit

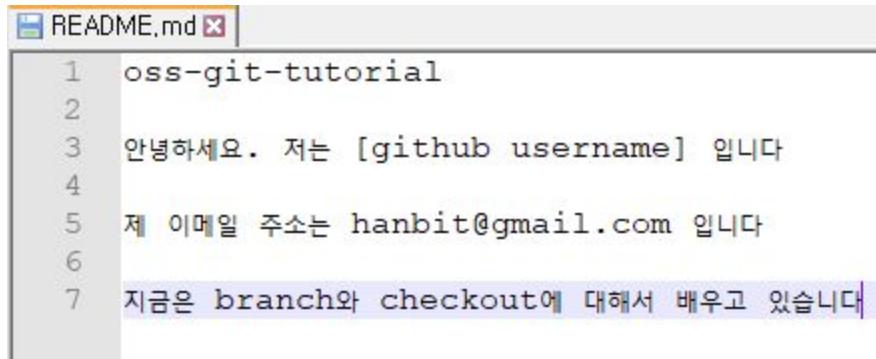


git branch - 마스터 브랜치(master)에서 commit



```
1 oss-git-tutorial
2
3 안녕하세요. 저는 [github username] 입니다
4
5 제 이메일 주소는 hanbit@gmail.com 입니다
```

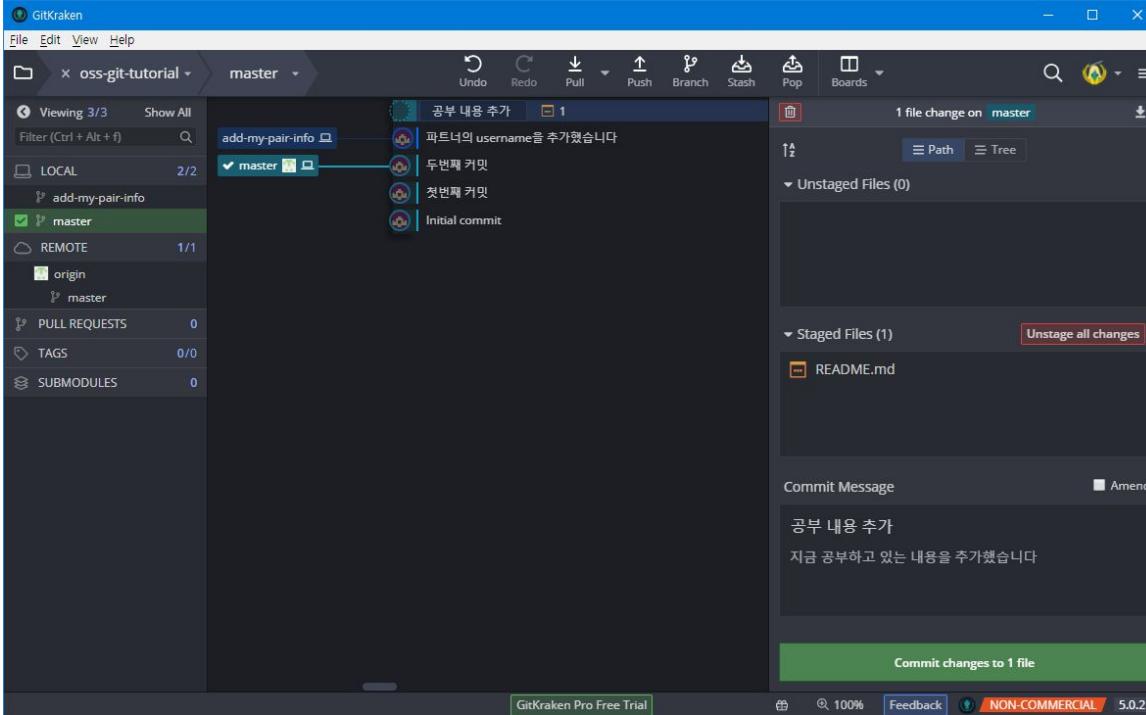
git branch - 마스터 브랜치(master)에서 commit



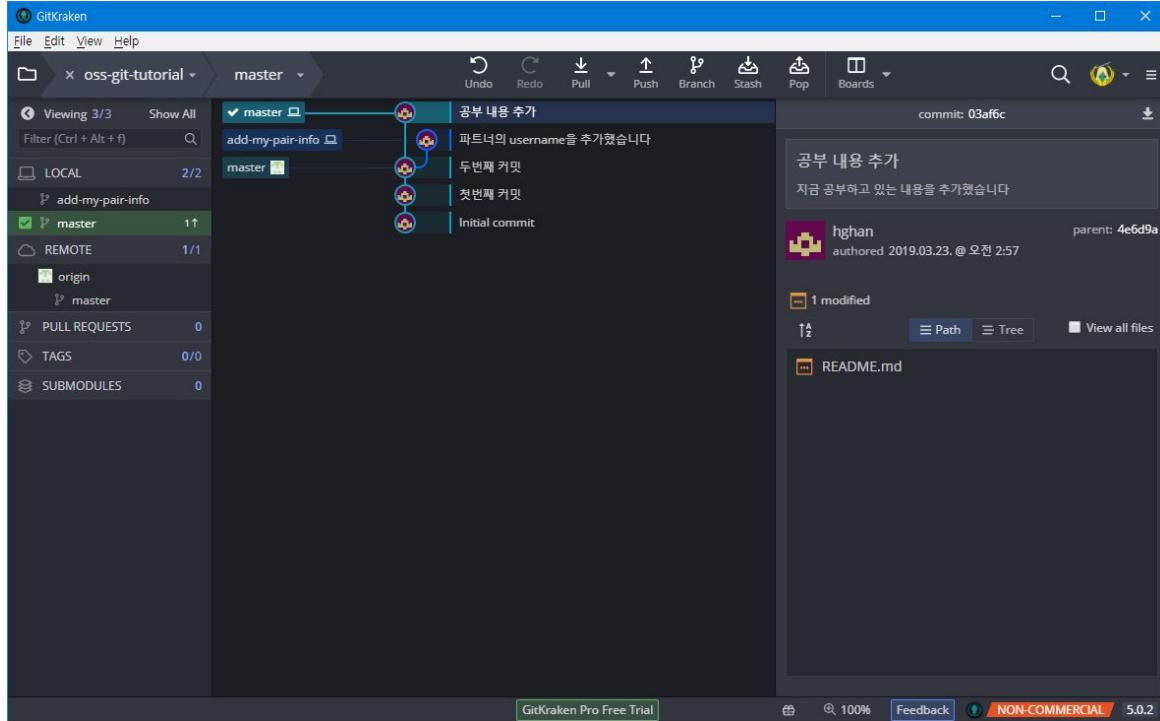
A screenshot of a code editor window titled "README.md". The content of the file is as follows:

```
1 oss-git-tutorial
2
3 안녕하세요. 저는 [github username] 입니다
4
5 제 이메일 주소는 hanbit@gmail.com 입니다
6
7 지금은 branch와 checkout에 대해서 배우고 있습니다
```

git branch - 마스터 브랜치(master)에서 commit



git branch - 마스터 브랜치(master)에서 commit



git merge

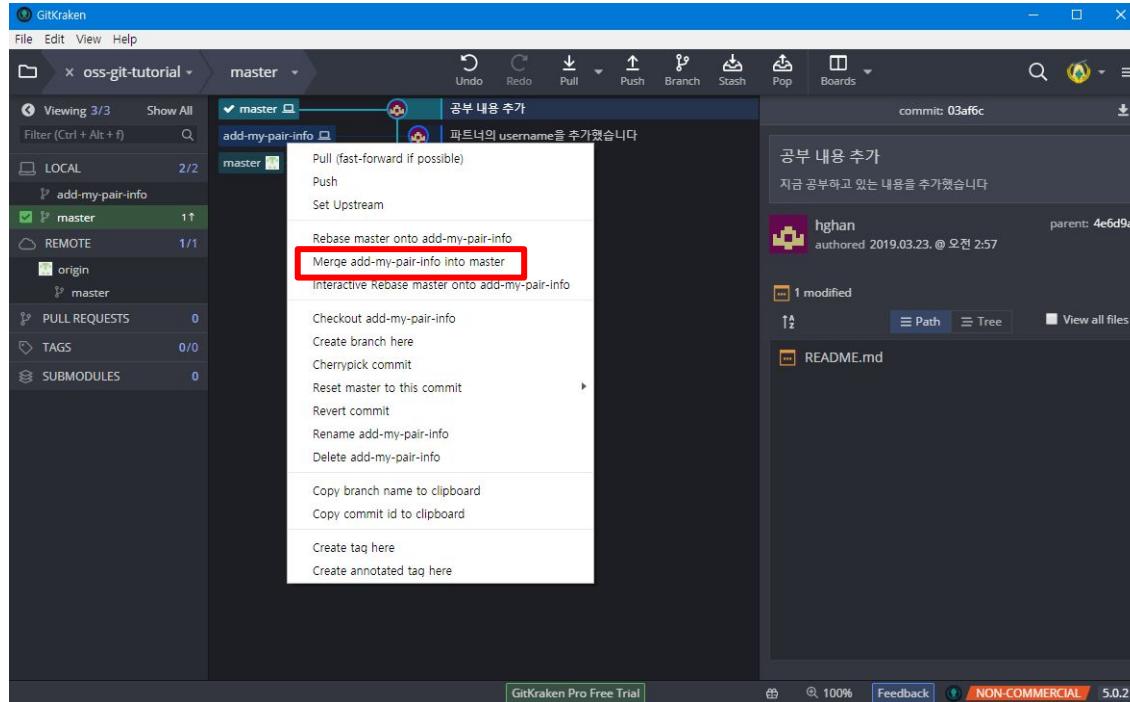
서로 다른 branch의 내용을 하나로 합치는 명령어

branch들의 내용이 모여야 하는 branch에서 실행

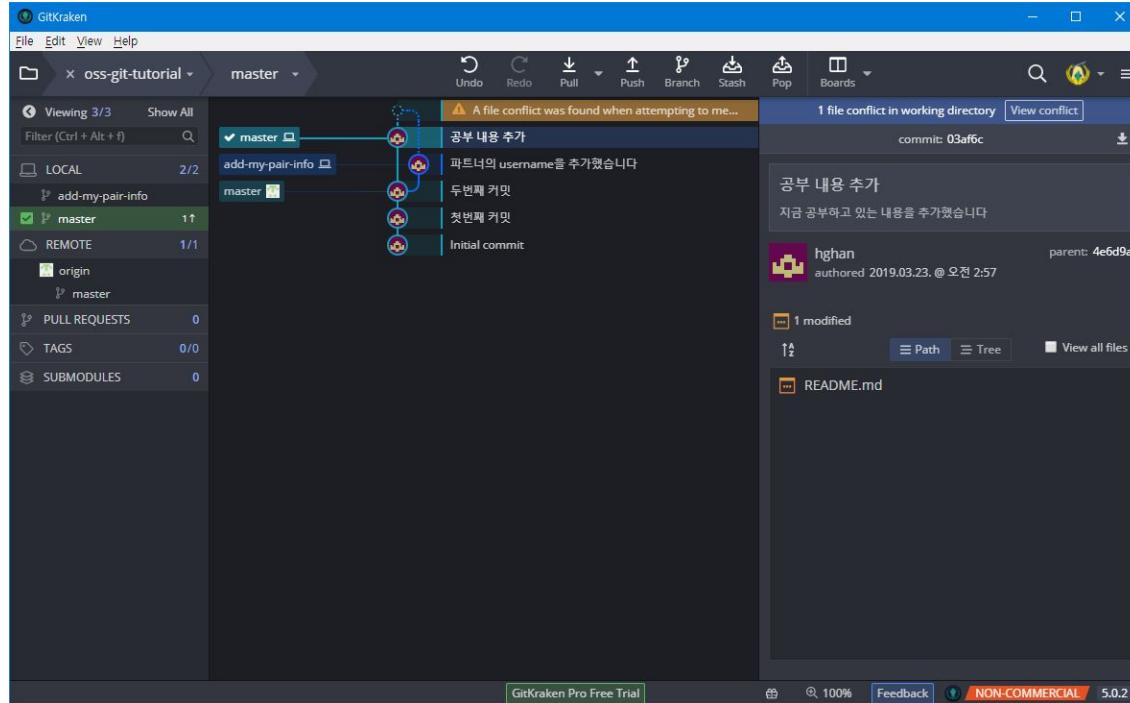
예) master 브랜치에 다른 브랜치의 내용을 합치고 싶다

→ master 브랜치에서 merge를 진행한다

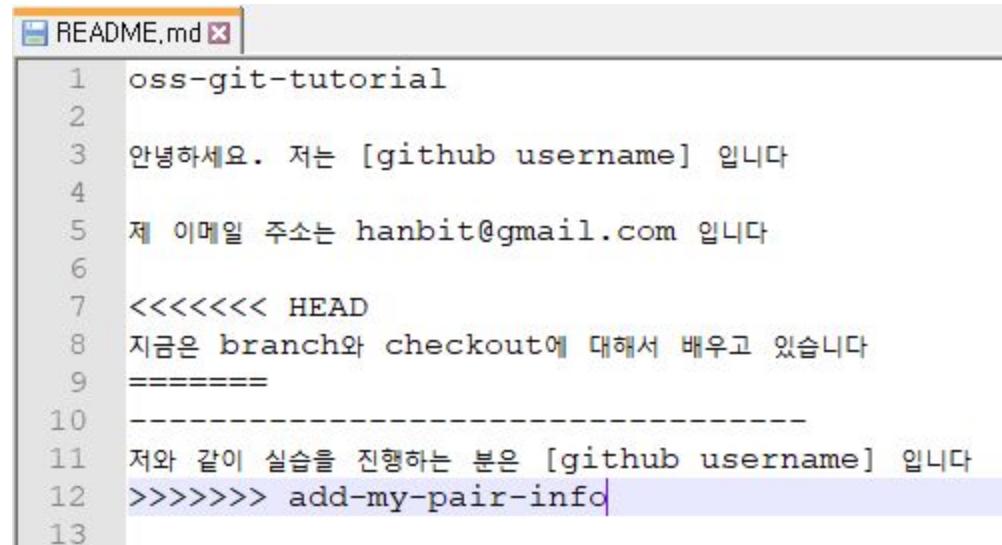
git merge



git merge - conflict(충돌)



git conflict



The screenshot shows a code editor window with the file 'README.md' open. The content of the file is as follows:

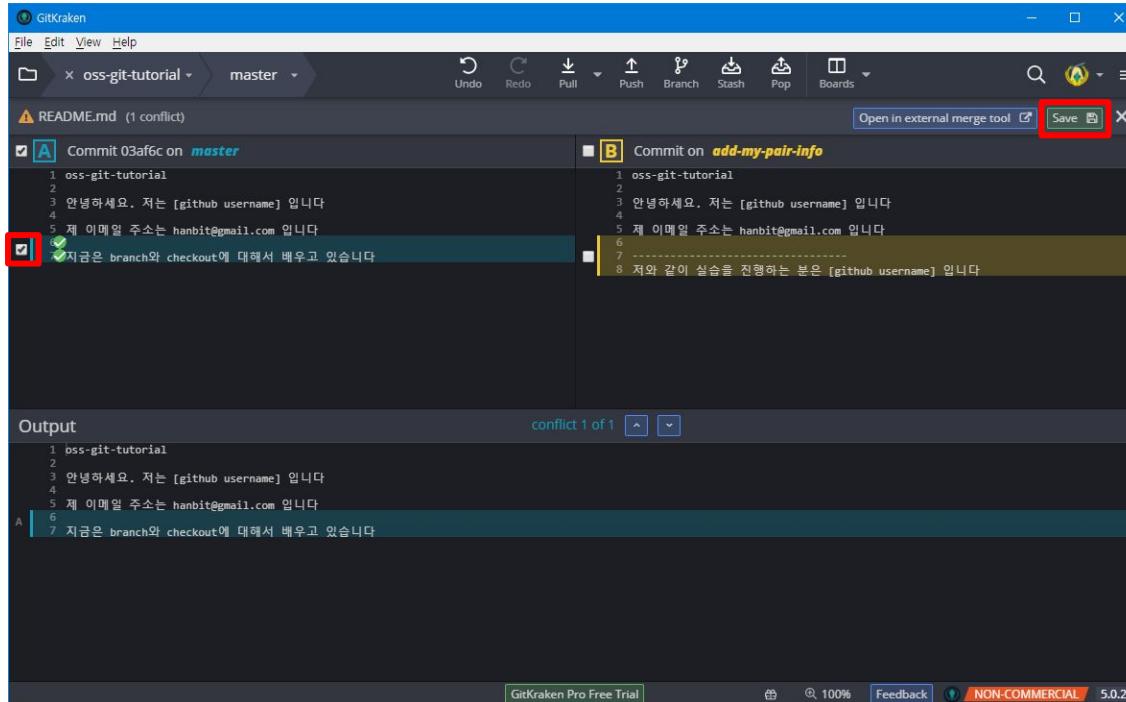
```
1 oss-git-tutorial
2
3 안녕하세요. 저는 [github username] 입니다
4
5 제 이메일 주소는 hanbit@gmail.com 입니다
6
7 <<<<< HEAD
8 지금은 branch와 checkout에 대해서 배우고 있습니다
9 =====
10 -----
11 저와 같이 실습을 진행하는 분은 [github username] 입니다
12 >>>> add-my-pair-info
13
```

The lines from 11 to 12 are highlighted with a light purple background, indicating a merge conflict between the 'HEAD' branch and the 'add-my-pair-info' branch.

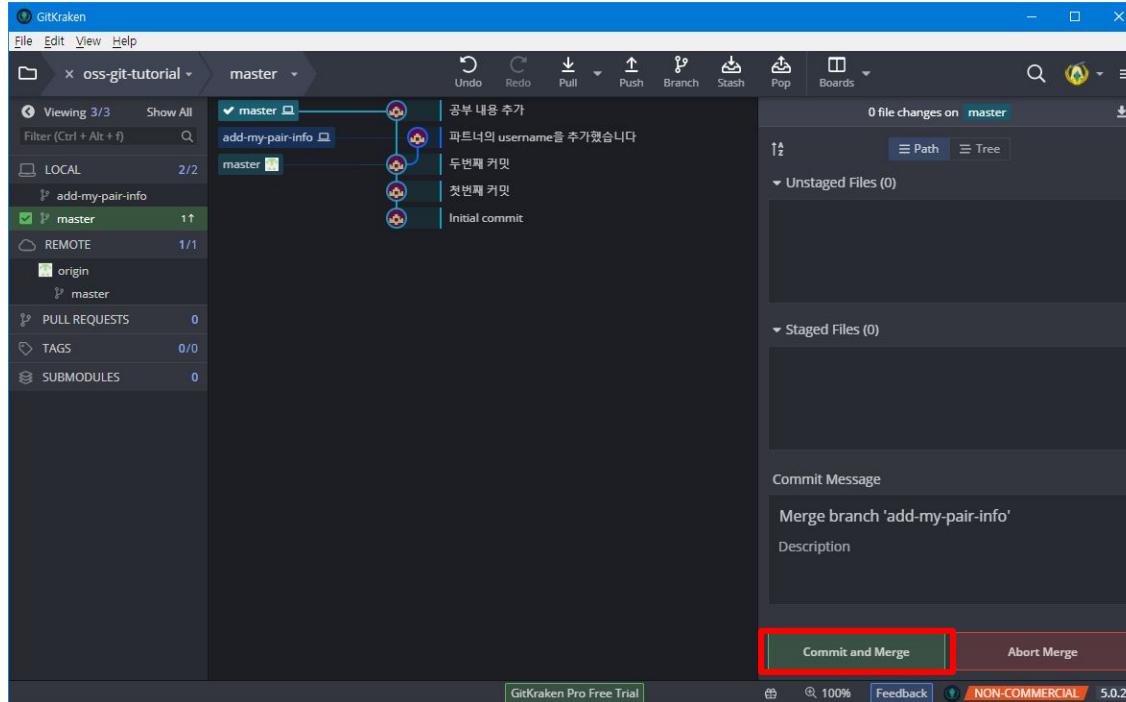
git conflict

나	기준	다른사람	3 way merge
A	A	A	A
B	B	B	B
1	C	2	?

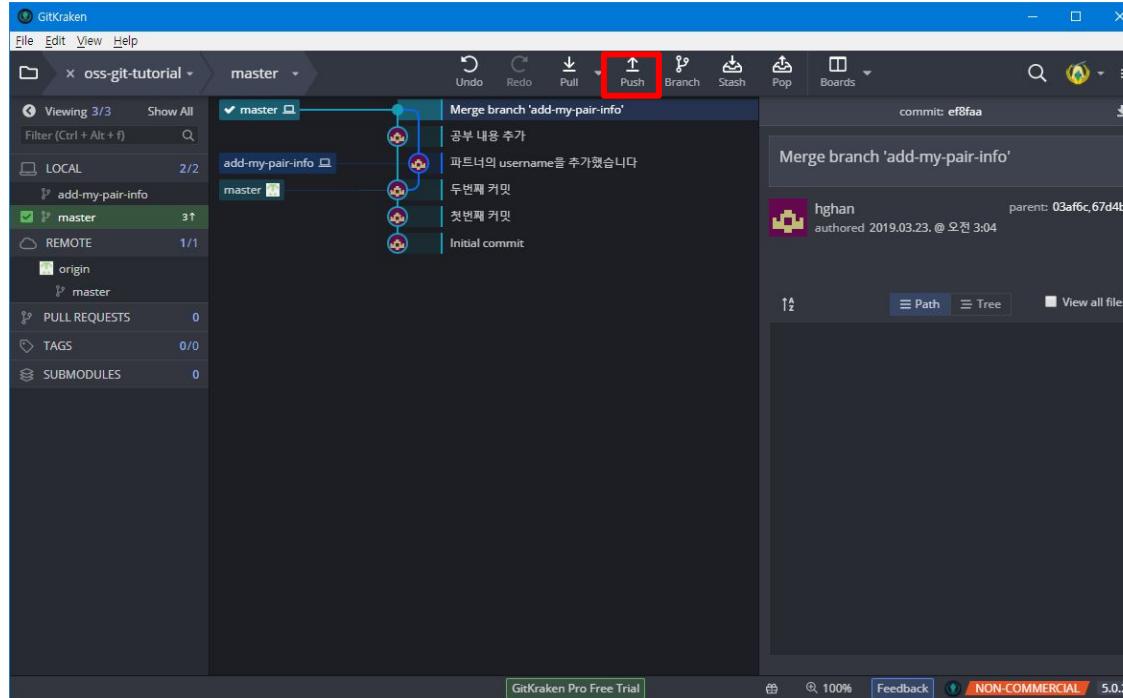
git conflict - resolve



git conflict - resolve



git push - remote repository로 commit들을 전달



쉬는시간

fork - 다른 사람의 프로젝트를 복제

<https://github.com/ossgituser1/pr-test>

fork - 다른 사람의 프로젝트를 복제

The screenshot shows a GitHub repository page for 'ossgituser1/pr-test'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The main header shows the repository name 'ossgituser1 / pr-test'. Below the header, there's a search bar and a 'Watch' button. The 'Fork' button is highlighted with a red box; it has a counter of '0' next to it. A 'Star' button with a counter of '0' is also present. The repository summary shows 1 commit, 1 branch, 0 releases, and 1 contributor. It includes a 'Clone or download' button and a 'New pull request' button. The repository history lists a single commit by 'erinyees' named 'init' made 2 minutes ago. The README.md file is shown below, containing the text 'PR-TEST'.

No description, website, or topics provided.

1 commit 1 branch 0 releases 1 contributor

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

erinyees init Latest commit 7b02a7f 2 minutes ago

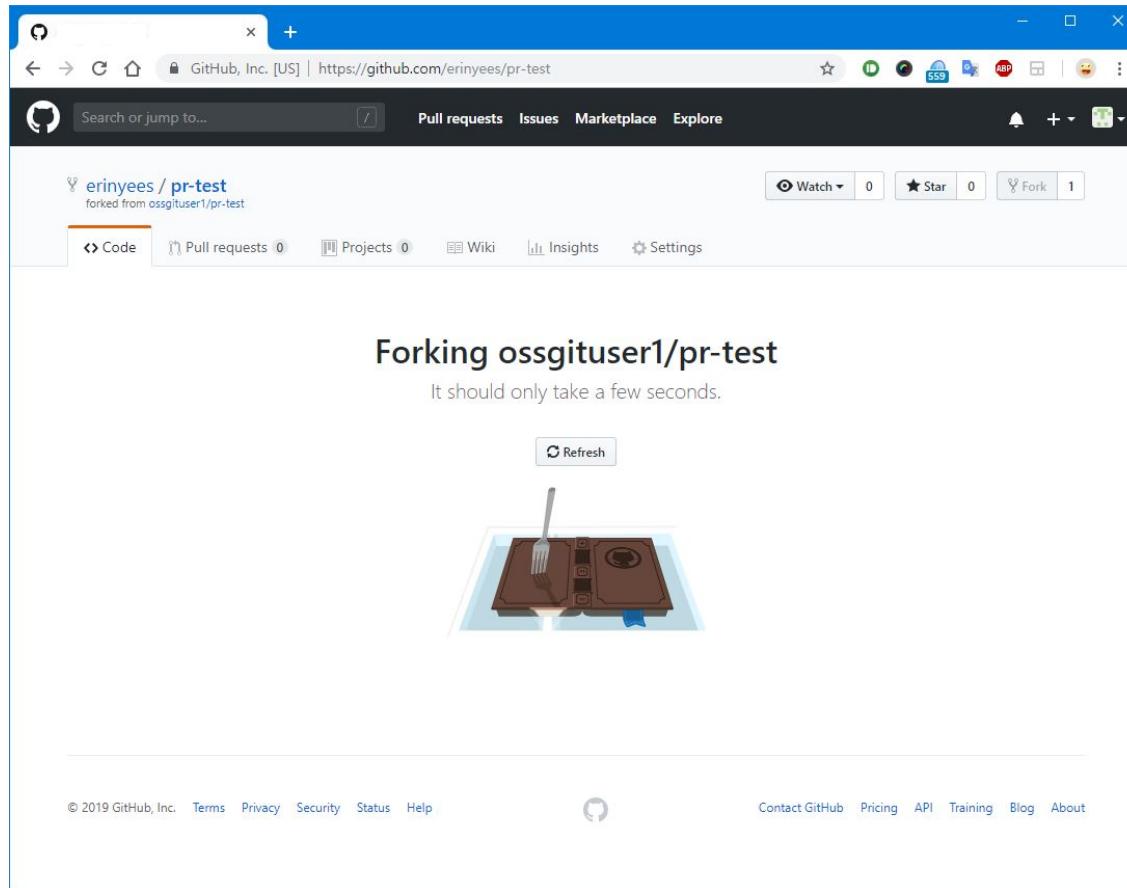
README.md init 2 minutes ago

README.md

PR-TEST

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

fork - 다른 사람의 프로젝트를 복제



fork - 다른 사람의 프로젝트를 복제

The screenshot shows a GitHub repository page for 'erinyees/pr-test'. The title bar indicates the repository is 'erinyees/pr-test' and it was forked from 'ossgituser1/pr-test'. The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a header with Watch (0), Star (0), Fork (1) buttons. Below the header, there's a 'Code' tab selected, followed by 'Pull requests (0)', 'Projects (0)', 'Wiki', 'Insights', and 'Settings' tabs. A note says 'No description, website, or topics provided.' with an 'Edit' button. Under 'Manage topics', there are statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. A 'Branch: master' dropdown and a 'New pull request' button are visible. Below these, a message says 'This branch is even with ossgituser1:master.' with 'Pull request' and 'Compare' buttons. The repository contains two files: 'README.md' (version init, committed 3 minutes ago) and another 'README.md' (version PR-TEST, uncommitted). At the bottom, there are links for GitHub terms, privacy, security, status, help, contact, pricing, API, training, blog, and about.

erinyees/pr-test

GitHub, Inc. [US] | https://github.com/erinyees/pr-test

erinyees / pr-test
forked from ossgituser1/pr-test

Watch 0 Star 0 Fork 1

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

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

This branch is even with ossgituser1:master. Pull request Compare

erinyees init Latest commit 7b02a7f 3 minutes ago

README.md init 3 minutes ago

README.md

PR-TEST

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

clone - 프로젝트를 로컬 디렉토리로 복사

A screenshot of a GitHub repository page for 'erinyees/pr-test'. The page shows basic repository statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. A green box highlights the 'Clone or download' button in the top right corner of the main content area. Below the button, there are two cloning options: 'Clone with HTTPS' and 'Use SSH', each with a corresponding URL and a copy icon. At the bottom of the page, there are links for 'Open in Desktop' and 'Download ZIP'.

No description, website, or topics provided. [Edit](#)

Manage topics

Branch: master [New pull request](#)

1 commit 1 branch 0 releases 1 contributor

This branch is even with ossguser1:master.

[erinyees init](#)

[README.md](#)

[README.md](#)

PR-TEST

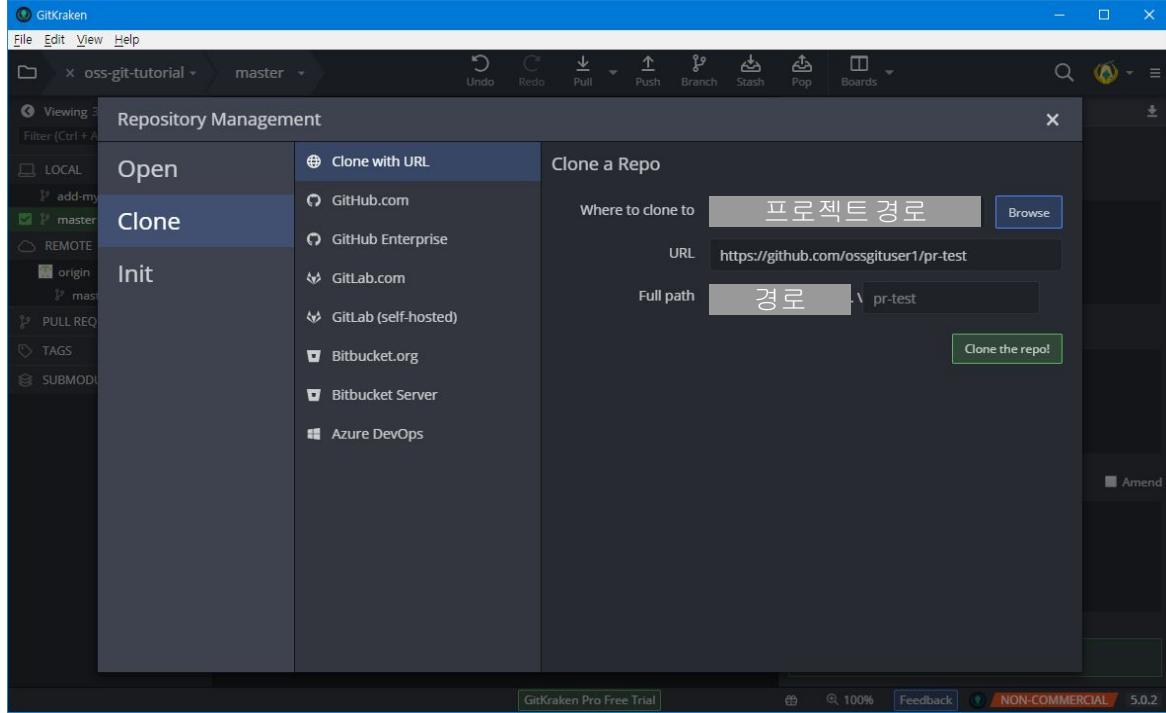
Clone with HTTPS <https://github.com/erinyees/pr-test.git> Use SSH

[Open in Desktop](#) [Download ZIP](#)

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

Contact GitHub [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

clone - 프로젝트를 로컬 디렉토리로 복사



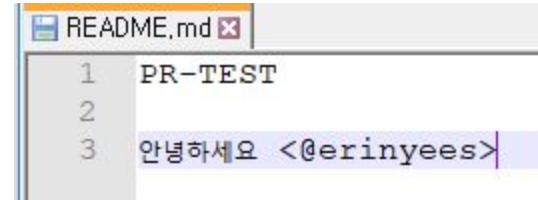
Pull Request

자신이 변경한 코드를 상대방의 리포지토리에 넣고 싶을 때 사용하는 기능

- gitlab에서는 merge request라는 이름으로 사용됨

pull request가 무엇인지 살펴본 뒤 실습을 진행합니다

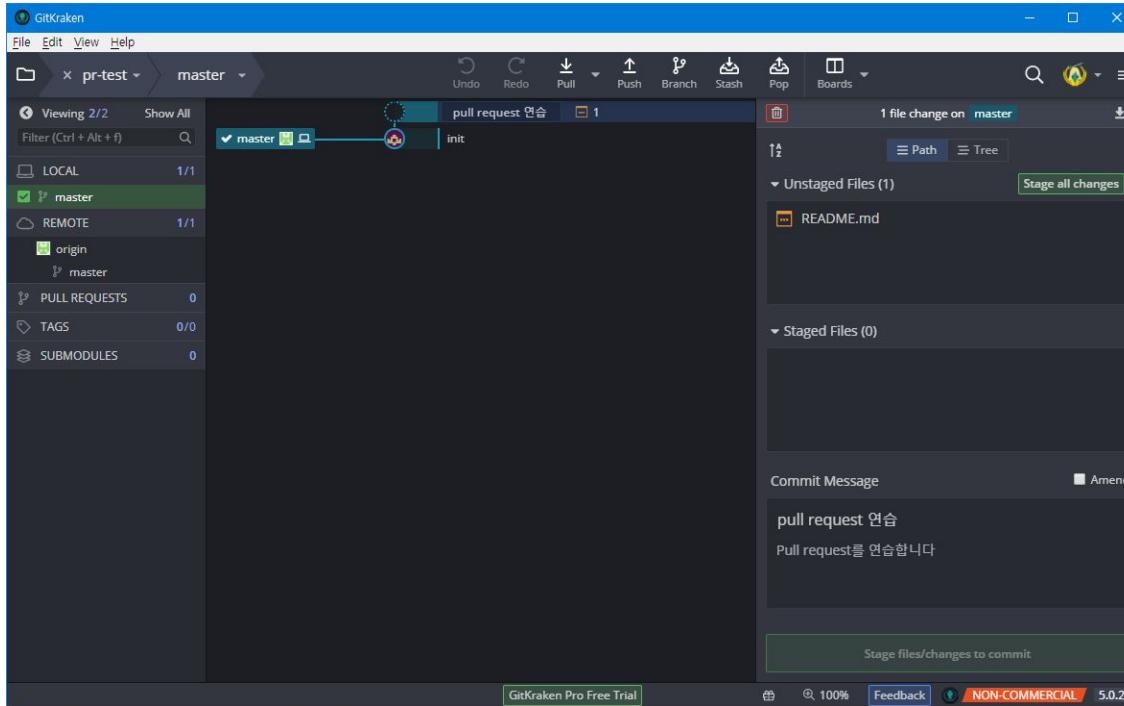
Pull Request



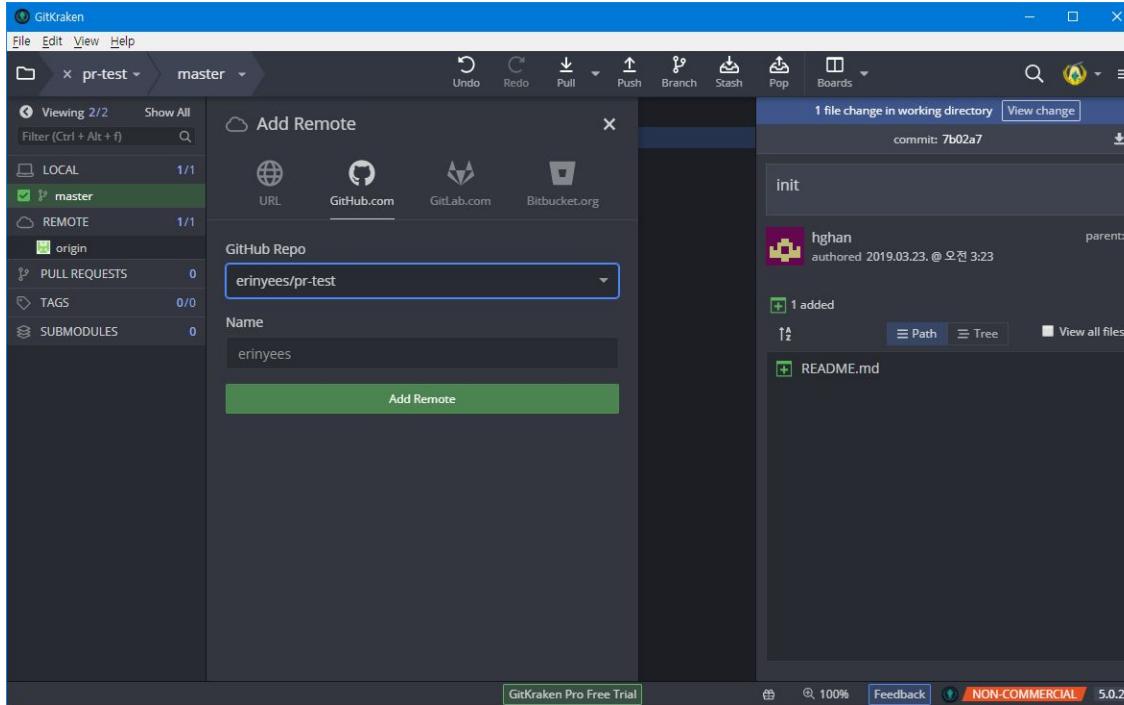
A screenshot of a code editor window titled "README.md". The content of the file is as follows:

```
1 PR-TEST
2
3 안녕하세요 <@erinyees>
```

Pull Request - commit 추가



Pull Request - remote repository(github)으로 push 준비



Pull Request

The screenshot shows a GitHub repository page for `erinyees/pr-test`. The repository is a fork from `ossgituser1/pr-test`. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the navigation, there are buttons for Watch (0), Star (0), and Fork (1). The repository has 2 commits, 1 branch, 0 releases, and 1 contributor.

A red box highlights the "New pull request" button, which is located next to the "Branch: master" dropdown menu. Other buttons in the row include Create new file, Upload files, Find File, and Clone or download.

The repository description states: "No description, website, or topics provided." There is a Manage topics link below the description.

The commit history shows one commit by `erinyees` titled "pull request 연습". The commit was made 2 minutes ago and is ahead of the `ossgituser1:master` branch. A "Pull request" button is available for this commit.

The README.md file contains the following content:

```
PR-TEST  
안녕하세요 <@erinyees>
```

At the bottom of the page, there are links for Contact GitHub, Pricing, API, Training, Blog, and About.

Pull Request

The screenshot shows a GitHub browser interface comparing branches between two repositories: `ossgituser1/pr-test` and `erinyees/pr-test`. The base repository is set to `ossgituser1/pr-test`, the base branch is `master`, the head repository is `erinyees/pr-test`, and the compare branch is also `master`. A green success message indicates that the branches are `Able to merge`.

Create pull request button (highlighted with a red box)

Commits on Mar 23, 2019:

- `erinyees` pull request 연습 ... 0d688ed

Showing 1 changed file with 2 additions and 0 deletions.

File	Additions	Deletions
README.md	2	0

```
@@ -1 +1,3 @@
1   PR-TEST
2 +
3 + 안녕하세요 <@erinyees>
```

Pull Request

The screenshot shows a GitHub pull request page for a repository named 'ossgituser1/pr-test'. The pull request is titled 'pull request 연습한 내용을 등록요청합니다 #1' and has been merged. A comment from 'erinyees' is visible, stating '안녕하세요.' and 'pull request 연습한 내용입니다. 추가부탁드립니다.' The page includes tabs for Code, Issues (0), Pull requests (1), Projects (0), Wiki, Insights, and Settings. On the right side, there are sections for Reviewers, Assignees, Labels, Projects, Milestone, and Notifications.

pull request 연습한 내용을 등록요청합니다 #1

Merged ossgituser1 merged 1 commit into ossgituser1:master from erinyees:master 10 seconds ago

erinyees commented a minute ago

안녕하세요.
pull request 연습한 내용입니다. 추가부탁드립니다.

ossgituser1 merged commit adbaaeef into ossgituser1:master just now

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

Pull Request

The screenshot shows a GitHub pull request page for the repository `ossgituser1/pr-test`. The repository has 3 commits, 1 branch, 0 releases, and 1 contributor. A merge pull request from `erinyees/master` was merged 44 seconds ago. The latest commit message is `안녕하세요 <@erinyees>`.

No description, website, or topics provided.

3 commits | 1 branch | 0 releases | 1 contributor

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

ossgituser1 Merge pull request #1 from erinyees/master ... | Latest commit adbaaeef 44 seconds ago

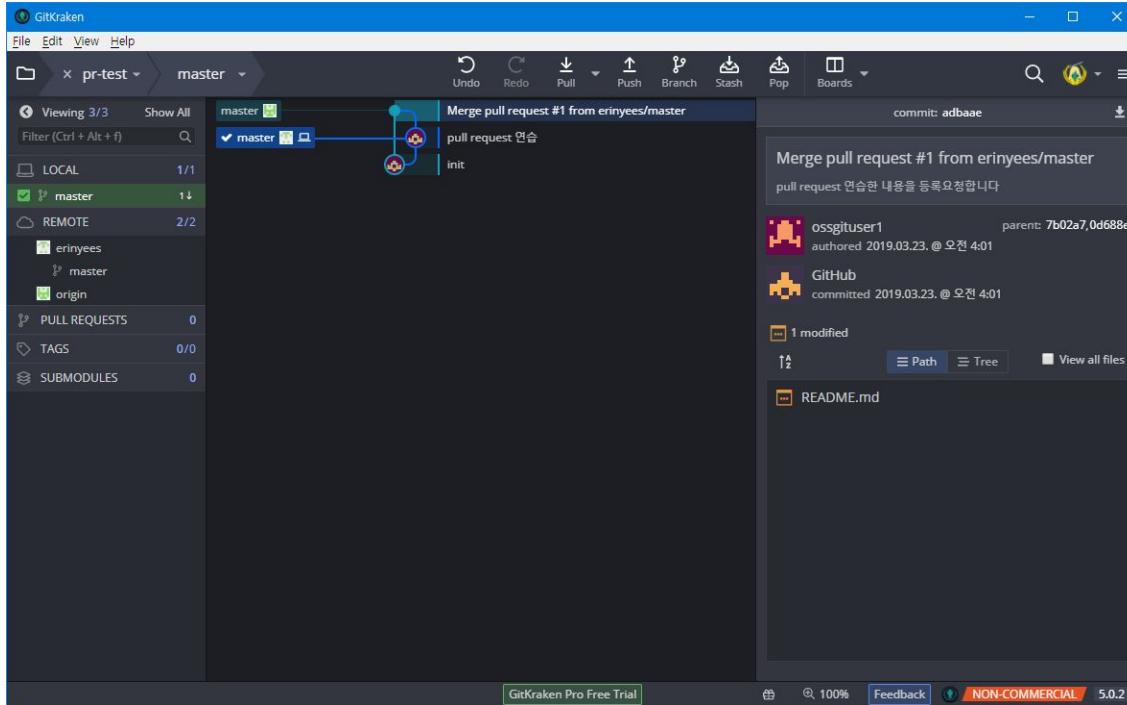
README.md | pull request 연습 | 5 minutes ago

README.md

PR-TEST

안녕하세요 <@erinyees>

Pull Request



Pull Request 실습

1) 같은 책상에 앉은 상대에게 pull request를 보내보세요

- 규칙

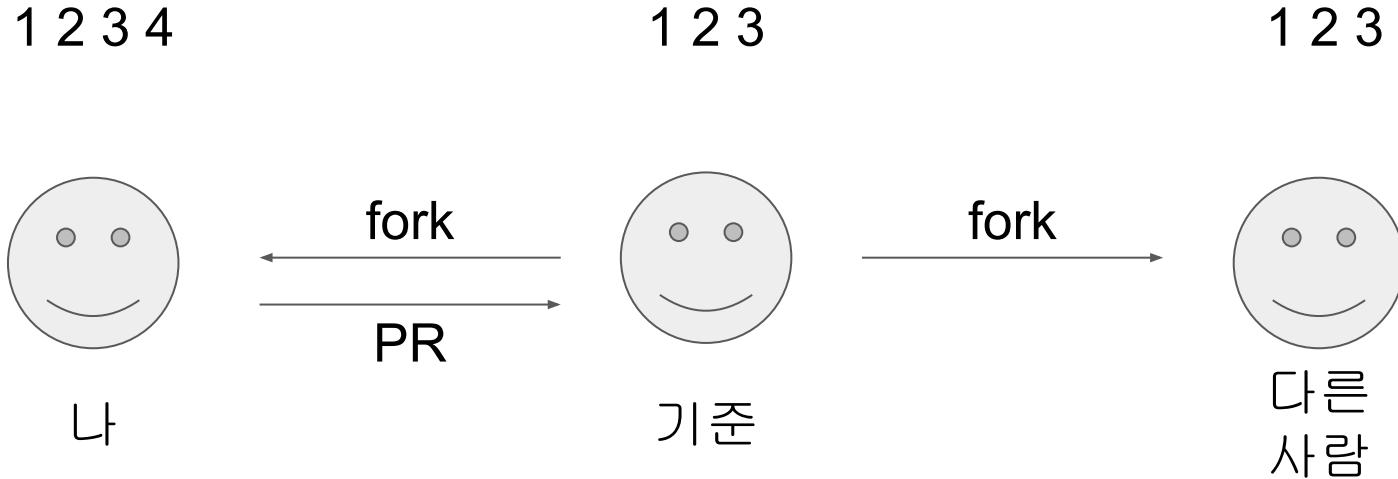
- 왼쪽 자리에 앉은 사람이 오른쪽 자리에 앉은 사람에게 pull request를 보냅니다
- 오른쪽 자리에 앉은 사람은 요청이 들어온 pull request를 merge 해봅니다
- merge가 잘 됐다면 서로 역할을 바꿔서 해봅니다

* 주의 : clone 할 때 경로가 중복되지 않도록 주의하세요

Pull Request에도 순서가 있습니다



Pull Request에도 순서가 있습니다



Pull Request에도 순서가 있습니다

1 2 3 4

1 2 3 4

1 2 3



나

fork
PR



기준

나의 PR을 merge 함



다른
사람

fork

Pull Request에도 순서가 있습니다

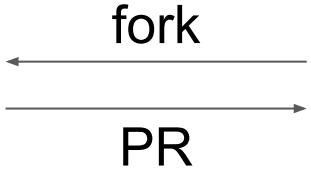
1 2 3 4

1 2 3 4

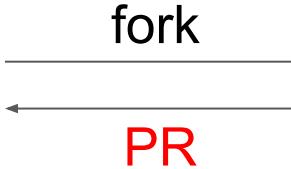
1 2 3 5



나



기준



다른 사람

다른 사람의 PR을 merge 시도
→ 실패

나의 PR이 merge되기 이전 상태

나	기준	다른사람	3 way merge
1	1	1	1
2	2	2	2
3	3	3	3
4			

나의 PR0| merge 된 상태

나	기준	다른사람	3 way merge
1	1	1	1
2	2	2	2
3	3	3	3
4	4		4

다른 사람이 PR을 보내려 할 때의 상태

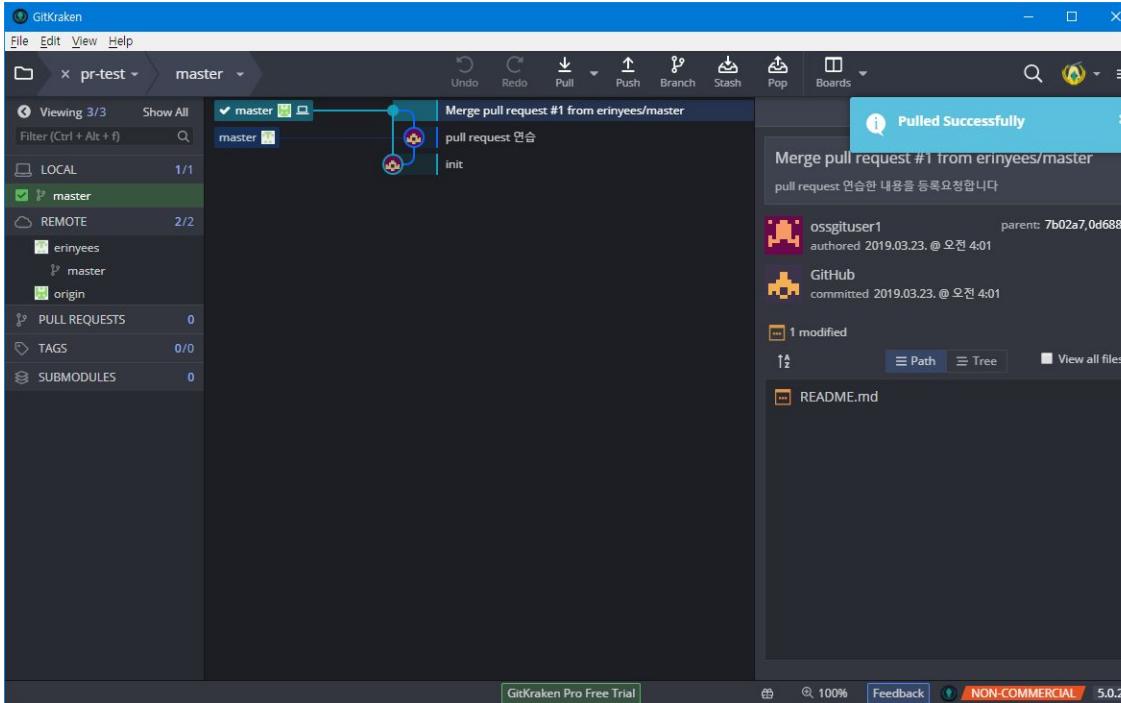
나	기준	다른사람	3 way merge
1	1	1	1
2	2	2	2
3	3	3	3
4	4	5	?

git fetch / pull

git fetch : 리포지토리에서 변경된 기록을 가져오되 **merge**는 하지 않음

git pull : 리포지토리에서 변경된 기록을 가져오고 **merge**도 바로 이루어짐

git pull



Pull Request에도 순서가 있습니다

1 2 3 4



나

1 2 3 4

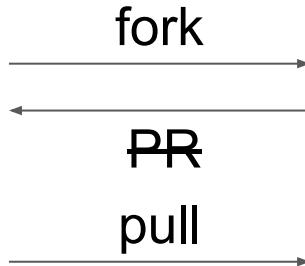
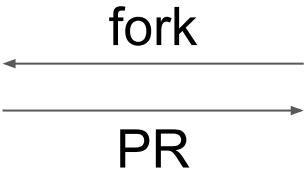


기준

1 2 3 [4 또는 5]



다른 사람



Pull Request에도 순서가 있습니다

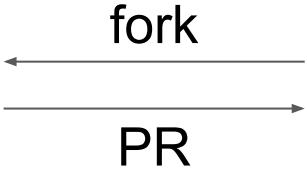
1 2 3 4



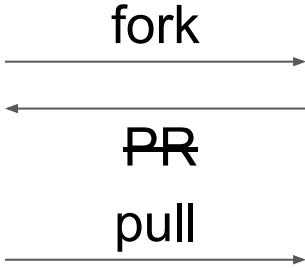
1 2 3 4



1 2 3 5



나



기준



다른 사람

conflict
해결

Pull Request에도 순서가 있습니다

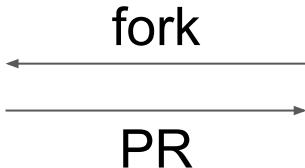
1 2 3 4

1 2 3 5

1 2 3 5

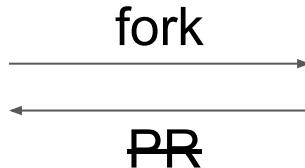


나

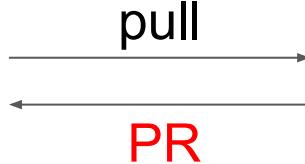


기준

B의 PR을 merge 함



다른 사람



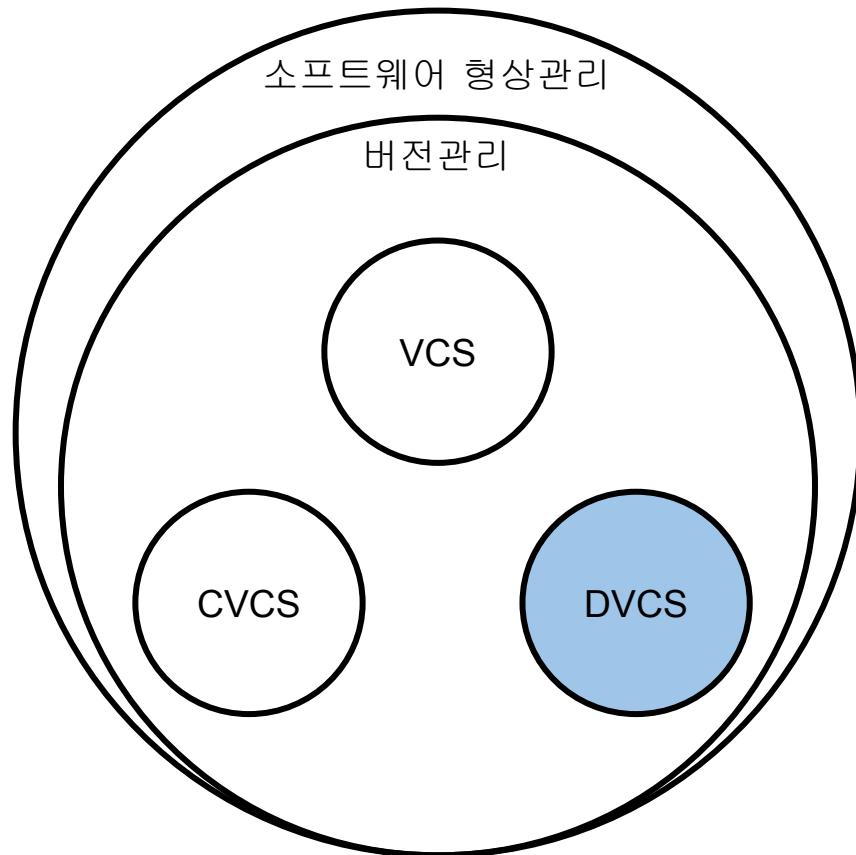
오늘 공부한 내용 훑어보기

소프트웨어 형상 관리

- 소프트웨어의 **변경사항을 체계적으로 추적하고 관리, 통제하는 것**
- 형상 관리 대상
 - 소스코드 그리고 그 외 많은 관리 대상들

버전 관리 시스템

- 소스코드의 변경사항을 체계적으로 추적하고 관리, 통제하는 시스템

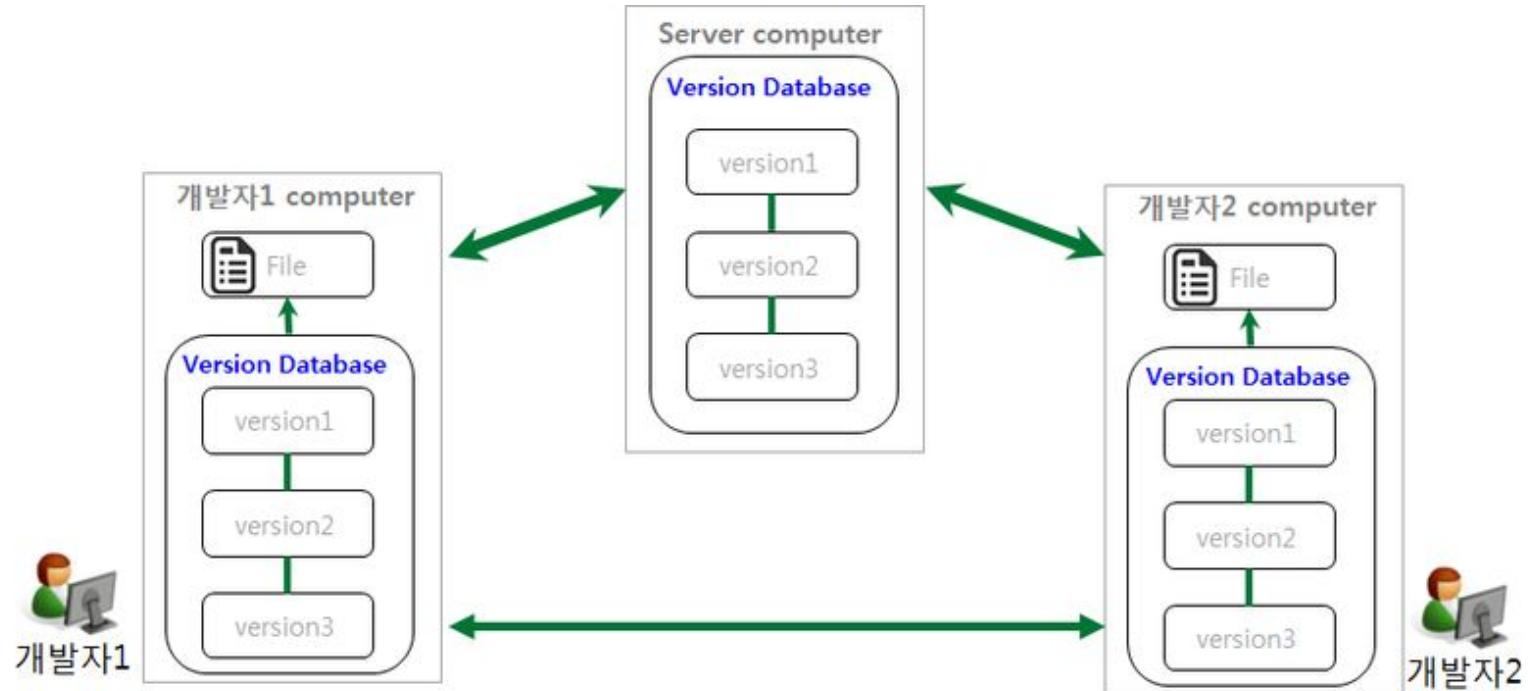


Git과 Github

Git : 버전 관리 시스템

Github : Git 호스팅 사이트

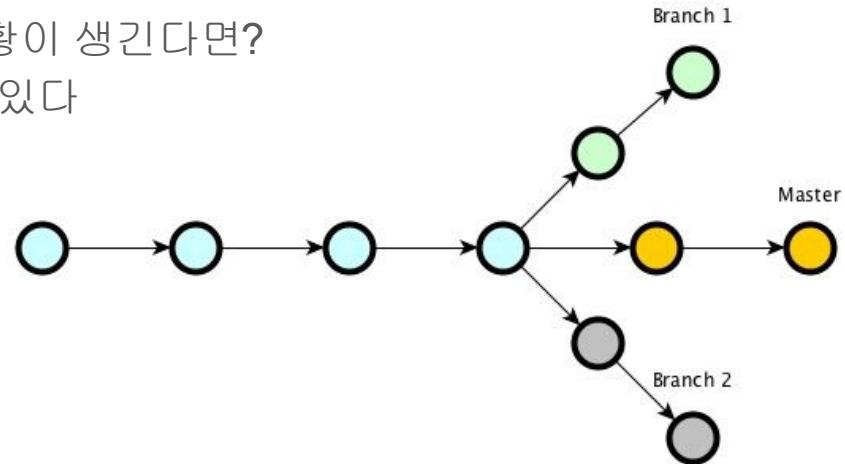
- 호스팅 : 웹의 공간을 빌려주거나 빌리는 것



add, commit 명령어를 이용해 기록 생성

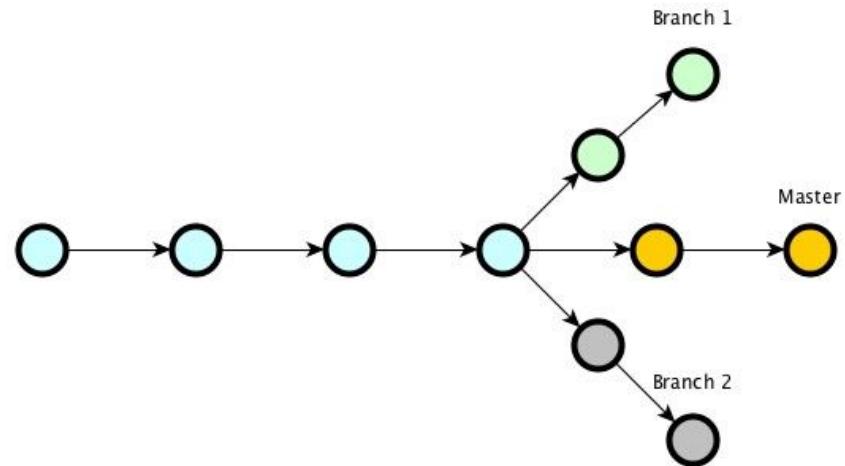
git branch

- 특정 기준에 따라 commit들을 모아두는 공간
 - 추가 기능 개발, 버그 수정 등
- 하나의 파일에서 두가지 일을 해야 하는 상황이 생긴다면?
→ 두 작업을 하다가 작업내용이 섞일 수 있다



conflict

- 같은 부분에 접근해서 변경사항이 발생할 때 생기는 문제



Pull Request

자신이 변경한 코드를 상대방의 리포지토리에 넣고 싶을 때 사용하는 기능

- gitlab에서는 merge request라는 이름으로 사용됨

Pull Request 도중 conflict가 발생하면?

- git fetch / git pull을 이용해서 최신 버전의 코드를 가져온 뒤 문제를 해결한다

GitHub 둘러보기

- 여러 프로젝트, 소식 살펴보기 : <https://github.com/explore>
- 오픈소스 둘러보기
 - <https://github.com/naver/pinpoint>
 - issue : <https://github.com/naver/pinpoint/issues/5374>
 - good first issue
 - vue.js : <https://github.com/vuejs/vue/labels/good%20first%20issue>
 - react : <https://github.com/facebook/react/labels/good%20first%20issue>
- 라이센스
 - <https://opensource.org/licenses>
- pricing plan
 - <https://github.com/pricing>

알아두면 좋은 것

- 생활코딩 - 지옥에서 온 Git(소스트리 활용) : <https://opentutorials.org/course/2708>
- 동네개발자형 교육방송 - git bisect : <https://www.youtube.com/watch?v=SYVUyNrQhAU>
- student pack
 - <https://education.github.com/pack>
- .gitignore
 - <https://www.gitignore.io/>
 - gitignore 적용하기 :
<http://www.itpaper.co.kr/gitignore-%ED%8C%8C%EC%9D%BC%EC%9D%84-%ED%86%B5%ED%95%9C-%EB%AC%B4%EC%8B%9C%EB%AA%A9%EB%A1%9D-%EC%84%A4%EC%A0%95%ED%95%98%EA%B8%B0/>

더 공부할 주제

- git lint
 - 소스코드 lint : <https://github.com/sk-/git-lint>
 - 커밋메시지 규칙 : <https://jorisroovers.github.io/gitlint/>
- git reset
- git revert
- git reflog
- git rebase
- git cherry-pick
- git tag
- label
- release
- git flow : https://danielkummer.github.io/git-flow-cheatsheet/index.ko_KR.html

참고하면 도움이 될 책

소프트웨어 개발의 모든 것

<http://www.yes24.com/Product/Goods/3920120?Acode=101>

성공으로 이끄는 팀 개발 실천 기술

<http://www.yes24.com/Product/Goods/14725219?Acode=101>

팀을 위한 git

<http://www.yes24.com/Product/Goods/30741673?Acode=101>

Q & A