

영상관리

2019 – (주)텔코코리아아이에스

2022 – (주)리모샷

2023 – 넥스트IT

강사: 서지민

연락처: 010-3376-1886

메일: jmseokm29a1@naver.com

<https://github.com/RadeonHD6750>



일상생활

원격 저장소



시간과 공간에 제약이 없다.

넥스트 IT 교육센터

여기서 작업하던 것을

학원에서도 그대로 이어서 가능

네이버, 구글 클라우드와 무슨 차이??



집에서도 그대로 이어서 가능

집에서 주말에 했던 것을

프로젝트 관리 개념

◆ 정의

프로젝트의 진행상황 및 변경사항 등을 체계적으로 추적하고 통제하는 것

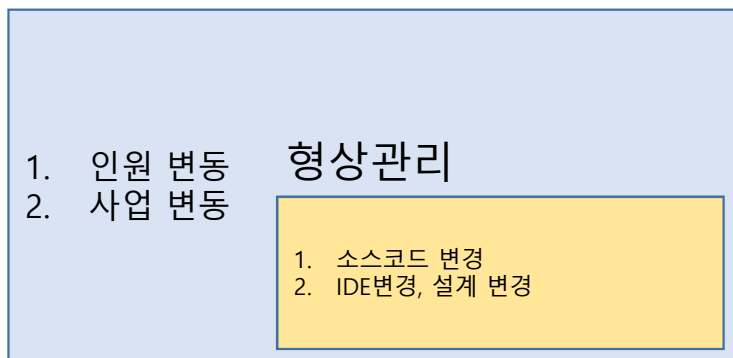
◆ 프로젝트의 변경사항 종류

- 소스코드 변경사항 -> 깃으로 관리
- IDE, 빌드구조, 설계 변동 등등
- 참여인원에 대한 변동 (근태, 인원수 변동, 인력 보충 등)
- 사업에 대한 변동 (요구사항 변동, 일정 변동 등)

◆ 용어 정리

- 변경관리 : 소스코드 변경사항에 대한 관리
- 버전관리 : 변경사항을 '버전'이란 개념을 통하여 관리
- 형상관리 : 위의 개념들을 포함하여 프로젝트와 관련된 모든 변경사항을 관리

프로젝트 관리



Git Client 설치하기



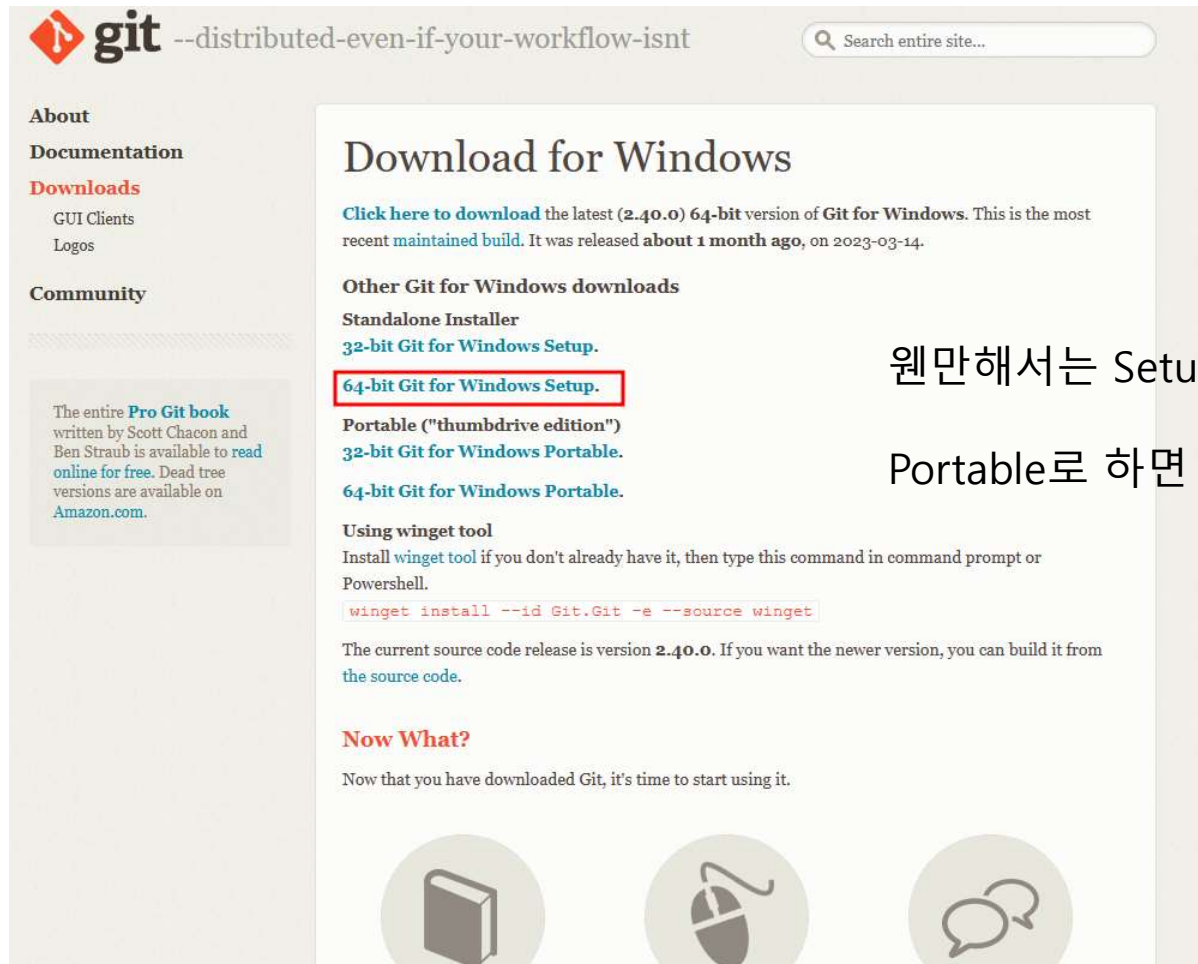
Git 설치 - 리눅스

```
jimin@jimin-meat: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

jimin@jimin-meat:~$ git
명령어 'git' 을(를) 찾을 수 없습니다. 그러나 다음을 통해 설치할 수 있습니다:
sudo apt install git
jimin@jimin-meat:~$ sudo apt install git
[sudo] jimin 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다:
  git-man liberror-perl
제안하는 패키지:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
다음 새 패키지를 설치할 것입니다:
  git git-man liberror-perl
0개 업그레이드, 3개 새로 설치, 0개 제거 및 224개 업그레이드 안 함.
4,121 k바이트 아카이브를 받아야 합니다.
이 작업 후 20.9 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.170
29-1 [26.5 kB]
받기:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2
.34.1-1ubuntu1.8 [953 kB]
```

1. Git이 있는지 터미널에 입력하여 존재여부 확인
2. 없다면 sudo apt install git으로 설치

Git 설치 - 윈도우



웹만해서는 Setup버전을 하자

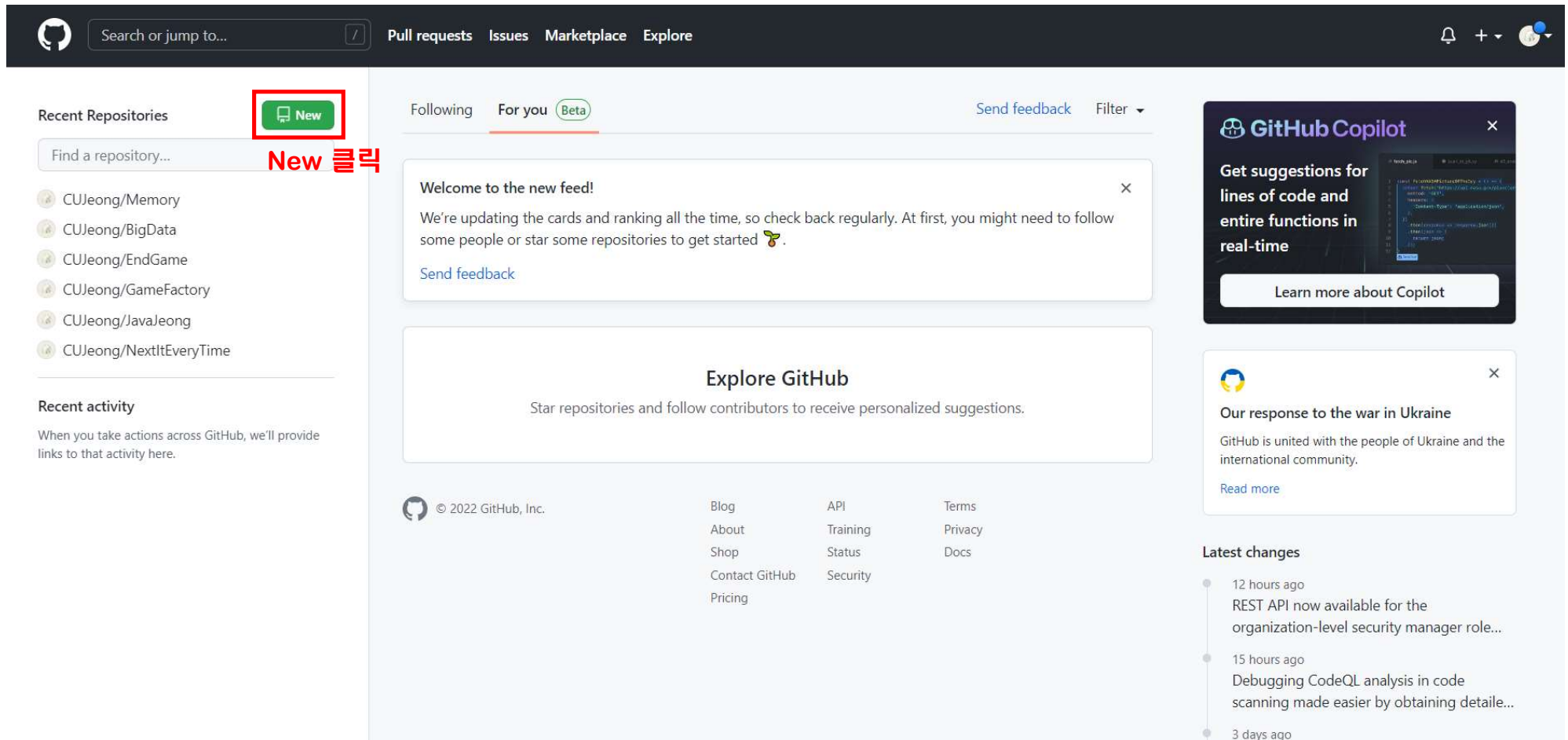
Portable로 하면 나중에 못 찾을

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

GitHub 저장소 생성



GitHub 저장소 생성



The screenshot shows the GitHub homepage. At the top, there is a navigation bar with the GitHub logo, a search bar, and links to Pull requests, Issues, Marketplace, and Explore. On the left sidebar, under 'Recent Repositories', there is a 'New' button highlighted with a red box. Next to it, the text 'New 클릭' is written in red. Below this, there is a list of recent repositories for the user 'CUJeong'. The main content area shows a 'Welcome to the new feed!' message, a 'Send feedback' link, and an 'Explore GitHub' section. On the right sidebar, there are two promotional cards: one for GitHub Copilot and another about GitHub's response to the war in Ukraine. At the bottom, there is a 'Latest changes' section with a list of recent updates.

Recent Repositories

New New 클릭

Find a repository...

- CUJeong/Memory
- CUJeong/BigData
- CUJeong/EndGame
- CUJeong/GameFactory
- CUJeong/JavaJeong
- CUJeong/NextItEveryTime

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Following For you **Beta** Send feedback Filter

Welcome to the new feed!

We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started 🍌.

[Send feedback](#)

Explore GitHub

Star repositories and follow contributors to receive personalized suggestions.

© 2022 GitHub, Inc.

Blog
About
Shop
Contact GitHub
Pricing

API
Training
Status
Security

Terms
Privacy
Docs

GitHub Copilot

Get suggestions for lines of code and entire functions in real-time

[Learn more about Copilot](#)

Our response to the war in Ukraine

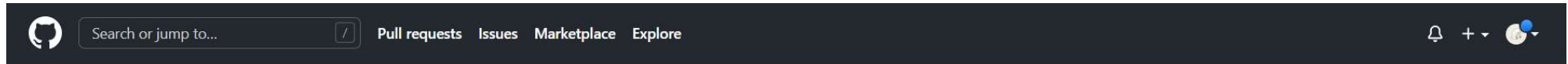
GitHub is united with the people of Ukraine and the international community.

[Read more](#)

Latest changes

- 12 hours ago
REST API now available for the organization-level security manager role...
- 15 hours ago
Debugging CodeQL analysis in code scanning made easier by obtaining detaile...
- 3 days ago

GitHub 저장소 생성



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * CUJeong / Repository name * MyProject **프로젝트명 설정**

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

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)



**README.md 차후 생성
지금은 X**

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

GitHub 저장소 생성

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

여기서는 README는 제외

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▼

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None** ▼

 You are creating a public repository in your personal account.

Create repository

Create repository 클릭

GitHub 저장소 생성

falsy / git-start-guide-1 Unwatch 1 Star 0 Fork 0

[Code](#)
[Issues 0](#)
[Pull requests 0](#)
[Projects 0](#)
[Wiki](#)
[Insights](#)
[Settings](#)

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

[Set up in Desktop](#) or
 [HTTPS](#)
[SSH](#)
<https://github.com/falsy/git-start-guide-1.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```

echo "# git-start-guide-1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/falsy/git-start-guide-1.git
git push -u origin master
        
```

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

```

git remote add origin https://github.com/falsy/git-start-guide-1.git
git push -u origin master
        
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

최초 생성된 상태

GitHub 저장소 생성

The screenshot shows the GitHub interface for a repository named 'RadeonHD6750 / JavaExample'. The repository is public and has 1 branch and 0 tags. The 'Code' dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI. The HTTPS URL is highlighted in a red box: `https://github.com/RadeonHD6750/JavaExample.g`. Other options include 'Open with GitHub Desktop' and 'Download ZIP'. The repository content shows a file named 'RadeonHD6750 Update README.md' and several folders: 'Libraries', 'ch01_helloworld', 'ch02_variable', 'ch03_operator', 'ch04_standardIO', 'ch05_controll', and 'ch06_method'. The right sidebar shows the 'About' section with details about the repository, including the license (GPL-3.0) and the number of stars (0).

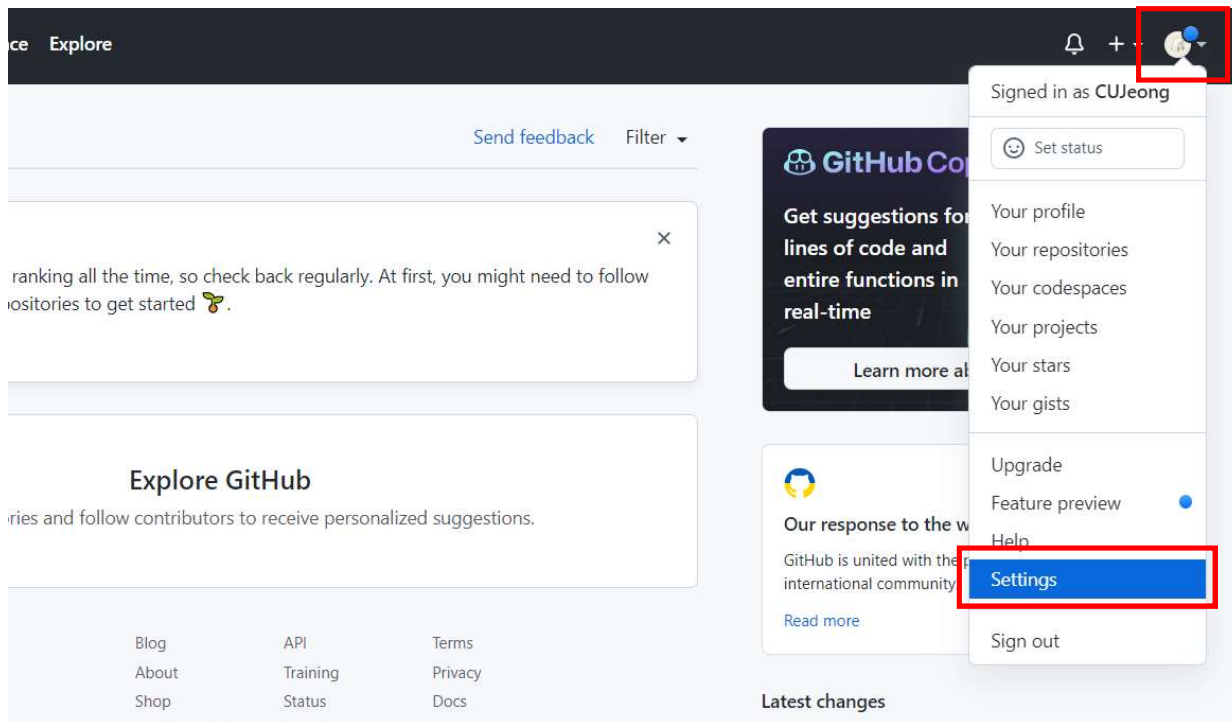
Git 주소 취득

GitHub 토큰 생성



GitHub 토큰 생성

◆ GitHub 토큰은 VS Code나 Eclipse에서 GitHub에 로그인 시 사용되는 비밀번호라고 보면 된다.



프로필 이미지 클릭

Settings 클릭

GitHub 토큰 생성



Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications
- Access
- Billing and plans
- Emails
- Password and authentication
- SSH and GPG keys
- Organizations
- Moderation

스크롤 내리기



- Code, planning, and automation
- Repositories
- Packages
- GitHub Copilot

Public profile

Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

You can @mention other users and organizations to link to them.

URL

Twitter username

Profile picture



Edit

GitHub 토큰 생성

Security

🔒 Code security and analysis

Integrations

🔌 Applications

🕒 Scheduled reminders

Archives

📖 Security log

📖 Sponsorship log

⏏ Developer settings

Developer settings
클릭

You can @mention your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & Activity

☐ Make profile private and hide activity Beta

Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

☐ Include private contributions on my profile

Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more](#).

Update preferences

Profile settings

☒ Show Achievements on my profile

Your achievements will be shown on your profile.

GitHub 토큰 생성



Search or jump to...



Pull requests

Issues

Codespaces

Marketplace

Explore



Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Beta

Tokens (classic)

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Tokens classic



© 2023 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)

GitHub 토큰 생성

Settings / Developer settings

- GitHub Apps
- OAuth Apps
- Personal access tokens
 - Fine-grained tokens Beta
 - Tokens (classic)


Personal access tokens (classic) Generate new token Revoke all

Tokens you have generated that can be used to access the Git

jiminToken — admin:enterprise, admin:gpg_key, admin:org, admin:org_

admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, gist, not,

write:packages

 This token has no expiration date.

Generate new token Beta

Fine-grained, repo-scoped

Generate new token (classic)

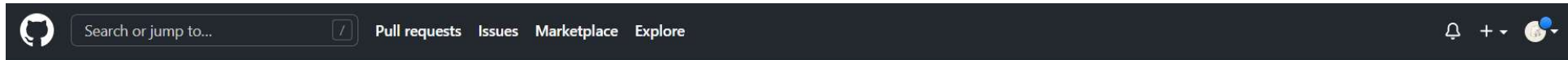
For general use

Delete

Generate new Token

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

GitHub 토큰 생성



Settings / Developer settings

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

myproject_token

토큰 이름 설정

What's this token for?

Expiration *

90 days

The token will expire on Mon, Oct 31 2022

엔간해서는 길게 길게 설정

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows

GitHub 토큰 생성

Select scopes 는 이 토큰을 통해 로그인했을시 이용 가능한 권한을 설정한다고 보면 된다.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org

스크롤 내리기

Delete 키워드가 들어간 항목을 제외하고 모두 체크

<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys (Developer Preview)
<input checked="" type="checkbox"/> write:gpg_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:gpg_key	Read public user GPG keys

Generate token Cancel

Generate token 클릭

GitHub 토큰 생성

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your new personal access token now. You won't be able to see it again!



토큰 생성됨



Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over [Basic Authentication](#).

**생성된 토큰을 VS Code나 Eclipse에서 로그인할 때 언제나 사용할 수 있도록
따로 복사해서 저장해놓는다. (로컬파일 뿐 아니라 본인 메일에도 저장하는 걸 추천)**

생성했던 토큰을 잊어버렸다면 그냥 다시 발급받으면 된다.

Code에 Github 연동



Github 최초 연동

ch01_helloworld > src > ch01_helloworld > HelloWorld.java

```

13  * 이름      | 주원
14  * 성명      | 서지민
15
16  https://github.com/RadeonHD6750/GitSample.git
17  *****/
18
19
20  //Commit 테스트
21  public class HelloWorld
22  {
23
24      public static void main(String[] args)
25      {
26          System.out.println("Hello World");
27      }
28
29
30

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> ls

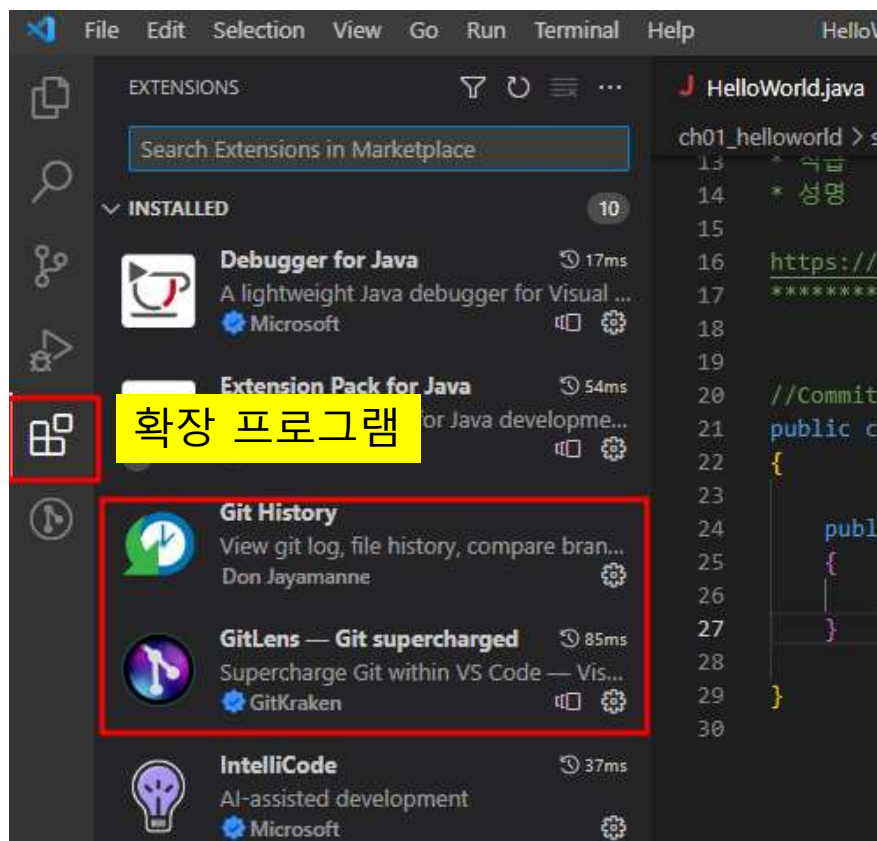
디렉터리: D:\Class\프로젝트 형상관리\GitTest\GitSample3

Mode	LastWriteTime	Length	Name
d----	2023-04-24 오전 10:43		ch01_helloworld

PS D:\Class\프로젝트 형상관리\GitTest\GitSample3>

git이 전혀 물어 있지 않은 순정상태의 code

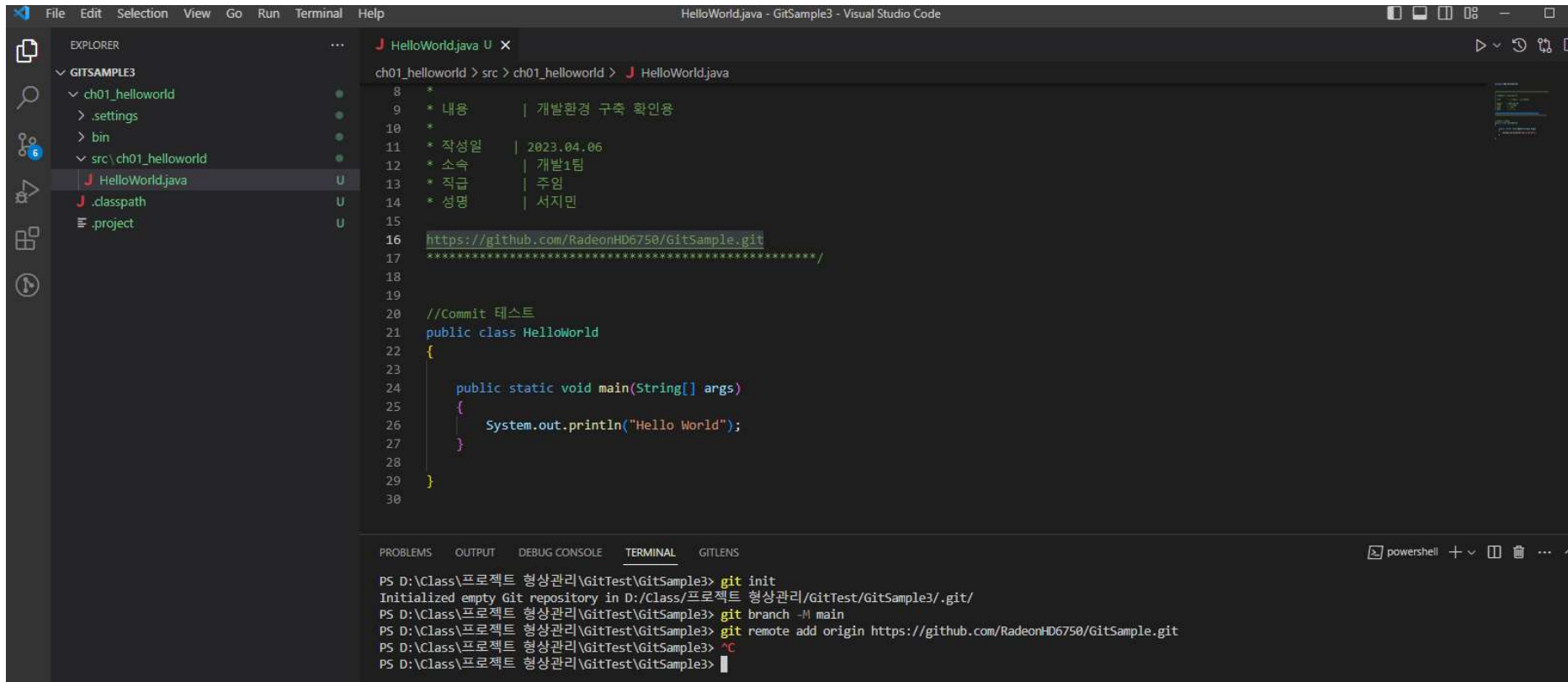
Github 최초 연동



1. 확장 프로그램 들어가기
2. Git History, GitLens, Graph 설치

필수는 아니지만, 삶을 좀 더 풍족하게 해준다.

Github 최초 연동



```
File Edit Selection View Go Run Terminal Help
HelloWorld.java - GitSample3 - Visual Studio Code

EXPLORER
GITSAMPLE3
├── ch01_helloworld
│   ├── .settings
│   ├── bin
│   └── src\ch01_helloworld
│       ├── HelloWorld.java
│       ├── .classpath
│       └── .project

ch01_helloworld > src > ch01_helloworld > HelloWorld.java

8
9 * 내용 | 개발환경 구축 확인용
10
11 * 작성일 | 2023.04.06
12 * 소속 | 개발1팀
13 * 직급 | 주임
14 * 성명 | 서지민
15
16 https://github.com/RadeonHD6750/GitSample.git
17 *****/
18
19
20 //Commit 테스트
21 public class HelloWorld
22 {
23
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28
29 }
30

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git init
Initialized empty Git repository in D:/Class/프로젝트 형상관리/GitTest/GitSample3/.git/
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git branch -M main
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git remote add origin https://github.com/RadeonHD6750/GitSample.git
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> ^C
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> |
```

1. git init
2. git branch -M main master가 아닌 main으로 하자
3. git remote add origin "깃허브 주소"

Github 최초 연동

```

J HelloWorld.java x
ch01_helloworld > src > ch01_helloworld > J HelloWorld.java
8  *
9  * 내용      | 개발환경 구축 확인용
10 *
11 * 작성일    | 2023.04.06
12 * 소속      | 개발1팀
13 * 직급      | 주임
14 * 성명      | 서지민
15
16 https://github.com/RadeonHD6750/GitSample.git
17 *****/
18
19
20 //Commit 테스트
21 public class HelloWorld
22 {
23
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git add .
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git commit -m "first"
[main (root-commit) d04fc5b] first
6 files changed, 66 insertions(+)
create mode 100644 ch01_helloworld/.classpath
create mode 100644 ch01_helloworld/.project
create mode 100644 ch01_helloworld/.settings/org.eclipse.core.resources.prefs
create mode 100644 ch01_helloworld/.settings/org.eclipse.jdt.core.prefs
create mode 100644 ch01_helloworld/bin/.gitignore
create mode 100644 ch01_helloworld/src/ch01_helloworld/HelloWorld.java
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (13/13), 1.63 KiB | 833.00 KiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RadeonHD6750/GitSample.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3>

```

3. git add .

4. git commit -m "메세지"

5. git push -u origin main

Github 최초 연동

RadeonHD6750 / **GitSample** Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file <> Code

RadeonHD6750 first d04fc5b 1 minute ago 1 commit

ch01_helloworld first 1 minute ago

Help people interested in this repository understand your project by adding a README. Add a README

About
깃 저장소 테스트용
0 stars
1 watching
0 forks

Releases
Create a new release

이제 간판 역할을 하는 README를 넣자

비어있던 github 원격 저장소에 작업중이던 내용물 업로드 완료


Github 최초 연동

```

1 # Gitsample
2 깃 저장소 테스트용
3
4
5 # 개정 이력
6
7 1. 2023-04-24, 최초 commit
    
```

READMD.md 파일 내용 작성

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



Commit new file

2023-04-24

최초 생성

☒ Commit directly to the `main` branch.
 ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file
Cancel

commit new file을 눌러 마저 진행

Github 최초 연동

RadeonHD6750 / **GitSample** Public
Pin Unwatch 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags
Go to file Add file Code

RadeonHD6750 2023-04-24 f3fd529 now 2 commits

ch01_helloworld	first	4 minutes ago
README.md	2023-04-24	now

README.md

GitSample

깃 저장소 테스트용

개정이력

1. 2023-04-24, 최초 commit

About

깃 저장소 테스트용

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

저장소에 간판역할을 하는 README.md 파일 생성완료

Github 최초 연동

```

File Edit Selection View Go Run Terminal Help
README.md - GitSample3 - Visual Studio Code

EXPLORER
GITSAMPLE3
├── ch01_helloworld
│   ├── .settings
│   ├── bin
│   └── src\ch01_helloworld
│       ├── HelloWorld.java
│       ├── .classpath
│       └── .project
└── README.md

HelloWorld.java README.md x

README.md > # 개정이력
You, 4 minutes ago | 1 author (You)
1 # GitSample
2 깃 저장소 테스트용
3
4
5 # 개정이력
6
7 1. 2023-04-24, 최초 commit You, 4 minutes ago + 2023-04-24 ...
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3> git pull origin
Updating d04fc5b..f3fd529
Fast-forward
 README.md | 7 ++++++
 1 file changed, 7 insertions(+)
 create mode 100644 README.md
PS D:\Class\프로젝트 형상관리\GitTest\GitSample3>
    
```

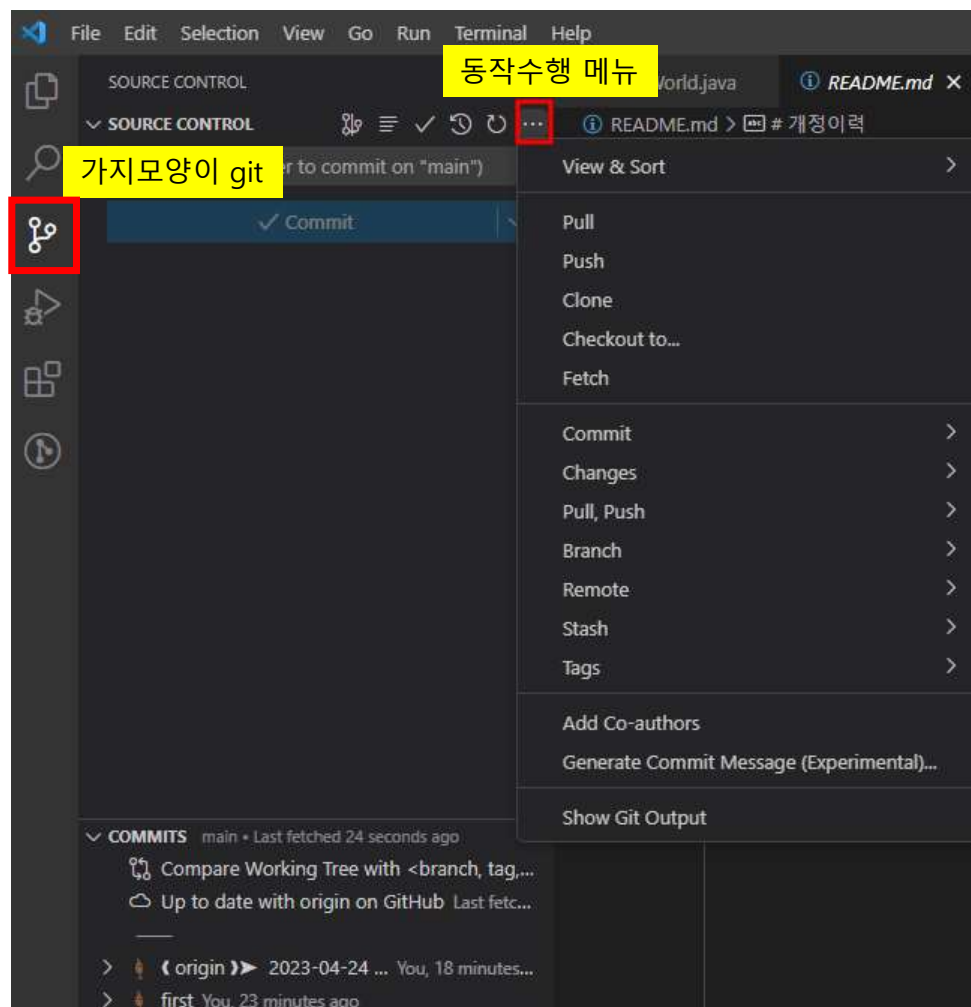
git pull origin을 하여 README.md 파일 다운로드 및 최종 동기화

이것을 안하면 나중에 reject의 원인으로 작용

Github으로 형상관리

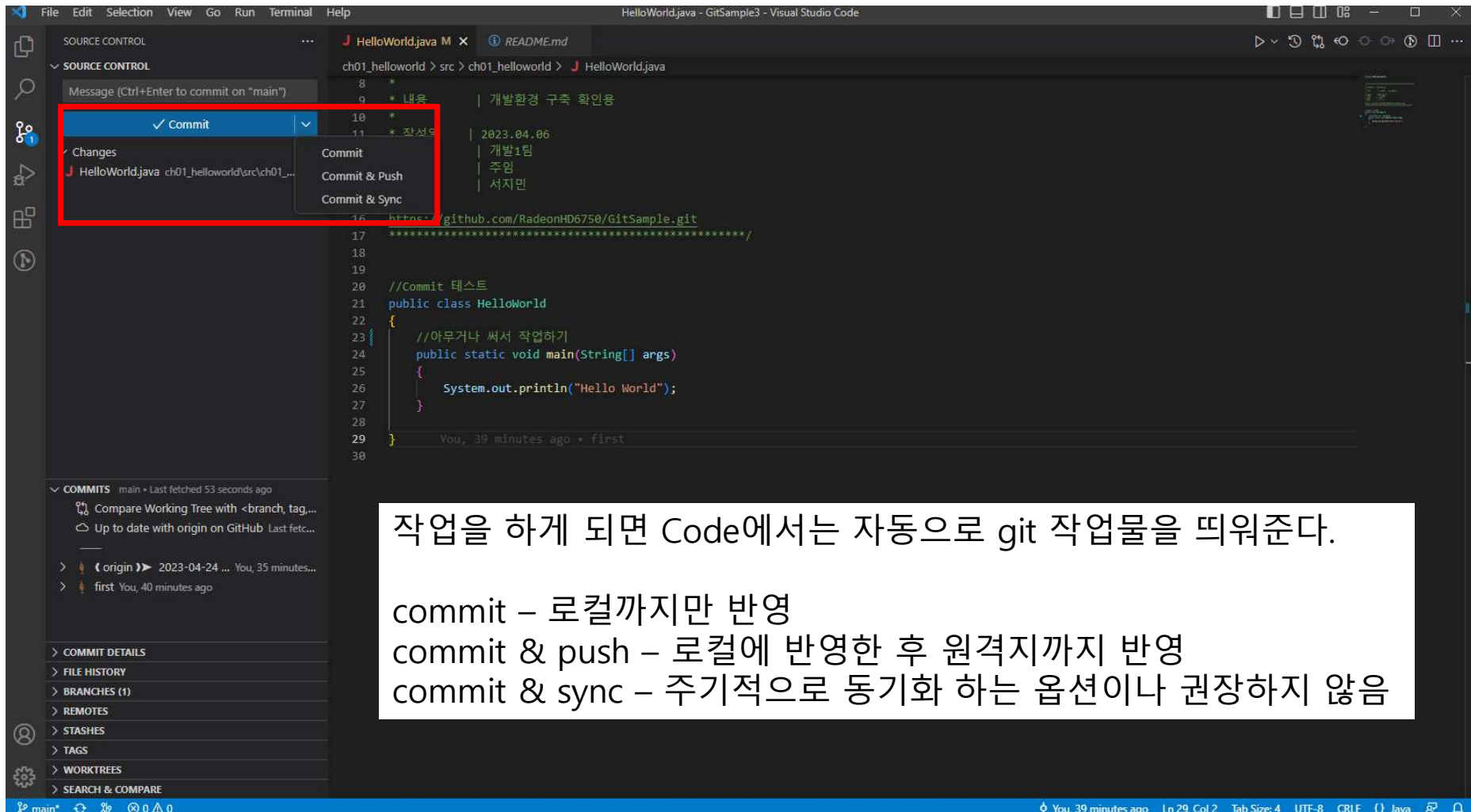


Visual Studio Code 활용



- Visual Studio Code에서는 기본적인 git GUI를 제공
- 앞서 했던 터미널 명령어는 초기설정에만 사용

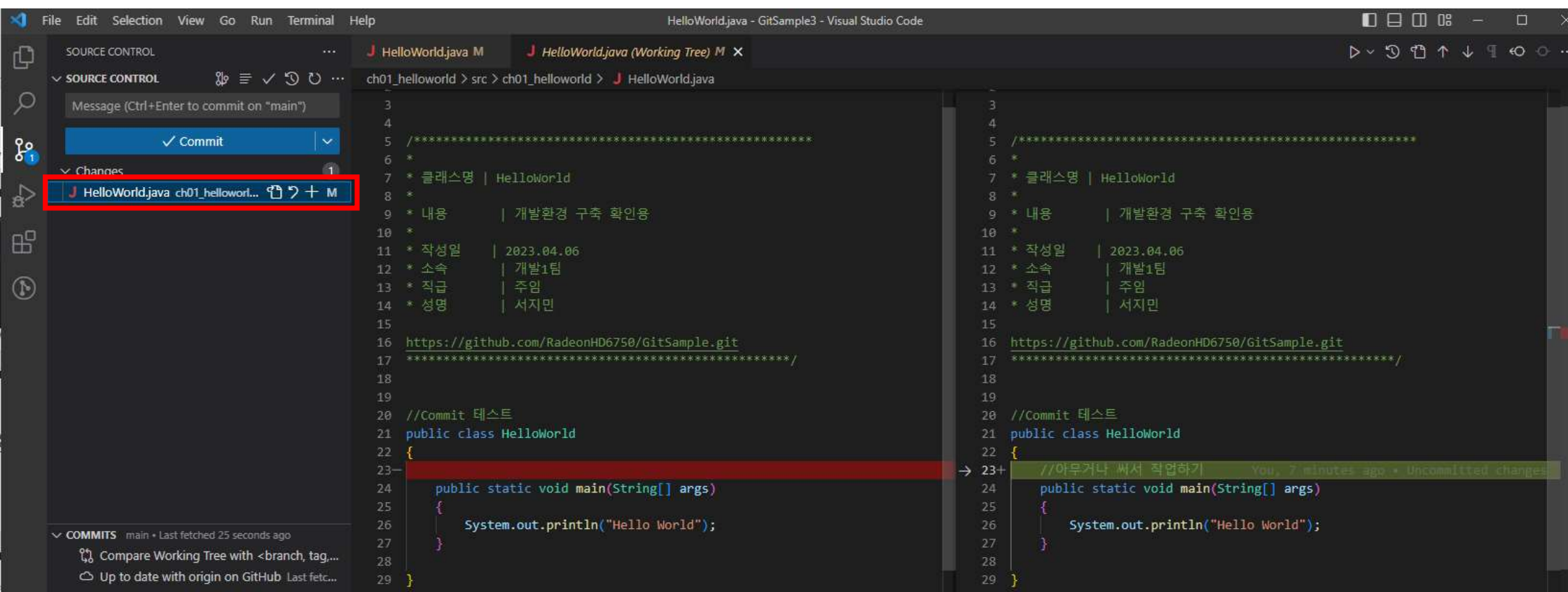
작업 후 업로드 하기



작업을 하게 되면 Code에서는 자동으로 git 작업물을 띄워준다.

- commit – 로컬까지만 반영
- commit & push – 로컬에 반영한 후 원격지까지 반영
- commit & sync – 주기적으로 동기화 하는 옵션이나 권장하지 않음

작업 후 업로드 하기



바로 commit & push하지 말고 Changes를 눌러 확인 후 진행

작업 후 업로드 하기

The screenshot shows the Visual Studio Code interface with the 'HelloWorld.java' file open. The left sidebar displays the 'Commit' button and a message field. The main editor shows the 'HelloWorld.java' file. A dialog box is open, asking 'There are no staged changes to commit. Would you like to stage all your changes and commit them directly?' with 'Yes', 'Always', 'Never', and 'Cancel' buttons. The 'Yes' button is highlighted.

commit 메시지는 친절하게

Message (Ctrl+Enter to commit on "main")

✓ Commit

Changes

1 HelloWorld.java ch01_helloworld...

COMMIT main • Last fetched 7 seconds ago

Compare Working Tree with <branch, tag...

Up to date with origin on GitHub Last fet...

Visual Studio Code

There are no staged changes to commit.

Would you like to stage all your changes and commit them directly?

Yes Always Never Cancel

한번에 commit 하고 바로 원격지까지 가겠냐는 질문인데 여기서는 yes

작업 후 업로드 하기

The screenshot shows the Visual Studio Code interface with the Source Control panel on the left. The commit message is "Commit 테스트" and the file "HelloWorld.java" is being committed to the "ch01_helloworld" branch. The commit is successful, as indicated by the green checkmark and the message "Commit successful".

```

3
4
5 /*****
6 *
7 * 클래스명 | HelloWorld
8 *
9 * 내용      | 개발환경 구축 확인용
10 *
11 * 작성일    | 2023.04.06
12 * 소속      | 개발1팀
13 * 직급      | 주임
14 * 성명      | 서지민
15
16 https://github.com/RadeonHD6750/GitSample.git
17 *****/
18
19
20 //Commit 테스트
21 public class HelloWorld
22 {
23
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28
29 }
30
  
```

commit이 완료된 모습

작업 후 업로드 하기

Repository: RadeonHD6750 / GitSample (Public)

Navigation: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Branch: main - GitSample / ch01_helloworld / src / ch01_helloworld / HelloWorld.java / <> Jump to -

Go to file ...

RadeonHD6750 2023-04-24 Latest commit bcbbe33 3 minutes ago History

1 contributor

29 lines (22 sloc) 546 Bytes

Raw Blame Edit Copy Delete

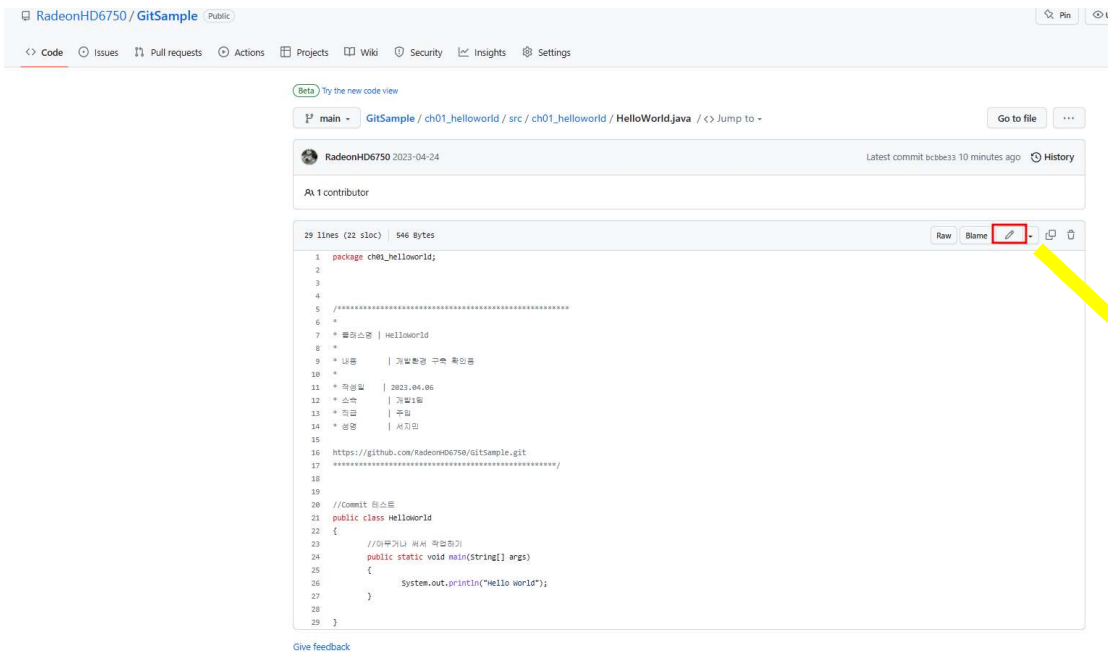
```

1 package ch01_helloworld;
2
3
4
5 /*****
6 *
7 * 클래스명 | HelloWorld
8 *
9 * 내용 | 개발환경 구축 확인용
10 *
11 * 작성일 | 2023.04.06
12 * 소속 | 개발1팀
13 * 직급 | 주임
14 * 성명 | 서지민
15
16 https://github.com/RadeonHD6750/GitSample.git
17 *****/
18
19
20 //Commit 테스트
21 public class HelloWorld
22 {
23     //이부분이나 써서 작업하기
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28 }
29

```

원격 저장소에도 반영된 모습

원격지에서 다운로드 받기



```

7  * 클래스명 | HelloWorld
8  *
9  * 내용    | 개발환경 구축 확인용
10 *
11 * 작성일   | 2023.04.06
12 * 소속    | 개발1팀
13 * 직급    | 주임
14 * 성명    | 서지민
15
16 https://github.com/RadeonHD6750/GitSample.git
17 *****
18
19
20 //Commit 테스트
21 public class HelloWorld
22 {
23     //아무거나 써서 작업하기
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28     //원격 저장소에서 직접 편집
29 }
30
31

```



Commit changes

2023-04-24, 원격지에서 편집

Add an optional extended description...

☒ Commit directly to the `main` branch.

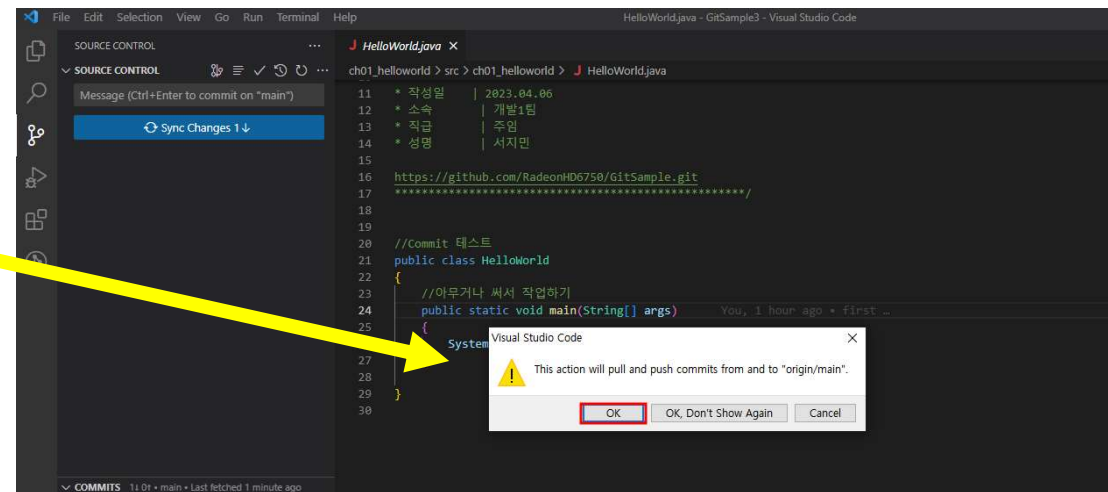
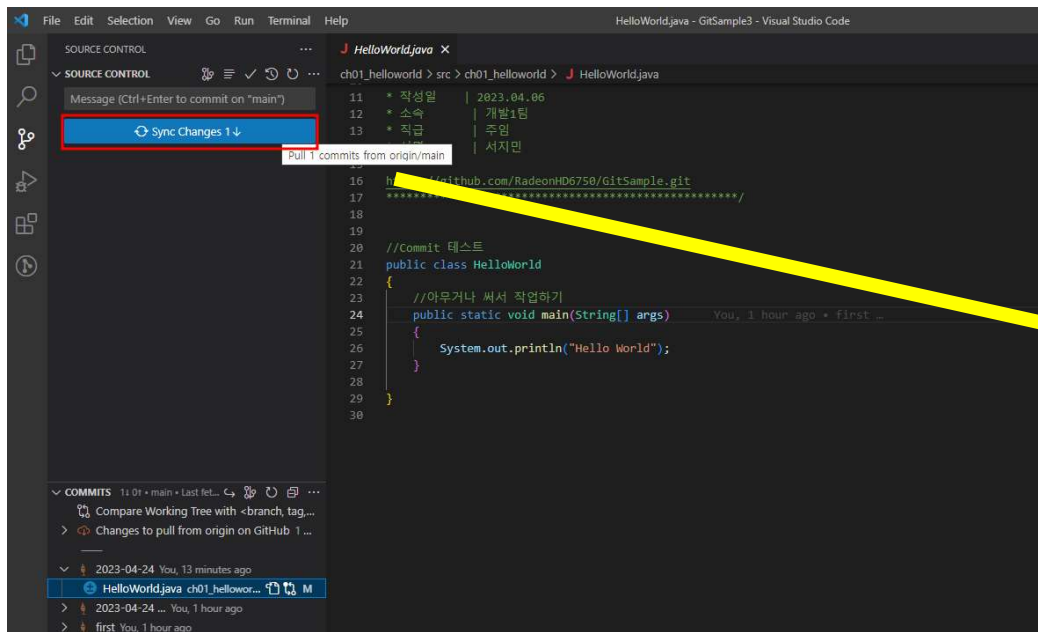
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

본래 다른 사람이나 다른 장소에서 commit을 했겠지만
여기서는 저장소에서 직접 편집

원격지에서 다운로드 받기



인터넷이 연결되어 있다면 자동으로 원격지와 비교 후 동기화 메시지

원격지에서 다운로드 받기

The screenshot shows the Visual Studio Code interface. On the left, the 'SOURCE CONTROL' panel is open, displaying a commit message dialog with the text 'Message (Ctrl+Enter to commit on "main")' and a 'Commit' button. The main editor area shows the file 'HelloWorld.java' with the following content:

```

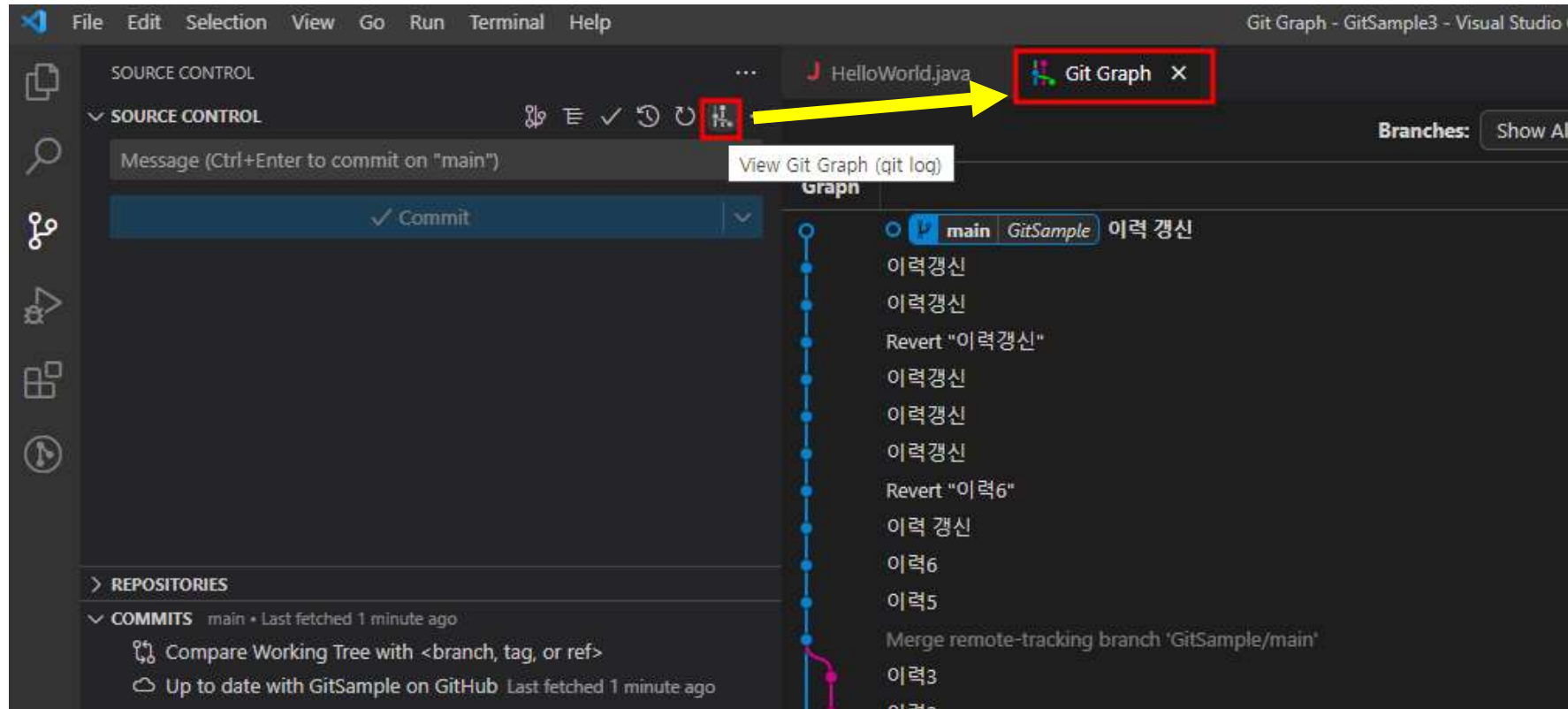
11  * 작성일      | 2023.04.06
12  * 소속        | 개발1팀
13  * 직급        | 주임
14  * 성명        | 서지민
15
16  https://github.com/RadeonHD6750/GitSample.git
17  *****/
18
19
20  //Commit 테스트
21  public class HelloWorld
22  {
23      //아무거나 써서 작업하기
24      public static void main(String[] args)
25      {
26          System.out.println("Hello World");
27      }
28      //원격 저장소에서 직접 편집
29  }
30
31
    
```

자동으로 동기화되어 다운로드 후 수정된 모습

Rollback

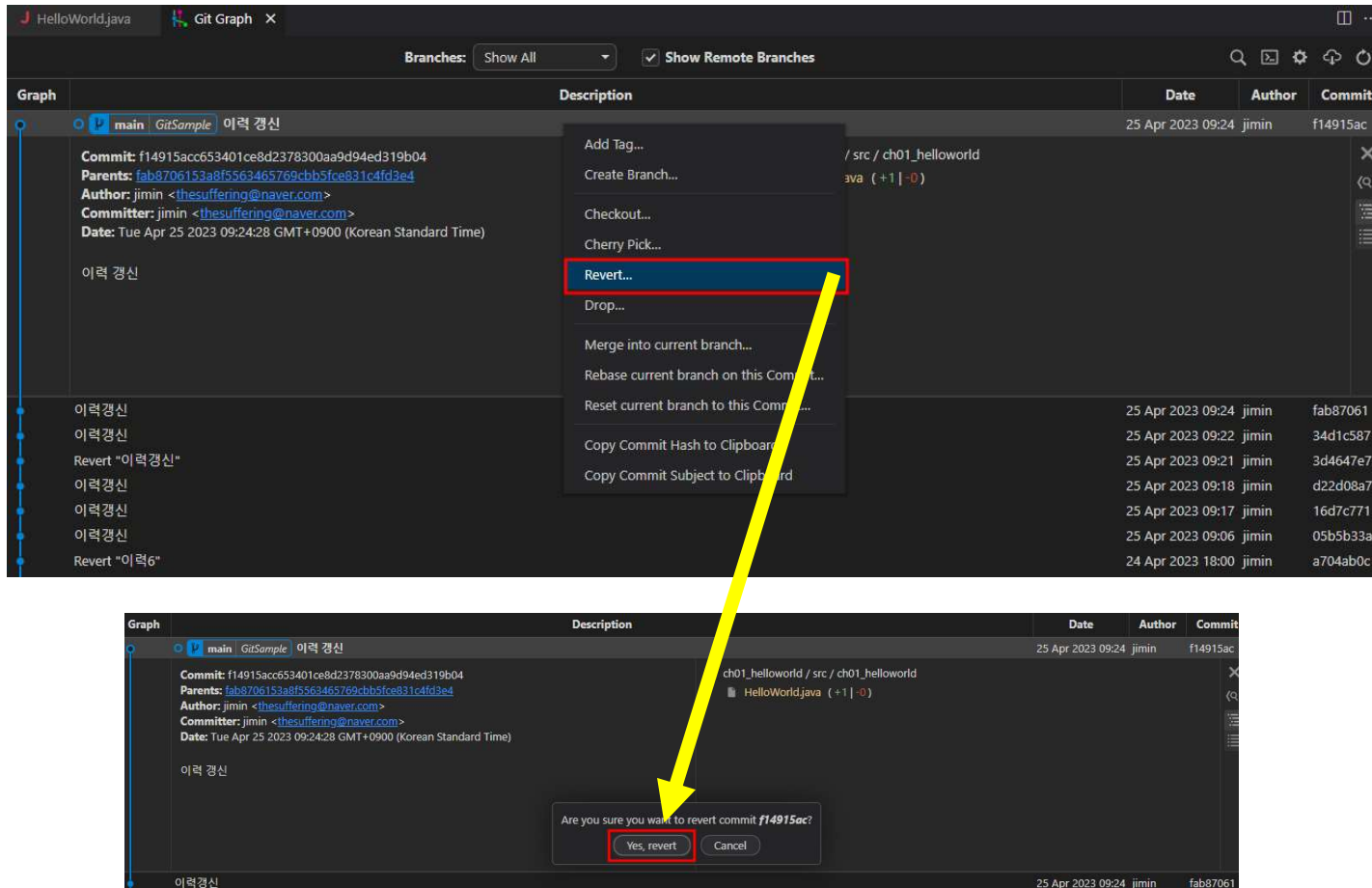


복구하기



확장 기능인 Git Graph를 사용하면 훨씬 편하게 복구가능

복구하기



취소하고 싶은 Commit에서 우클릭 후 Revert

복구하기

The screenshot shows the Visual Studio Code interface with the Git Graph extension. The left sidebar contains the 'SOURCE CONTROL' view with a 'Sync Changes 1 ↑' button highlighted by a red box. The main area displays the 'Git Graph' view for the 'main' branch, showing a commit titled 'Revert "이력 갱신"' also highlighted by a red box. The commit details include the hash, parents, author, committer, and date. The right sidebar shows the file 'HelloWorld.java' with a diff of +0 lines and -1 line.

Sync Changes 누르면 자동으로 원격 저장소까지 반영

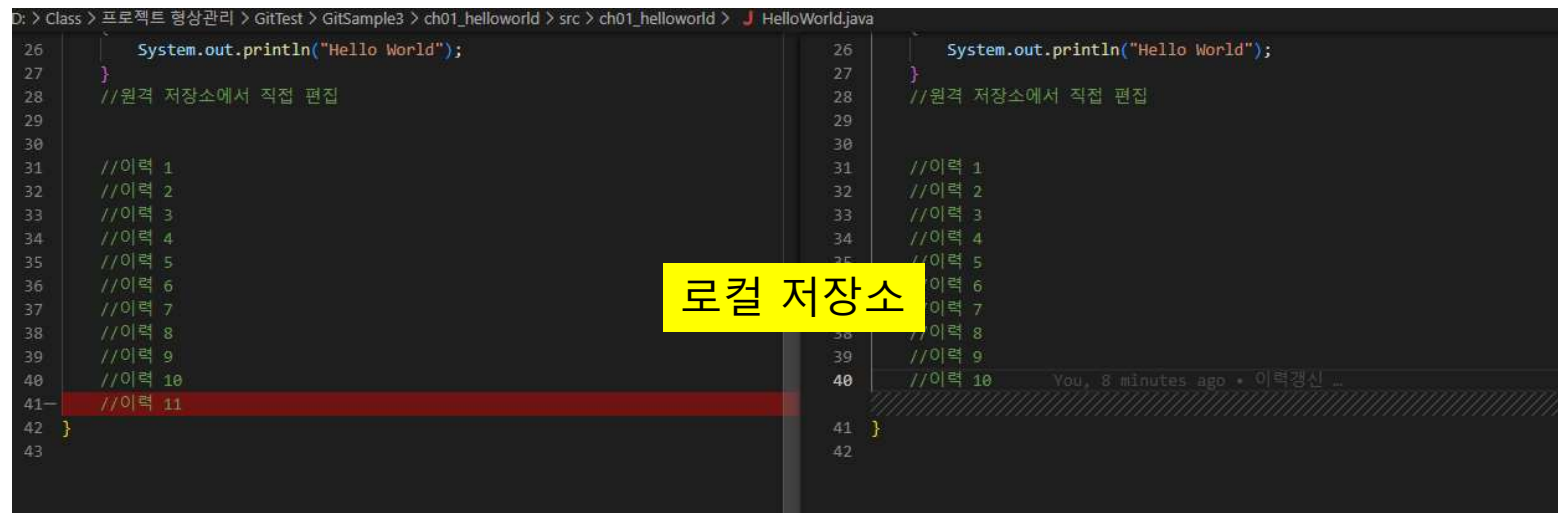
복구하기

```

20 //commit 테스트
21 public class HelloWorld
22 {
23     //아무거나 써서 작성하기
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28     //원격 저장소에서 직접 편집
29
30
31     //이력 1
32     //이력 2
33     //이력 3
34     //이력 4
35     //이력 5
36     //이력 6
37     //이력 7
38     //이력 8
39     //이력 9
40     //이력 10
41 }

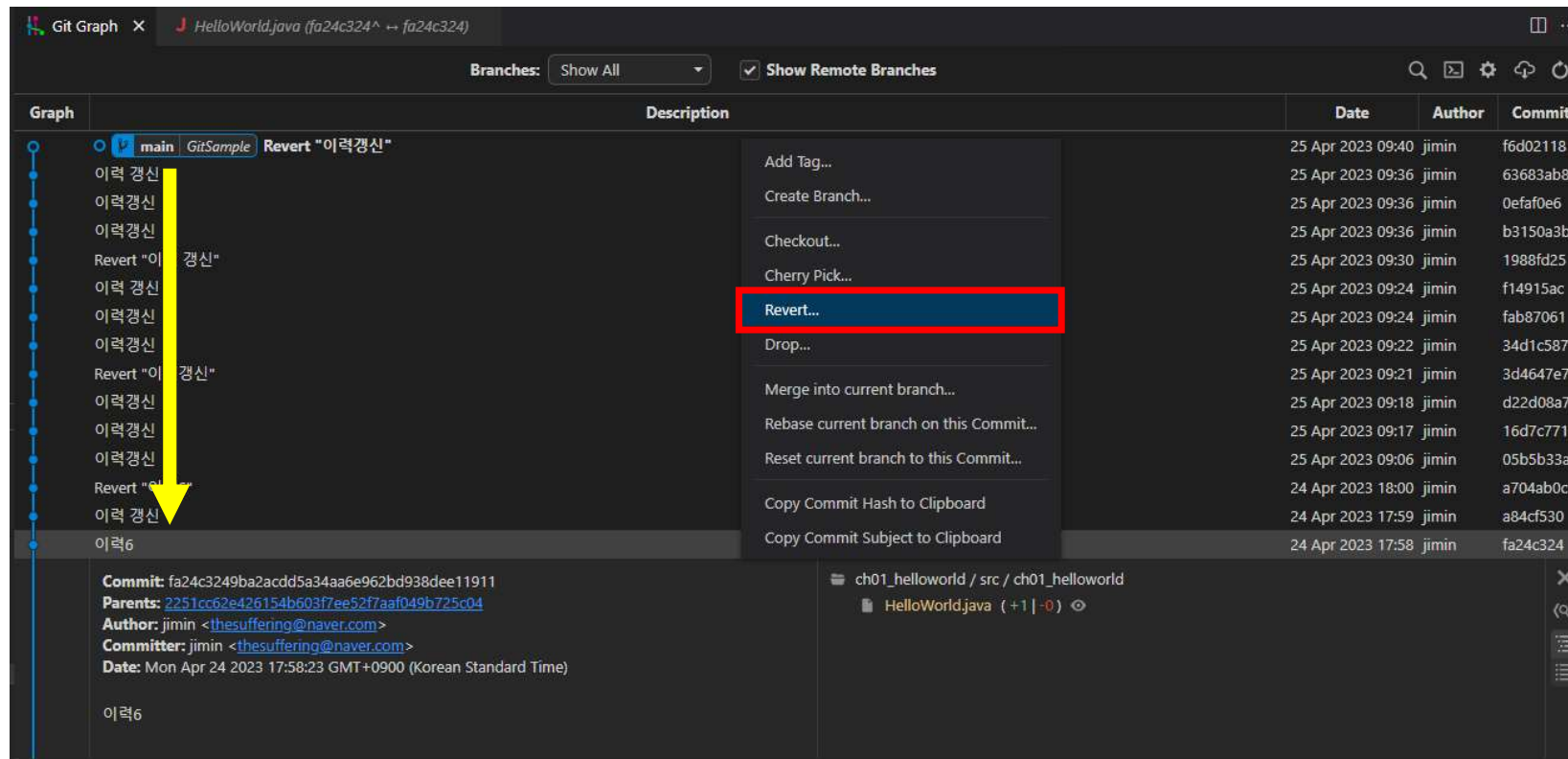
```

원격 저장소



로컬 저장소

머나먼 과거로 돌아가기



직전 단계 뿐 아니라 아예 머나먼 과거로 까지 돌아갈 수 있다.

마찬가지로 돌아가고자 하는 commit 바로 직후의 commit을 취소 시키면 된다.

머나먼 과거로 돌아가기

The screenshot shows the Visual Studio Code interface with the Git Graph extension. The left sidebar shows the 'SOURCE CONTROL' view with a list of commits. The main area shows the 'Git Graph' view with a commit history graph. A commit titled 'Revert "이력6"' is selected. The right sidebar shows the 'Description' of the selected commit, which includes an error message: 'Error: Unable to Revert Commit'. The error message states: 'error: could not revert fa24c32... 이력6', followed by hints for resolving conflicts and skipping the commit. The commit details at the bottom show the commit hash, parents, author, committer, and date.

Git Graph - GitSample3 - Visual Studio Code

Source Control

- ✓ SOURCE CONTROL
 - Revert "이력6"
 - This reverts commit fa24c3249ba2acdd5a34aa6e962bd938dee11911.
 - ✓ Commit
- ▼ Merge Changes
 - ch01_helloworld\src\ch01_helloworld
 - HelloWorld.java
- ▼ Changes

REPOSITORIES

- ▼ COMMITS main • Last fetched 5 seconds ago
 - Compare Working Tree with <branch, tag, or ref>
 - Up to date with GitSample on GitHub Last fetched 5 seconds ago

Git Graph

Graph

- main GitSample Revert "이력갱신"
- 이력 갱신
- 이력갱신
- 이력갱신
- Revert "이력 갱신"
- 이력 갱신
- 이력갱신
- 이력갱신
- 이력갱신
- 이력갱신
- Revert "이력6"
- 이력 갱신
- 이력6

Description

Error: Unable to Revert Commit

error: could not revert fa24c32... 이력6
 hint: After resolving the conflicts, mark them with
 hint: "git add/rm <pathspec>", then run
 hint: "git revert --continue".
 hint: You can instead skip this commit with "git revert --skip".
 hint: To abort and get back to the state before "git revert",
 hint: run "git revert --abort".
 Auto-merging
 ch01_helloworld/src/ch01_helloworld/HelloWorld.java
 CONFLICT (content): Merge conflict in
 ch01_helloworld/src/ch01_helloworld/HelloWorld.java

Commit: fa24c3249ba2acdd5a34aa6e962bd938dee11911
Parents: 2251cc62e426154b603f7ee52f7aaf049b725c04
Author: jimin <thesuffering@naver.com>
Committer: jimin <thesuffering@naver.com>
Date: Mon Apr 24 2023 17:58:23 GMT+0900 (Korean Standard Time)

ch01_helloworld / src / ch01_helloworld

HelloWorld.java (+1 | -0)

Revert 하자마자 위와 같은 오류가 발생할 것이다.

바로 너무 먼 과거여서 현재와 충돌한 것이다.

머나먼 과거로 돌아가기

The screenshot shows the Visual Studio Code interface with the following components:

- Source Control Panel (Left):** Displays the commit history. The commit '이력6' (fa24c3249ba2acdd5a34aa6e962bd938dee11911) is selected. The 'Merge Changes' section shows 'ch01_helloworld' and 'HelloWorld.java' with a conflict icon.
- Git Graph (Top):** Shows the commit history graph.
- Code Editor (Center):** Displays the 'HelloWorld.java' file. The code is as follows:


```

21 public class HelloWorld
22 {
23     //아무거나 써서 작업하기
24     public static void main(String[] args)
25     {
26         System.out.println("Hello World");
27     }
28     //원격 저장소에서 직접 편집
29
30
31     //이력 1
32     //이력 2
33     //이력 3
34     //이력 4
35
36     //이력 5
37     //이력 6
38     //이력 7
39     //이력 8
40     //이력 9
41     //이력 10
42
43     =====
44     >>>>>> parent of fa24c32 (이력6) (Incoming Change)
45
46 
```
- Conflict Resolution:** A yellow box highlights the conflict area (lines 35-41) with the text '충돌이 발생한 지점'. A red box highlights the 'Resolve in Merge Editor' button at the bottom right.

현재의 코드와 과거의 코드가 충돌한 것이니 Merge Editor를 사용하여 조정해주자

머나먼 과거로 돌아가기

Git Graph | HelloWorld.java ! | Merging: HelloWorld.java !

ch01_helloworld > src > ch01_helloworld > HelloWorld.java

Incoming | ... | Current

30
31 //이력 1
32 //이력 2
33 변경되는 지점
34
35
36 }
37

Accept Incoming | Accept Combination | Ignore

30
31 //이력 1
32 //이력 2
33 //이력 3
34 //이력 4
35
36
37
38
39
40
41 }
42

Accept Current | Accept Combination | Ignore

현재 상태

Result ch01_helloworld\src\ch01_helloworld\HelloWorld.java

30
31 //이력 1
32 //이력 2
33 //이력 3
34 //이력 4
35 No Changes Accepted
36 //이력 5
37 }
38

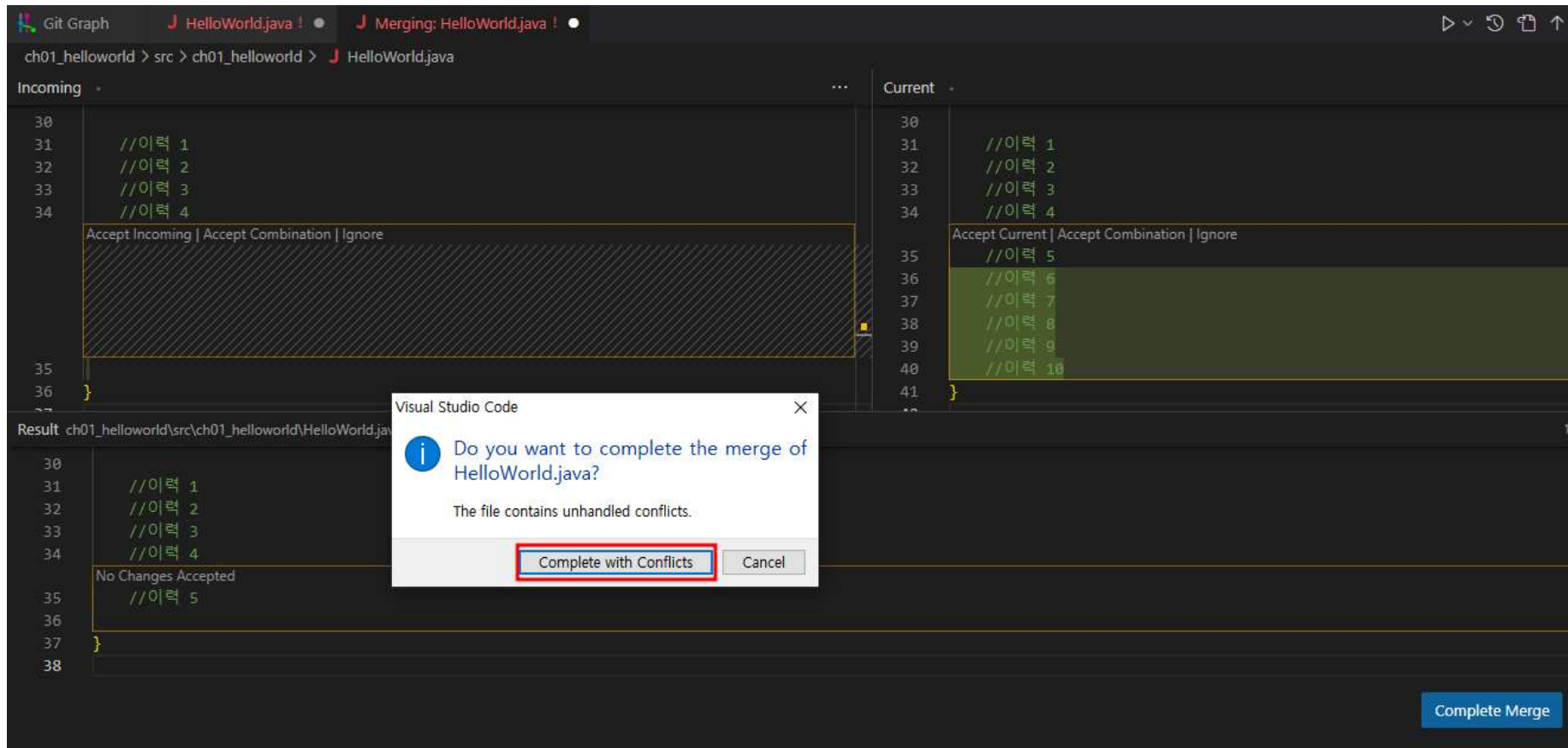
최종 결과

1 Conflict Remaining

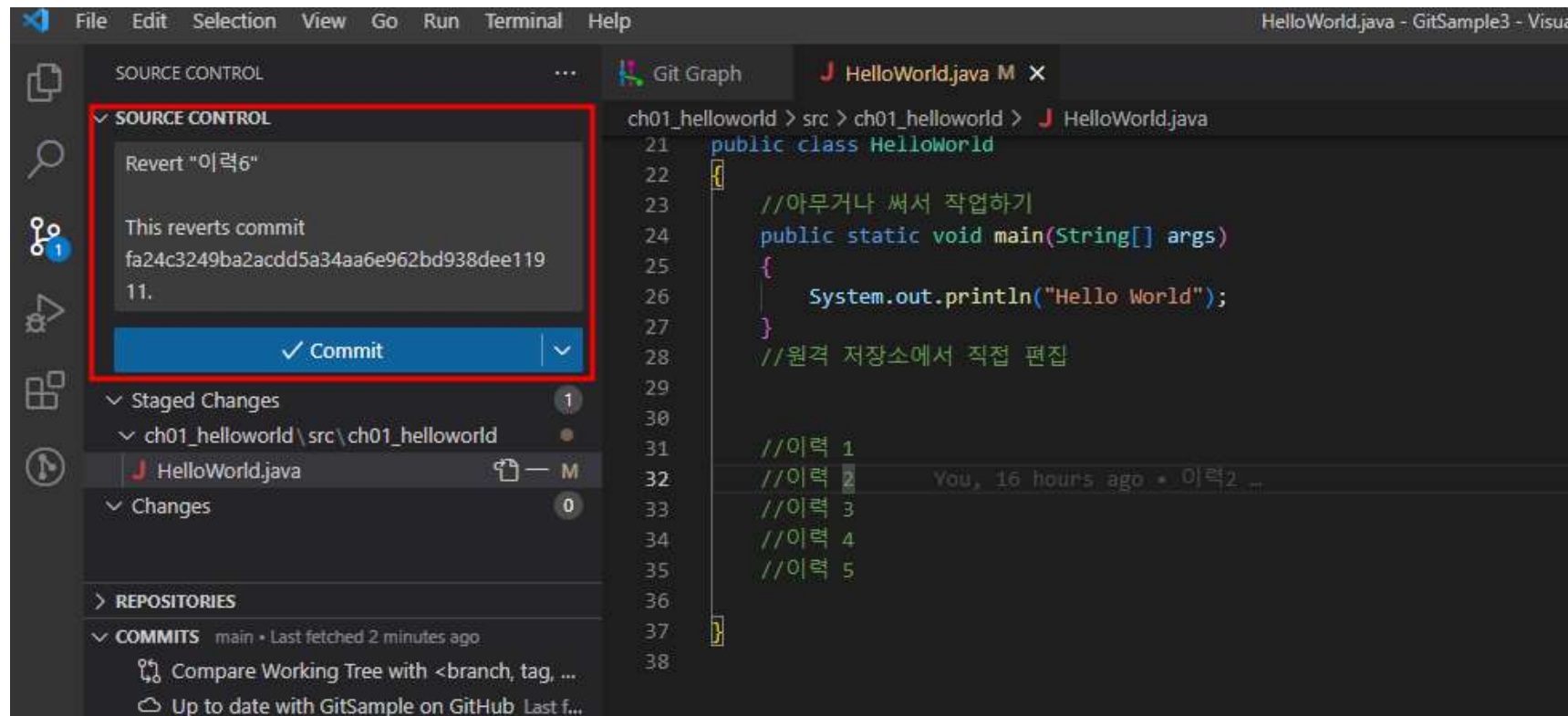
Complete Merge

현재 상태를 편집하거나 아니면 자동으로 반영된 결과를 보고 Merge를 하자

머나먼 과거로 돌아가기



머나먼 과거로 돌아가기



merge 이후에 마저 commit & push 해주도록 하자

머나먼 과거로 돌아가기

```
//Commit 테스트
public class HelloWorld
{
    //아무거나 써서 작업하기
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
    //원격 저장소에서 직접 편집

    //이력 1
    //이력 2
    //이력 3
    //이력 4
    //이력 5

}
```

이력 10까지 있었다가 훨씬 더 과거인 이력5로 돌아간 모습이다.

터미널로 직접 다루는 것 보다 어느 정도 절차가 짜여진 GUI가 실수를 방지해준다.