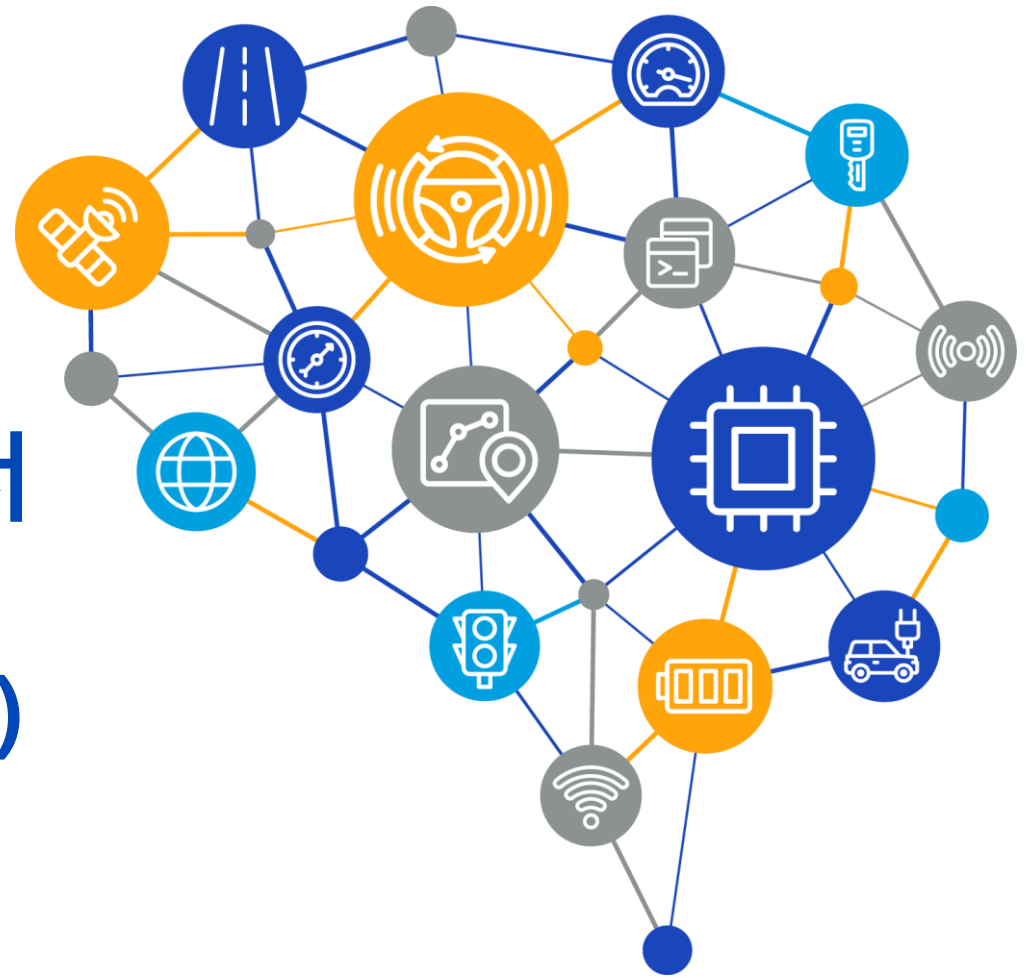


비주얼 스튜디오 코드에서 깃허브로 게시와 깃허브 저장소 복제(clone)

강환수 교수



- 학습 개요

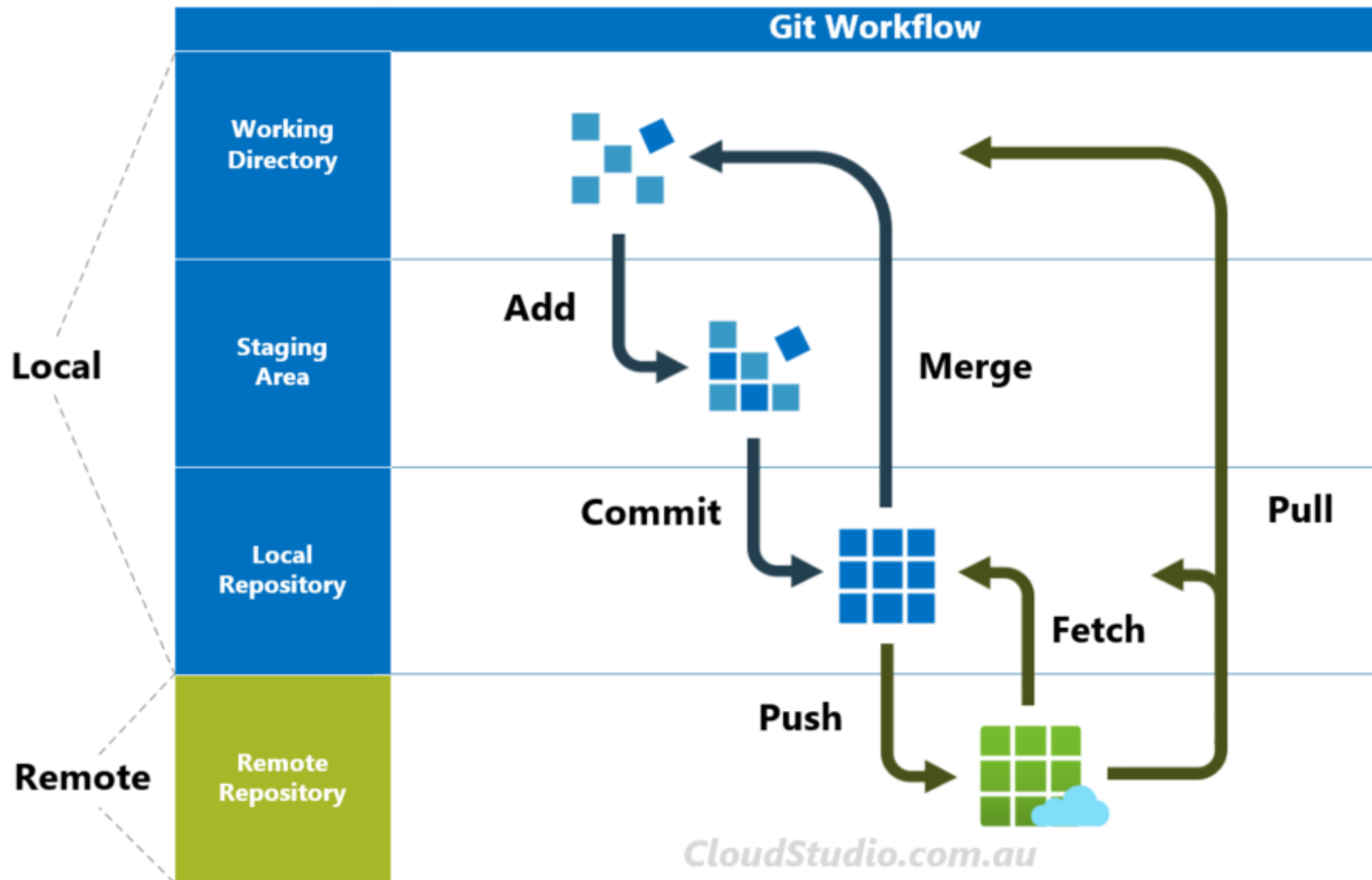
- Vs code로 지역 저장소를 깃허브에 게시
 - 게시: 깃허브에 처음으로 원격 저장소를 생성해 지역저장소 내용을 복사

- 학습 목표

- 깃과 깃허브의 연동을 vs code로 쉽게 이해
- 다음 용어의 이해
 - 게시
 - Push / pull
 - 클론

Push Pull

깃과 깃허브 Python language



AI Experts
Who Lead
The Future

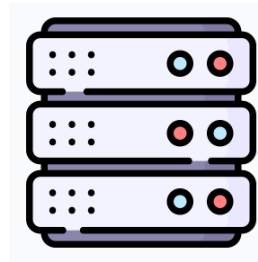
01

깃허브에 publish와 pull



- **Publish**

- 지역 저장소의 내용을 처음으로 원격 저장소를 생성해 보내는 작업



깃허브(서버) 저장소

처음으로 깃허브(서버) 저장소를 생성해 지역 저장소를 서버로 보내는 작업



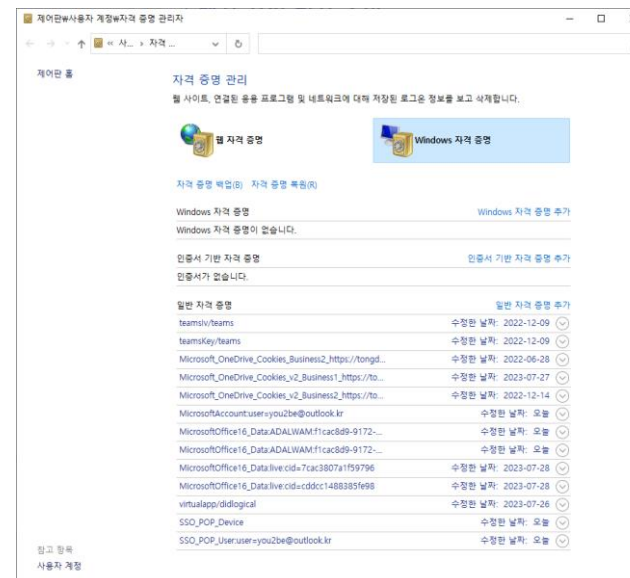
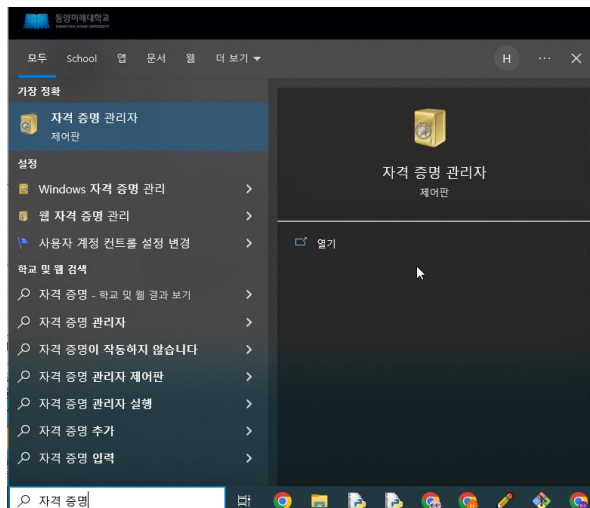
지역 저장소

0. 현재 상황 점검

- Vs code에서 깃 배시 실행 확인
- 윈도우 '자격 증명 관리' 확인
 - 기존의 '일반 자격 증명'에서 다음 2개 제거
 - git:https://github.com
 - Vscodevscode.github-authentication/github.auth

0. vs code 설정 초기화

- 확장 등 설정 폴더 삭제
 - C:\Users\[사용자계정]\.vscode
 - C:\Users\[사용자계정]\AppData\Roaming\Code 삭제



- 1. vs code에서 준비 폴더에서 파일 작업
 - repo-publish
- 2. 깃 배시를 열어, 깃 사용자와 전자메일 설정
 - \$ git config --global user.name chatkang
 - \$ git config --global user.email chatkang@gmail.com
 - 나머지 4개
- 3. 먼저 깃허브에 로그인 한 후
 - 저장소로 활용할 깃허브 계정에 로그인
- 4. 활동바 '소스 제어'에서 Publish To GitHub 클릭
 - 깃허브에서 확인
- 5. 깃허브에서 readme.md 파일 작성
 - 커밋 한 후
- 6. 다시 vs code에서 pull

```
$ git config --global user.name ai7dnn
```

```
$ git config --global user.email ai7dnn@gmail.com
```

```
$ git config --global core.autocrlf true
```

```
$ git config --global core.safecrlf false
```

```
$ git config --global core.editor 'code --wait'
```

```
$ git config --global init.defaultBranch main
```

0. 현재 상황 점검

깃과 깃허브 Python language

- 검색 창 '자격 증명 관리' 로 검색
 - 기존의 '일반 자격 증명'에서 다음 2개 제거
 - git:https://github.com
 - Vscodevscode.gihub-authentication/github.auth

자격 증명 관리

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그온 정보를 보고 삭제합니다.



웹 자격 증명



Windows 자격 증명

자격 증명 백업(B) 자격 증명 복원(R)

Windows 자격 증명

Windows 자격 증명 추가

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명

인증서 기반 자격 증명 추가

인증서가 없습니다.

일반 자격 증명

일반 자격 증명 추가

vscodevscode.github-authentication/github.auth

수정한 날짜: 2023-05-01

git:https://github.com

수정한 날짜: 2023-03-20

virtualapp/didlogical

수정한 날짜: 2023-02-20

SSO_POP_Device

수정한 날짜: 오늘

XboxLive

수정한 날짜: 2023-05-04

일반 자격 증명

일반 자격 증명 추가

vscodevscode.github-authentication/github.auth

수정한 날짜: 2023-05-01

인터넷 또는 네트워크 주소:

vscodevscode.github-authentication/github.auth

사용자 이름: github.auth

암호:

지속성: 엔터프라이즈

편집 제거

자격 증명 백업(B) 자격 증명 복원(R)

Windows 자격 증명

Windows 자격 증명 추가

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명

인증서 기반 자격 증명 추가

인증서가 없습니다.

일반 자격 증명

일반 자격 증명 추가

vscodevscode.github-authentication/github.auth

수정한 날짜: 2023-05-01

git:https://github.com

수정한 날짜: 2023-03-20

인터넷 또는 네트워크 주소: git:https://github.com

사용자 이름: satcop-daum

암호:

지속성: 로컬 컴퓨터

편집 제거

virtualapp/didlogical

수정한 날짜: 2023-02-20

SSO_POP_Device

수정한 날짜: 오늘

XboxLive

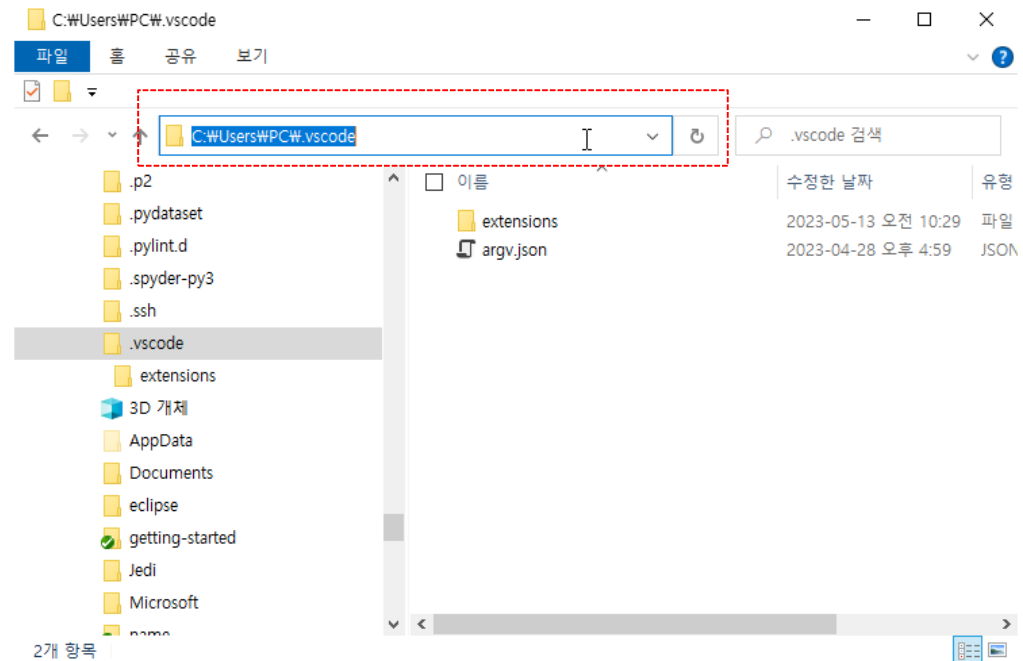
수정한 날짜: 2023-05-04

0. vs code 설정 초기화

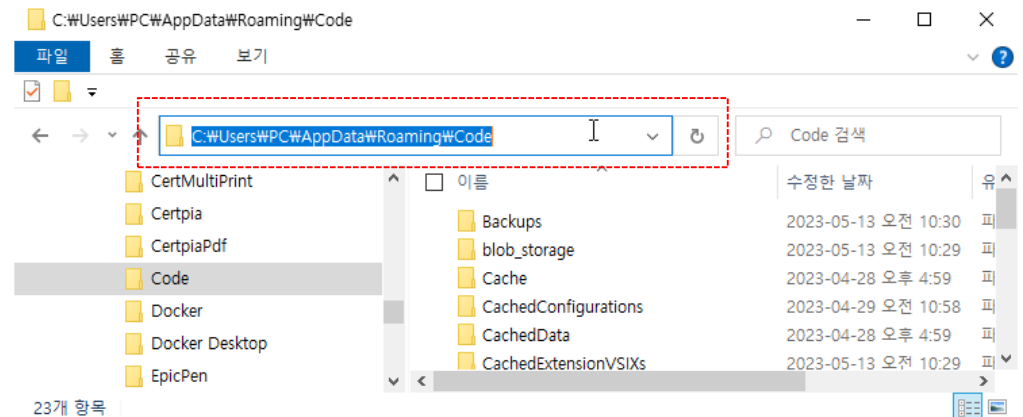
깃과 깃허브 Python language

• 확장 등 설정 폴더 삭제

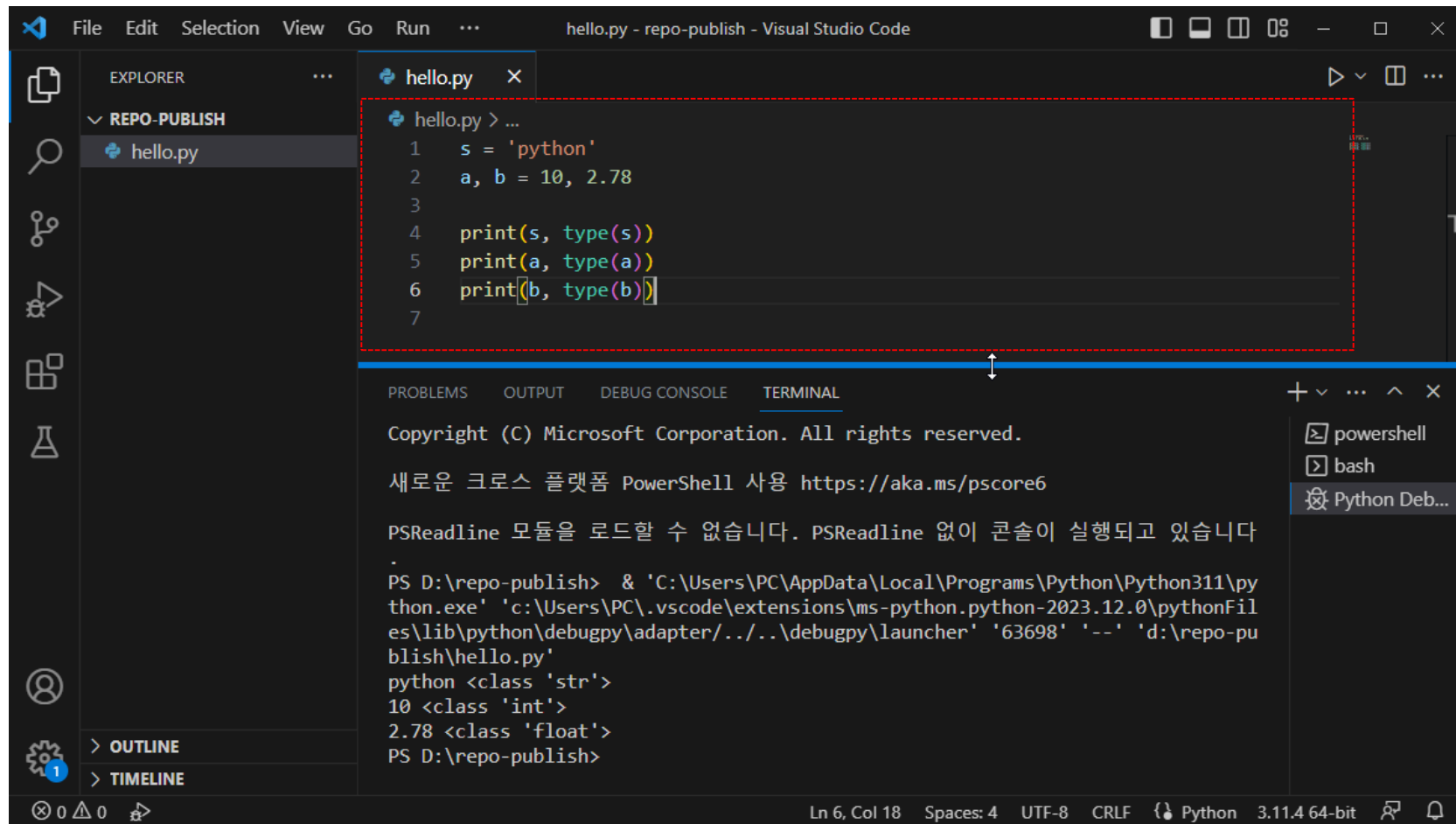
- C:\Users\W[사용자계정]\W.vscode



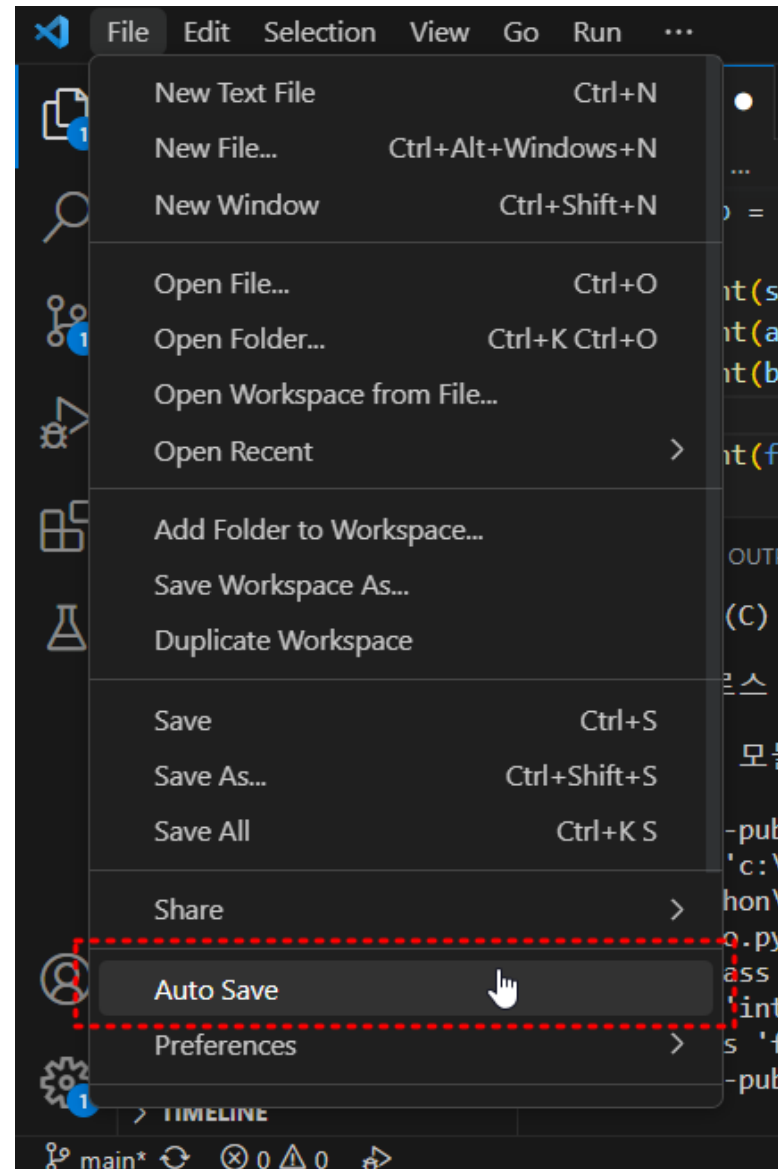
- C:\Users\W[사용자계정]\AppData\Roaming\Code 삭제



- 폴더 repo-publish 생성
 - 파일 생성 후 실행, ctrl + F5



- Auto Save



- 깃 배시 이용

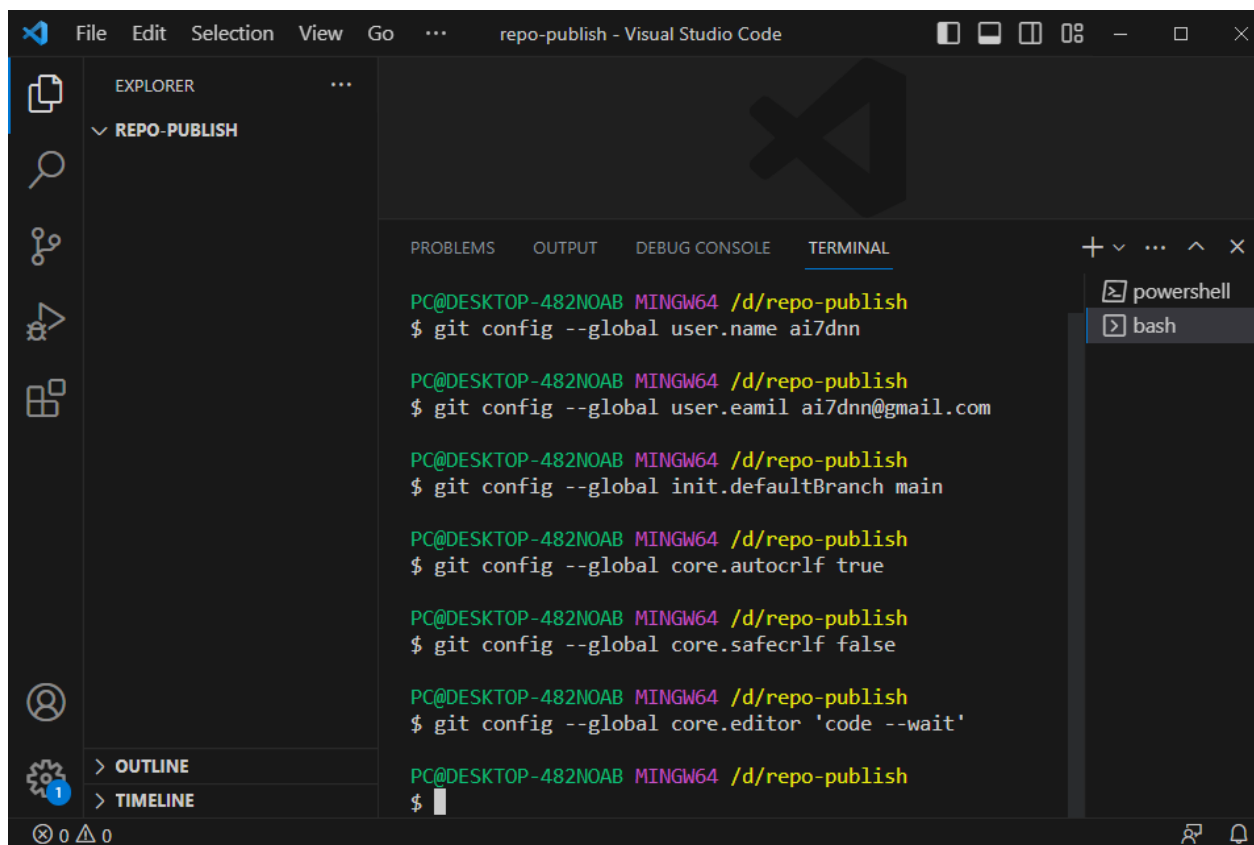
- \$ git config --global user.name
- \$ git config --global user.email
- \$ git config --global core.autocrlf true
- \$ git config --global core.safecrlf false
- \$ git config --global core.editor 'code --wait'
- \$ git config --global init.defaultBranch main



```
MINGW64:/  
  
112@102-PRO MINGW64 /  
$ git config --global user.name  
chatkang  
  
112@102-PRO MINGW64 /  
$ git config --global user.email  
chatkang@gmail.com  
  
112@102-PRO MINGW64 /  
$
```

- 깃 사용자와 전자메일 등 설정

- \$ git config --global user.name
- \$ git config --global user.email
- \$ git config --global core.autocrlf true
- \$ git config --global core.safecrlf false
- \$ git config --global core.editor 'code --wait'
- \$ git config --global init.defaultBranch main



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and their outputs:

```
PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global user.name ai7dnn

PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global user.eamil ai7dnn@gmail.com

PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global init.defaultBranch main

PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global core.autocrlf true

PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global core.safecrlf false

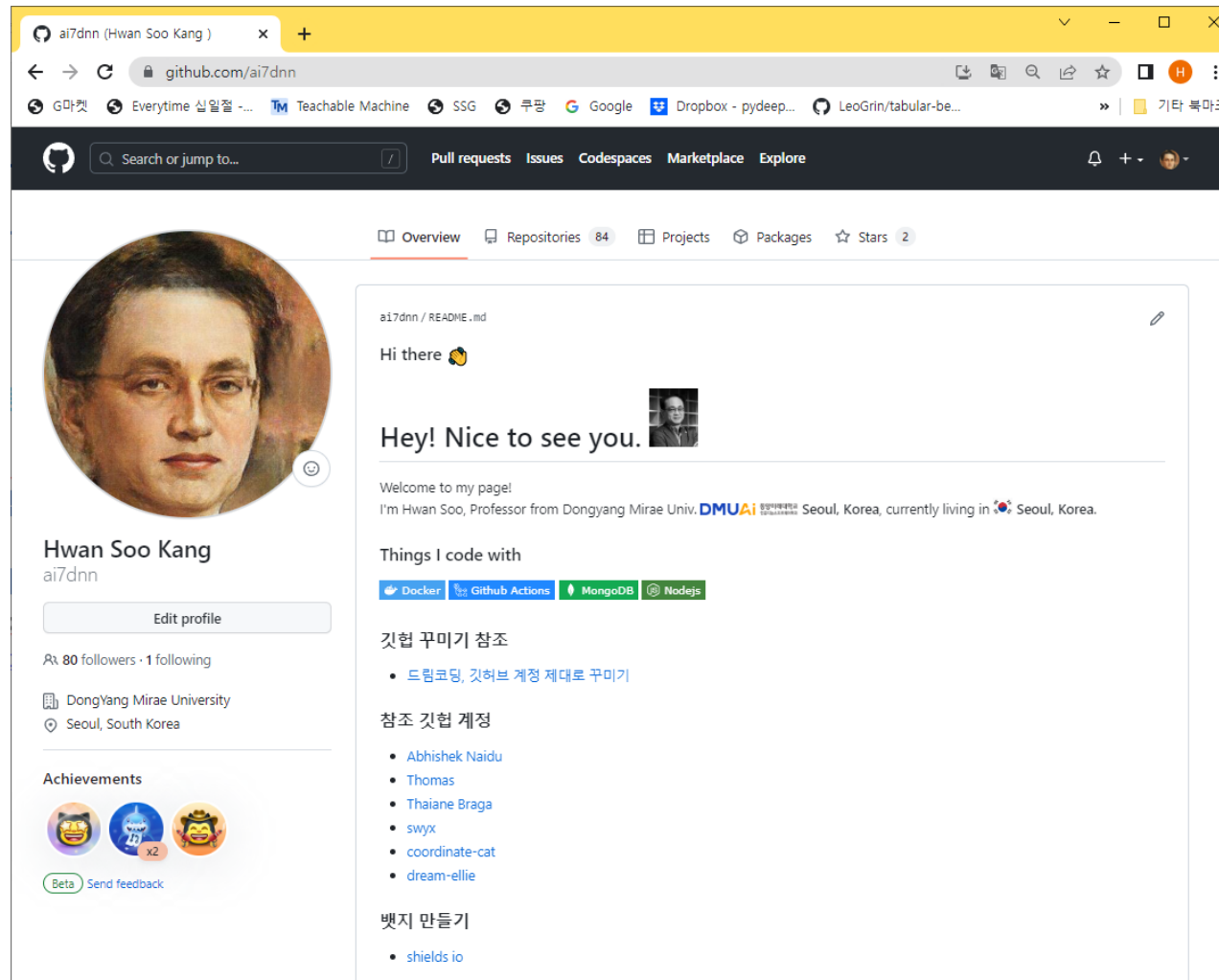
PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$ git config --global core.editor 'code --wait'

PC@DESKTOP-482NOAB MINGW64 /d/repo-publish
$
```

깃허브에 로그인 한 후 게시 시작

깃과 깃허브 Python language

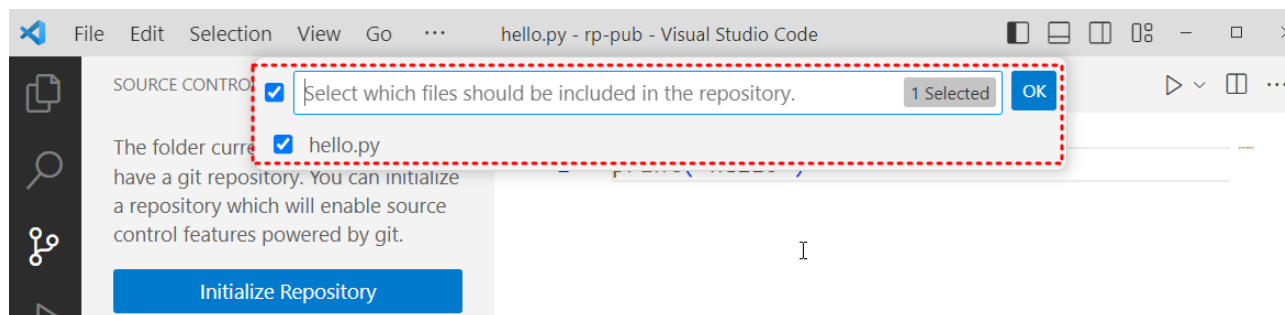
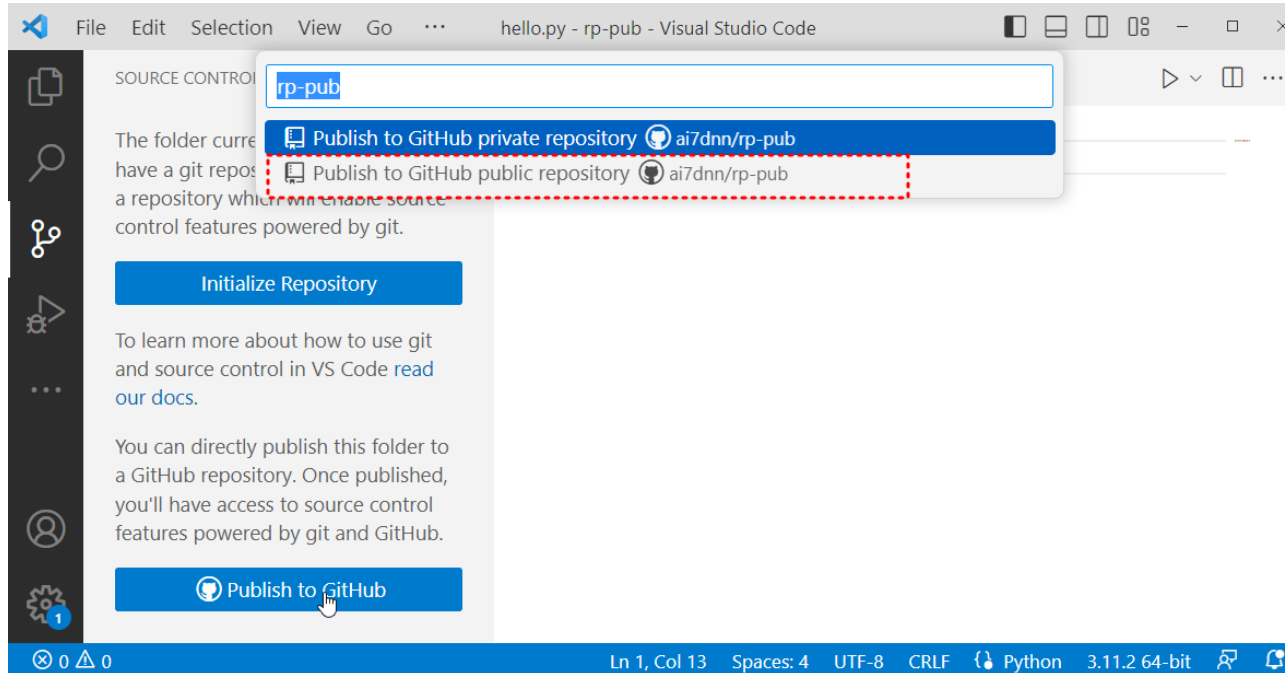
- 먼저 자신의 깃허브에 로그인



바로 퍼블리시: 가능 하나 커밋하고 하는 방향으로

깃과 깃허브 Python language

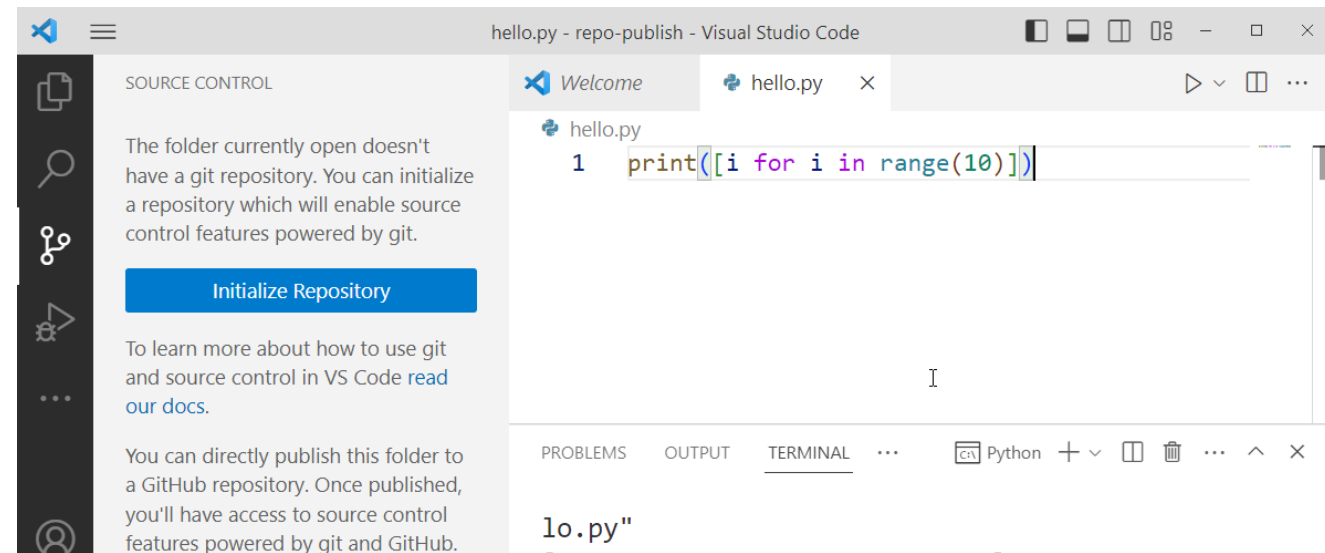
- 파란색 Publish to GitHub 클릭
 - 팔레트: Publish to GitHub public repository ... 클릭



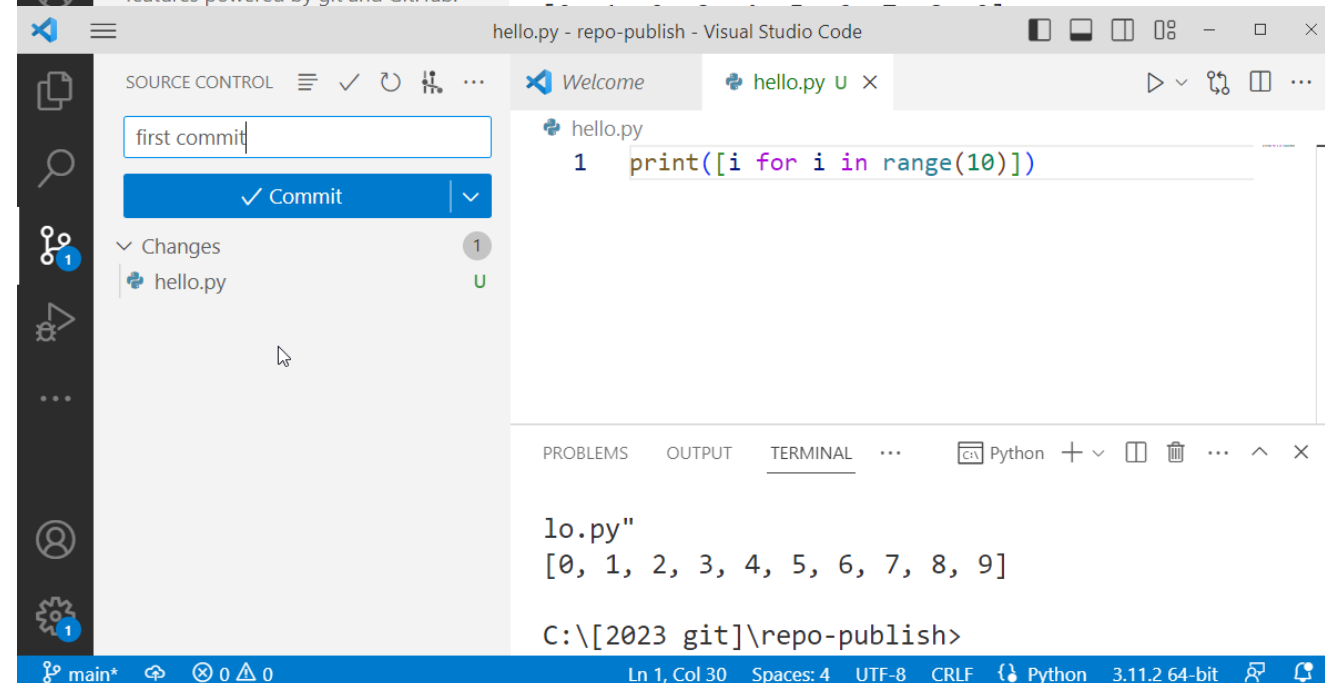
저장소 초기화와 파일 커밋 후 퍼블리시

깃과 깃허브 Python language

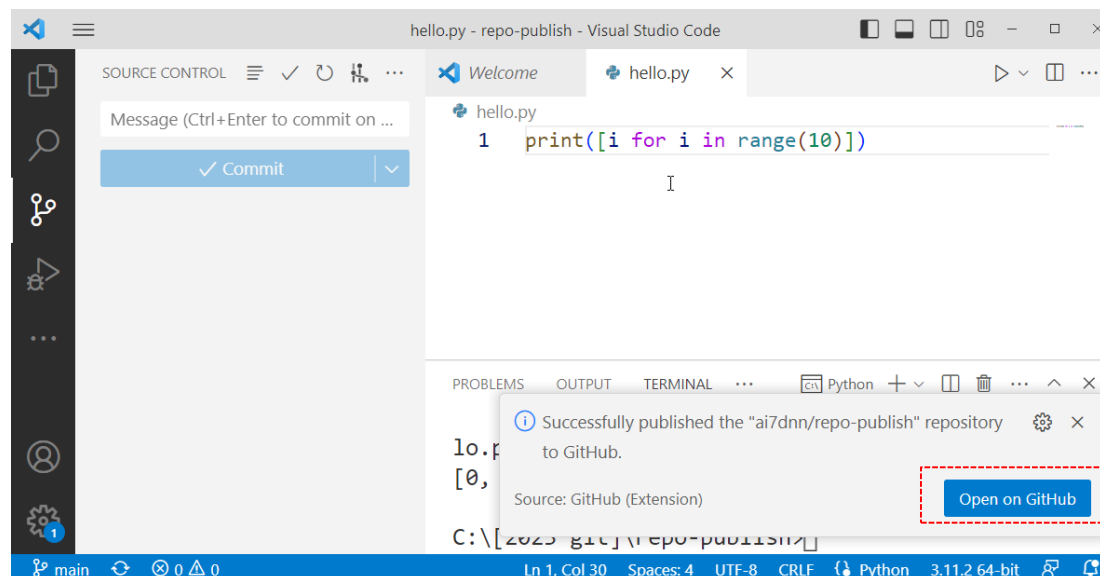
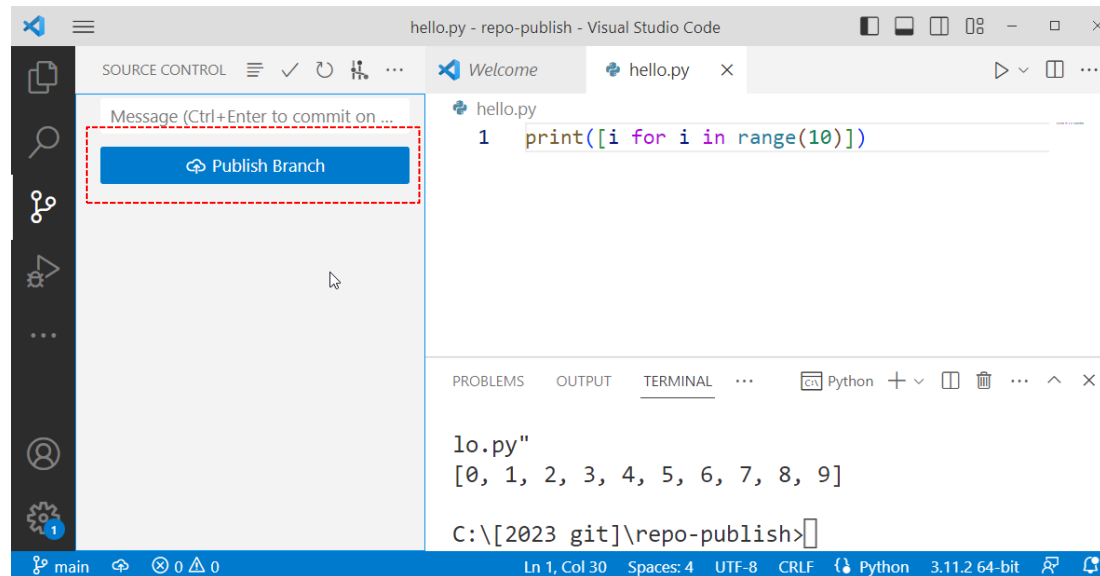
- Initialize Repository



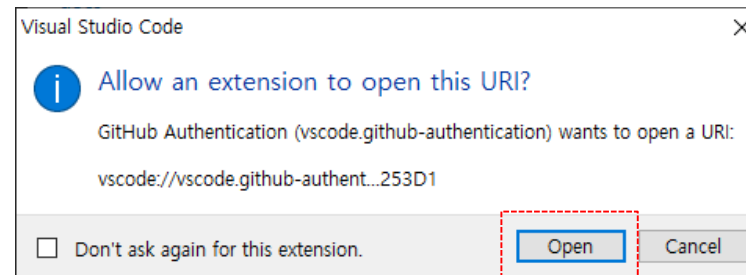
- Commit



- Publish To Branch 클릭



- Open



- 성공 후 자격 증명 관리자 확인

- Vscodevscode.github-authentication/github.auth 추가됨

일반 자격 증명

일반 자격 증명 추가

vscodevscode.github-authentication/github.auth

수정한 날짜: 오늘 ^

인터넷 또는 네트워크 주소:

vscodevscode.github-authentication/github.auth

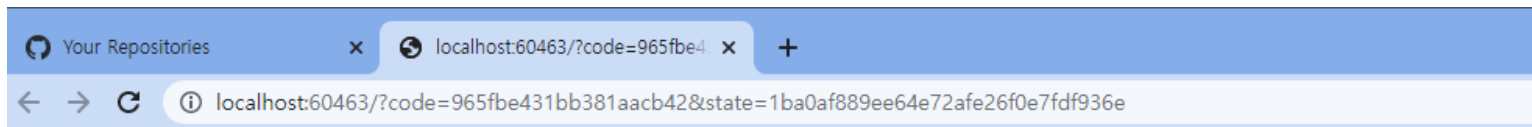
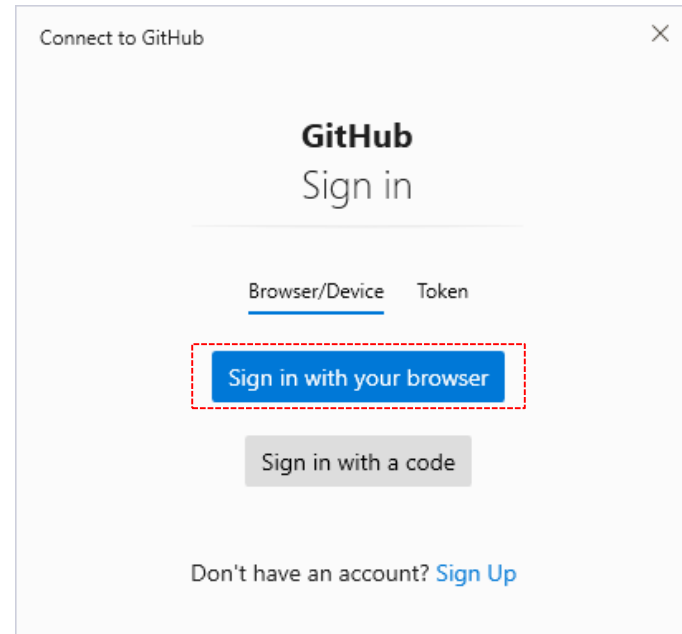
사용자 이름: github.auth

암호:

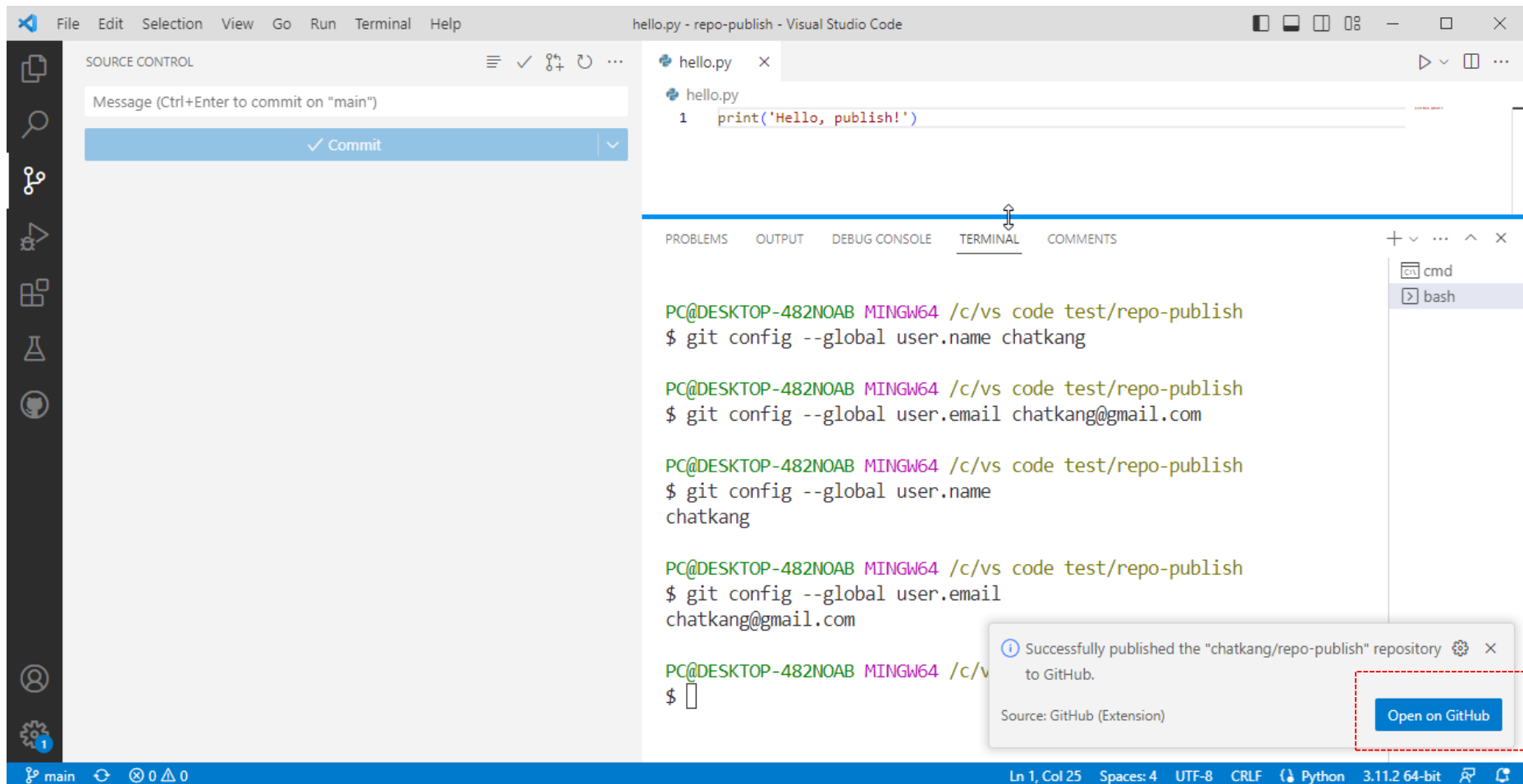
지속성: 엔터프라이즈

편집 제거

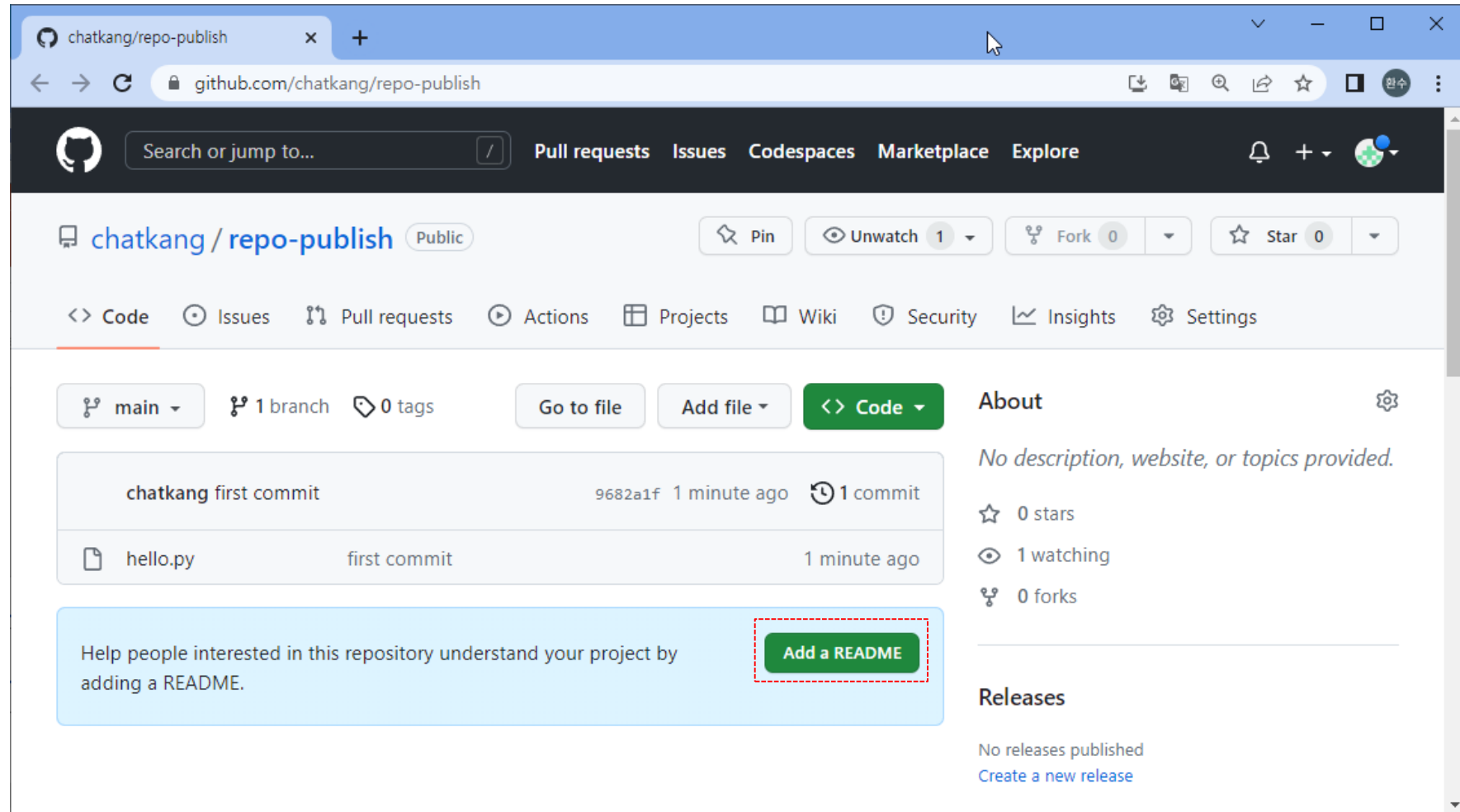
- 이미 로그인 한 이후이므로 빈 브라우저 실행



- 게시 성공 후
 - Open on github 클릭



- Add a README 클릭



- Readme.md 파일 편집 후 commit

The image consists of two overlapping screenshots of the GitHub web interface. The left screenshot shows the 'New File' page for the repository 'chatkang / repo-publish'. The 'README.md' file is selected, and the 'Edit new file' tab is active. The content of the file is:

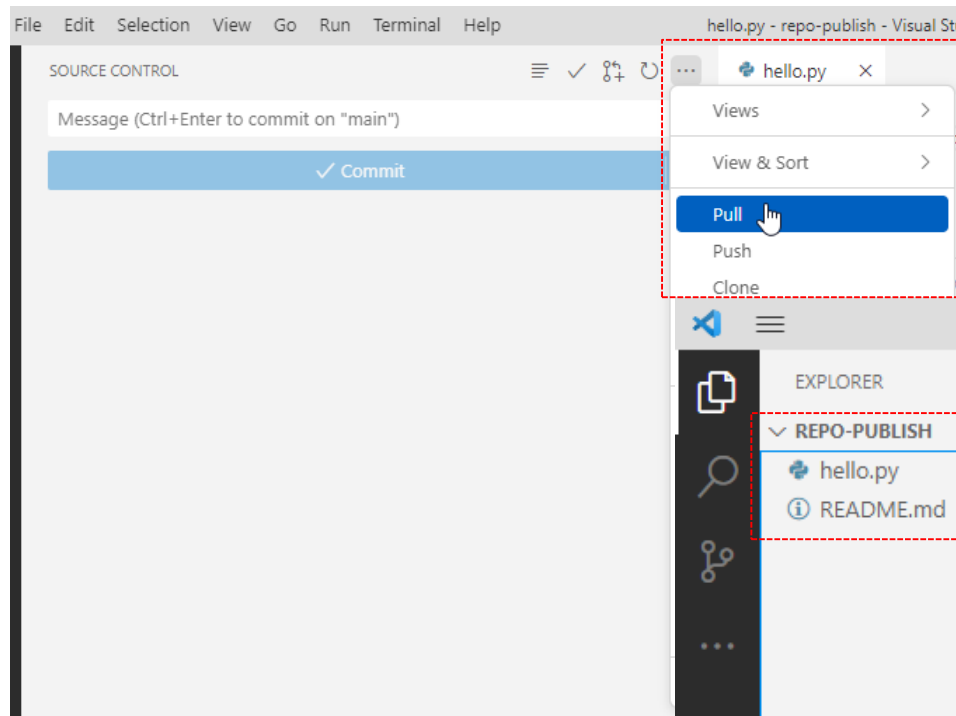
```
1 # 저장소 repo-publish
2
3 ```퍼블리쉬 점검```
```

 The right screenshot shows the repository page for 'chatkang / repo-publish'. The 'README.md' file is highlighted in the file list. The preview of the README.md file shows the text:

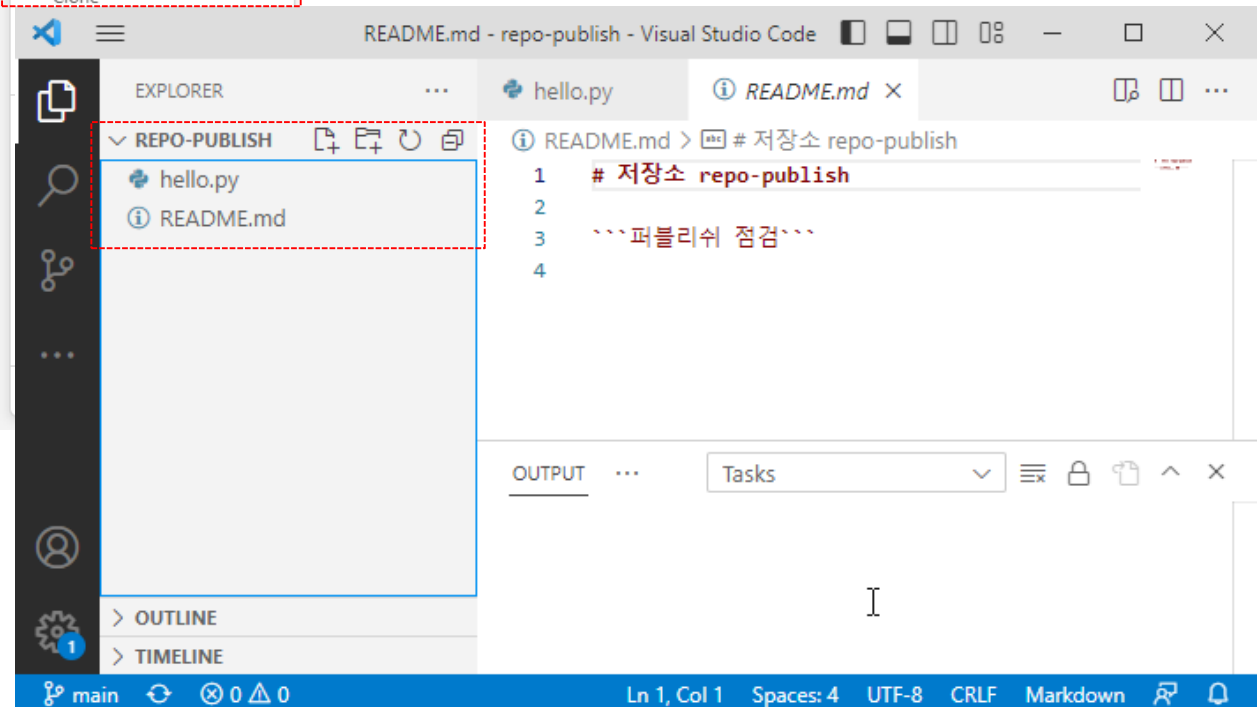
저장소 repo-publish

 and a button labeled '퍼블리쉬 점검'.

- 현재 상황이 깃허브 저장소가 지역 저장소보다 더 수정된 상태
 - Pull
 - 원격 저장소 수정을 지역으로 가져오기



지역 저장소는 Push보다
항상 Pull을 먼저 수행해야 함



- Git Graph 보기
- \$ git branch -a -vv
- 옵션 -r

The screenshot shows the Visual Studio Code interface with the Git Graph extension open. The Git Graph view displays a commit history table with columns: Graph, Description, Date, Author, and Commit. The first commit is highlighted, showing it was created on 4 Jun 2023 at 13:36 by Hwan Soo Kang with commit hash 17685689. The second commit is also shown, created on 4 Jun 2023 at 13:32 by ai7dnn with commit hash dde3d1df. Below the Git Graph view, the terminal window shows the output of the command `$ git log --oneline --all`, which lists the commits. The output is highlighted with a red dashed box. The terminal also shows the output of the command `$ git branch -vv -a`, which lists the branches and their associated commits. The output is also highlighted with a red dashed box. The terminal window is titled 'cmd' and 'bash'.

Graph	Description	Date	Author	Commit
1768568 (HEAD -> main, origin/main)	Create README.md	4 Jun 2023 13:36	Hwan Soo Kang	17685689
dde3d1d	first commit	4 Jun 2023 13:32	ai7dnn	dde3d1df

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (main)
$ git log --oneline --all
1768568 (HEAD -> main, origin/main) Create README.md
dde3d1d first commit

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (main)
$ git branch -vv -a
* main 1768568 [origin/main] Create README.md
remotes/origin/main 1768568 Create README.md

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (main)
$
```


- 다음에 다른 사용자로 접속하려면
 - 다음을 제거하고 다시 시도

자격 증명 관리

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.



웹 자격 증명



Windows 자격 증명

[자격 증명 백업\(B\)](#) [자격 증명 복원\(R\)](#)

Windows 자격 증명

[Windows 자격 증명 추가](#)

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명

[인증서 기반 자격 증명 추가](#)

인증서가 없습니다.

일반 자격 증명

[일반 자격 증명 추가](#)

vscodevscode.github-authentication/github.auth

수정한 날짜: 오늘 ⌵

인터넷 또는 네트워크 주소:
vscodevscode.github-authentication/github.auth
사용자 이름: github.auth
암호:
지속성: 엔터프라이즈

[편집](#) [제거](#)

git:https://github.com

수정한 날짜: 오늘 ⌵

인터넷 또는 네트워크 주소: git:https://github.com
사용자 이름: chatkang
암호:
지속성: 로컬 컴퓨터

[편집](#) [제거](#)

virtualapp/didlogical

수정한 날짜: 2023-02-20 ⌵

SSO_POP_Device

수정한 날짜: 오늘 ⌵

XboxLive

수정한 날짜: 2023-05-04 ⌵

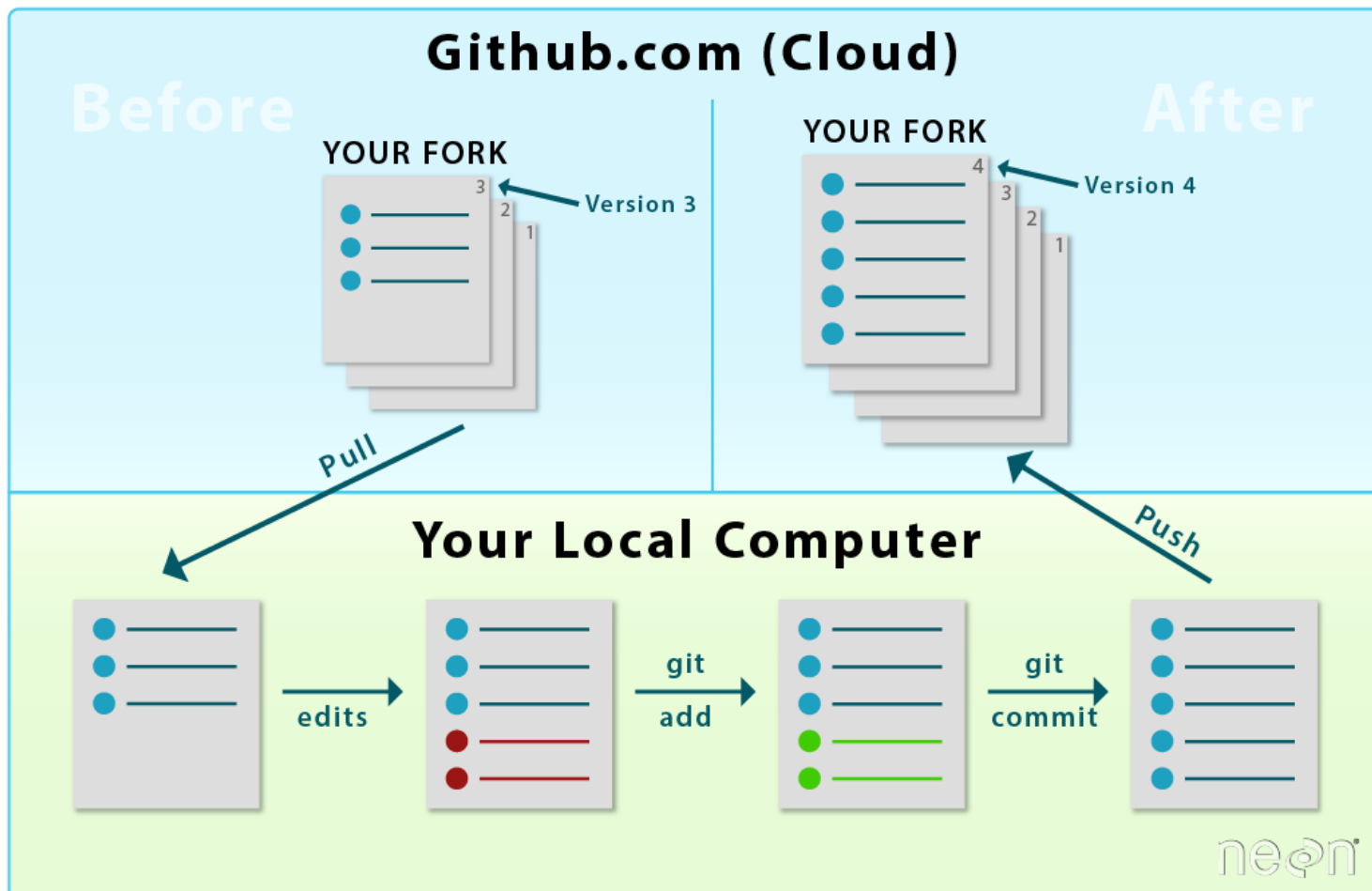
AI Experts
Who Lead
The Future

02

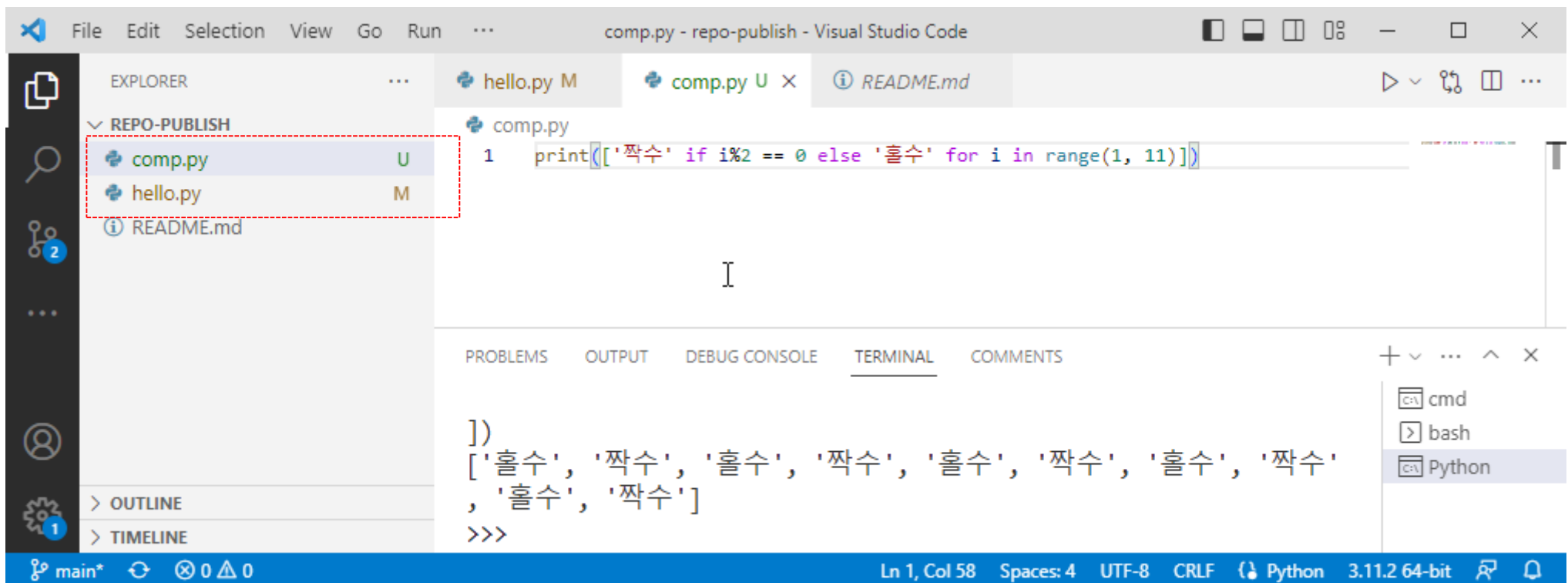
지역 저장소 수정 후 push



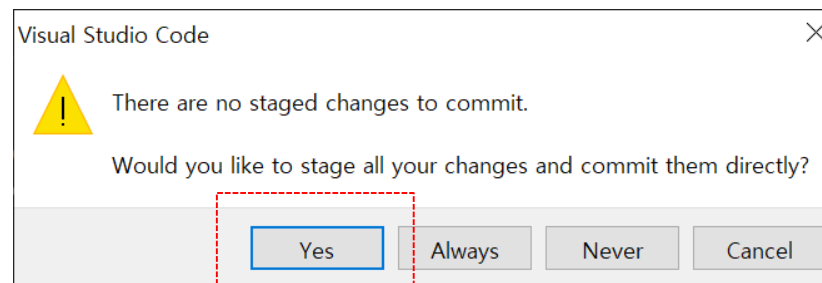
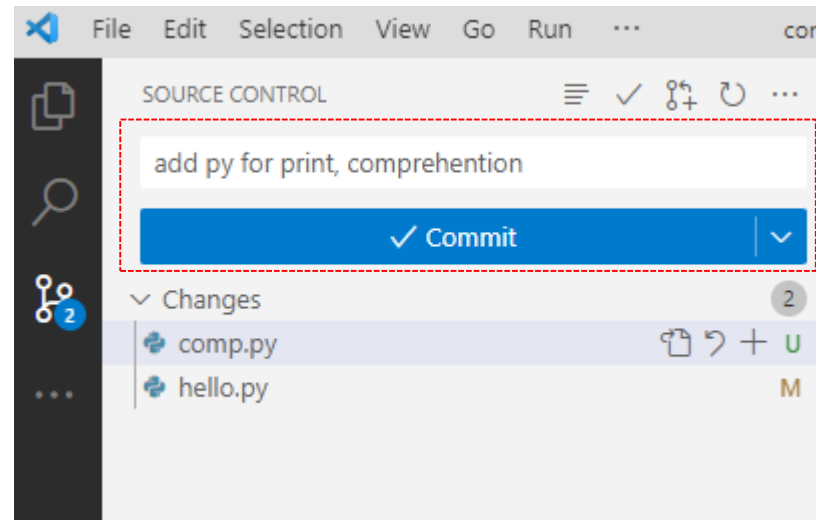
- 깃 저장소에 저장하는 절차
 - Add
 - Commit



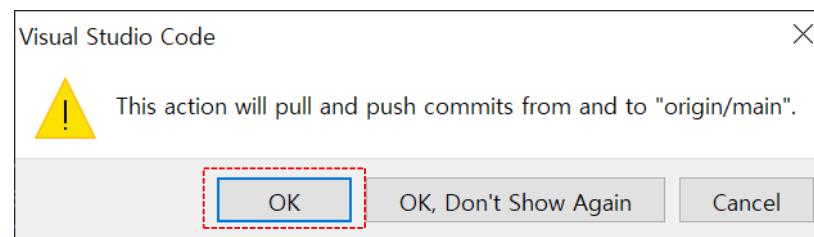
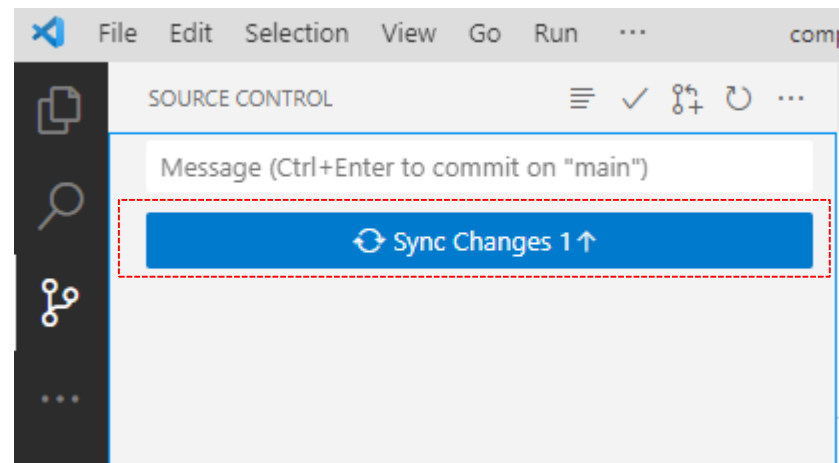
- 파일 수정과 추가
 - U: untracked
 - 새로 추가된 파일
 - M: modified
 - 수정된 파일

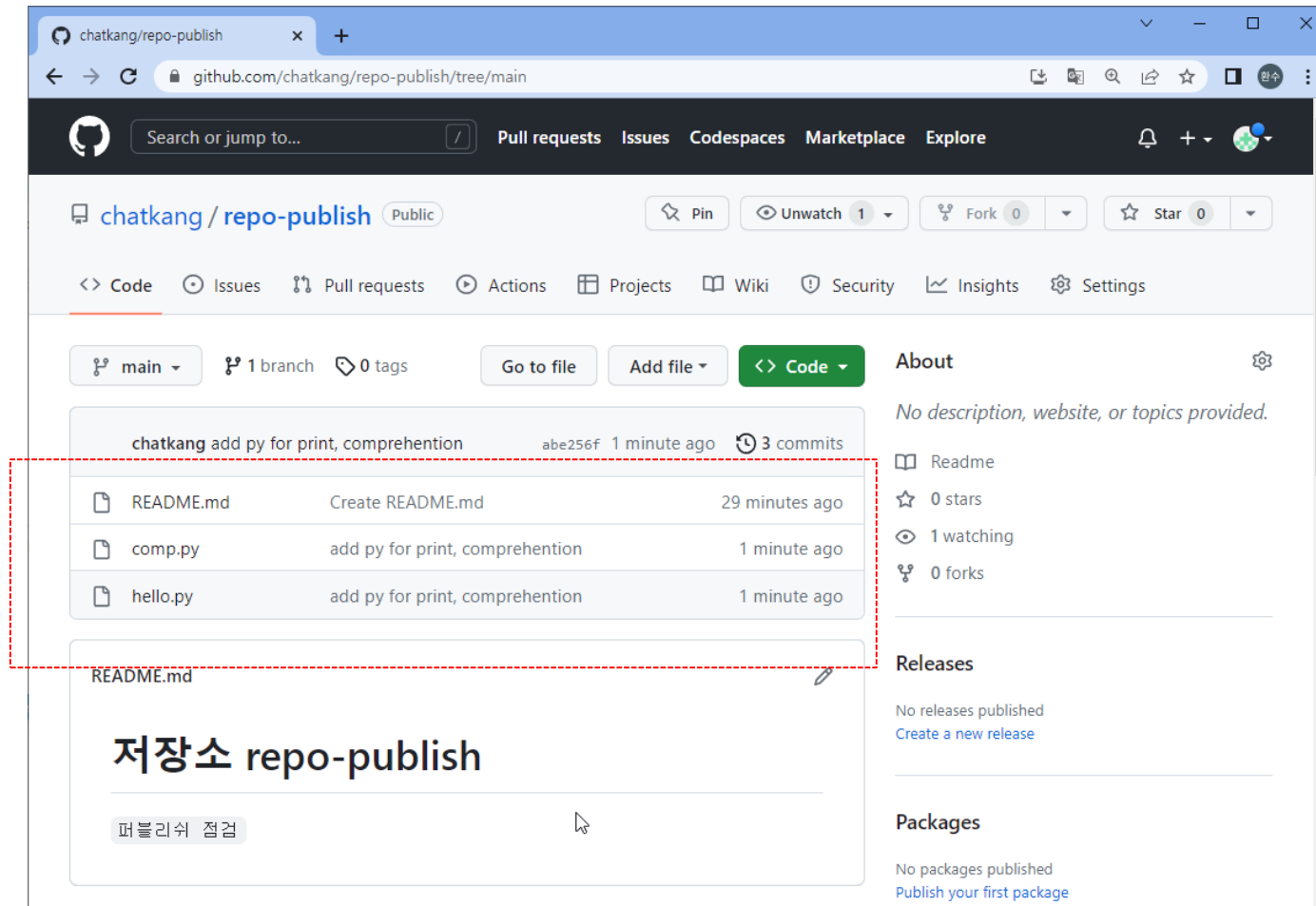


- 파일 수정과 추가
- 커밋
 - Commit 버튼 누르기
 - 메시지 작성 후
 - 버전 관리를 위해 현재 상태를 저장소에 저장하는 과정

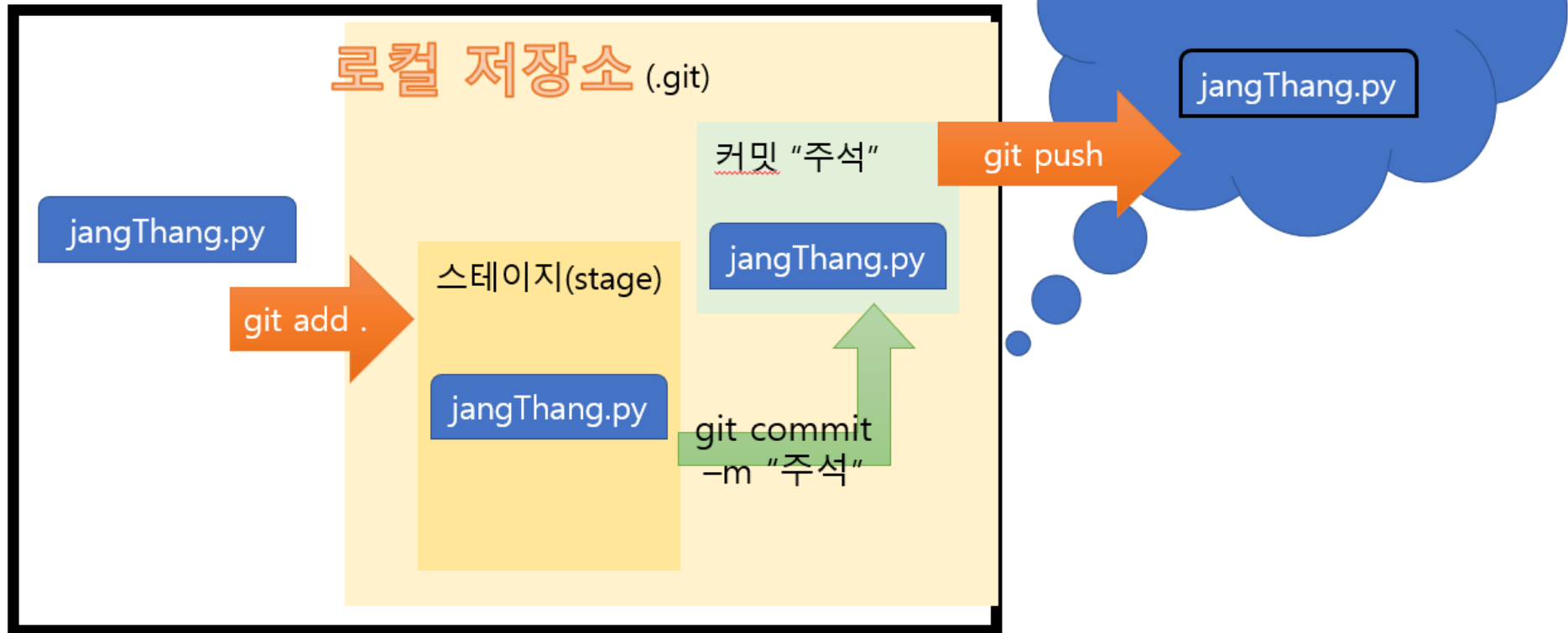


- **Push**
 - **Sync Changes**
 - 지역 저장소의 수정을 원격 저장소에 반영





작업공간 (프로젝트 폴더, Working Directory)



AI Experts
Who Lead
The Future

03

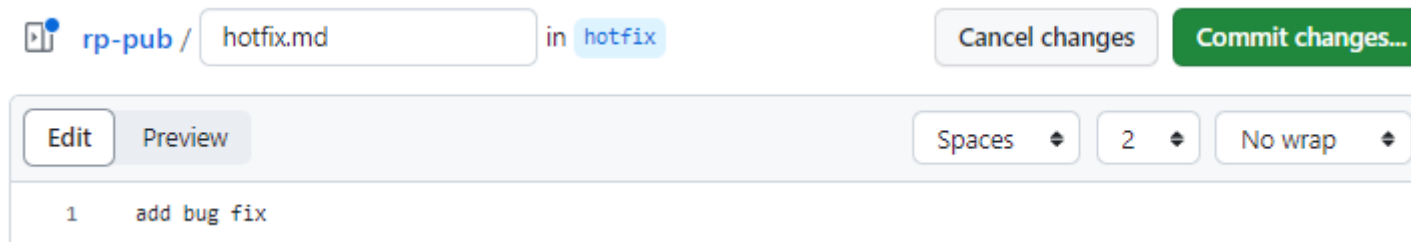
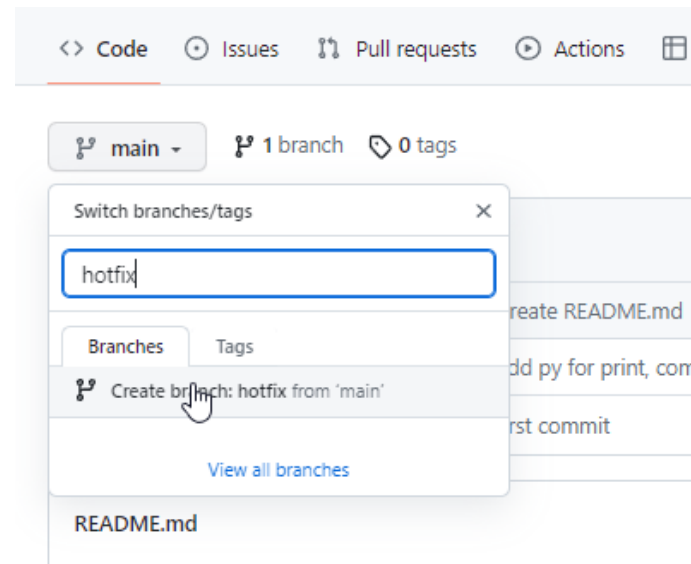
깃허브 새로운 브랜치 복제



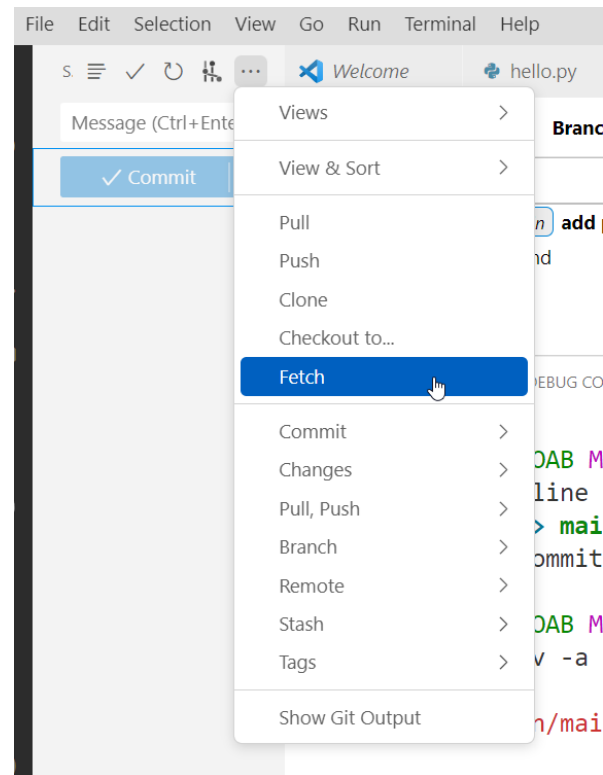
깃허브에서 새로운 브랜치 hotfix 생성

깃과 깃허브 Python language

- 파일도 생성해 커밋



- 메뉴 fetch



- Refresh Git graph

A screenshot of the Git Graph view in Visual Studio Code. The view shows a list of commits with columns for Graph, Description, Date, Author, and Commit. The 'Show Remote Branches' checkbox is checked. The commits listed are:

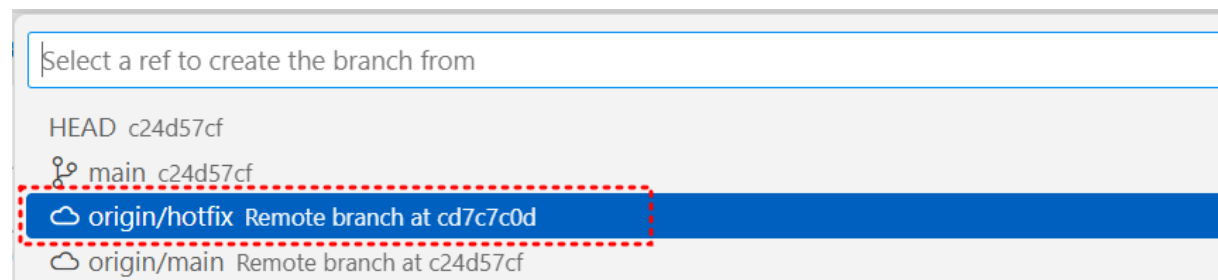
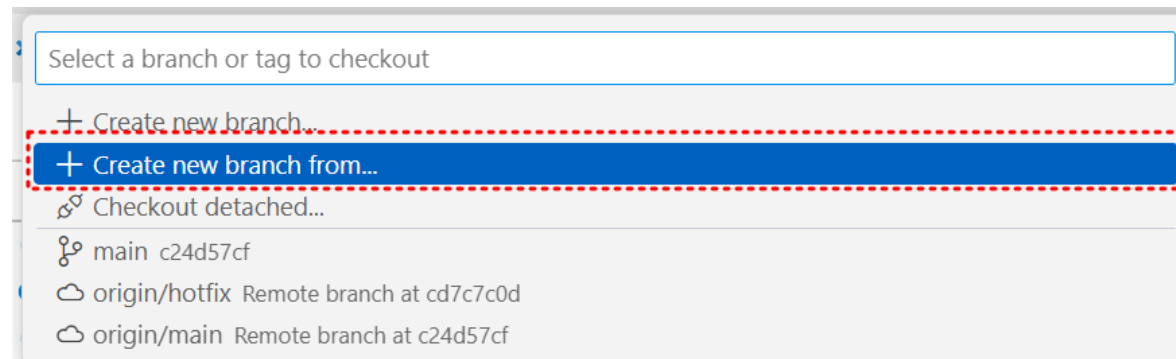
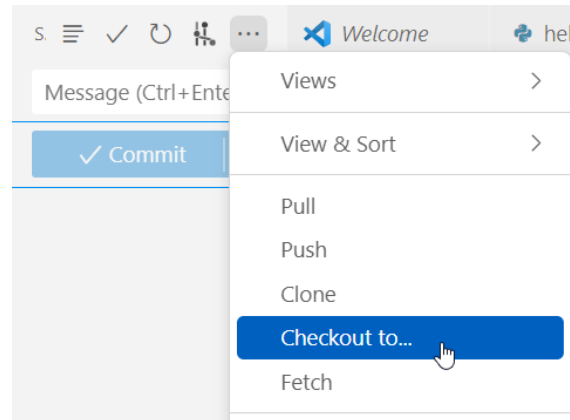
Graph	Description	Date	Author	Commit
origin/hotfix	Create hotfix.md	4 Jun 2023 13:52	Hwan Soo Kang	cd7c7c0d
main origin	add py for print, comprehensi...	4 Jun 2023 13:47	ai7dnn	c24d57cf
	Create README.md	4 Jun 2023 13:36	Hwan Soo Kang	17685689
	first commit	4 Jun 2023 13:32	ai7dnn	dde3d1df

```
$ git branch -vv -a
* main                c24d57c [origin/main] add py for print, comprehension

remotes/origin/hotfix cd7c7c0 Create hotfix.md
remotes/origin/main   c24d57c add py for print, comprehension

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (main)
$ git log --oneline --all
cd7c7c0 (origin/hotfix) Create hotfix.md
c24d57c (HEAD -> main, origin/main) add py for print, comprehension
1768568 Create README.md
dde3d1d first commit
```

- 메뉴 Checkout to



- 아직 origin/hotfix와 추적은 안됨
 - 왼쪽 파란색 Publish Branch 클릭

The screenshot shows the Visual Studio Code interface with the Git Graph extension. The 'Publish Branch' button is highlighted with a red dashed box. The terminal window shows the output of the `git branch -vv -a` command, with the hotfix branch details highlighted by a red dashed box.

Graph	Description	Date	Author	Commit
hotfix origin	Create hotfix.md	4 Jun 2023 13:52	Hwan Soo Kang	cd7c7c0d
main origin	add py for print, comprehension	4 Jun 2023 13:47	ai7dnn	c24d57cf
	Create README.md	4 Jun 2023 13:36	Hwan Soo Kang	17685689
	first commit	4 Jun 2023 13:32	ai7dnn	dde3d1df

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (main)
$ git branch -vv -a
* main c24d57c [origin/main] add py for print, comprehension
remotes/origin/hotfix cd7c7c0 Create hotfix.md
remotes/origin/main c24d57c add py for print, comprehension

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (hotfix)
$
```

Message (Ctrl+Enter) [Commit]

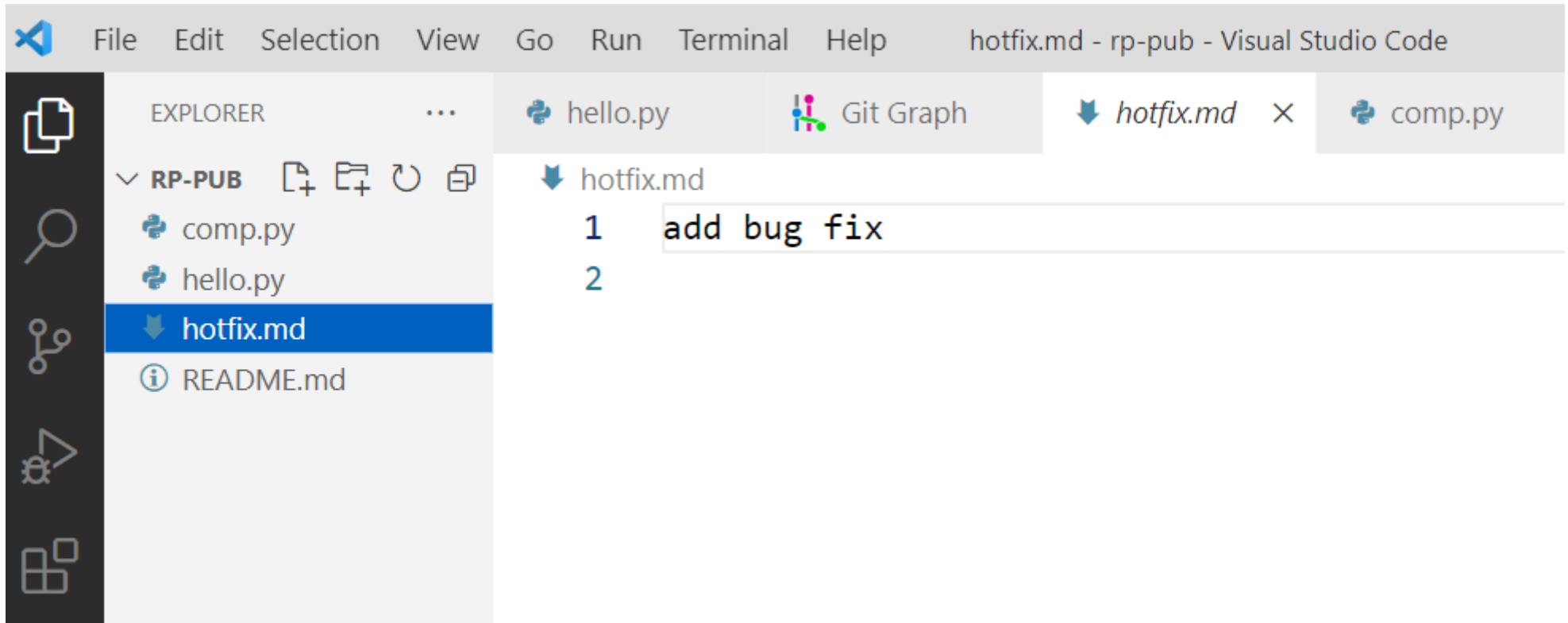
Branches: Show All [Show Remote Branches]

Graph	Description	Date	Author	Commit
hotfix origin	Create hotfix.md	4 Jun 2023 13:52	Hwan Soo Kang	cd7c7c0d
main origin	add py for print, comprehension	4 Jun 2023 13:47	ai7dhn	c24d57cf
	Create README.md	4 Jun 2023 13:36	Hwan Soo Kang	17685689
	first commit	4 Jun 2023 13:32	ai7dhn	dde3d1df

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

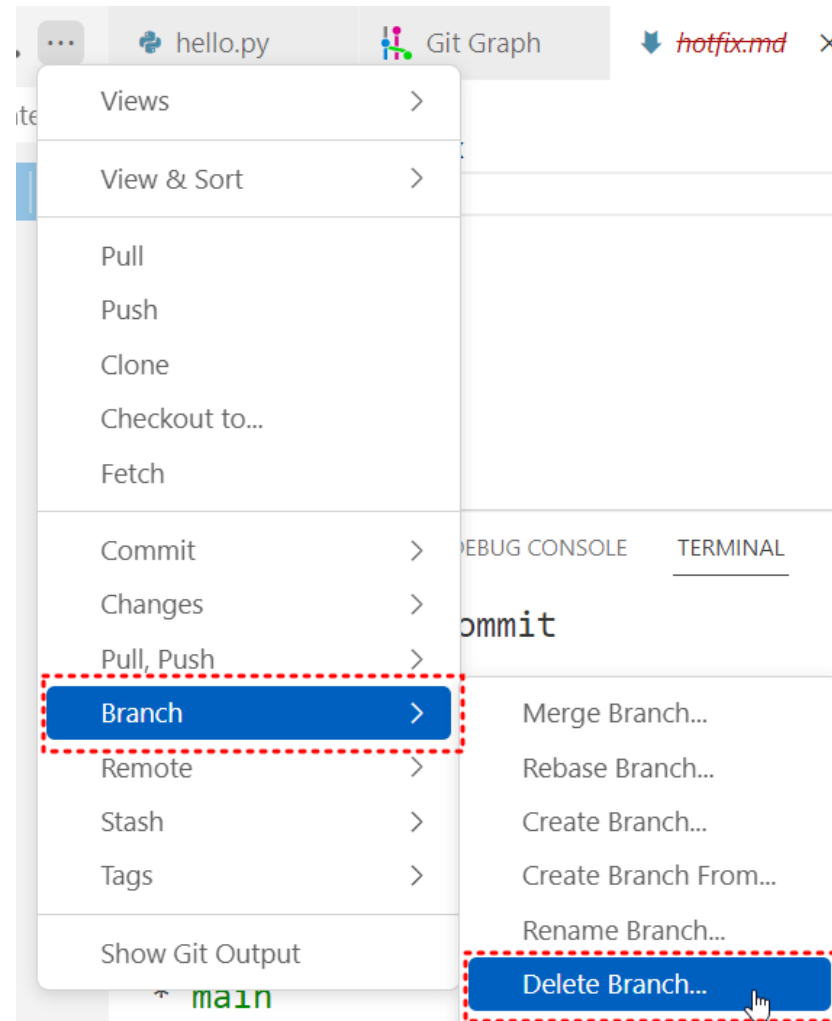
```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (hotfix)
$ git branch -vv -a
* hotfix          cd7c7c0 [origin/hotfix] Create hotfix.md
main             c24d57c [origin/main] add py for print, comprehension
remotes/origin/hotfix cd7c7c0 Create hotfix.md
remotes/origin/main  c24d57c add py for print, comprehension

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/rp-pub (hotfix)
$ git log --oneline --all
cd7c7c0 (HEAD -> hotfix, origin/hotfix) Create hotfix.md
c24d57c (origin/main, main) add py for print, comprehension
1768568 Create README.md
dde3d1d first commit
```



- 먼저 checkout to main

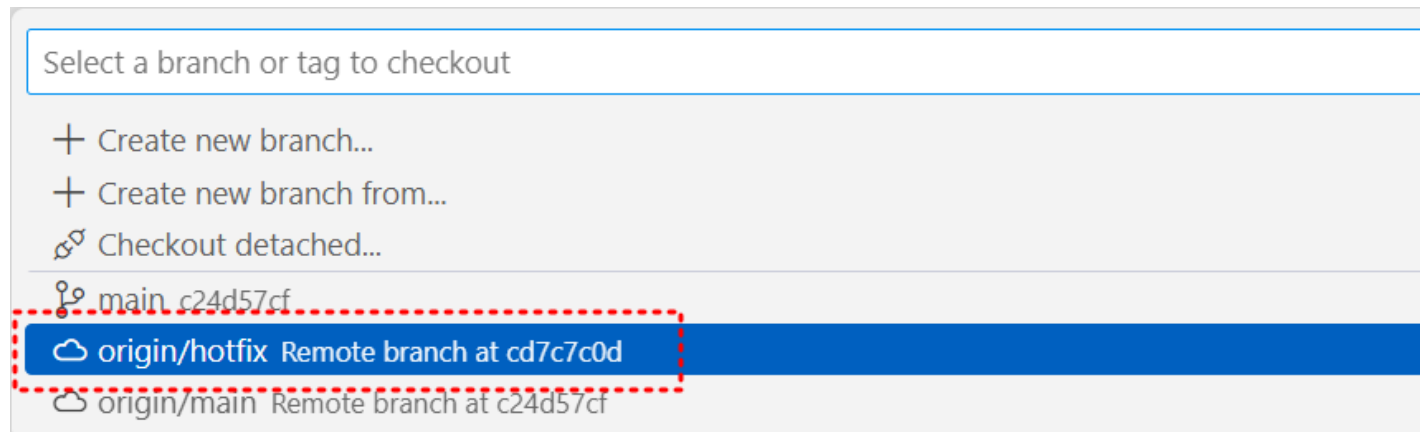
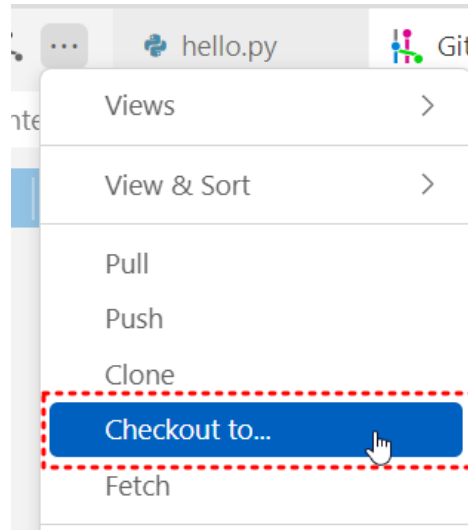
- 메뉴 Branch | Delete Branch



- `$ git checkout hotfix`

바로 원격 브랜치로 체크아웃해도 생성 및 체크아웃 가능

깃과 깃허브 Python language



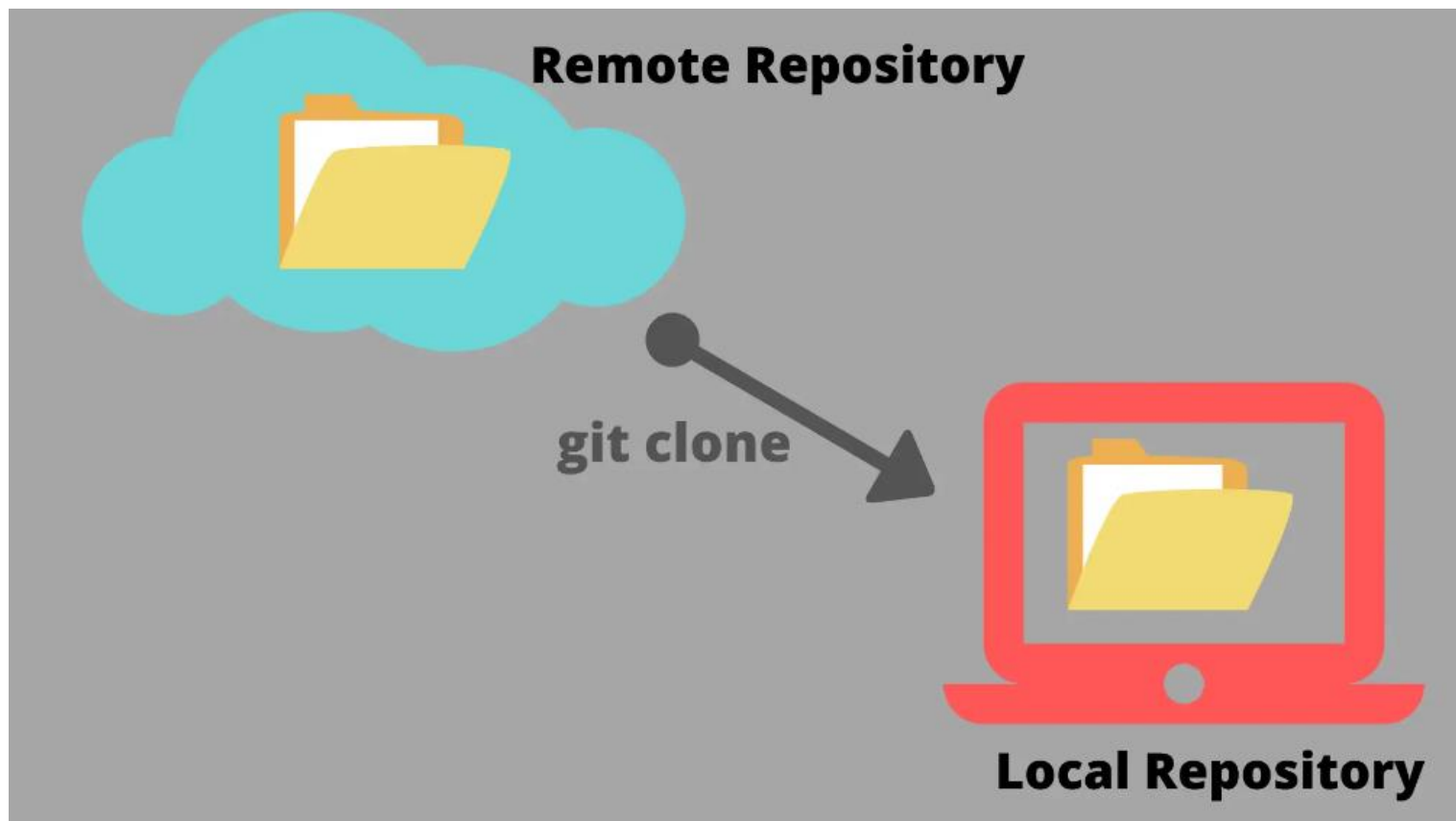
AI Experts
Who Lead
The Future

04

깃허브 저장소 복제



- 공개된 원격 저장소를 자신의 PC 저장소로 가져오기
 - 자신의 저장소나 타인의 저장소 모두 가능

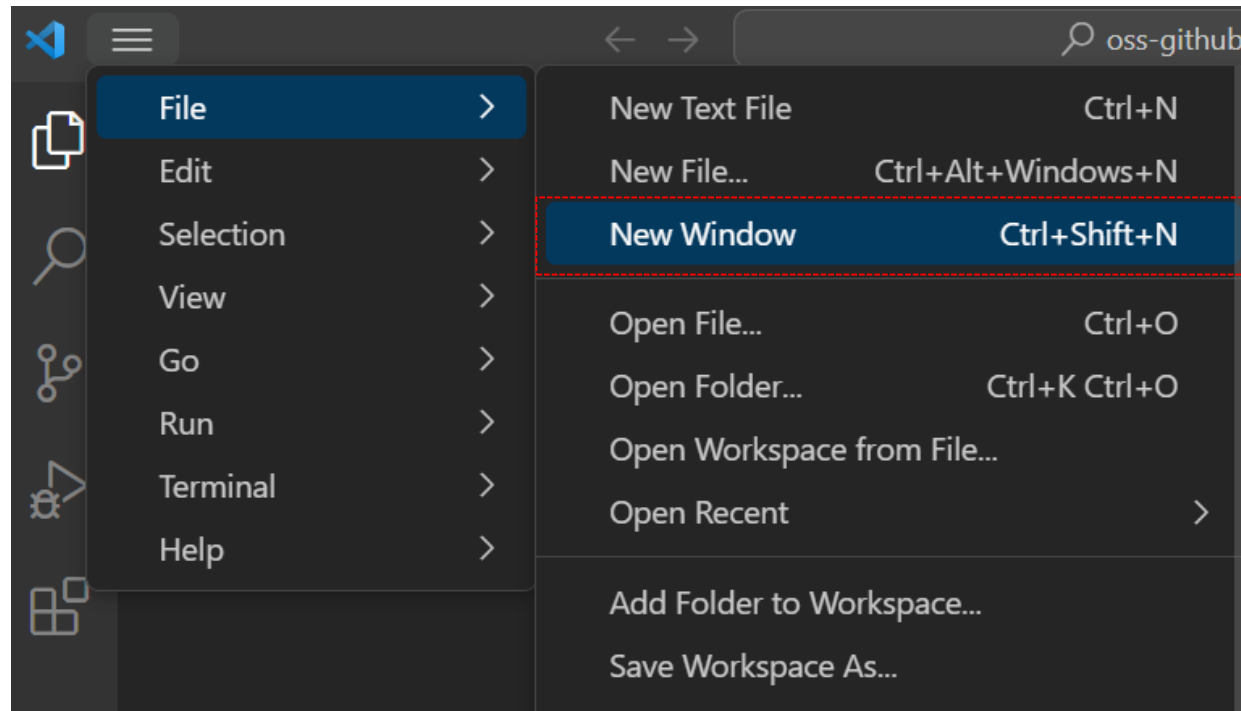


- 유명 공개 저장소 주소
 - 깃
 - <https://github.com/git/git.git>
 - Vs code
 - <https://github.com/microsoft/vscode.git>
 - 쿠버네티스
 - <https://github.com/kubernetes/kubernetes.git>
 - 텐서플로
 - <https://github.com/tensorflow/tensorflow.git>
 - 파이토치
 - <https://github.com/pytorch/pytorch.git>
 - 딥드림
 - <https://github.com/google/deepdream.git>
 - 리눅스
 - <https://github.com/torvalds/linux.git>

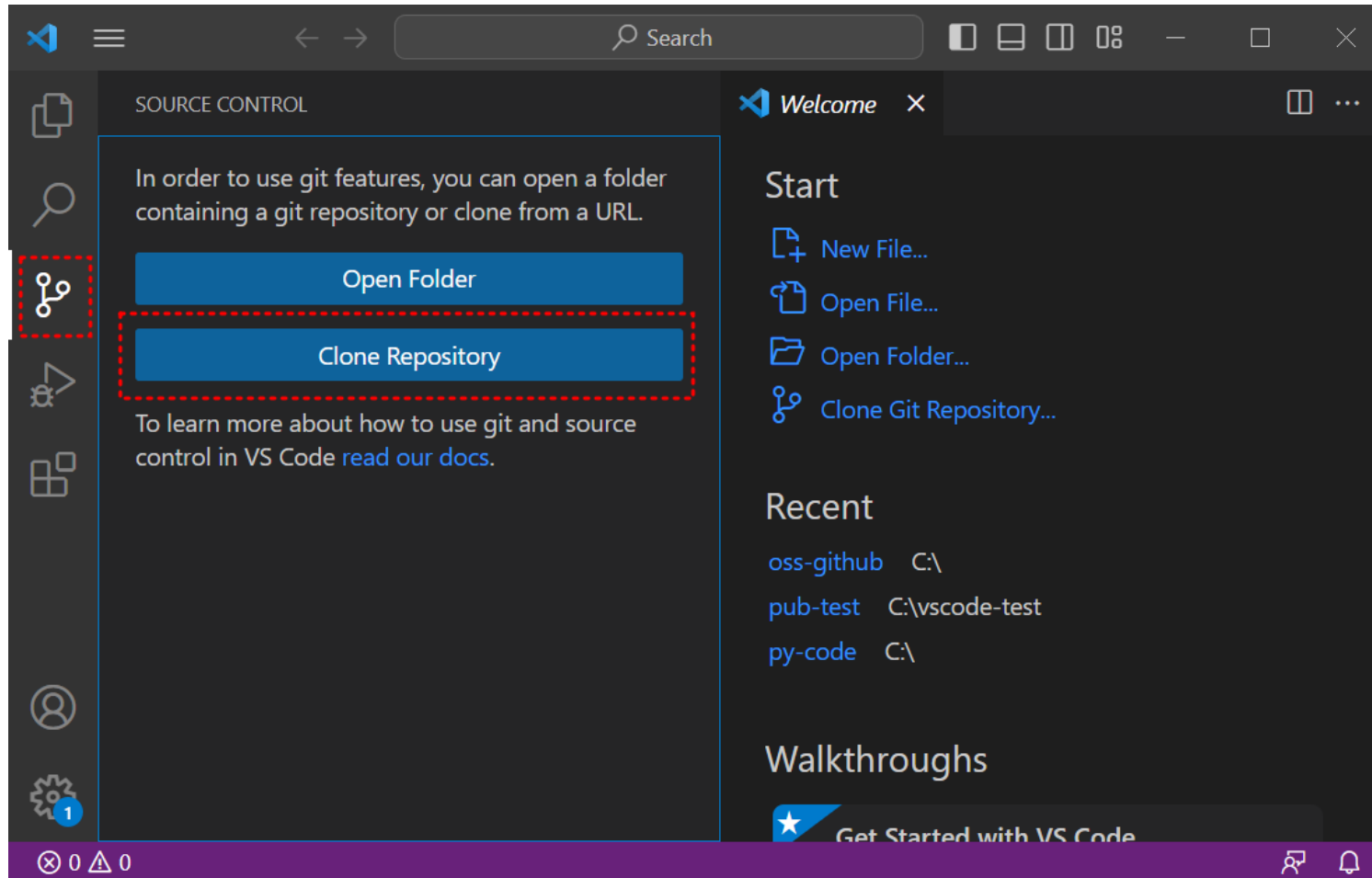
방법 1: 새로운 윈도우에서

깃과 깃허브 Python language

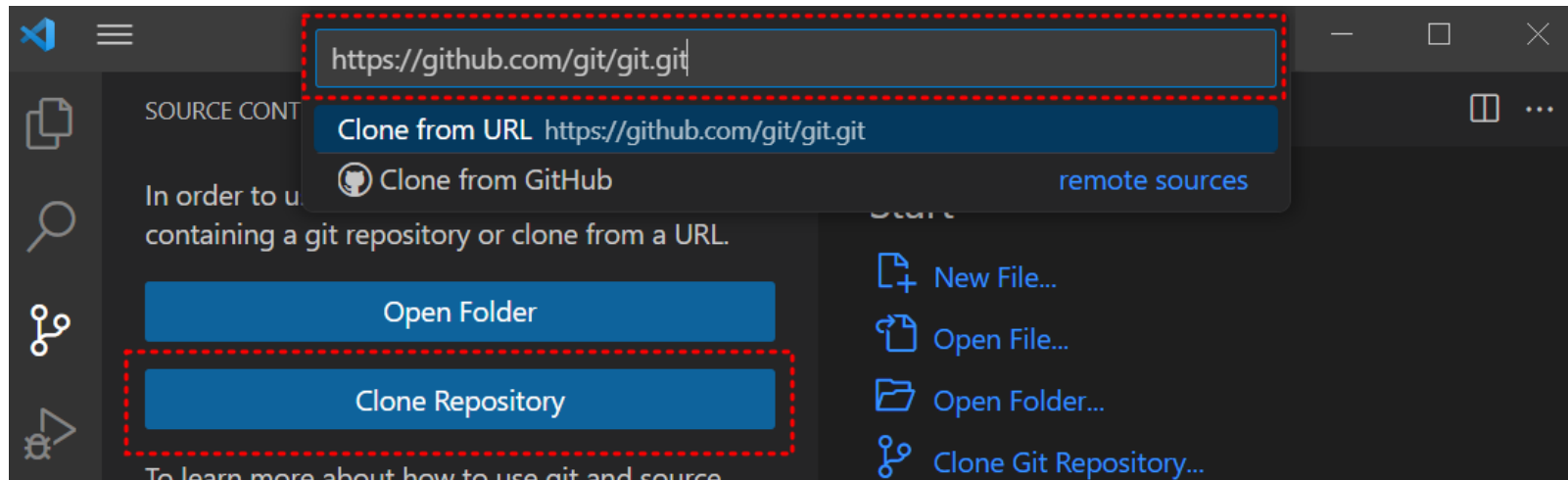
- 복제할 저장소의 주소를 복사
 - <https://github.com/git/git.git>
- 메뉴 File | New Window
 - 단축키 Ctrl + Shift + N



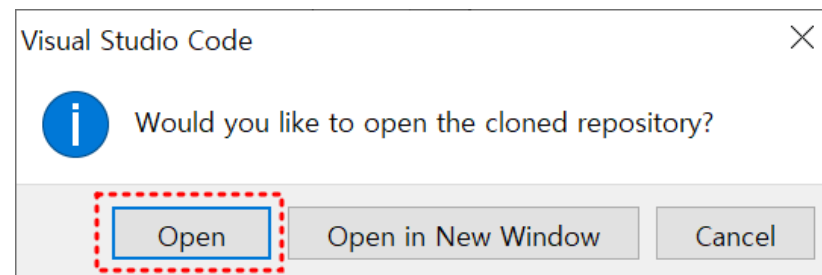
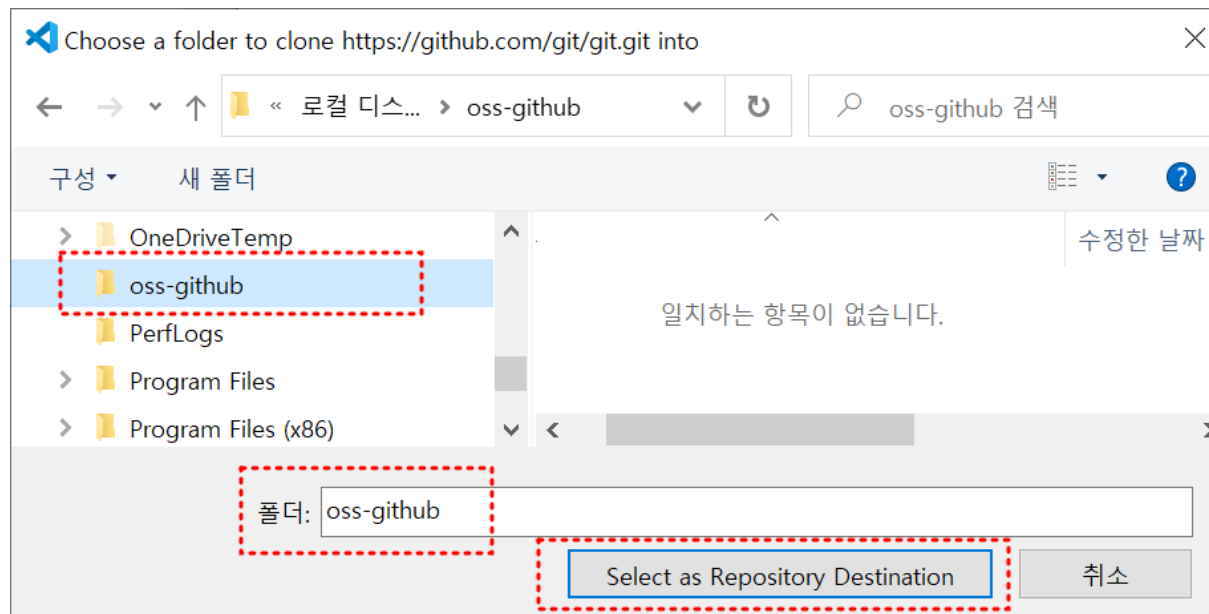
- Clone Repository 클릭



- 편집창에 붙이기

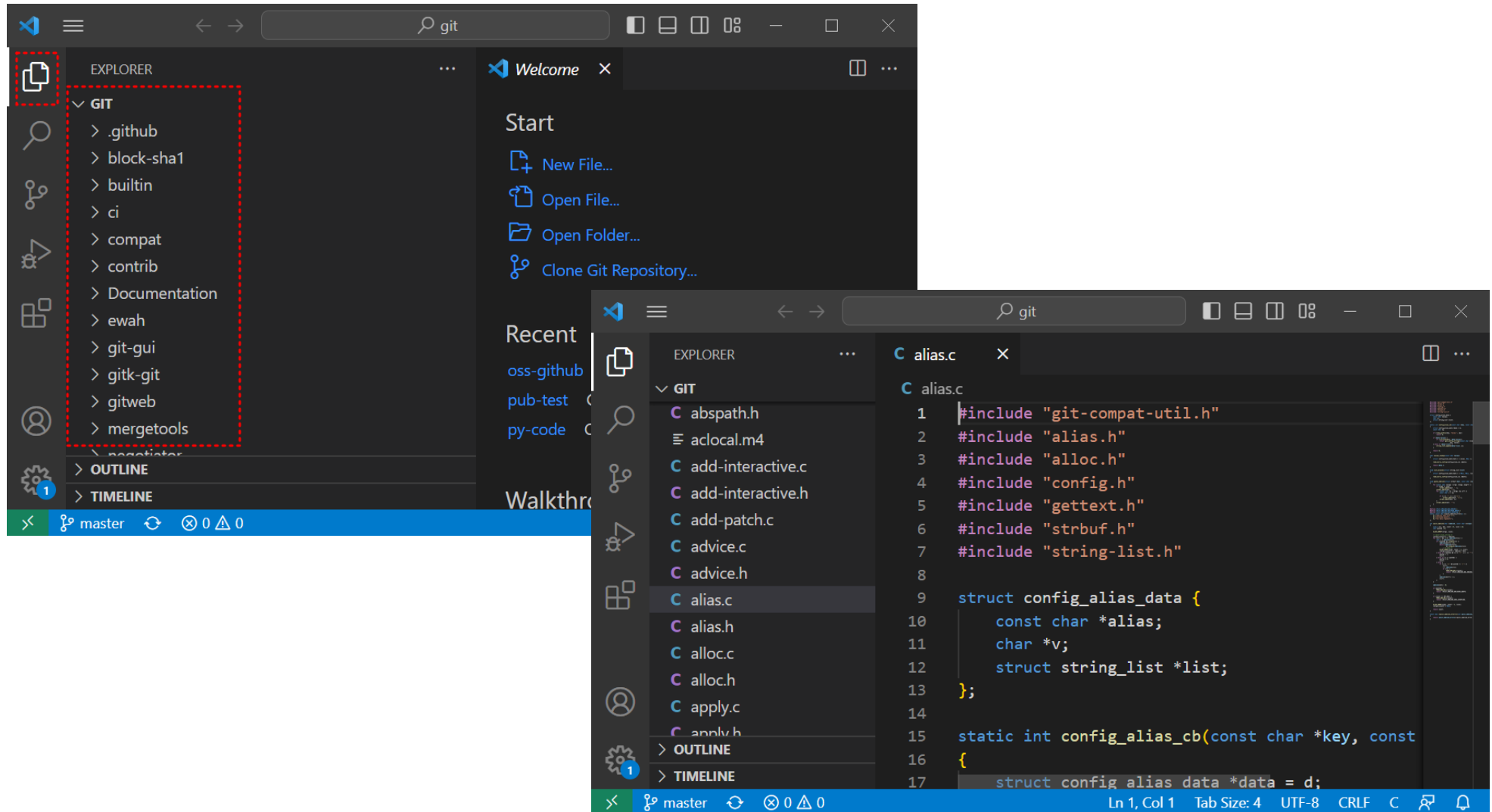


- 상위 폴더가 됨
 - 폴더 oss-github 하부에 설치
 - oss-github/git 폴더



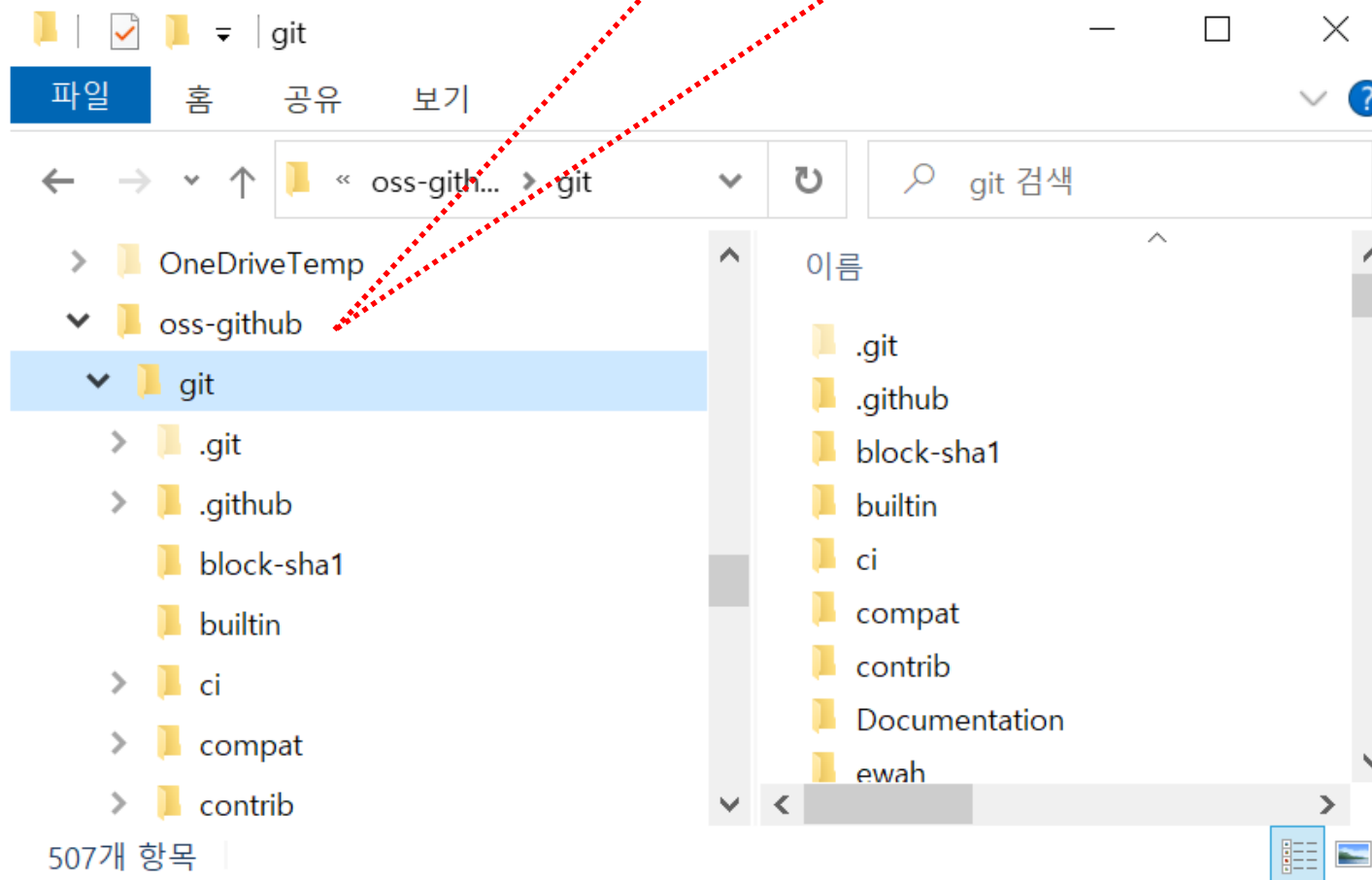
복제된 지역 저장소를 vs code로 보는 화면

깃과 깃허브 Python language



- 탐색기

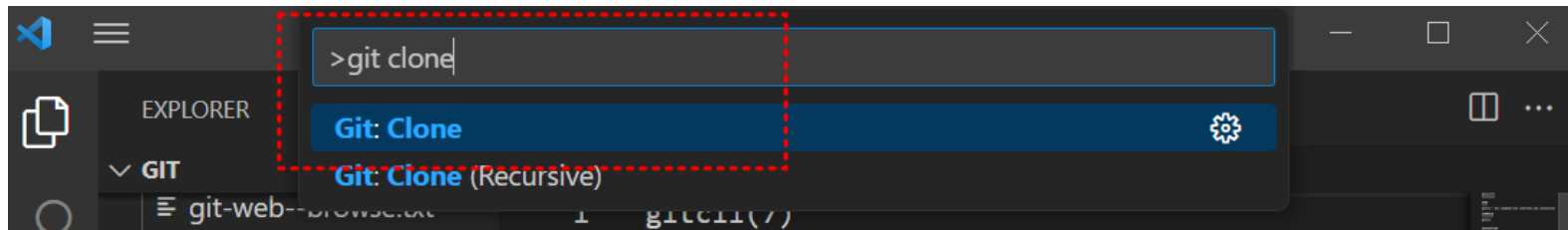
지정한 폴더가 저장소 폴더의 상위 폴더



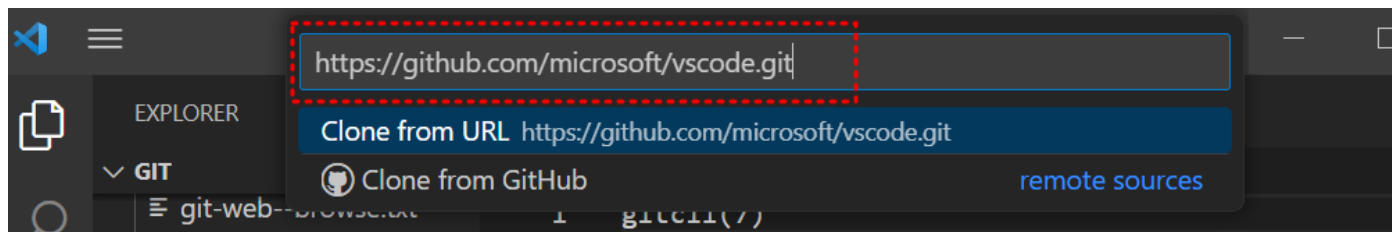
방법 2: 기존의 윈도우에서

깃과 깃허브 Python language

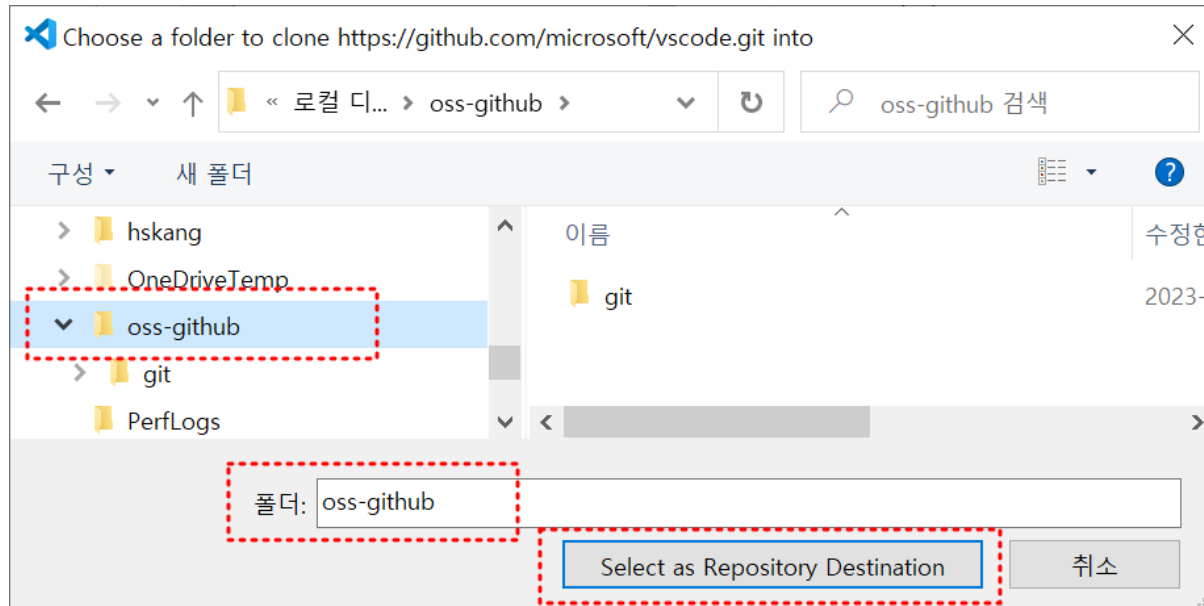
- 팔레트 열어서 하는 방법
 - Ctrl + Shit + P
 - 입력 git clone 후 선택



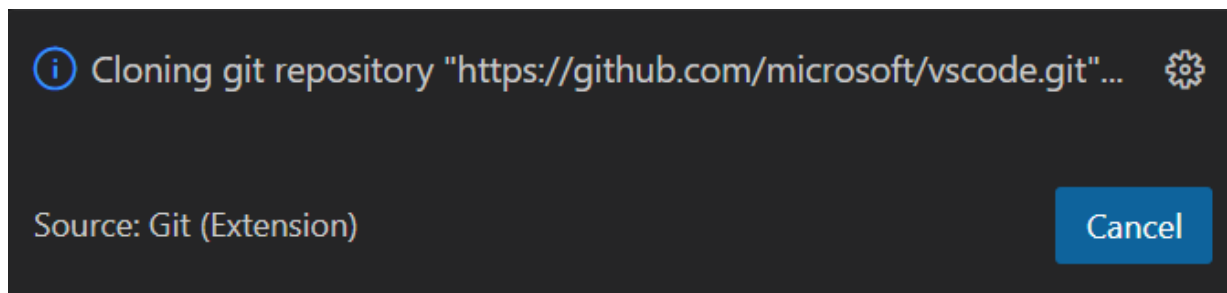
- 주소 붙이기, 후 선택



- 상위 폴더

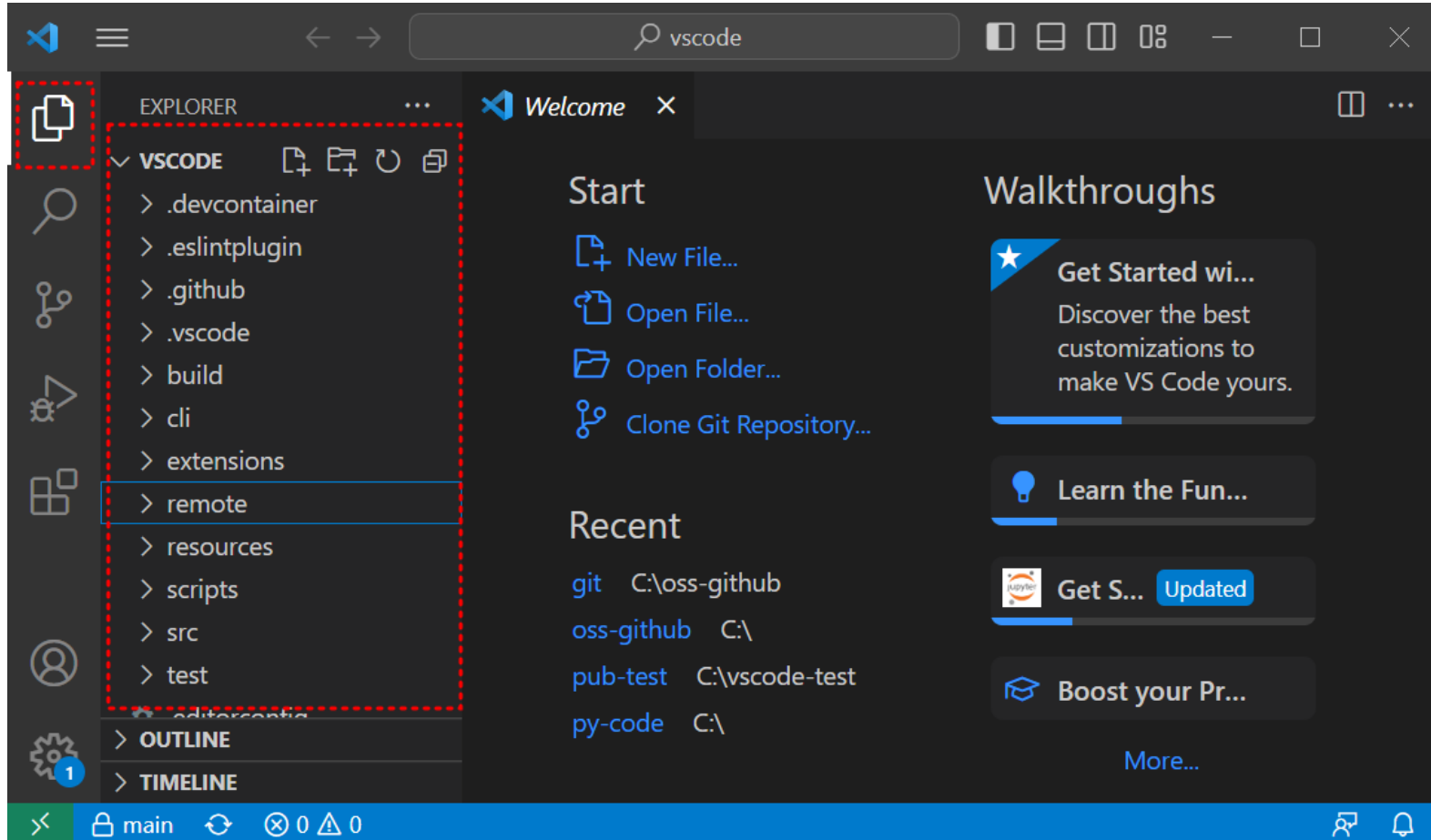


- 화면 우측 하단에 복제 중 표시

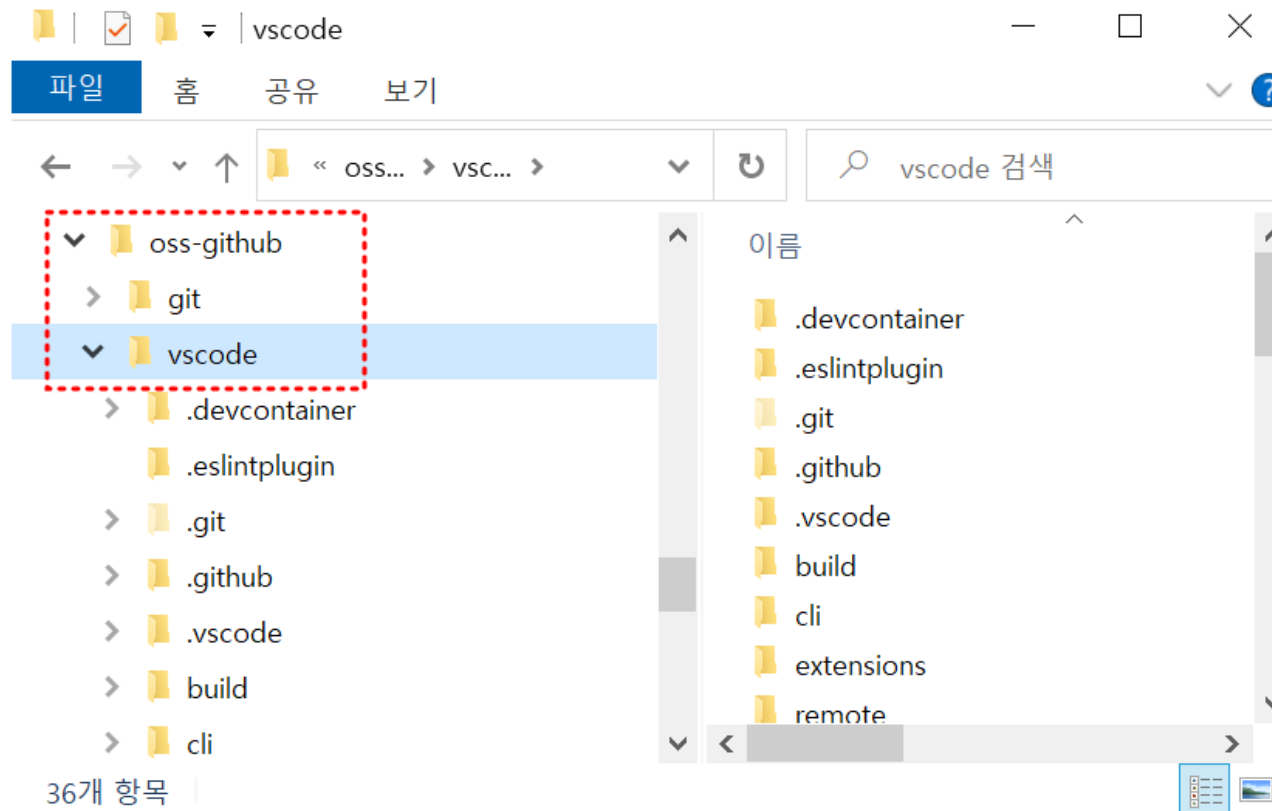


복제된 지역 저장소를 vs code로 보는 화면

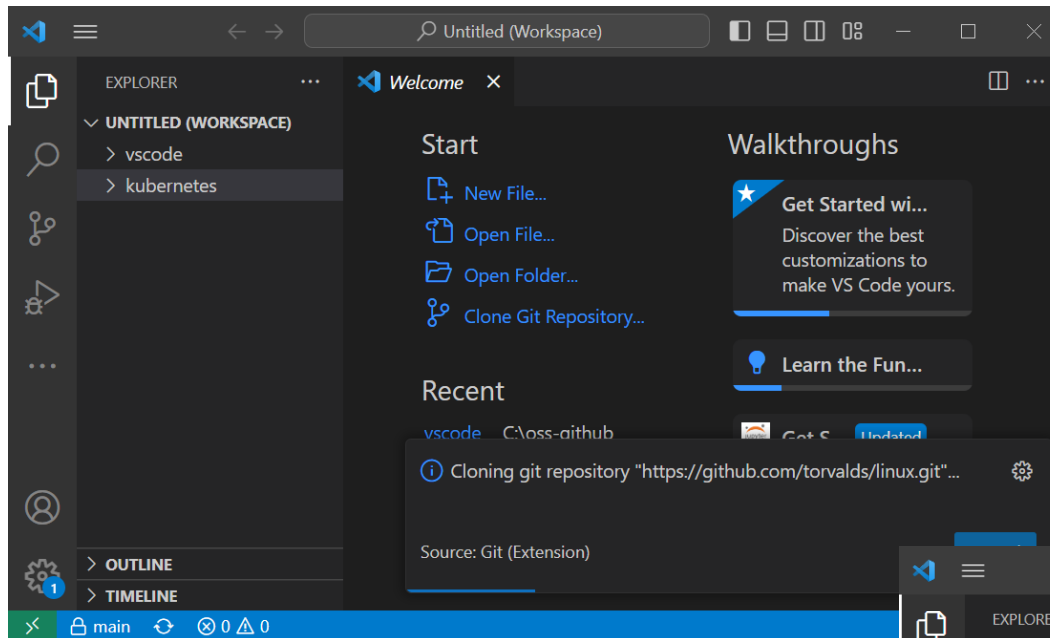
깃과 깃허브 Python language



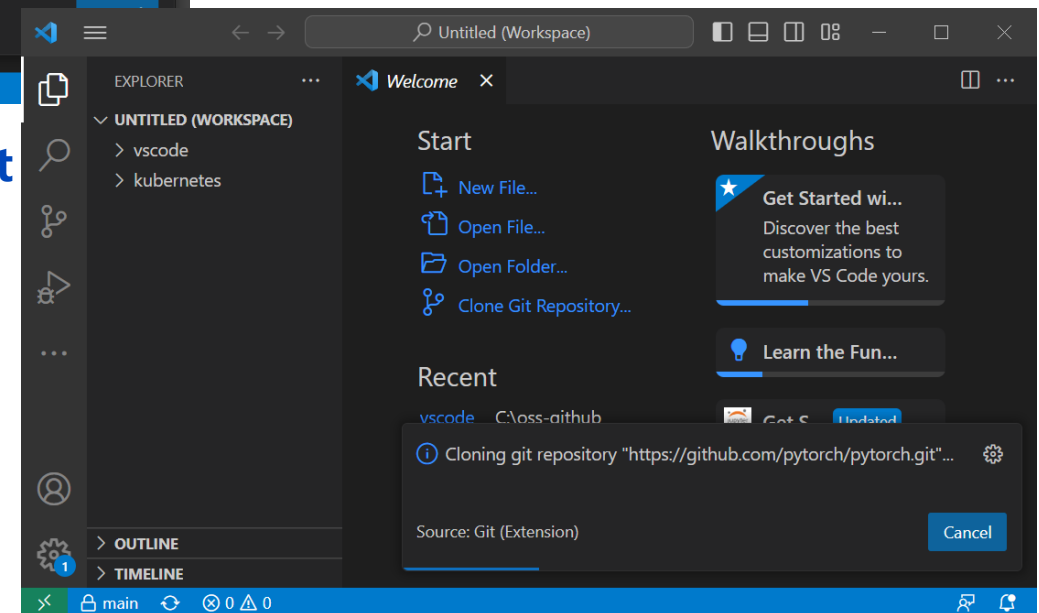
- 탐색기



- <https://github.com/torvalds/linux.git>



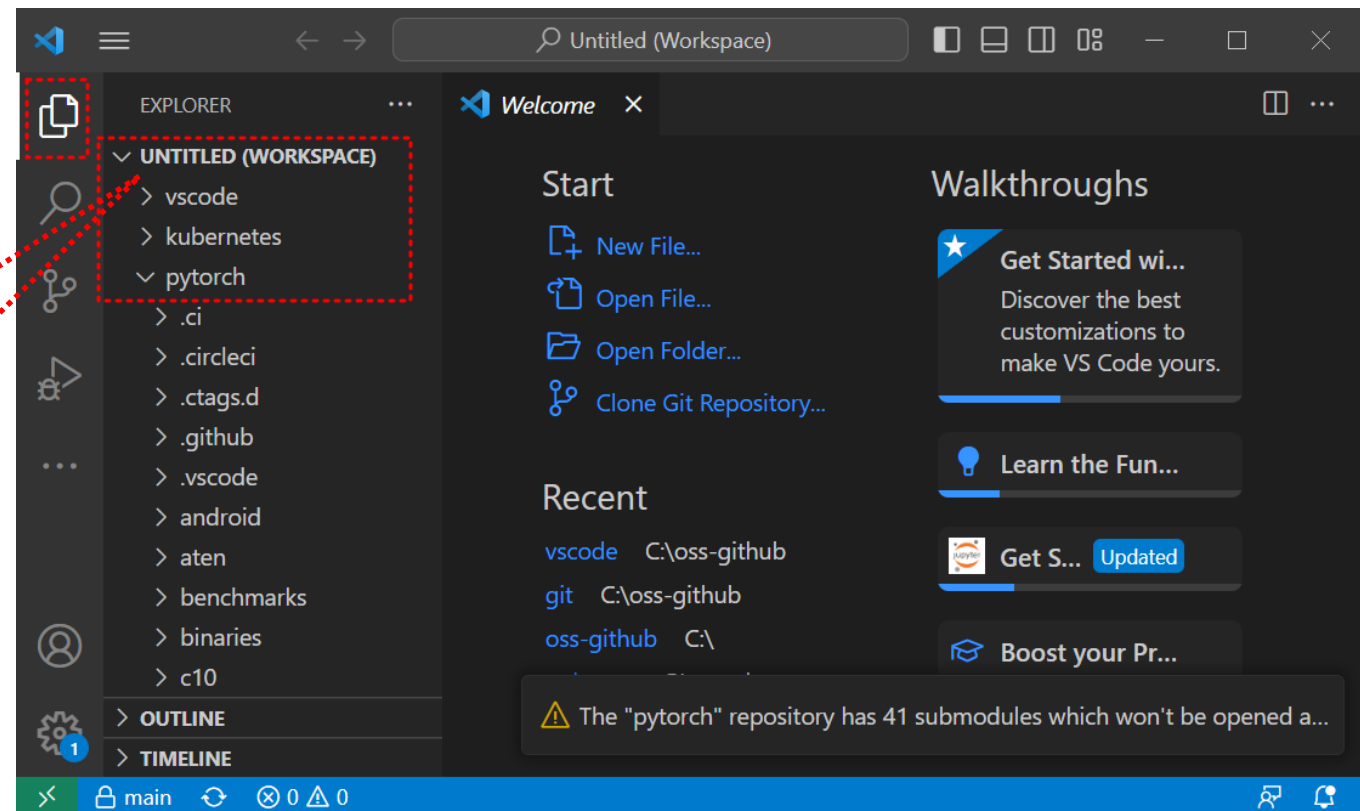
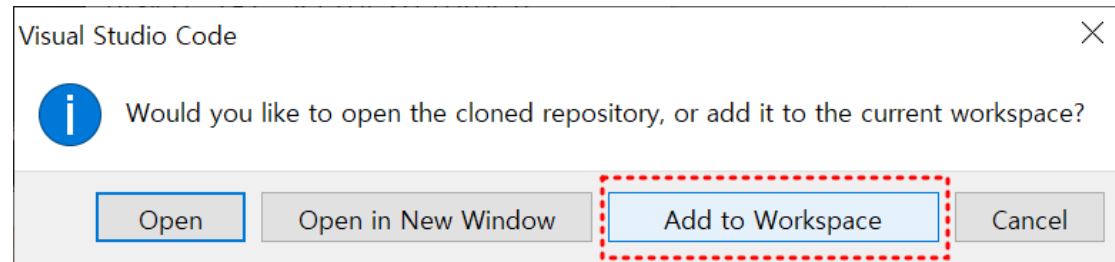
- <https://github.com/pytorch/pytorch.git>



기존의 vs code 윈도우에 workspace 하부에 생성

깃과 깃허브 Python language

- 선택 Add to Workspace
 - 가상의 작업공간 하부
 - 복제된 폴더를 생성



UNTITLED [WORKSPACE]는
다른 저장소를 하위로 갖는
가상의 작업 공간

