

Git + GitHub + SourceTree

셋팅 방법과 commit, push, pull 하는 방법

이클립스 세팅

나눠준 zip파일 압축해제 후 폴더 상태로 저장할 위치에 저장
⇒ 본인 pc 드라이브에 앞으로 공부할 때 사용할

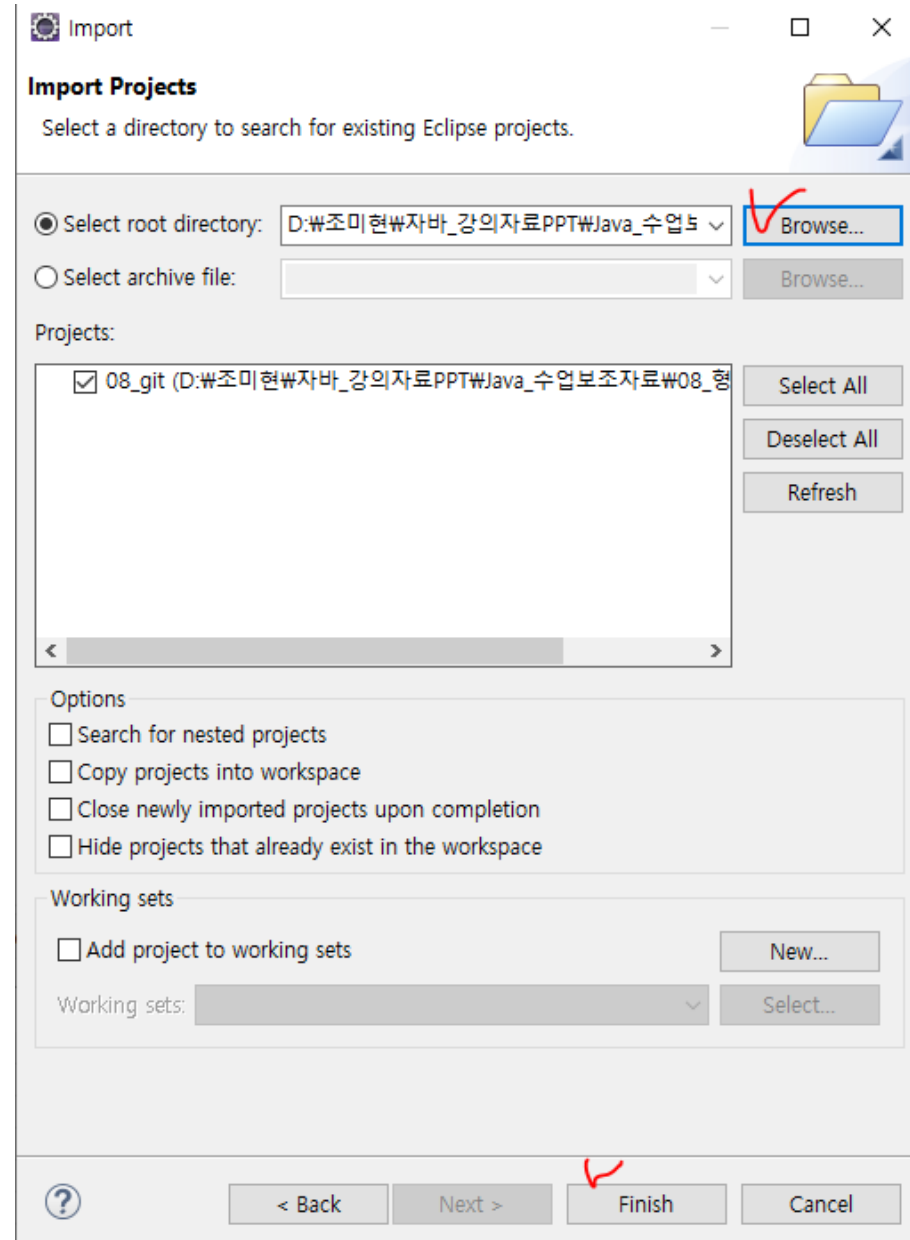
workspace 폴더 생성 후 그 안에 넣어두기

1. File > import..
2. Browse 선택 후 workspace 폴더에서 해당 폴더 선택
3. Projects란에 체크박스 선택 확인 후 Finish
4. 파일 안에 src > edu.kh.git.practice.GitPracticeRun.java 확인

.settings	2022-11-23 오후 4:28	파일 폴더		
bin	2022-11-23 오후 4:28	파일 폴더		
src	2022-11-23 오후 4:28	파일 폴더		
.classpath	2022-11-23 오후 4:28	CLASSPATH 파일	1KB	
.project	2022-11-23 오후 4:28	PROJECT 파일	1KB	

처음에 .git 폴더가 없는 상태

=> **Local repository가 생성되지 않은 상태**



Local repository에서 git 사용

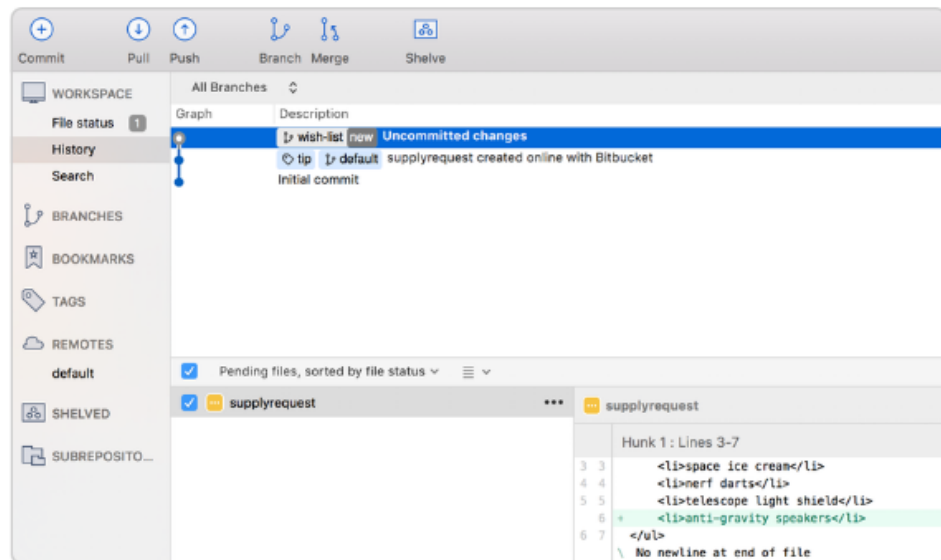
With Sourcetree



소스트리 셋팅하기

<https://www.sourcetreeapp.com/> 접속 및 다운로드

Next up: Learn Sourcetree with Bitbucket



[Step 1: Create a Git repository](#)

[Step 2: Copy your repository and add files](#)

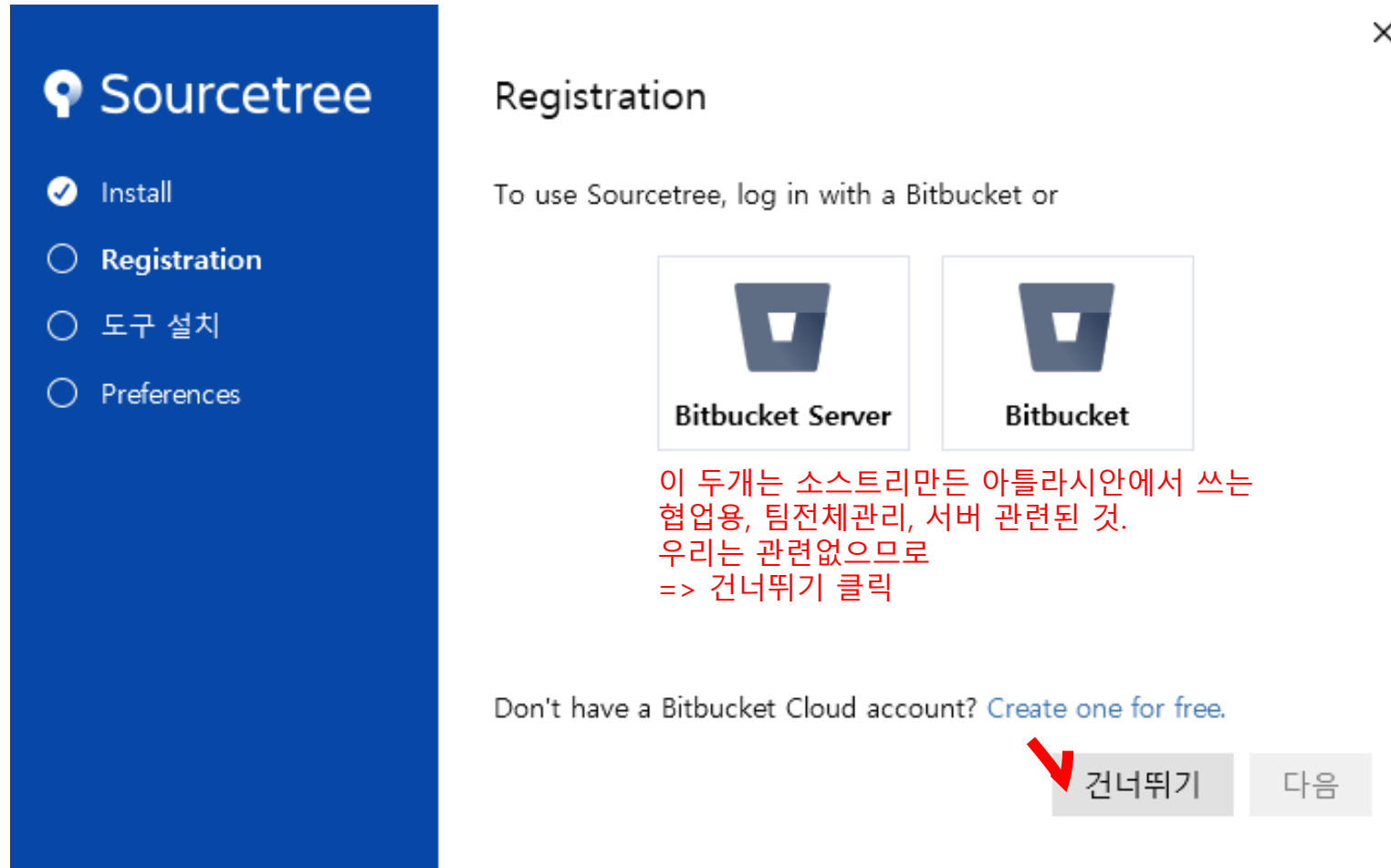
[Step 3: Pull changes from your repository](#)

[Step 4: Use Sourcetree branches to merge an update](#)

사용법

1. 깃 레파지토리 생성
2. 저장소 복사 및 파일 추가
3. 저장소에서 변경사항 가져오기
4. 소스트리 분기 사용하여 업데이트를 병합

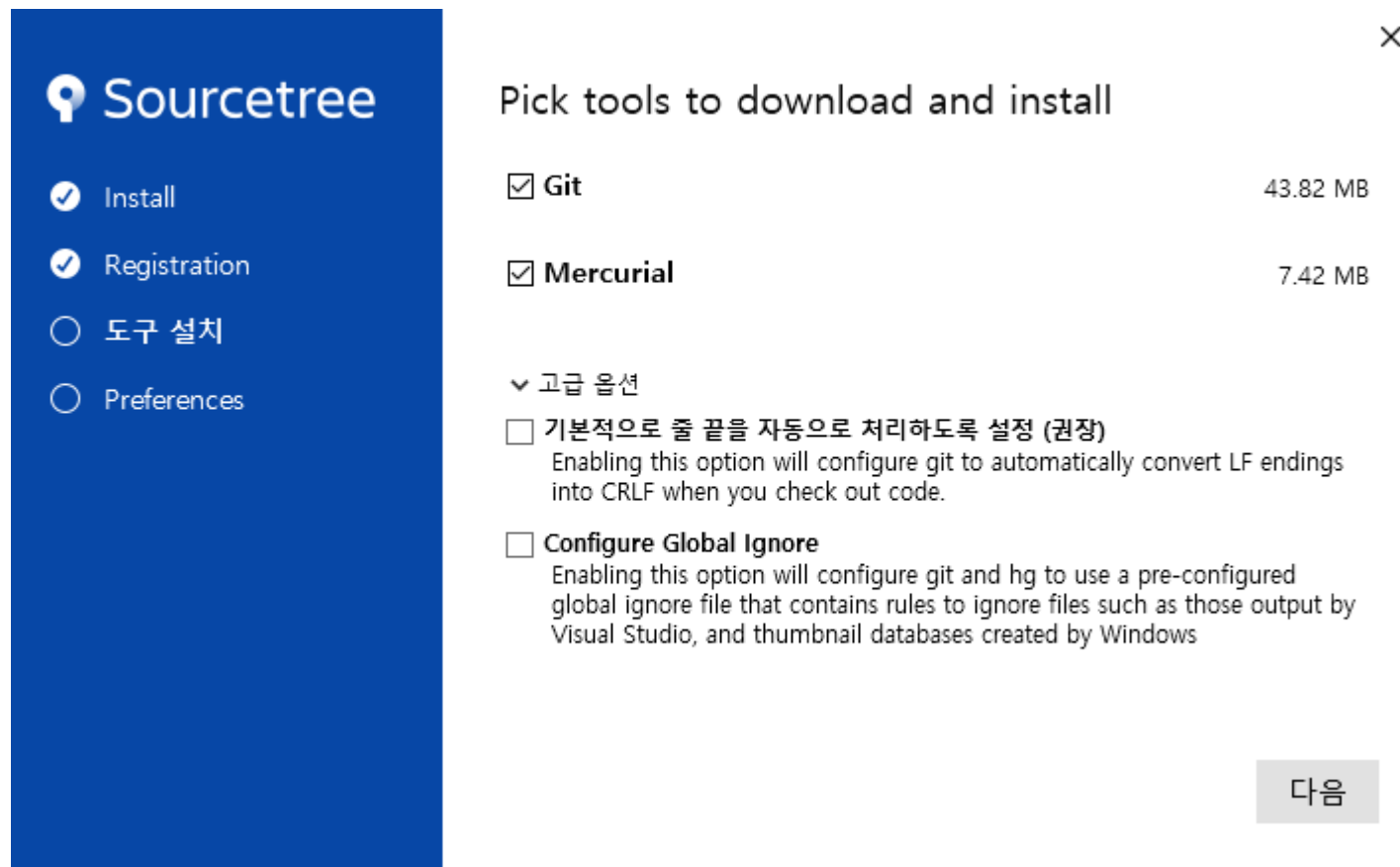
CLI보다 쉽게 GUI형태로 git을 사용하게 해줌



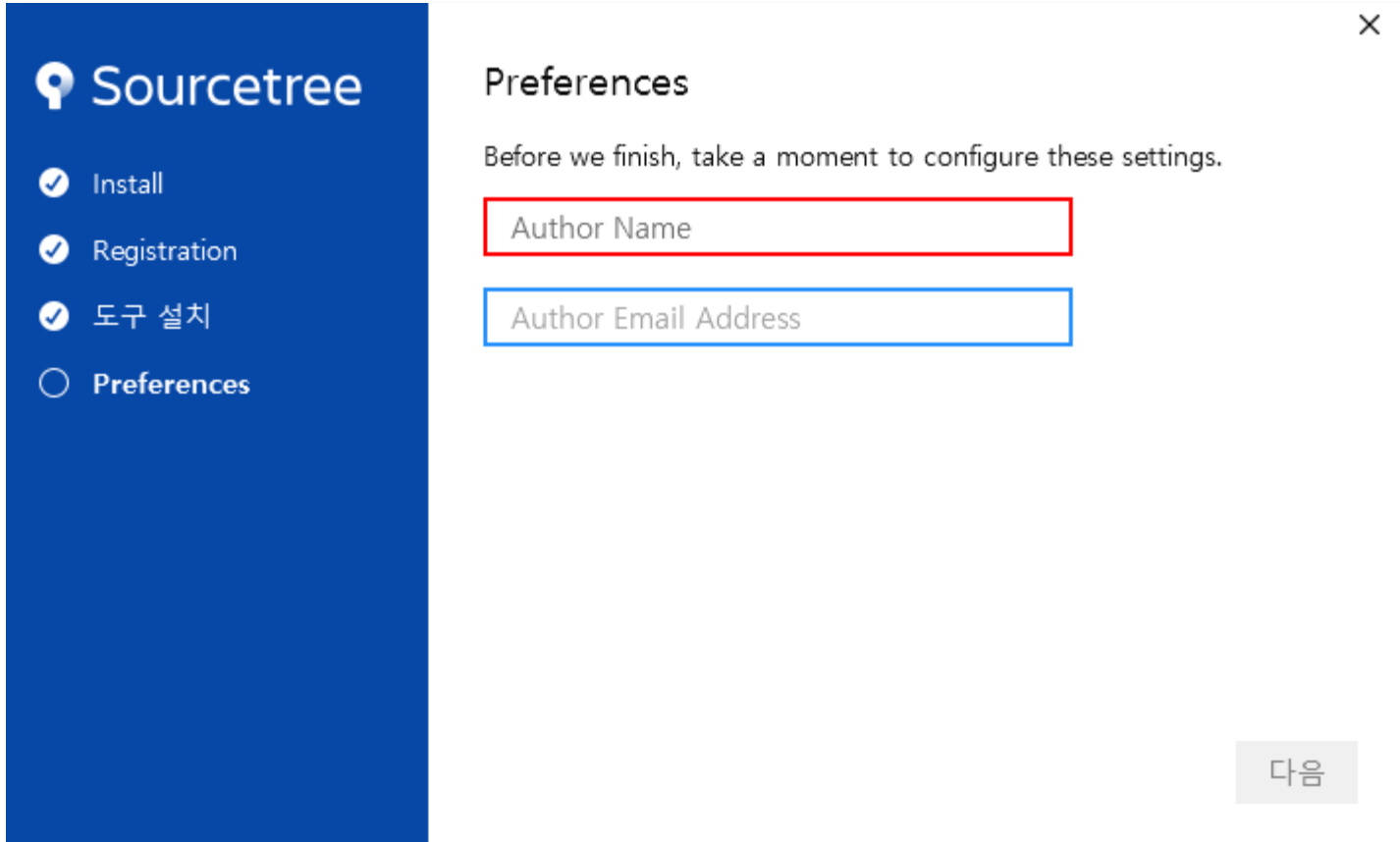
해당 창이 뜨면 Good!

고급옵션 추가 X Git, Mercurial만 체크확인하기

- 중간에 에러 발생 시 취소 or 다음 클릭하면 다시 실행.
=> 머큐리얼 관련된 에러 (중요하지 않음.)



Github 사용할 때 변경내역을 누가 올렸는지 내역에 함께 올라갈 이름과 이메일.

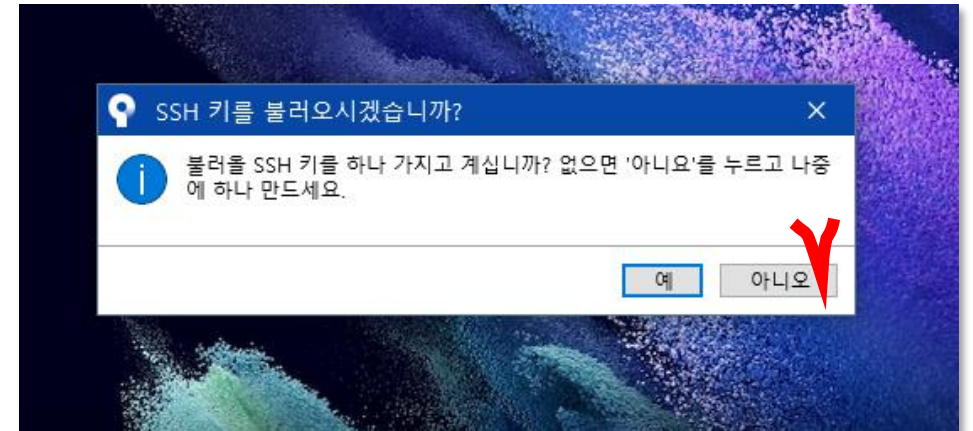


이름과 이메일 주소 작성 후 다음 누르면
아래처럼 SSH키 알림창이 뜨는데 여기서
'아니오' 클릭
⇒ 우리는 OAUTH 사용

Note!

시큐어 셸(Secure SHell, **SSH**) :
보안 용어로, 암호화 기술이 적용.
윈도우 자체 기능이나 별도 프로그램기능을 이용해서 SSH Key를 생성할 수 있는데,
나를 인증할 수 있는 인증수단이 저장되어있는 파일이 만들어짐.

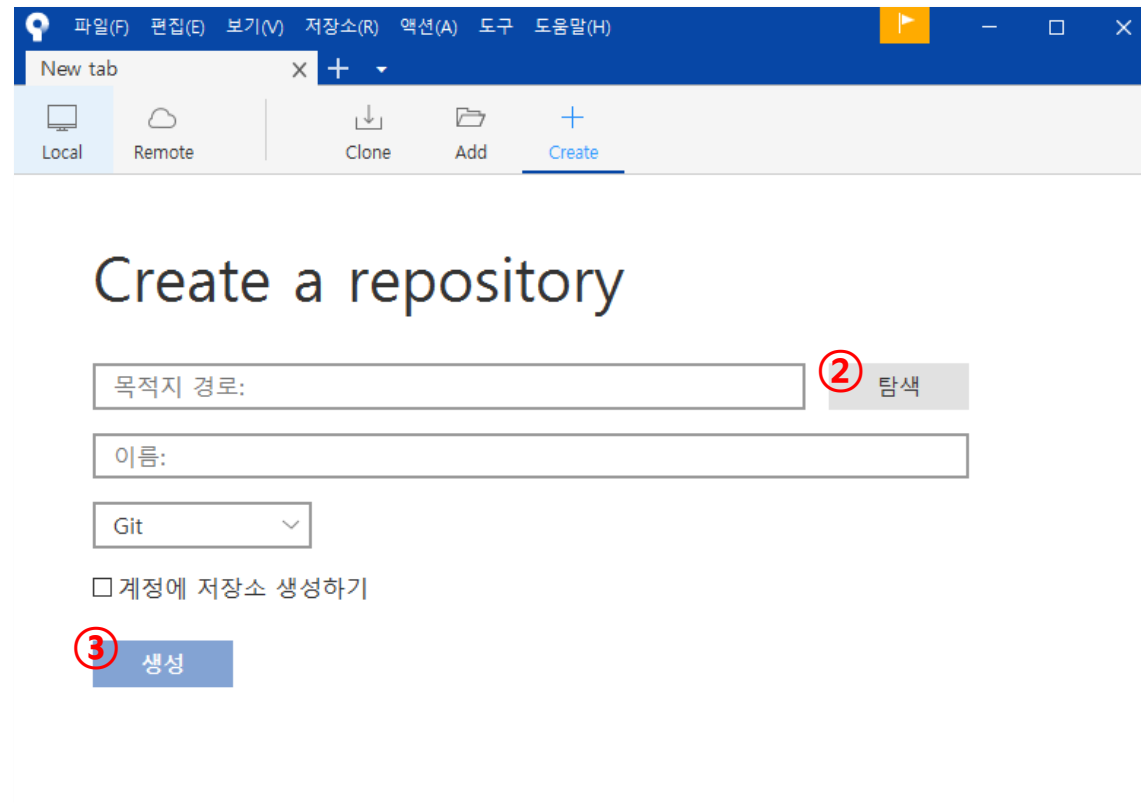
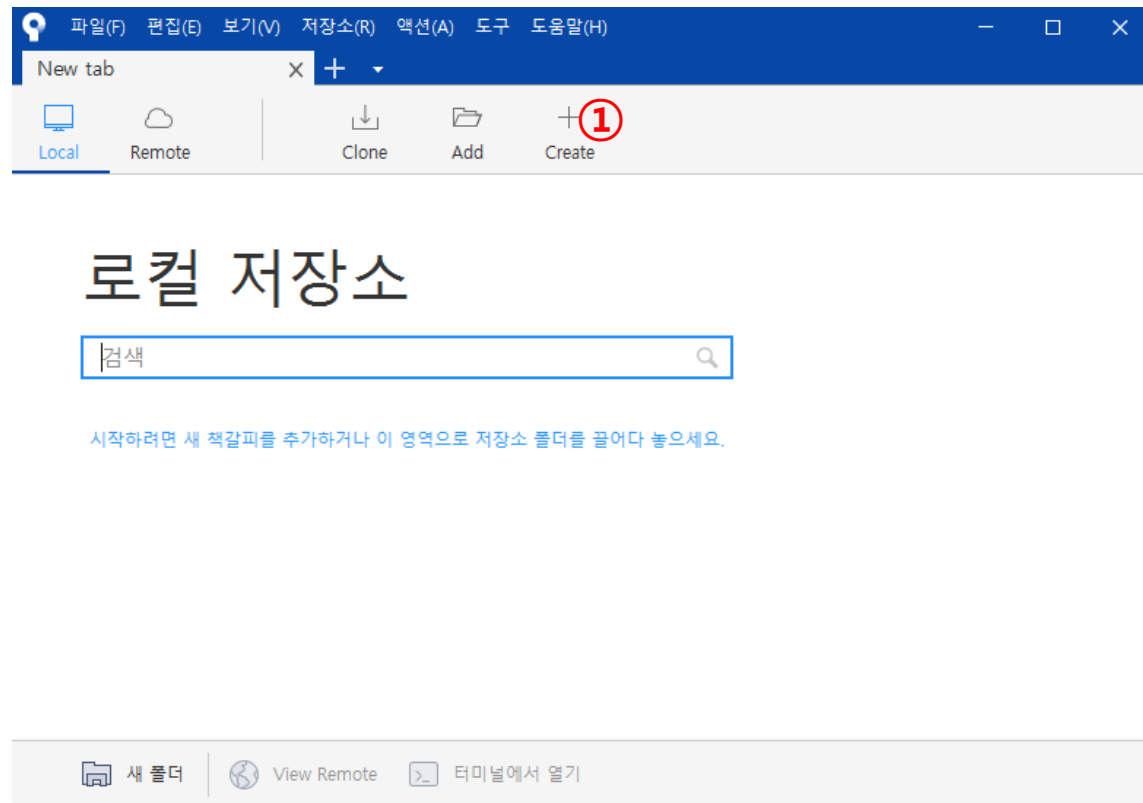
- ⇒ 이 파일을 불러오는 순간 아이디나 패스워드 없이도 자동으로 '나'라는 것이 인증됨.
- ⇒ 이것을 github에 로그인안하고 연결할 수 있다는 알림창임
- ⇒ 요즘은 패스워드만으로는 보안이 약해서 ssh나 oauth같은 인증방식을 따로 사용하는데 우리는 ssh말고 이 oauth를 사용함.

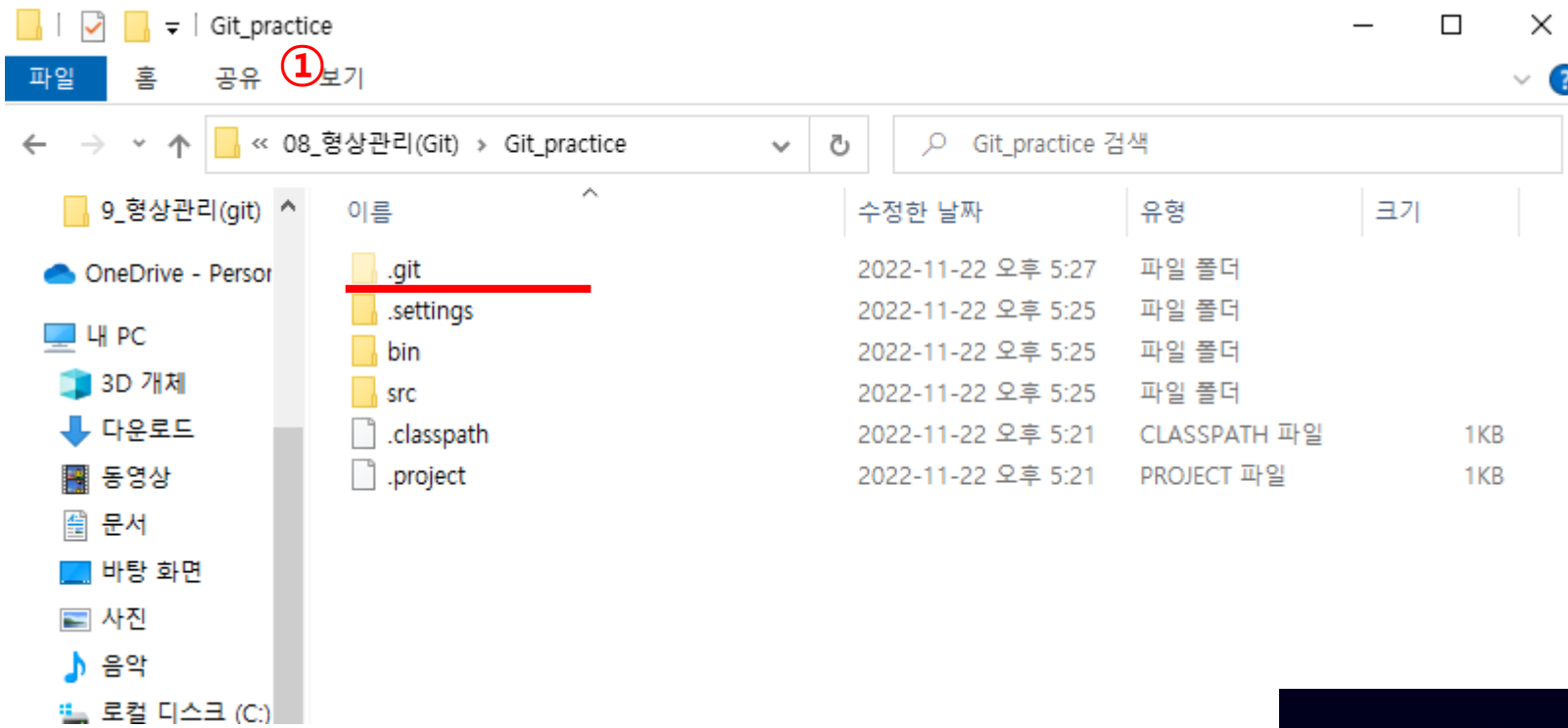


현재 컴퓨터에 저장: local
Github(원격저장소): remote
지금은 둘 다 없는 상태

컴퓨터에 local 저장소 만들기

1. Create 버튼 클릭
 2. 탐색 버튼 클릭 후 git에 올려 **형상관리 할 폴더 선택.(workspace에 저장된 폴더)**
 3. 계정에 저장소 생성하기 **체크하지 말고** 생성버튼 클릭
- > 경로상의 이름과 저장소 이름이 같아서 뜨는 알림창 '예' 누르기

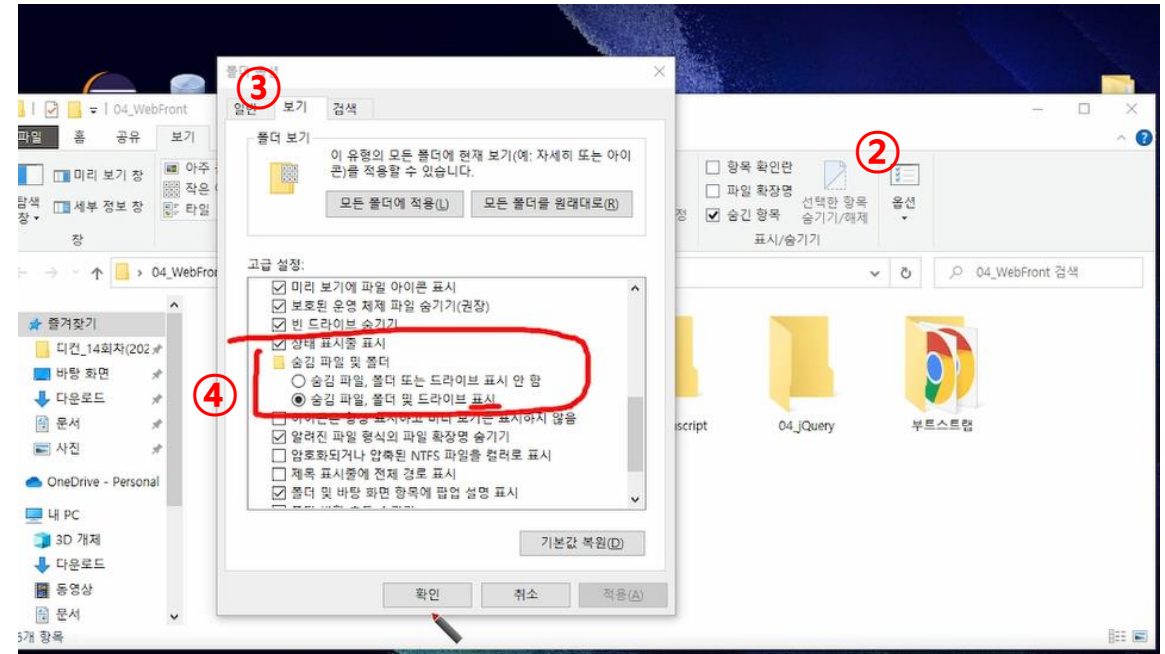




.git 생성됨

보이지 않으면 숨겨진 상태(. 점은 숨김파일이라는 뜻)

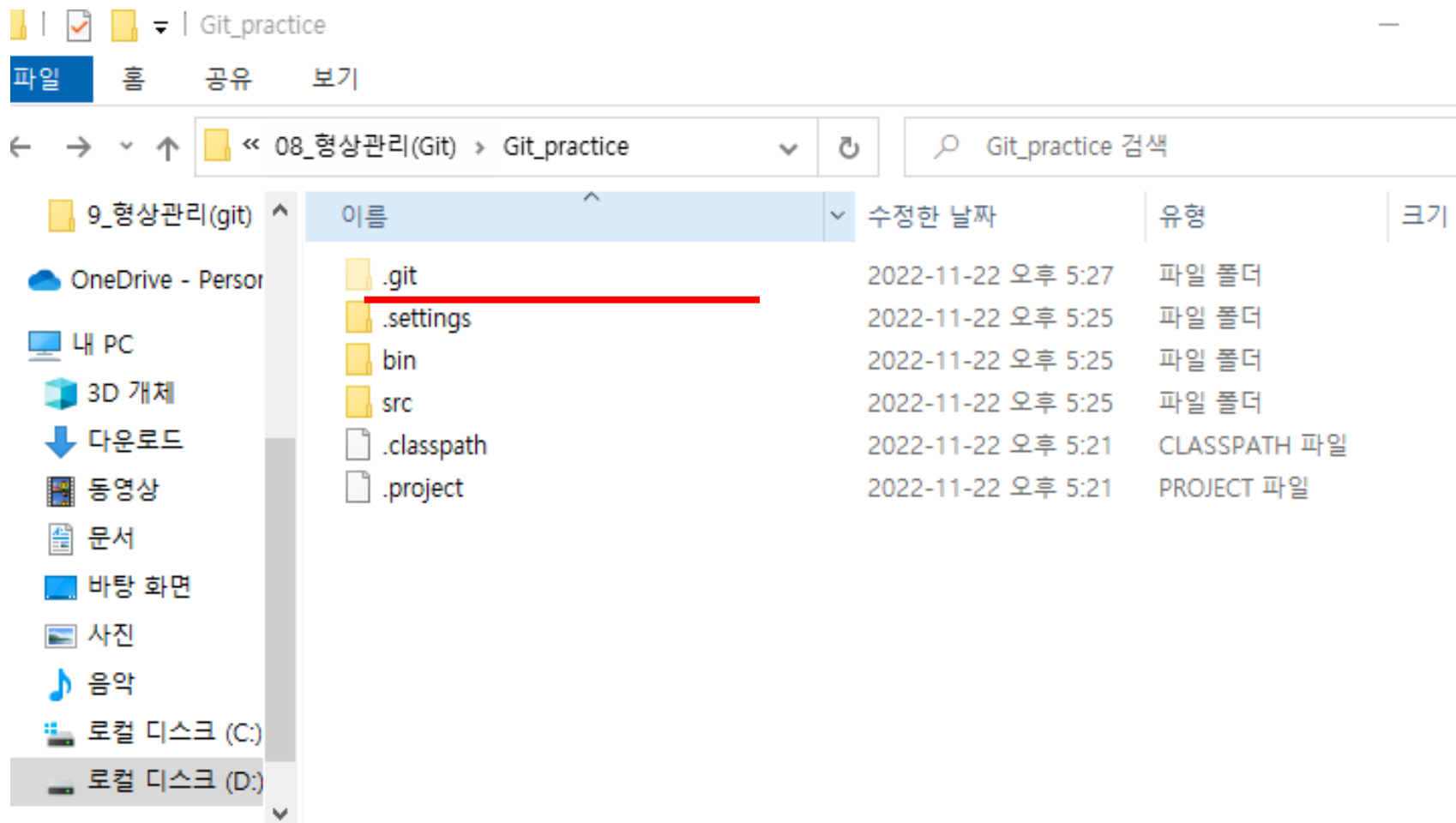
- ⇒ 윈도우는 위에
- ⇒ 1. 보기 탭 클릭
- ⇒ 2. 옵션 클릭
- ⇒ 3. 뜬 창에서 다시 보기 탭 클릭
- ⇒ 4. 고급설정 아래쪽에 숨김 파일 및 폴더에서 드라이브 표시로 바꿈



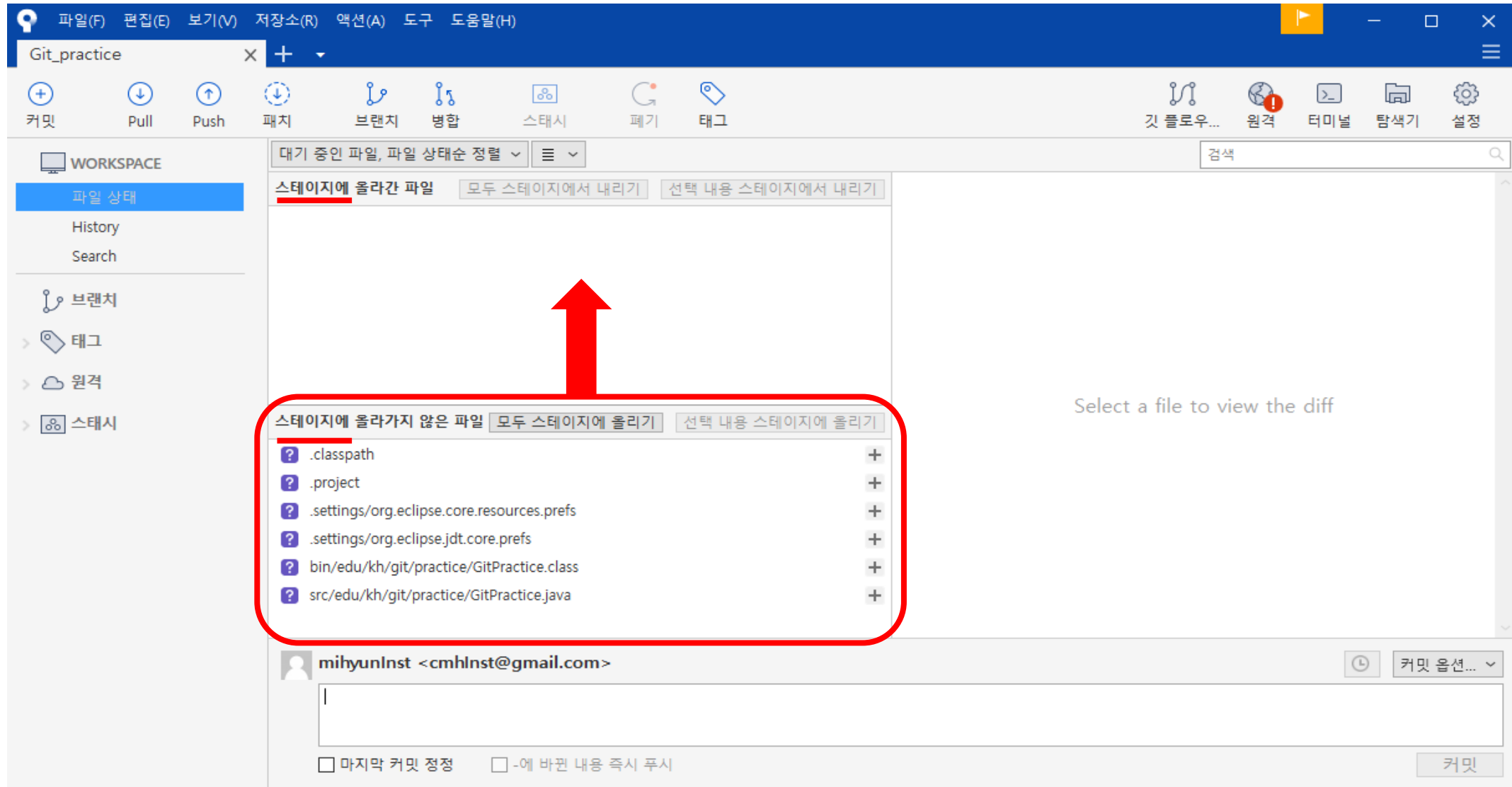
.git 파일 안에는 여러가지 파일들이 있는데, 어떤 내용이 누구에 의해 언제 변경되었는가?에 대한 내역 즉, 히스토리가 저장된 local repository가 됨.

.git 파일 지워지면 local repository에 저장된 내역 모두 삭제됨. 삭제하지 않도록 하기(건들지않기)

=> 여기까지 완료되면 local repository가 만들어지고, 해당 폴더가 앞으로 git에 의해 형상관리가 된다



소스트리에서 commit 하기



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

Git_practice

커밋 Pull Push 패치 브랜치 병합 스테시 폐기 태그

깃 플로우... 원격 터미널 탐색기 설정

WORKSPACE

파일 상태
History
Search

브랜치
태그
원격
스테시

대기 중인 파일, 파일 상태순 정렬

스테이지에 올라간 파일
모두 스테이지에서 내리기
선택 내용 스테이지에서 내리기

.classpath

.project

.settings/org.eclipse.core.resources.prefs

.settings/org.eclipse.jdt.core.prefs

bin/edu/kh/git/practice/GitPractice.class

src/edu/kh/git/practice/GitPractice.java

스테이지에 올라가지 않은 파일
모두 스테이지에 올리기
선택 내용 스테이지에 올리기

Select a file to view the diff

mihyunInst <cmhlnt@gmail.com>

커밋 메세지는 최대한 간결하되, 어떤 작업을 했는지 알 수 있도록 작성.

커밋 옵션...

커밋 메시지(commit message)를 꼭 작성하셔야 합니다~!
=> "First commit"

마지막 커밋 정보
이제 다른 내용 즉시 푸시

커밋

Git_practice

커밋 Pull Push 패치 브랜치 병합 스테시 페기 태그

WORKSPACE
파일 상태
History
Search

브랜치
master
태그
원격
스테시

모든 브랜치 원격 브랜치 날짜순으로 정렬

그래프 설명 날짜 작성자 커밋

master First commit 22 11 2022 17:58 mihyunInst <cmhInst@gmail.com> dc642a5

Master 브랜치 :저장소 최상위 관리자

파일 상태순 정렬

커밋: dc642a5c144eb891bebbce1211230dcd796eab72 [dc642a5]
작성자: mihyunInst <cmhInst@gmail.com>
날짜: 2022년 11월 22일 화요일 오후 5:58:23
커밋한 사람: mihyunInst

First commit

1 .classpath
2 .project
3 .settings/org.eclipse.core.resources.prefs
4 .settings/org.eclipse.jdt.core.prefs
5 bin/edu/kh/git/practice/GitPractice.class
6 src/edu/kh/git/practice/GitPractice.java

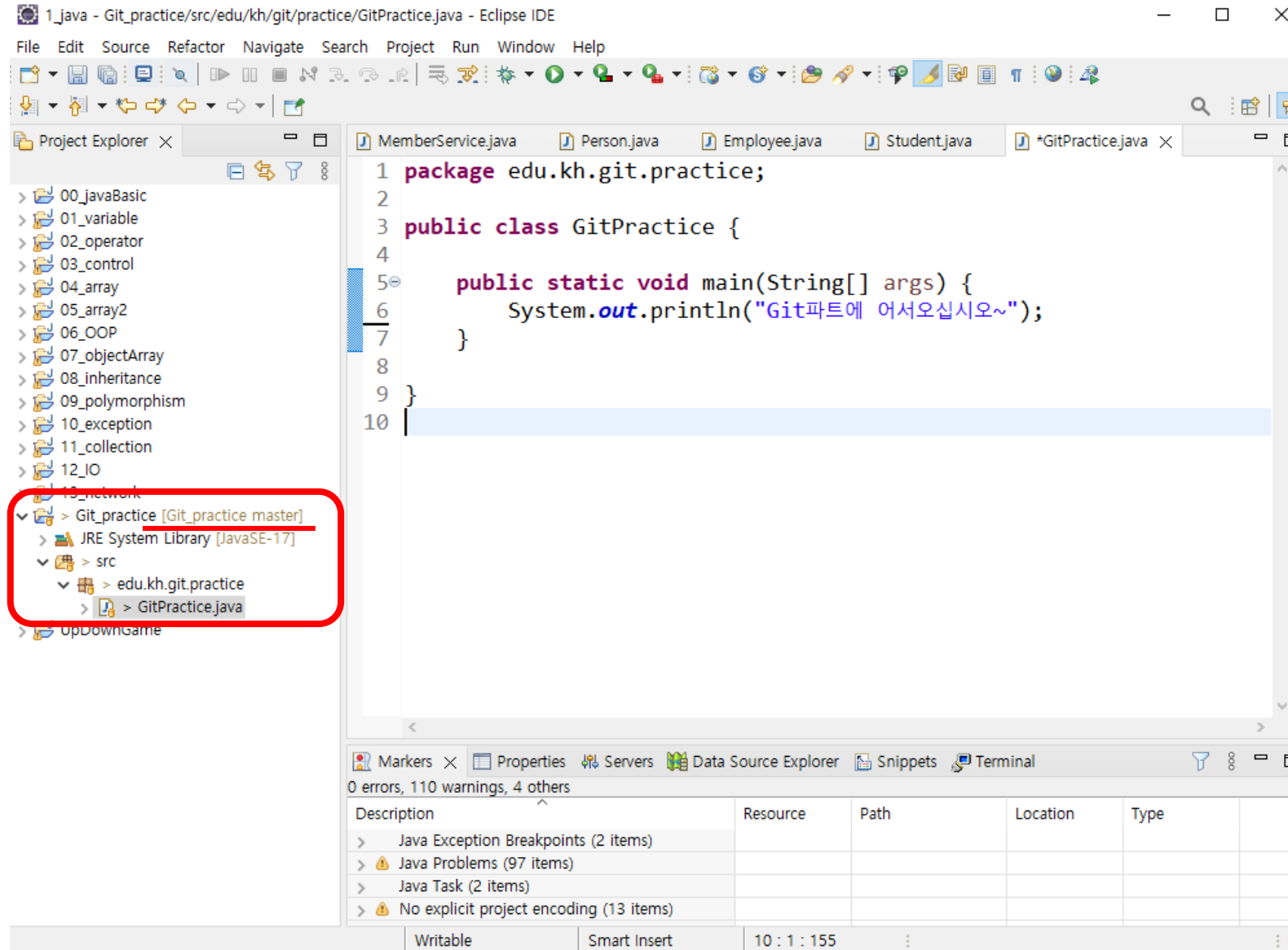
어떤 파일을
어떤 커밋메세지로
누가,
언제,
무슨내용을
커밋했는지 히스토리를 확인할 수 있음

.classpath

1 +<?xml version="1.0" encoding="UTF-8"?>
2 +<classpath>
3 +.....<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_
4 +.....<attributes>
5 +.....<attribute name="module" value="true"/>
6 +.....</attributes>
7 +.....</classpathentry>
8 +.....<classpathentry kind="src" path="src"/>
9 +.....<classpathentry kind="output" path="bin"/>
10 +</classpath>

추가된 내용은 초록색,
제거된 내용은 빨간색 섹션으로 나타남

이클립스에 올려둔 git 연동된 파일 수정해보기



1.java - Git_practice/src/edu/kh/git/practice/GitPractice.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project ... x MemberService.java Person.java Employee.java Student.java GitPractice.java x

```
1 package edu.kh.git.practice;
2
3 public class GitPractice {
4
5     public static void main(String[] args) {
6         System.out.println("Git파트에 어서오십시오~");
7         System.out.println("Hello World!");
8     }
9
10 }
11
```

수정해보기

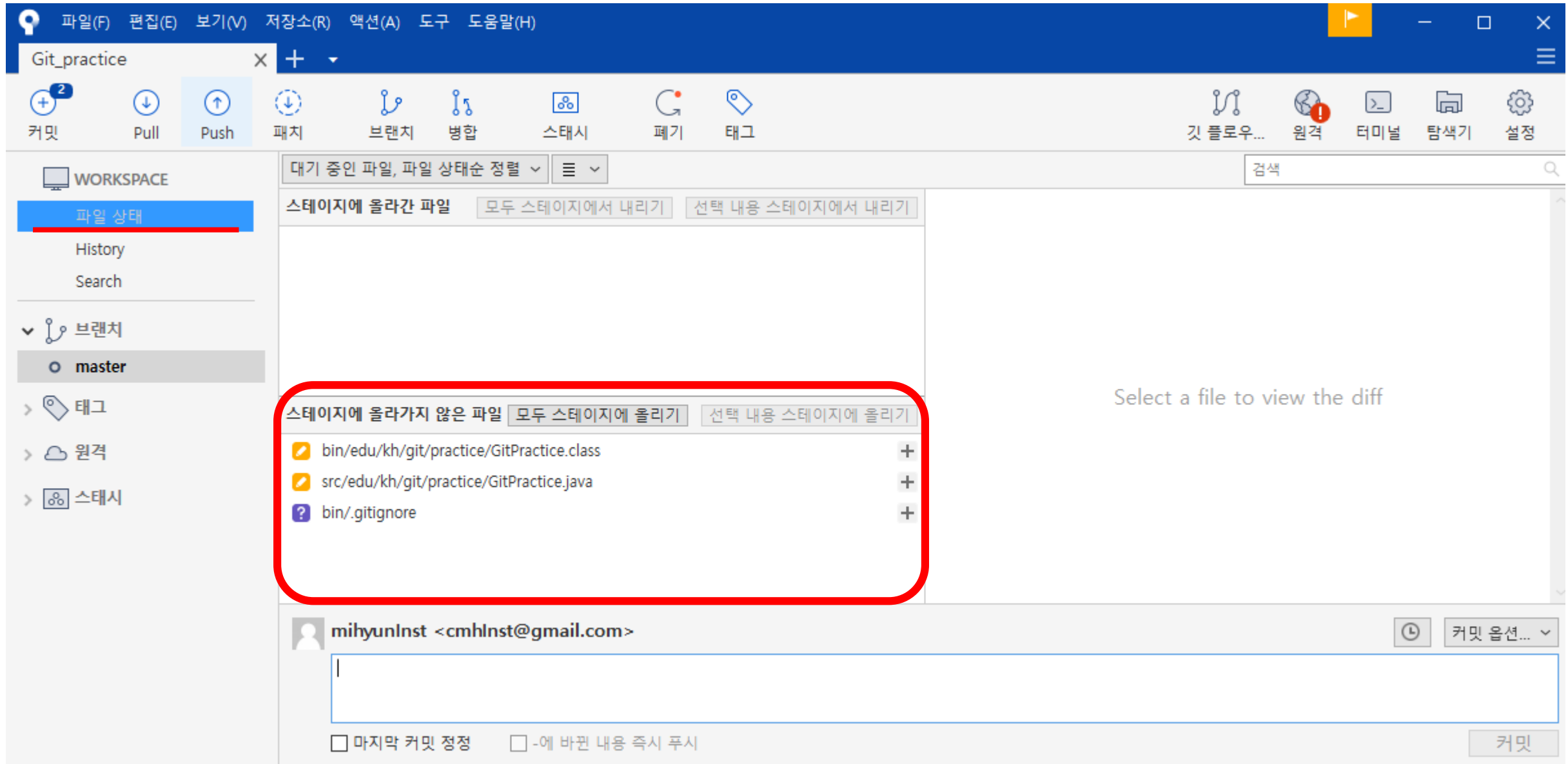
Markers x Properties Servers Data Source Explorer Snippets Terminal

0 errors, 110 warnings, 4 others

Description	Resource	Path	Location	Type
> Java Exception Breakpoints (2 items)				
> ⚠ Java Problems (97 items)				
> Java Task (2 items)				
> ⚠ No explicit project encoding (13 items)				

Writable Smart Insert 10 : 2 : 192

왼쪽 Workspace에서 [파일 상태]에 가보면 [스테이지에 올라가지 않은 파일]에 **방금 작업한 내용이 추가됨**



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

Git_practice

커밋

Pull

Push

패치

브랜치

병합

스태시

폐기

태그

깃 플로우...

원격

터미널

탐색기

설정

WORKSPACE

파일 상태

History

Search

브랜치

master

태그

원격

스태시

대기 중인 파일, 파일 상태순 정렬

스태시에 올라간 파일

모두 스태이지에서 내리기

선택 내용 스태이지에서 내리기

bin/edu/kh/git/practice/GitPractice.class

src/edu/kh/git/practice/GitPractice.java

스태시에 올라가지 않은 파일

모두 스태이지에 올리기

선택 내용 스태이지에 올리기

src/edu/kh/git/practice/GitPractice.java

1 부분 : 4-10 줄

코드웅치 스태이지에서 내리기

```
...
.....public static void main(String[] args) {
.....    System.out.println("Git [?] [?] [?] [?] 'y~'");
- -
+.....    System.out.println("Hello World!");
.....}
...
}
```

mihyunInst <cmhInst@gmail.com>

커밋 옵션...

Add: 프린트 구문 한 줄 추가

☐ 마지막 커밋 정정

☐ -에 바뀐 내용 즉시 푸시

커밋

Local repository에 업데이트 된 상태

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

Git_practice

+

-

커밋

Pull

Push

패치

브랜치

병합

스태시

폐기

태그

깃 플로우...

원격

터미널

탐색기

설정

WORKSPACE

파일 상태

History

Search

브랜치

master

태그

원격

스태시

모든 브랜치

원격 브랜치

날짜순으로 정렬

그래프

설명

날짜

작성자

커밋

○	○ master	Add: 프린트 구문 한 줄	22 11 2022 18:24	mihyunInst <cmhInst@gmail.com>	6fa69e1
●	First commit		22 11 2022 17:58	mihyunInst <cmhInst@gmail.com>	dc642a5

파일 상태순 정렬

☰

src/edu/kh/git/practice/GitPractice.java

1 부분 : 4-10 줄

코드문치 되돌리기

커밋: 6fa69e1c49e25f209093f7b569342b4f629e9b95 [6fa69e1]

상위 항목: dc642a5c14

작성자: mihyunInst <cmhInst@gmail.com>

날짜: 2022년 11월 22일 화요일 오후 6:24:03

커밋한 사람: mihyunInst

Add: 프린트 구문 한 줄 추가

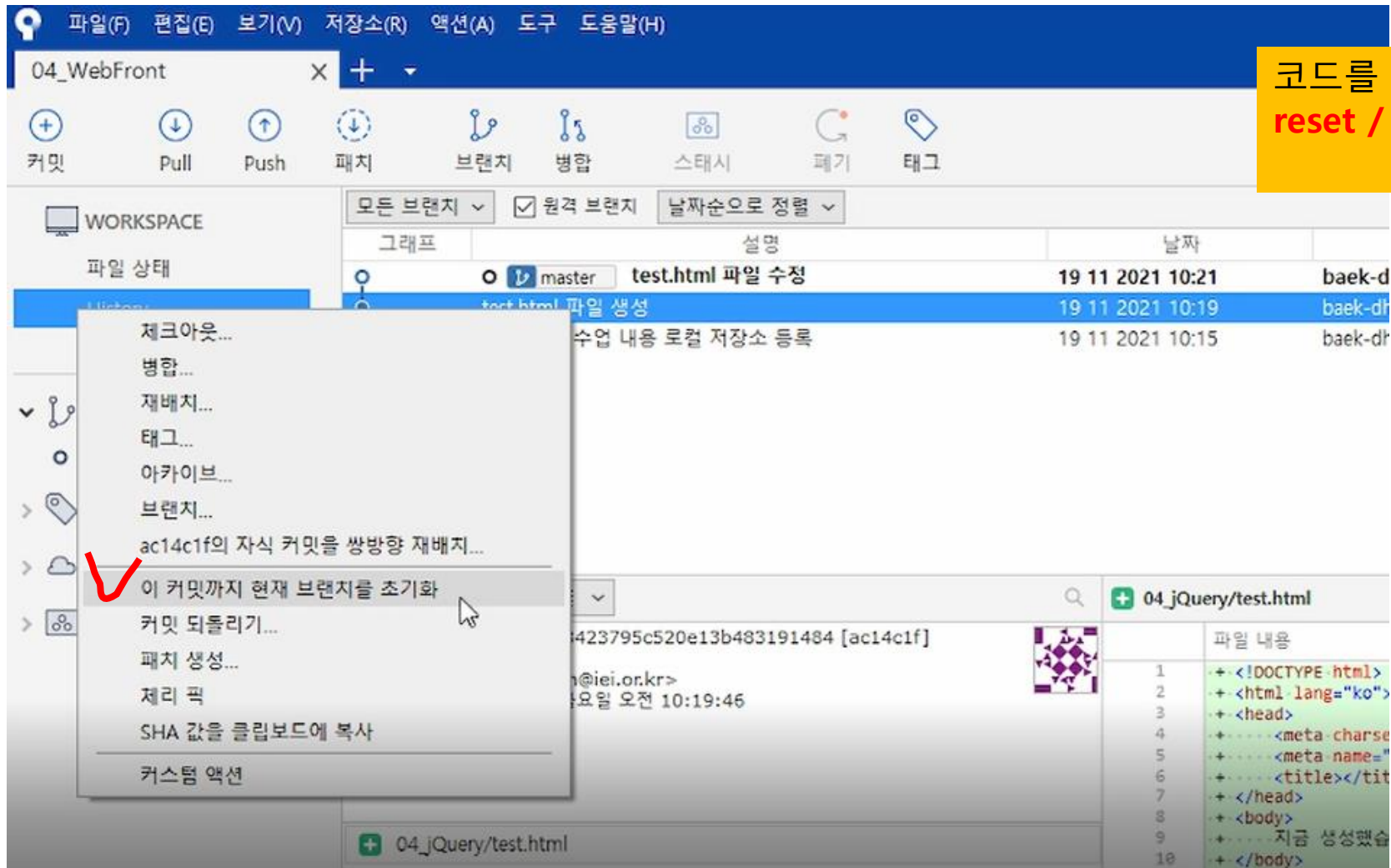
bin/edu/kh/git/practice/GitPractice.class

src/edu/kh/git/practice/GitPractice.java

```

4 4 ...
5 5 ..... public static void main(String[] args) {
6 6 ..... System.out.println("Git🔥🔥🔥🔥🔥🔥🔥🔥");
7 7 .....
7 7 ..... System.out.println("Hello World!");
8 8 ..... }
9 9 .....
10 10 ...}

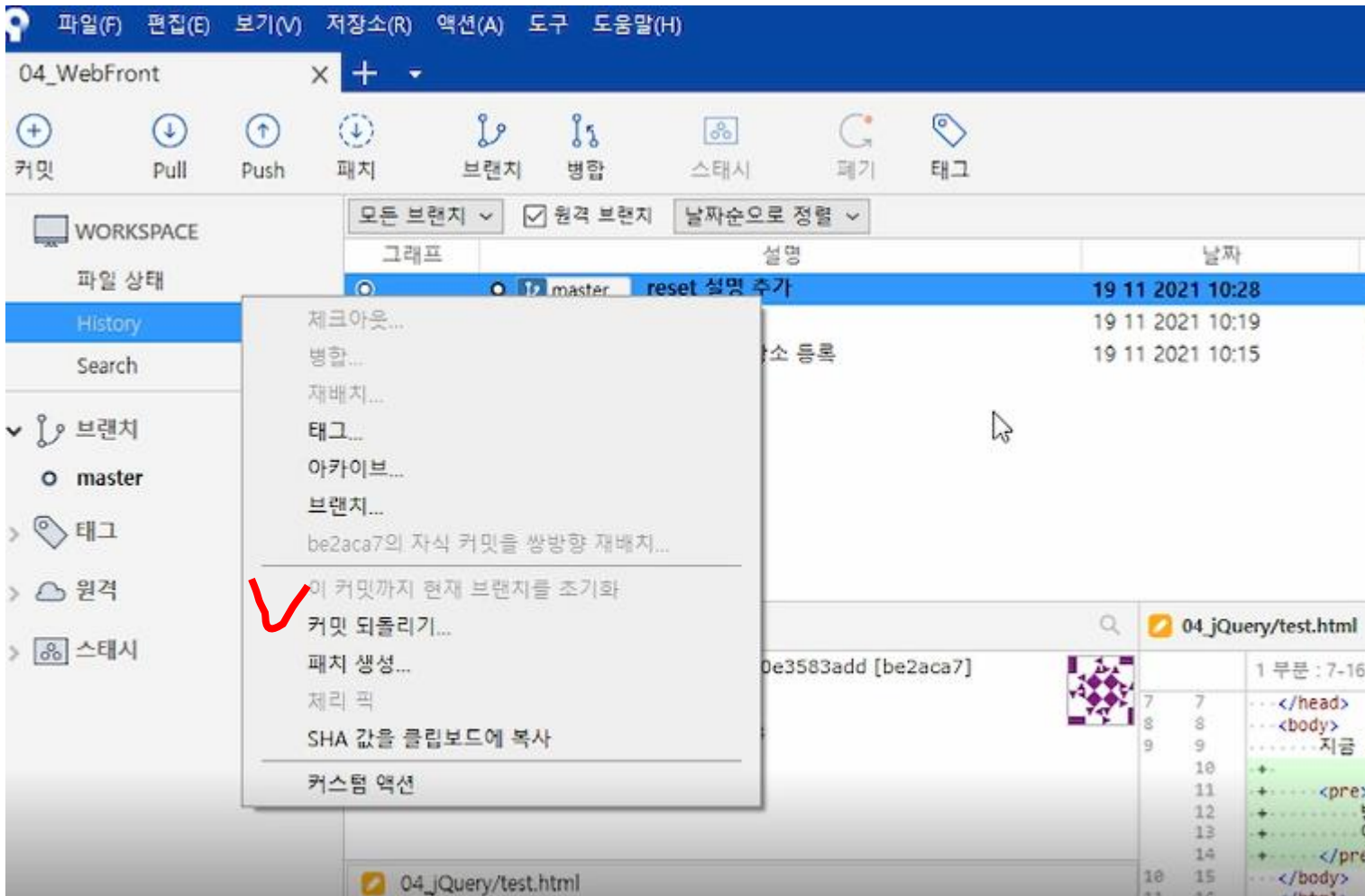
```



코드를 어떤 시점으로 되돌리고 싶을 때
reset / revert

(**reset**: 중대한 실수를 해서 코드를 초기로 돌려야 할 때 사용.)

1. history에서 되돌리고싶은 시점 위에서 오른쪽마우스 클릭 > '이 커밋까지 현재 브랜치를 초기화' 클릭
2. 완전 초기화(이후 시점은 삭제하겠다)는 Hard선택 > 경고(파괴적인 작업)창 무시해도 괜찮아요. '예' 클릭
3. 위에있던 히스토리가 사라지고 연결된 에디터에서도 내용이 사라진다.



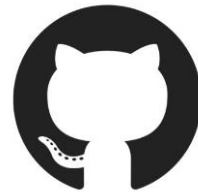
Revert의 revert를 하면 삭제되었던 내용이 다시 돌아온다

(**revert**: 커밋 되돌리기 => reset 과 다르게 기록이 남는다)

1. history에서 되돌리고 싶은 시점 위에서 오른쪽마우스 클릭 > '커밋 되돌리기' 클릭
2. '예' 클릭
3. 되돌리기 한 커밋 위에 **Revert** ""... 라는 문구 추가 확인=> 에디터 삭제 확인

Remote 에서 사용

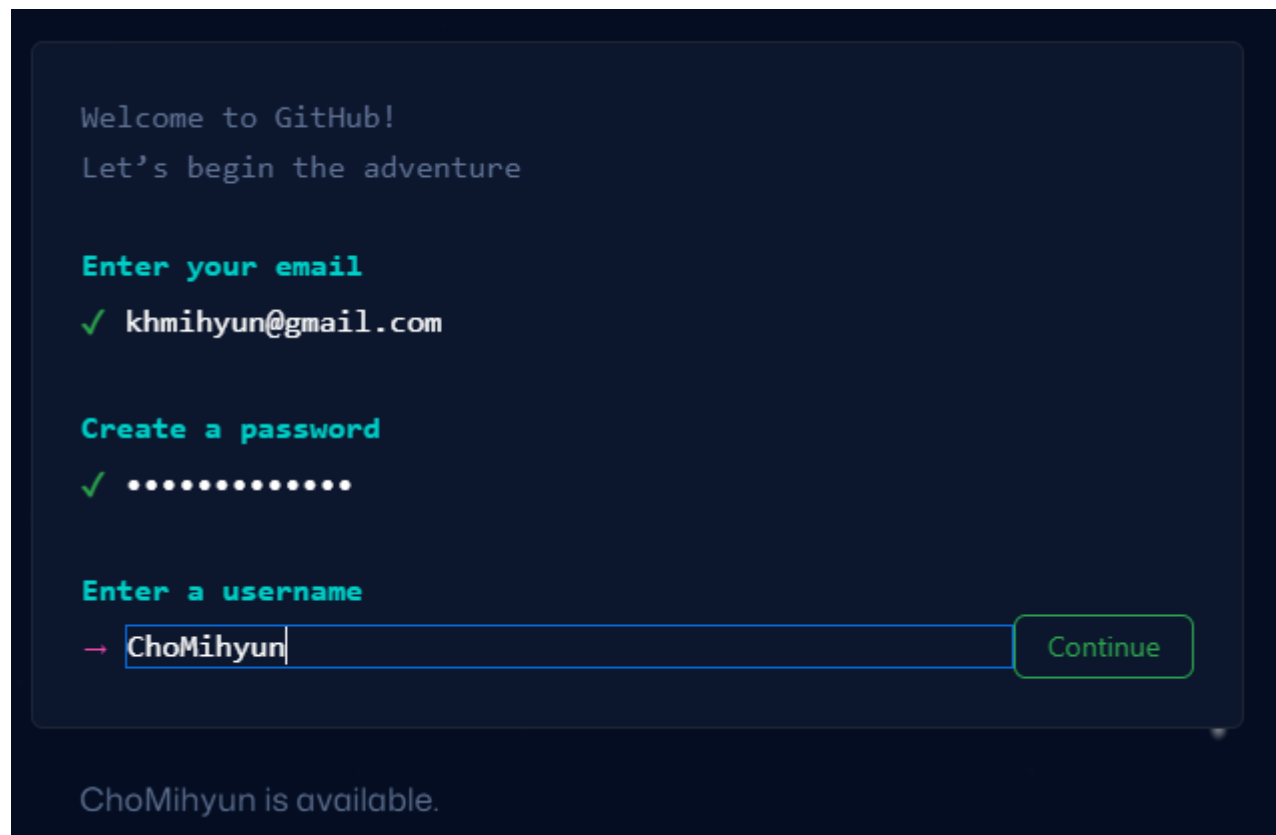
With GitHub



GitHub

<https://github.com/> 접속

1. 오른쪽상단 Sign Up 버튼 클릭 후
회원가입 진행 (이메일/패스워드/유저이름)
2. 인증 진행 후 로그인
(1. 로봇테스트 2.이메일로 인증번호 전송)
3. Free 버전으로 사용(Pro는 유료)



Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ khmihyun@gmail.com

Create a password
✓

Enter a username
→ ChoMihyun Continue

ChoMihyun is available.


- New repository
- Import repository
- New codespace
- New gist
- New organization
- New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *

 mihyunInst

Repository name * 저장소 이름

영문 소문자권장

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

Description (optional)



Public

용량제한없는 무제한 업로드 가능: 누구나 접근가능하고 코드를 발췌 가능

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



Private

용량제한있음, 아무나 접근 X. 허용한사람만 가능

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 마크다운 파일

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

Add .gitignore Github에 올라오지 못하게할 파일을 정할 수 있음(개인설정파일 등)

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

.gitignore template: None

Choose a license



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 마크다운 파일(README 필요시 체크)

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

Add .gitignore Github에 올라오지 못하게할 파일을 정할 수 있음(개인설정파일 등)

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

.gitignore template: None ▼

Choose a license

라이선스에 관한 곳

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

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

Set up in Desktop or HTTPS SSH

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

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

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

...or import code from another repository

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

Import code

Remote 계정추가



원격 저장소

+ 계정 추가...

계정 변경

호스팅 계정 편집

Host

호스팅 서비스: ☒ GitHub

호스트 URL:

선택 프로토콜: ☒ HTTPS

Credentials

인증: ☒ OAuth

사용자명:

☒ OAuth 토큰 새로고침

[Need help logging into your account?](#)

Github에서 인증방식 basic(id,pw)을 금지(보안상 취약)
OAuth로만 가능

OAuth 토큰 새로고침 클릭 하면 로그인된 크롬창에서 자동으로 연결



Authorize SourcetreeForWindows



SourcetreeForWindows by Atlassian
wants to access your mihyunInst account



GPG Keys
Full control of your GPG keys.



Personal user data
Full access



Cancel

Authorize atlassian

Authorizing will redirect to
<http://localhost:34106>

인증: OAuth

사용자명: mihyunInst

OAuth 토큰 새로고침

[Need help logging into your account?](#)

☒ 인증 성공

[확인](#) [취소](#)

아틀라시안에서 소스트리와 깃허브를 인증 후
자동으로 연동

** Readme file 만들었을때는 아래화면과 다르게 뜨는데,
오른쪽 위에 초록색 버튼 Code있는데 클릭하면 https 주소 복사 할 수 있음

github.com/mihyunInst/gitpractice 인터넷에서 접속할 때 사용하는 주소


언터테이너 Tool 스테디 알고리즘 외부소스사이트 GitHub HRD-Net 한기대보수교육 React Javascript

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

mihyunInst / gitpractice Public Pin Unwatch 1 Fork 0

Issues Pull requests Actions Projects Wiki Security Insights Settings

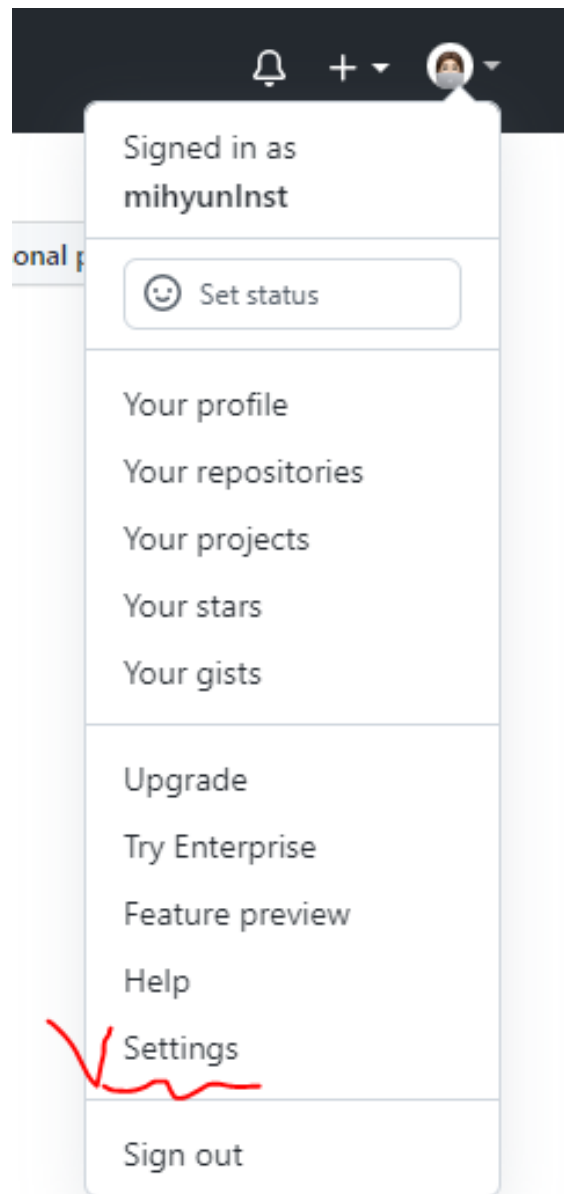
Quick setup — if you've done this kind of thing before Github 연동 시 사용하는 주소

Set up in Desktop or HTTPS SSH https://github.com/mihyunInst/gitpractice.git 

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

클릭하면 복사됨

** Github 접근 토큰 발급



Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Beta

Tokens (classic)

Personal access tokens (classic)

Generate new token ▼

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

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](#).

드롭다운 클릭 후 classic 선택

New personal access token (classic)

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](#).

Note

소스트리 연동용

What's this token for?

Expiration *

No expiration

The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure.

[Learn more](#)

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 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 |
| <input type="checkbox"/> write:packages | Upload packages to GitHub Package Registry |
| <input type="checkbox"/> read:packages | Download packages from GitHub Package Registry |
| <input type="checkbox"/> delete:packages | Delete packages from GitHub Package Registry |
| <input type="checkbox"/> admin:org | Full control of orgs and teams, read and write org projects |
| <input type="checkbox"/> write:org | Read and write org and team membership, read and write org projects |
| <input type="checkbox"/> read:org | Read org and team membership, read org projects |
| <input type="checkbox"/> manage_runners:org | Manage org runners and runner groups |
| <input type="checkbox"/> admin:public_key | Full control of user public keys |
| <input type="checkbox"/> write:public_key | Write user public keys |
| <input type="checkbox"/> read:public_key | Read user public keys |

- | | |
|---|---|
| <input checked="" type="checkbox"/> admin:repo_hook | Full control of repository hooks |
| <input checked="" type="checkbox"/> write:repo_hook | Write repository hooks |
| <input checked="" type="checkbox"/> read:repo_hook | Read repository hooks |
| <input type="checkbox"/> admin:org_hook | Full control of organization hooks |
| <input type="checkbox"/> gist | Create gists |
| <input type="checkbox"/> notifications | Access notifications |
| <input type="checkbox"/> user | Update ALL user data |
| <input type="checkbox"/> read:user | Read ALL user profile data |
| <input type="checkbox"/> user:email | Access user email addresses (read-only) |
| <input type="checkbox"/> user:follow | Follow and unfollow users |
| <input checked="" type="checkbox"/> delete_repo | Delete repositories |
| <input type="checkbox"/> write:discussion | Read and write team discussions |
| <input type="checkbox"/> read:discussion | Read team discussions |
| <input type="checkbox"/> admin:enterprise | Full control of enterprises |
| <input type="checkbox"/> manage_runners:enterprise | Manage enterprise runners and runner groups |
| <input type="checkbox"/> manage_billing:enterprise | Read and write enterprise billing data |
| <input type="checkbox"/> read:enterprise | Read enterprise profile data |
| <input type="checkbox"/> audit_log | Full control of audit log |
| <input type="checkbox"/> read:audit_log | Read access of audit log |
| <input type="checkbox"/> codespace | Full control of codespaces |
| <input type="checkbox"/> codespace:secrets | Ability to create, read, update, and delete codespace secrets |
| <input type="checkbox"/> project | Full control of projects |
| <input type="checkbox"/> read:project | Read access of projects |
| <input type="checkbox"/> admin:pgp_key | Full control of public user GPG keys |
| <input type="checkbox"/> write:pgp_key | Write public user GPG keys |
| <input type="checkbox"/> read:pgp_key | Read public user GPG keys |
| <input type="checkbox"/> admin:ssh_signing_key | Full control of public user SSH signing keys |
| <input type="checkbox"/> write:ssh_signing_key | Write public user SSH signing keys |
| <input type="checkbox"/> read:ssh_signing_key | Read public user SSH signing keys |

Generate token

Cancel

저장소의 접근 권한

*****발급된 토큰 복사 해서 바로 어딘가 저장해두기!!
5분 뒤 사라지고 두번 다시 찾을 수 없음**

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 [GitHub API](#).

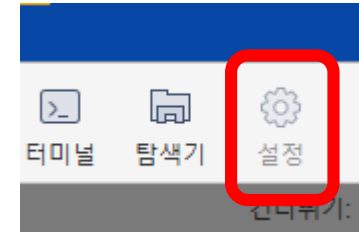
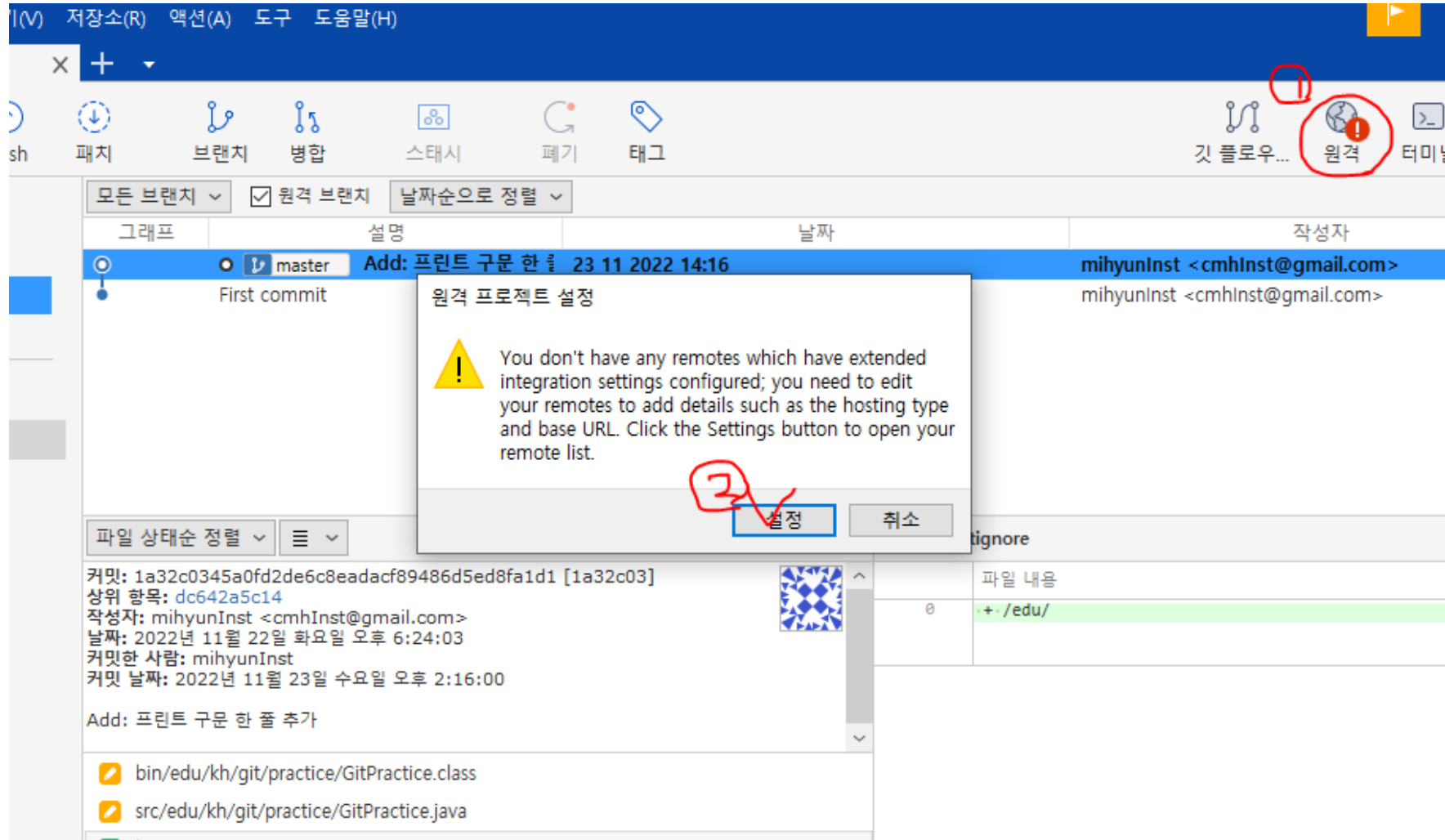
Make sure to copy your personal access token now. You won't be able to see it again!

✓ [REDACTED] Delete

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](#).

© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

**Remote 연동 작업



설정 버튼 눌러도 같은 창 뜸

이후 뜨는 창에서 "추가" 버튼 클릭

**Github에서 복사한 레파지토리 주소
붙혀넣기 후 아래 형태로 수정하기**

https://유저네임:토큰값@github.com/레파지토리주소.git

원격 저장소 정보


필요한 정보

원격 이름: ☒ 디폴트 원격

URL / 경로:

추가 확장 통합

Remote Account:

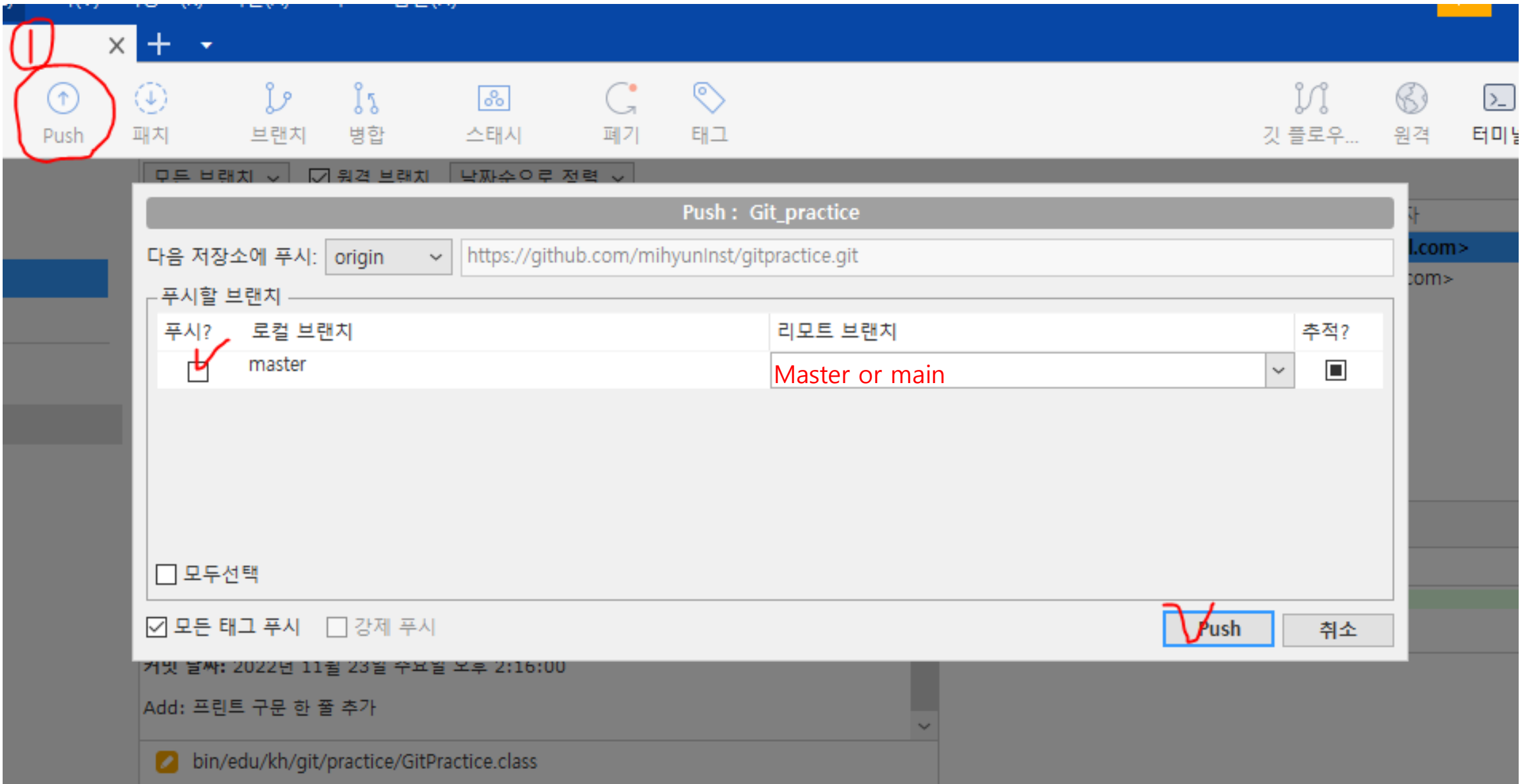
☒  mihyunInst GitHub **드롭다운 클릭하면 연동된 github 계정클릭**

확장 통합을 사용하면 Bitbucket 과 같은 외부 호스팅과 연계할 수 있습니다. 예를들어, 사이트에서 링크 기존의 클론을 찾아 표시하거나, 그대로 pull 요청을 만들 수 있습니다.

저장소 설정

원격 저장소 경로

이름	경로
origin	https://mihyunInst:ghp_100...@github.com/mihyunInst



Github에 해당 레파지토리 화면이 변경되어있음
⇒ 로컬에서 한 커밋 내역들이 확인된다.

⇒ Remote로 연동이 완료됨.

The screenshot shows the GitHub interface for the repository `mihyunInst / gitpractice`, which is public. The repository has 1 branch (master) and 0 tags. The commit history shows a recent commit by `mihyunInst` with the message "Add: 프린트 구문 한 줄 추가" (Add: Print statement one line add), committed 30 minutes ago. The commit includes files `.settings`, `bin`, `.classpath`, and `.project`. The repository also has a "Add a README" button. On the right sidebar, the "About" section shows 0 stars, 1 watching, and 0 forks. The "Releases" section shows no releases published, and the "Packages" section shows no packages published.

mihyunInst / gitpractice Public

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

master 1 branch 0 tags Go to file Add file <> Code

mihyunInst Add: 프린트 구문 한 줄 추가 1bdc0c4 30 minutes ago 2 commits

.settings	First commit	22 hours ago
bin	Add: 프린트 구문 한 줄 추가	30 minutes ago
.classpath	Add: 프린트 구문 한 줄 추가	30 minutes ago
.project	First commit	22 hours ago

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

About
remote 연결용
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package