

Git & GitHub

◆ VSCODE에서 깃 사용하기

정수아

Contents

01 환경 설정

02 VSCODE에서 파일 커밋

03 원격 저장소에 push

04 README 파일 작성

05 원격 저장소에서 pull

06 원격 저장소 복제

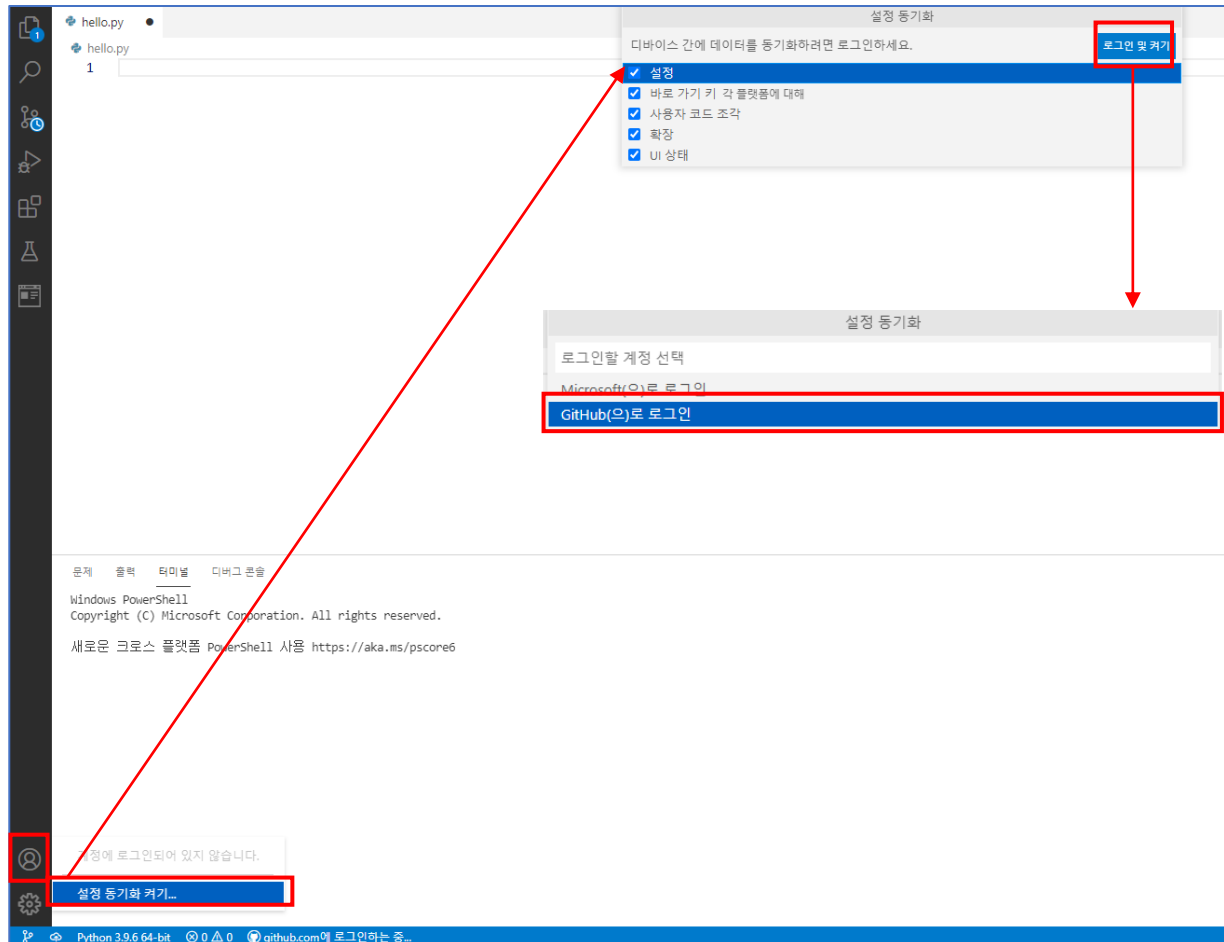


01

환경 설정

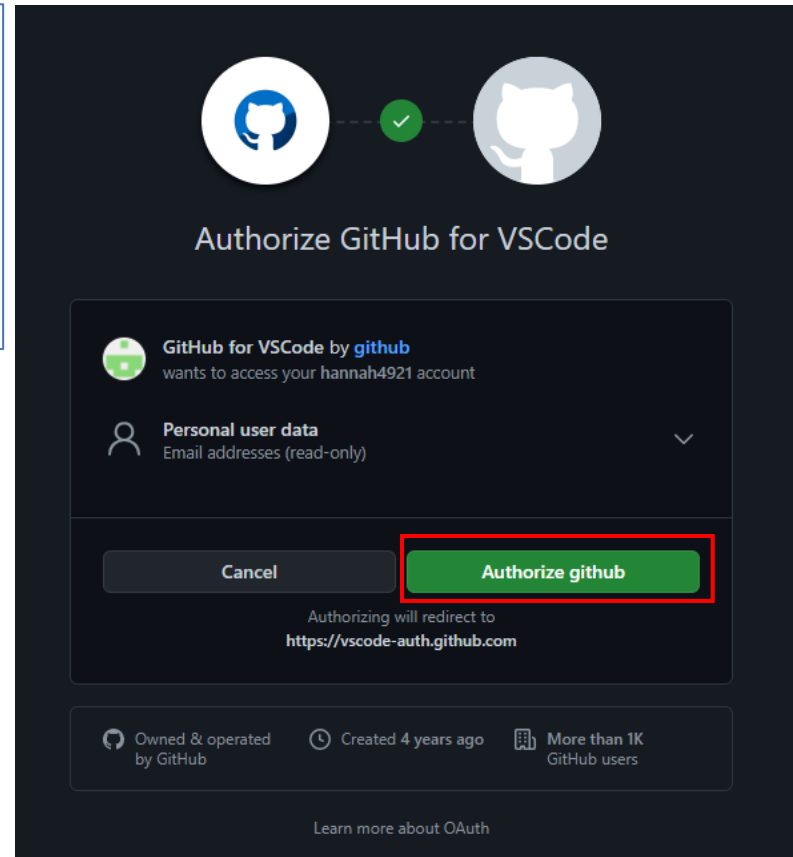
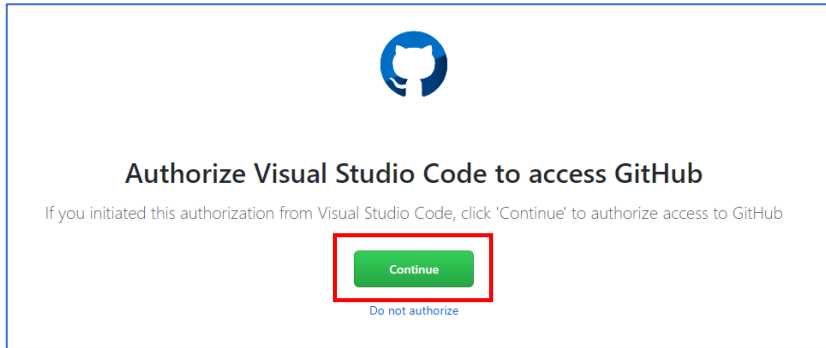
환경 설정

❖ 깃허브 계정 로그인하기



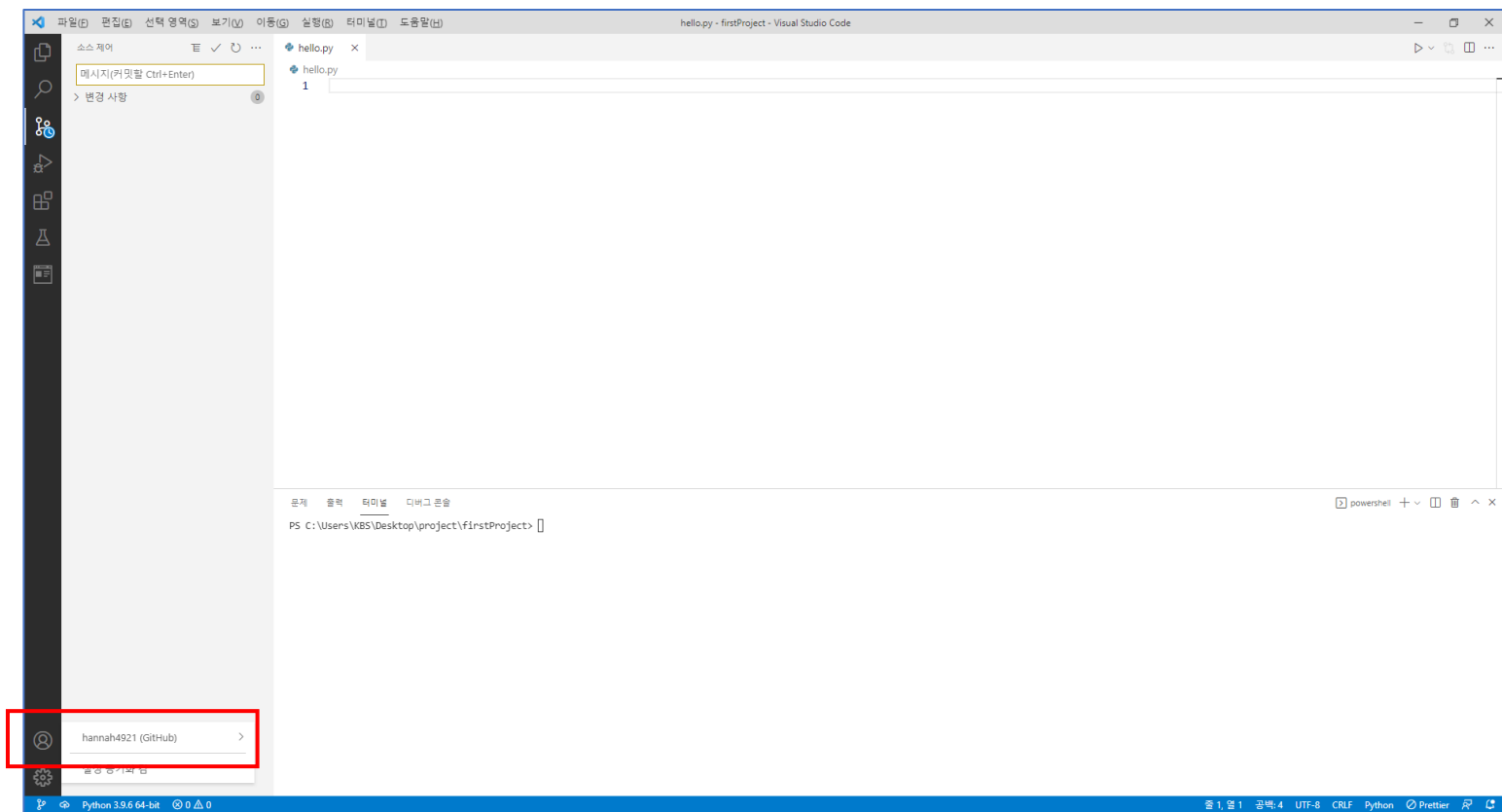
환경 설정

❖ 권한 수락



환경 설정

❖ 로그인 확인





02

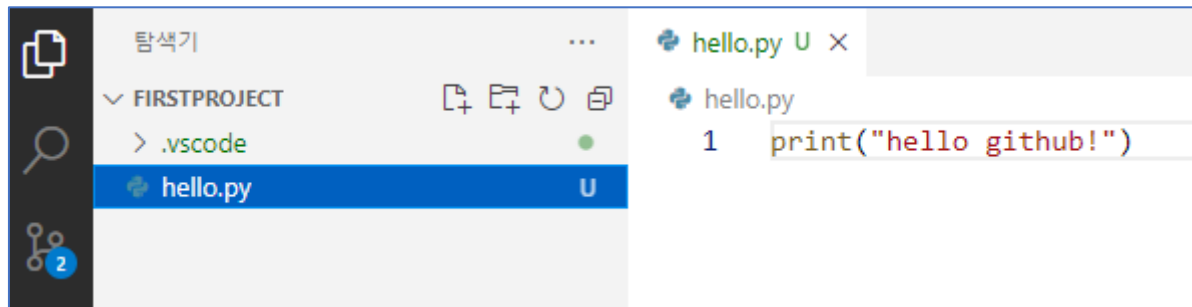
VSCODE에서 파일 커밋

VSCODE에서 파일 커밋

❖ 새 프로젝트 생성

- firstProject 폴더
- hello.py 파일

```
print("hello github!")
```



VSCODE에서 파일 커밋

❖ 깃 저장소 초기화

```
$ git init
```

❖ 터미널에 사용자 정보 입력

- 사용자 이름

```
$ git config --global user.name "Sooa"
```

- 사용자 이메일

```
$ git config --global user.email "Sooa@gmail.com"
```

VSCODE에서 파일 커밋

❖ 스테이지에 파일 올리기

```
$ git add hello.py
```

❖ 커밋하기

```
$ git commit -m "first commit"
```

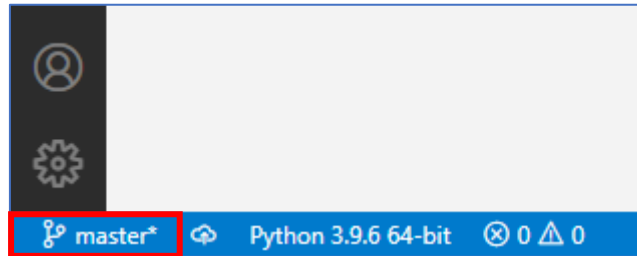


03

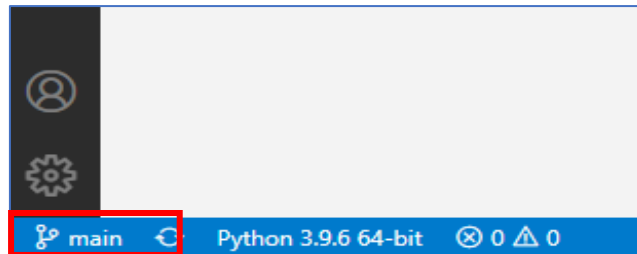
원격 저장소에 push

원격 저장소에 push하기

❖ main 브랜치 생성 및 변경(master → main)



```
$ git branch -m main
```



원격 저장소에 push하기

❖ 원격 저장소 생성하기 - firstProject

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *

HelloCoding22

Repository name *

firstProject

✔ firstProject is available.

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

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:

☐ Add a README file

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

Add .gitignore

.gitignore template: None

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

Choose a license

License: None

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

📘 You are creating a public repository in your personal account.

Create repository

원격 저장소에 push하기

❖ 지역 저장소와 원격 저장소 연결하기

```
$ git remote add origin 원격저장소_주소
```

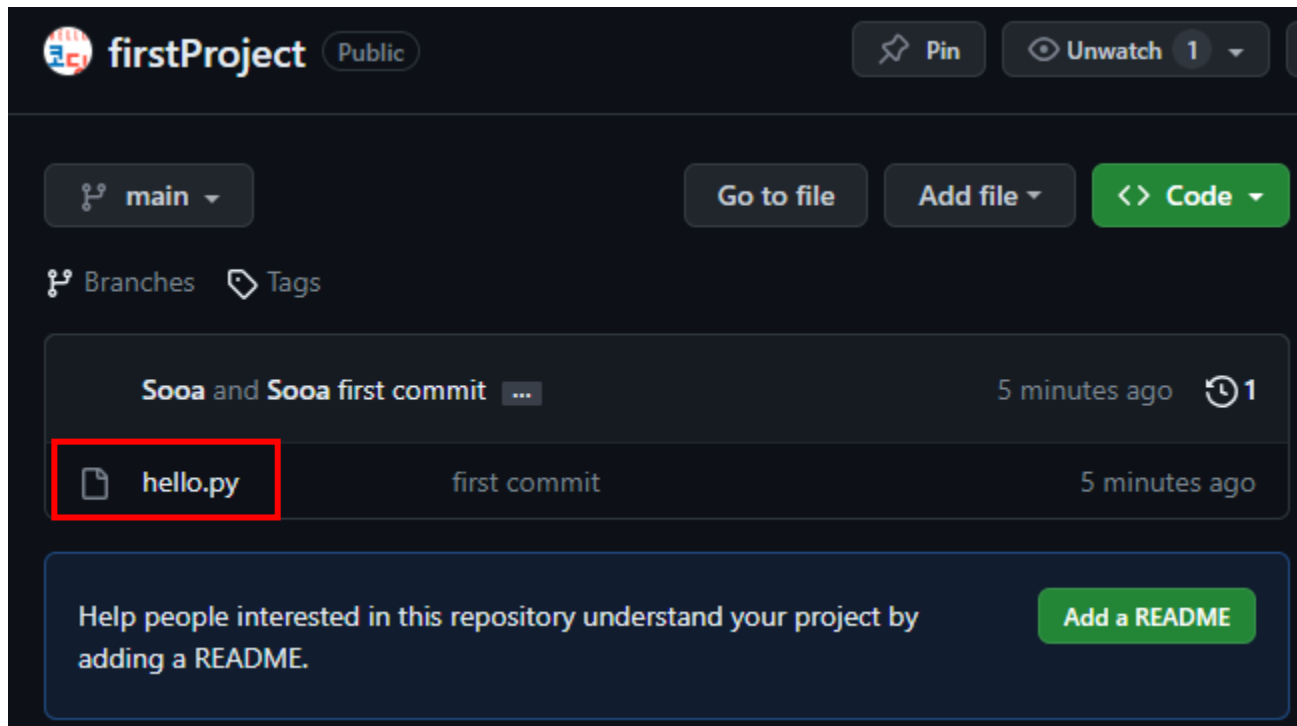
❖ 연결 확인하기

```
$ git remote -v
```

원격 저장소에 push하기

❖ 원격 저장소에 push하기

```
$ git push -u origin main
```



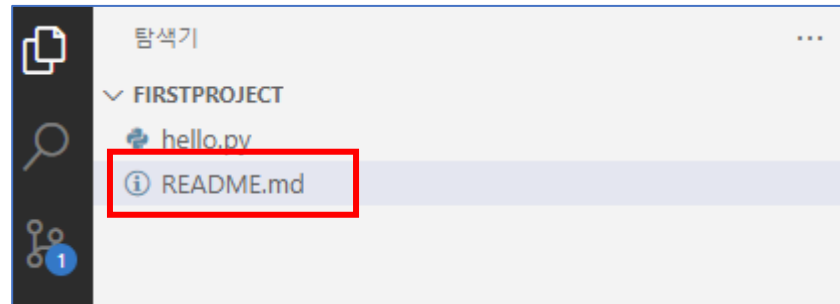


04

README 파일 작성

README 파일 작성

❖ README.md 파일 생성



README 파일 작성

❖ README 파일 작성

```
# 깃허브 연습
```

```
비주얼 스튜디오 코드를 사용해서 깃허브를 연습합니다.
```

```
**README.md 파일**을 작성 중입니다.
```

README 파일 작성

❖ 스테이지에 올리기

```
$ git add README.md
```

❖ 커밋하기

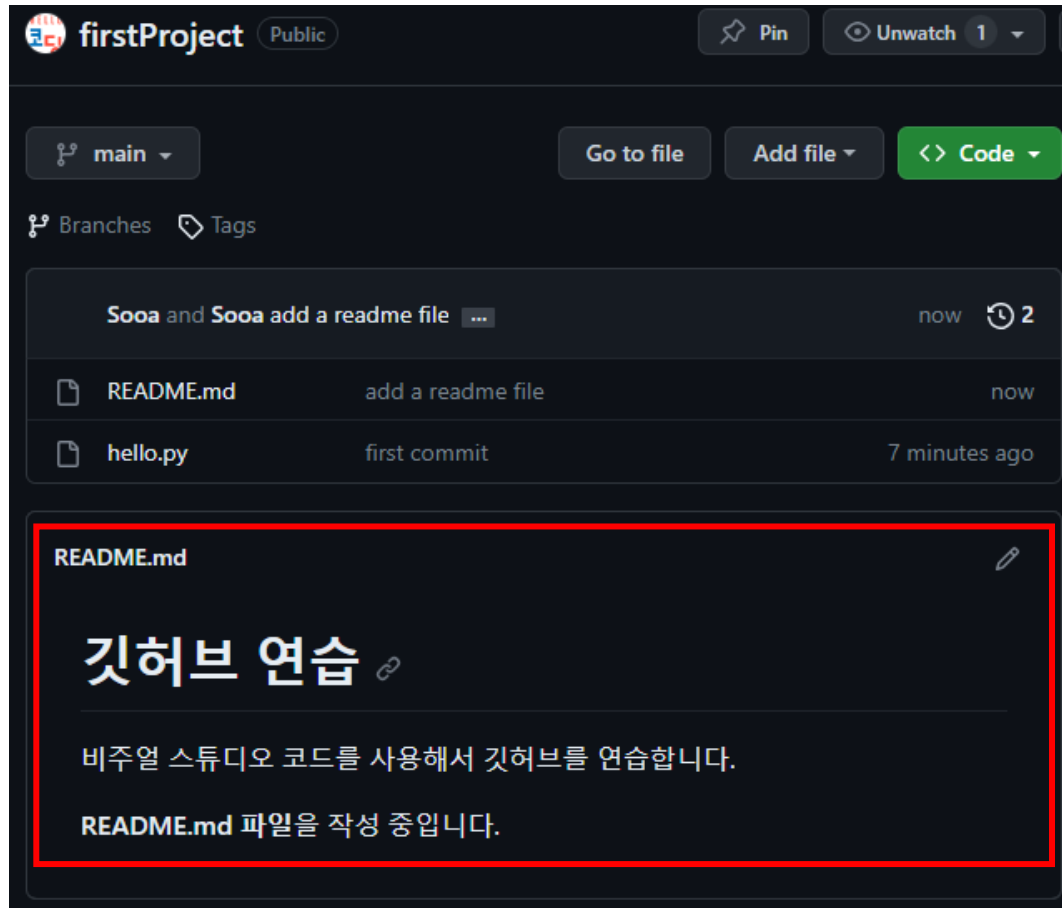
```
$ git commit -m "add a readme file"
```

❖ 원격 저장소에 push하기

```
$ git push
```

README 파일 작성

❖ README 파일 push 완료





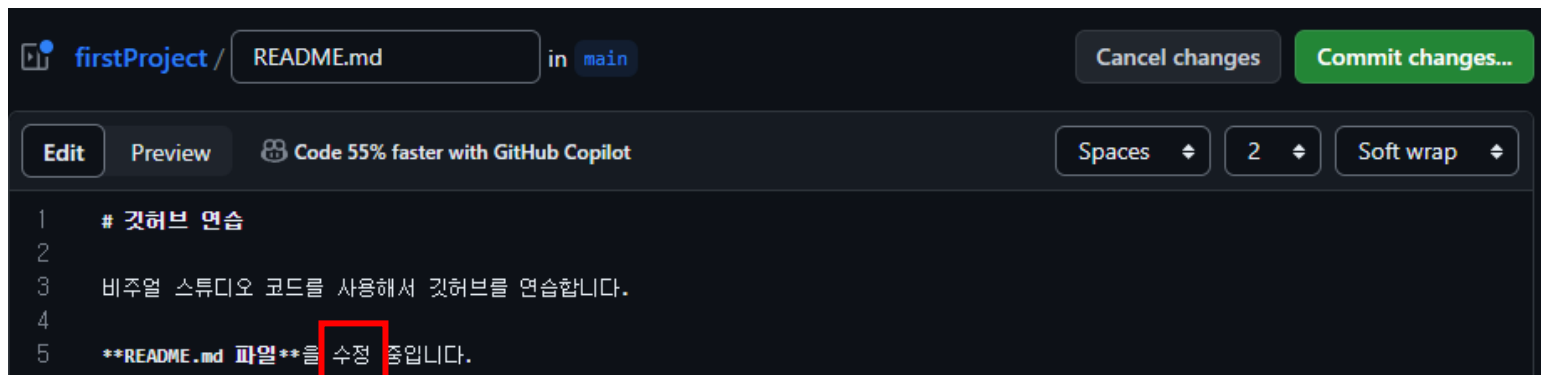
05

원격 저장소에서 pull

원격 저장소에서 pull하기

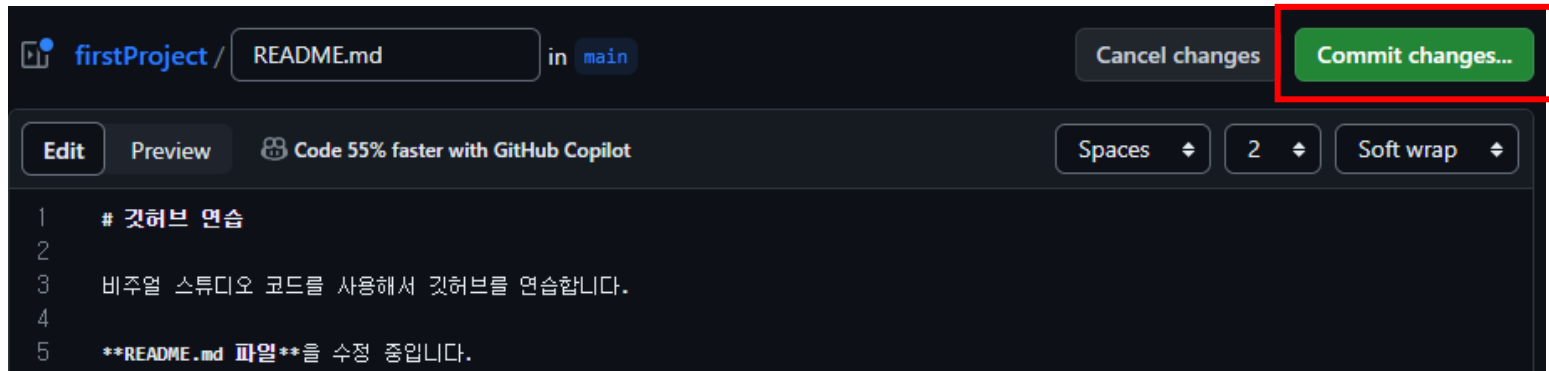
❖ 원격 저장소에서 pull하기

- 원격 저장소를 pull하기 위해서는 원격 저장소에 변경 사항이 있어야 함
 - 연결된 원격 저장소에서 직접 파일 수정 후 커밋하기
- 예) README.md 파일 수정



원격 저장소에서 pull하기

❖ 커밋하기

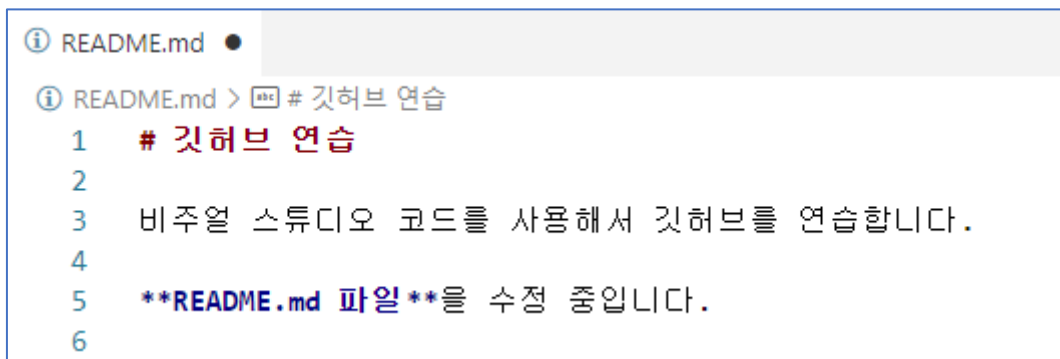


원격 저장소에서 pull하기

❖ 원격 저장소 pull하기

```
$ git pull
```

❖ 원격 저장소에서 수정한 내용이 즉시 반영



```
① README.md ●  
① README.md > # 깃허브 연습  
1  # 깃허브 연습  
2  
3  비주얼 스튜디오 코드를 사용해서 깃허브를 연습합니다.  
4  
5  **README.md 파일**을 수정 중입니다.  
6
```




06

원격 저장소 복제

원격 저장소 복제하기

❖ 원격 저장소 복제하기

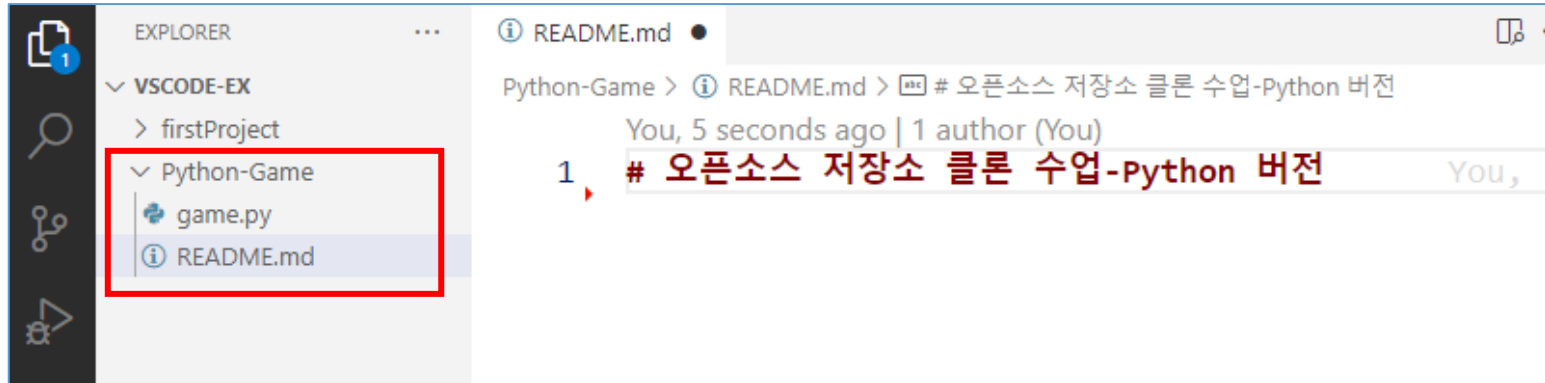
```
$ git clone 복제할_저장소_주소
```

- 예시

```
$ git clone https://github.com/HelloCoding22/Python-Game.git
```

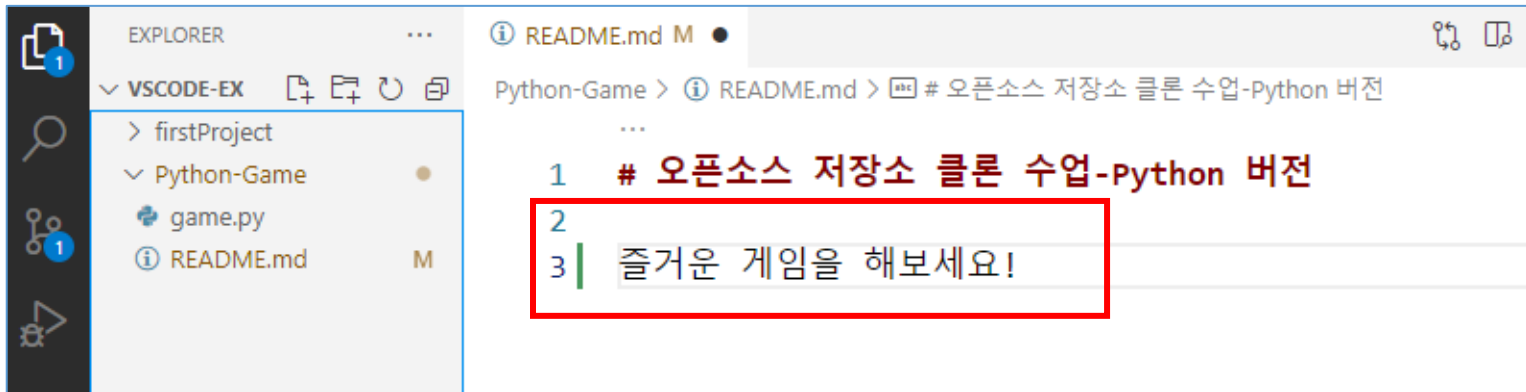
원격 저장소 복제하기

❖ 원격 저장소 복제 완료



지역 저장소에서 파일 수정하기

❖ 지역 저장소에서 복제한 파일 수정하기



원격 저장소에 push하기

❖ 스테이지에 올리기

```
$ git add README.md
```

❖ 커밋하기

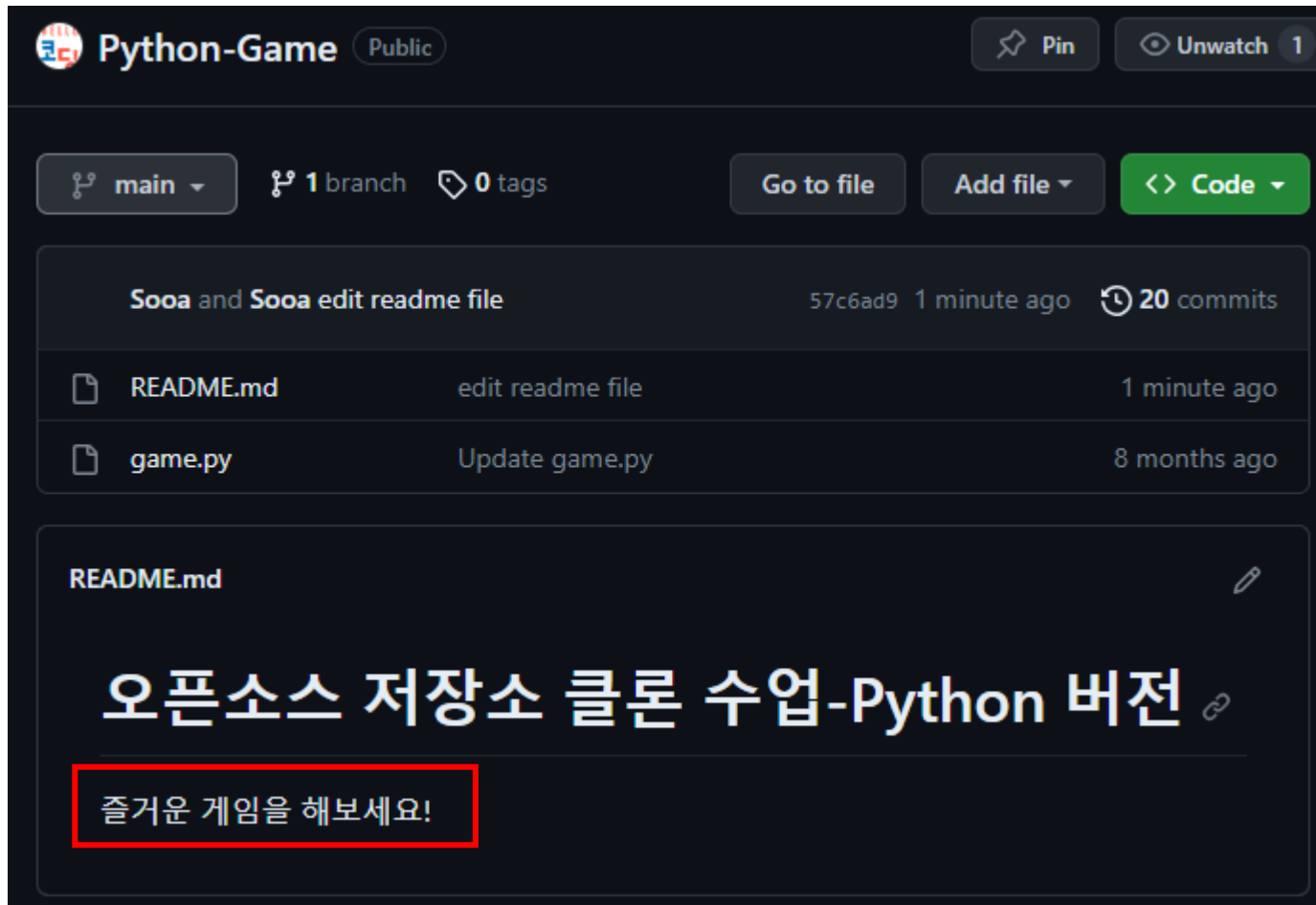
```
$ git commit -m "edit readme file"
```

❖ 원격 저장소에 push하기

```
$ git push
```

원격 저장소에 push하기

❖ 원격 저장소 확인하기



THANK 😊 YOU