Git 및 VS Code 설치 및 설정 가이드

git 강의 준비사항

1. 로컬 노트북에 VS Code 설치하기

1.1. Windows에서 설치하기

1. 공식 사이트 접속: code.visualstudio.com에 접속한다
2. 설치 파일 다운로드: 메인 화면의 'Download for Windows' 버튼을 클릭하여 .exe 파일을 받는다
3. 설치 관리자 실행: 다운로드된 파일을 실행하고 다음 단계를 진행한다
4. 추가 작업 선택 (중요!): 설치 중간에 '추가 작업 선택' 화면이 나오면 아래 항목들을 체크하는 것이 매우 편리하다
   - 'Code(으)로 열기' 액션을 Windows 탐색기 파일 상황에 맞는 메뉴에 추가
   - 'Code(으)로 열기' 액션을 Windows 탐색기 디렉터리 상황에 맞는 메뉴에 추가
   - PATH에 추가 (다시 시작한 후 사용 가능) - 이걸 체크해야 터미널에서 code라고 쳐서 바로 실행할 수 있다.
5. 완료: '설치'를 누르고 기다리면 끝난다.

1.2. Mac(macOS)에서 설치하기

1. 공식 사이트 접속: code.visualstudio.com에서 **'Download for Mac'**을 클릭한다.
2. 압축 해제: 다운로드된 .zip 파일의 압축을 풀면 'Visual Studio Code' 앱 파일이 나온다.
3. 응용 프로그램 폴더로 이동: 이 앱 파일을 Finder의 응용 프로그램(Applications) 폴더로 드래그해서 옮긴다. (이 과정을 건너뛰면 업데이트가 안 될 수 있다)
4. 터미널 명령어 등록 (선택사항): * VS Code를 실행한다.
   - Cmd + Shift + P를 눌러 명령 팔레트를 엽니다.
   - shell command를 입력한 뒤, **'Shell Command: Install 'code' command in PATH'**를 선택하여 설치한다. 이제 터미널에서 `code .`만 치면 현재 폴더가 바로 열린다.

---

2. 로컬 노트북에 git 설치하기

2.1. Windows에서 Git 설치하기

- Windows는 기본적으로 Git이 포함되어 있지 않기 때문에, 공식 설치 파일을 다운로드해야 한다.

1. 공식 사이트 접속: Git 에 접속하여 Download for Windows를 클릭한다.
2. 설치 파일 실행: 다운로드된 .exe 파일을 실행한다
3. 설치 옵션 선택: 대부분 **'Next'**를 눌러 기본값으로 진행해도 무방하지만, 아래 두 가지는 확인하는 것이 좋다.
   - Choosing the default editor: 익숙한 에디터(VS Code 등)가 있다면 선택하고, 잘 모르면 기본값인 'Vim' 그대로 둡니다.
   - Adjusting your PATH environment: 'Git from the command line and also from 3rd-party software'를 선택해야 터미널에서 바로 사용할 수 있다.
4. 설치 완료: 'Finish'를 누르면 설치가 끝난다.

2.2 Mac(macOS)에서 Git 설치하기(mac 인경우)

- Mac에는 Git이 이미 깔려 있을 확률이 높다. 터미널(Terminal.app)을 열고 다음 명령어를 입력한다

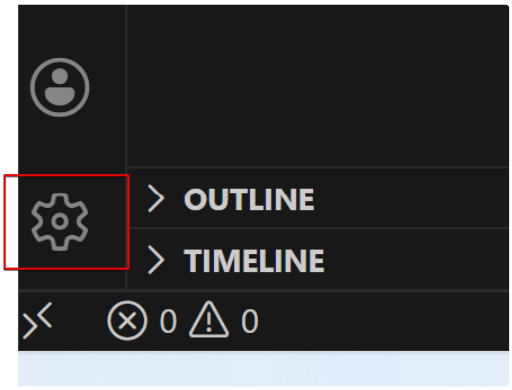
```
git --version

```

```
## 설치되어 있지 않다면 아래 명령어로 설치한다
brew install git

```

2.3 git 설정 및 VScode git bash 터미널을 디폴트로 설정하기

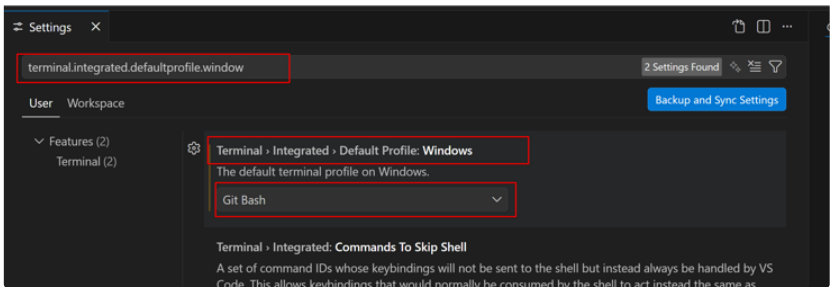
```
## [사용자메일, 사용자명] 부분을 변경하고 실행 한다
git config --global user.email "사용자메일"
git config --global user.name "사용자명"

```

1. VScode 화면의 좌측 아래 설정 버튼을 클릭하여 Settins 메뉴를 클릭한다

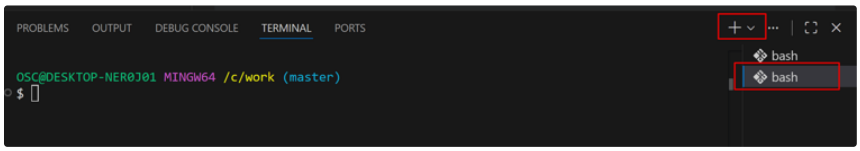


2. 상단의 검색창에 `terminal.integrated.defaultprofile.windows` 를 입력한다

- Defalut Profile Windows 셀을 그림과 같이 null --> Git Bash로 변경한다



1. VScode 터미널 창에서 + 클릭하면 Git Bash로 전환 된다



---

3. Git 교육 사전 준비 가이드

성공적인 교육 진행을 위해 아래 3가지 사항을 교육 전까지 반드시 완료할 것.

3.1. GitHub 계정 생성

Git으로 협업하기 위한 온라인 저장소 서비스인 GitHub 계정이 필요하다.

- **사이트:** github.com 접속
- **방법:** 'Sign up' 버튼을 눌러 가입 진행
- **주의:** 가입 후 입력한 이메일로 발송된 **인증 메일(Verify email)**을 반드시 확인해야 정상적으로 기능을 사용할 수 있다.

3.2. GitHub 저장소(Repository) 생성

교육에서 실습용으로 사용할 프로젝트 공간을 온라인에 직접 만든다.

1. GitHub 로그인 후 우측 상단 **[+]** 아이콘 클릭 → **[New repository]** 선택
2. **Repository name:** `my-test-lab` (반드시 이 이름으로 설정)
3. **Configuration 항목 > Choose visibility:** 'Public' 선택확인 (교육용이므로 공개 권장 default는 public)
4. **Configuration 항목 > Add README :** 항목에 On 으로 선택 (체크해야 초기 복제가 간편함)
5. 맨 아래 **[Create repository]** 클릭하여 생성 완료

3.3. 로컬 PC에 프로젝트 복제(Clone)

온라인에 만든 저장소를 내 노트북으로 내려받는 과정이다.

1. 생성된 `my-test-lab` 저장소 메인 화면에서 초록색 **[<> Code]** 버튼 클릭
2. **HTTPS** 탭에 있는 URL 주소를 복사 (예: `https://github.com/아이디/my-test-lab`)
3. 내 노트북에서 터미널(Mac) 또는 Git Bash(Windows) 실행
4. 교육을 위한 work 폴더 생성 (d: work 생성)
5. 프로젝트를 저장할 폴더(d: work)로 이동한 뒤, 아래 명령어를 입력

```
git clone [복사한 주소]

```

3.4 Git 기본 명령어 미리 학습하기

교육의 원활한 진행을 위해 아래 주요 명령어의 용도를 미리 읽고 올 것.(명령어 실행 실습도 해보기)

3.4.1 기초 명령어

- git init: 새로운 로컬 저장소를 생성한다.
- git status: 현재 작업 디렉터리의 상태(변경사항 등)를 확인한다.
- git add: 변경된 파일을 스테이징 영역(Staging Area)에 올린다.
- git commit -m "메시지": 스테이징 영역에 있는 파일들을 기록(저장)한다.
- git push: 로컬에서 커밋한 내용을 원격 저장소(GitHub)에 업로드한다.
- git pull: 원격 저장소의 최신 변경 내용을 로컬로 가져와 합친다.
- git log: 지금까지 만든 커밋 기록을 확인한다.

3.4.2 브랜치(Branch) 관련 명령어

- git branch: 현재 브랜치 목록을 확인한다.
- git branch [브랜치명]: 새로운 브랜치를 생성한다.
- git switch [브랜치명]: 해당 브랜치로 이동한다. (과거 명령어 git checkout과 동일)
- git switch -c [브랜치명]: 브랜치를 생성함과 동시에 해당 브랜치로 즉시 이동한다.
- git merge [브랜치명]: 현재 브랜치에 다른 브랜치의 변경 사항을 병합한다.

4. git push 하기

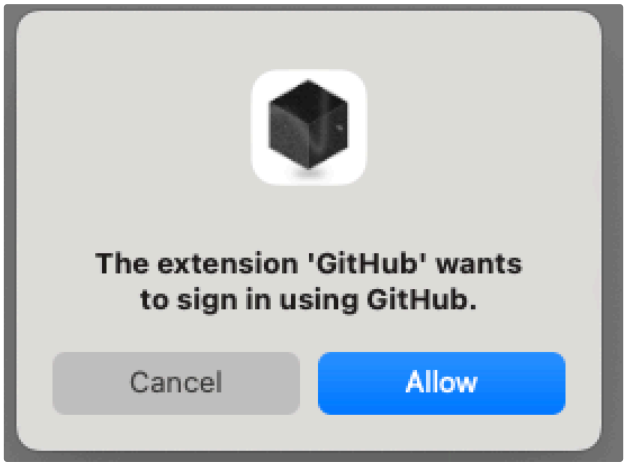
- my-test-lab 폴더를 vscode 로 열기
- Home - Chess online - Dealsbe 클릭하여 수정한다 : `# my-test-lab 입니다`

```
## 수정한 터미널을 열고
cd d:/work/my-test-lab
git add . ; git commit -m 'README.md 수정';
git push -u origin main
```

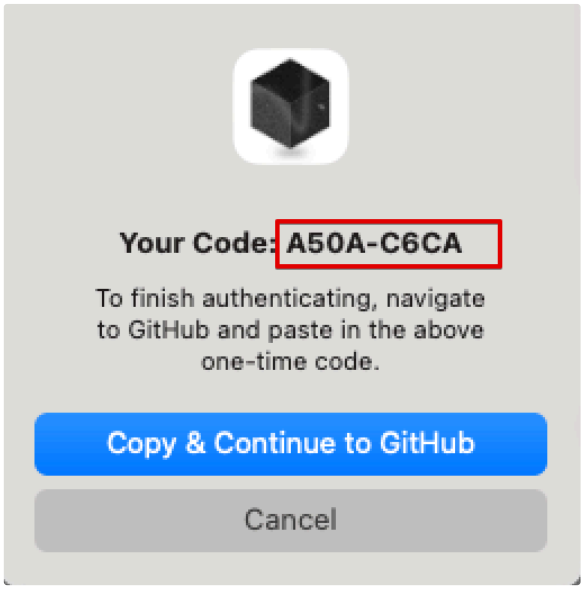
```
5
```

- repository가 public으로 설정 했지만 push는 repo 권한자만 가능하다
- 따라서 아래 사항을 한번만 해놓으면 다음 부터는 같은 작업을 할 필요가 없다
- 처음 push를 하게 되면 vscode에서 아래와 같이 extention 뜬다

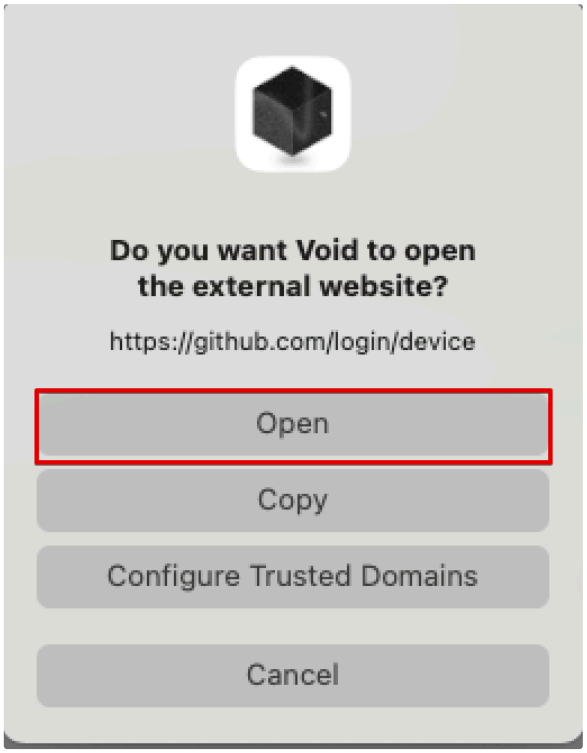
1. Allow 선택 한다



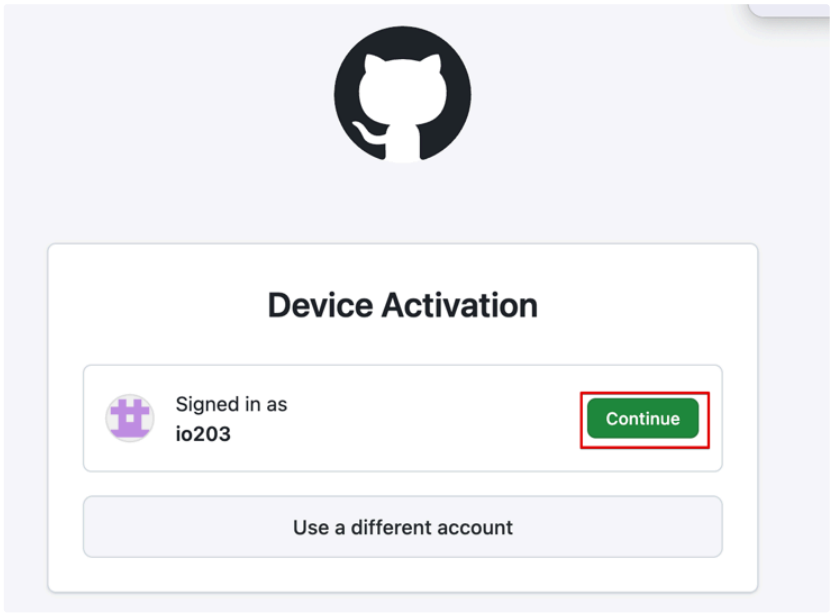
2. Copy & Continue to Gihub 버튼을 클릭하면 빨간라인의 Your Code 값이 복사 되는데 따로 파일에 붙현 넣기하여 저장해 둔다



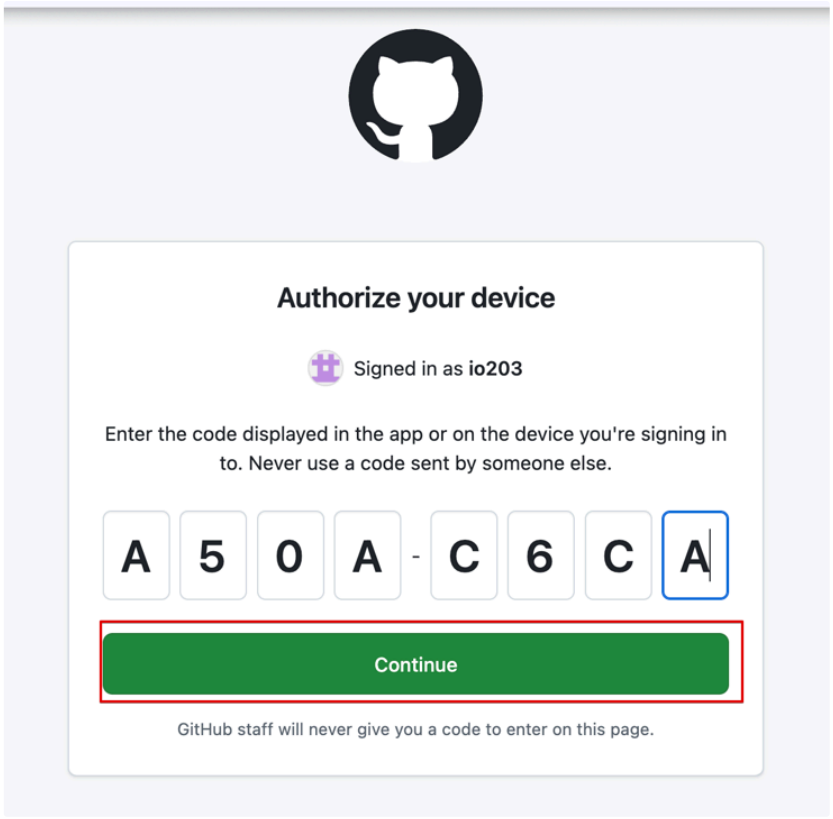
3. Open을 클릭한다



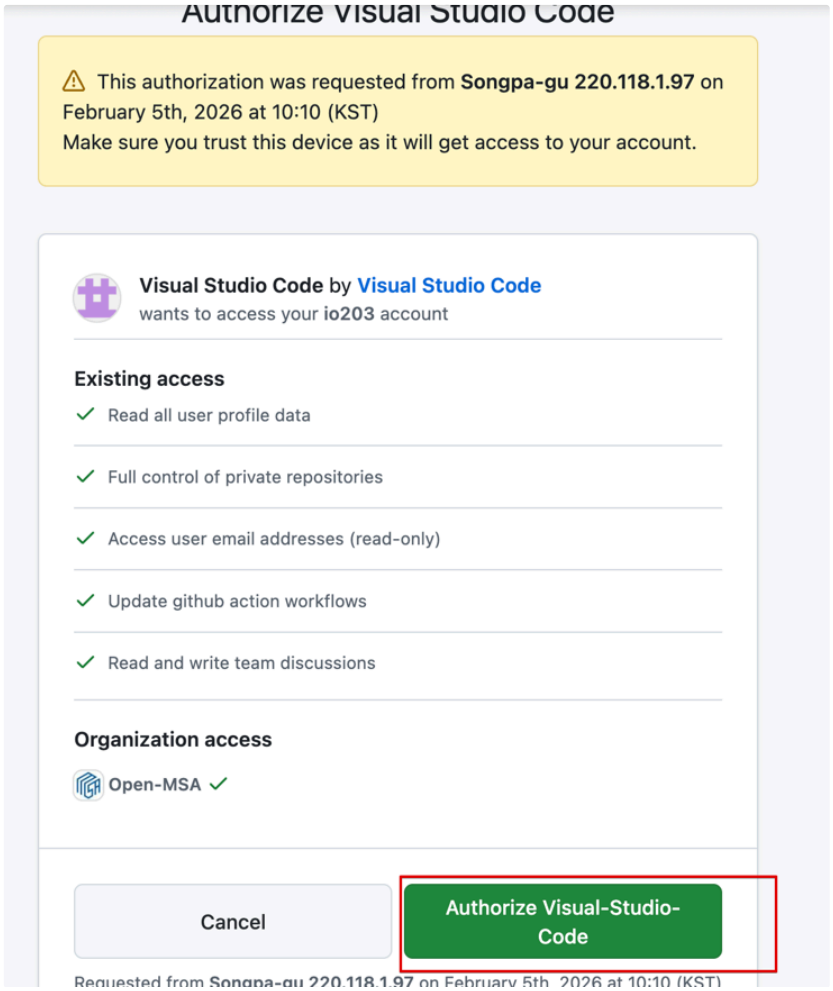
4. Continue 버튼을 클릭한다



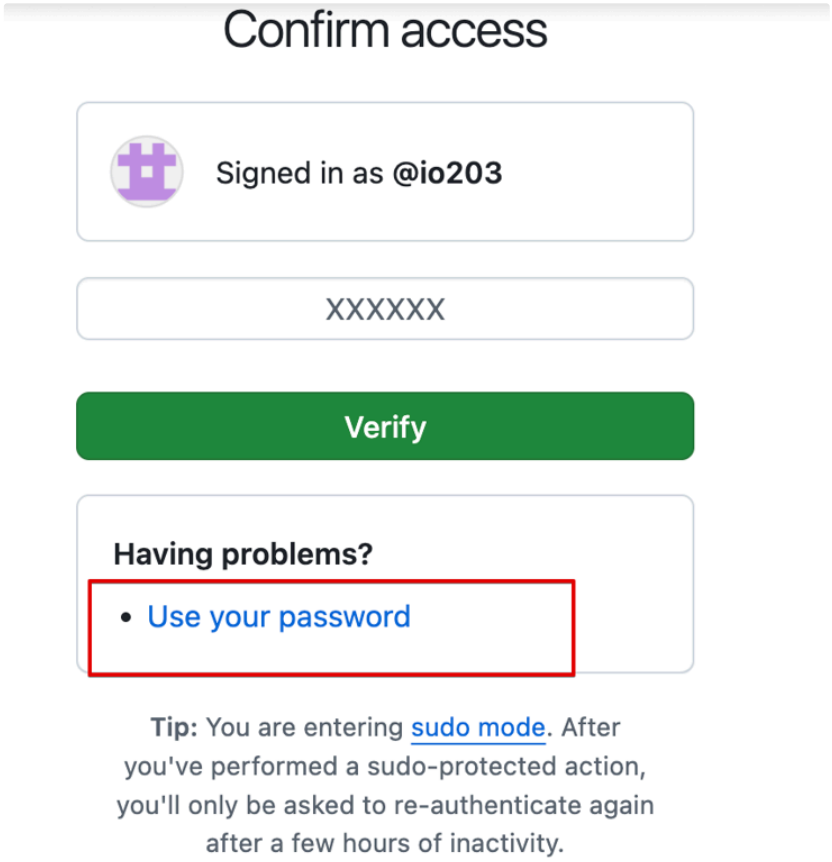
5. 복사해 놓은 코드를 입력하고 아래 Continue 버튼을 클릭한다



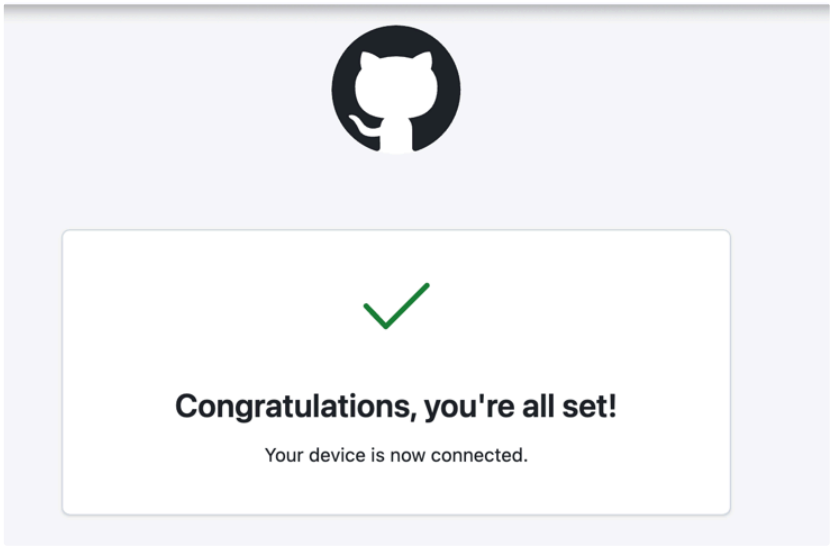
6. VScode에 대한 인증을 한다(Authorize Visual-studio-code)



7. 2차 인증앱을 요구하는데 (보통 핸드폰에 Autht앱을 설치한다) 앱을 설치 하지 않고 Use your password를 클릭하여 비번으로 인증도 가능하다 클릭하여 비번을 입력한다

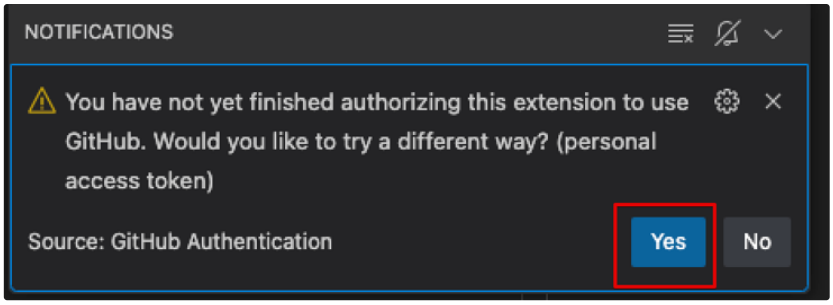


8. 성공하면 아래와 같이 설정이 완료 되었다는 화면이 뜬다

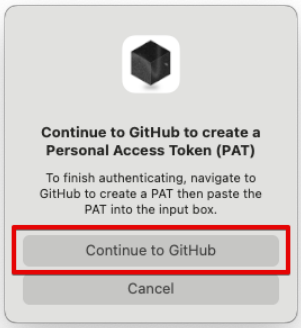


9. VScode 화면에서 오른쪽 하단에 PAT(personal access token)을 사용하는 방법을 요구한다

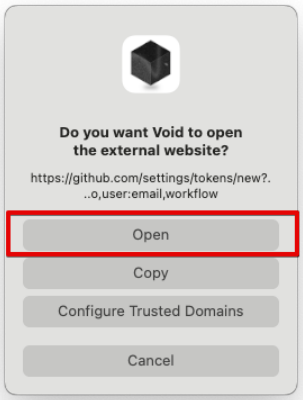
- 브라우저에서 github 접속 외에 git으로 clone push 등 git으로 연동시에는 패스워드는 사용할수 없고 PAT로 사용해야 가능하다 따라서 생성해 놓고 사용해야 한다 Yes 클릭하여 PAT를 생성한다



10. Contine to GitHub 버튼을 클릭하여 이동한다

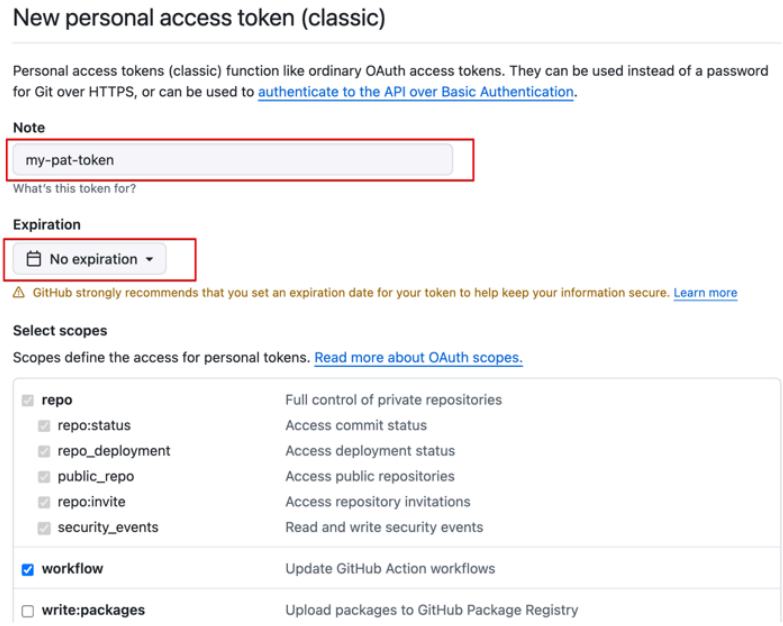


11. Open 버튼을 클릭한다

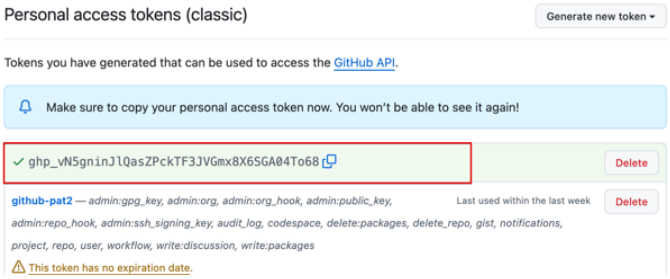


12. github web ui에서 상단 오른쪽의 `사용자이미지 클릭> settings > Deveoper settings > Personal access tokens 클릭> Tokens(classic) 이다

- note: token 이름을 입력 한다
- expire: No expiration 선택한다
- repo 체크박스는 이미 디폴트로 선택 되어 있다(비활성) 만약 선택 가능하다면 선택해야 한다
- 나머지 모두 기본 디폴트 체크된 상태로 두고 하단에 Generate token 버튼을 클릭하여 생성한다

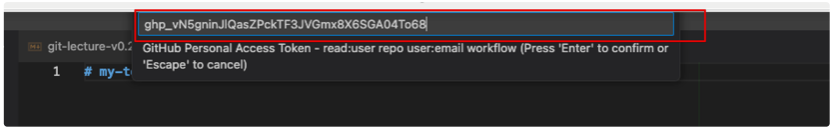


13. 생성된 토큰값은 복사하여 따로 보관 해야 한다 다시는 볼수 없다



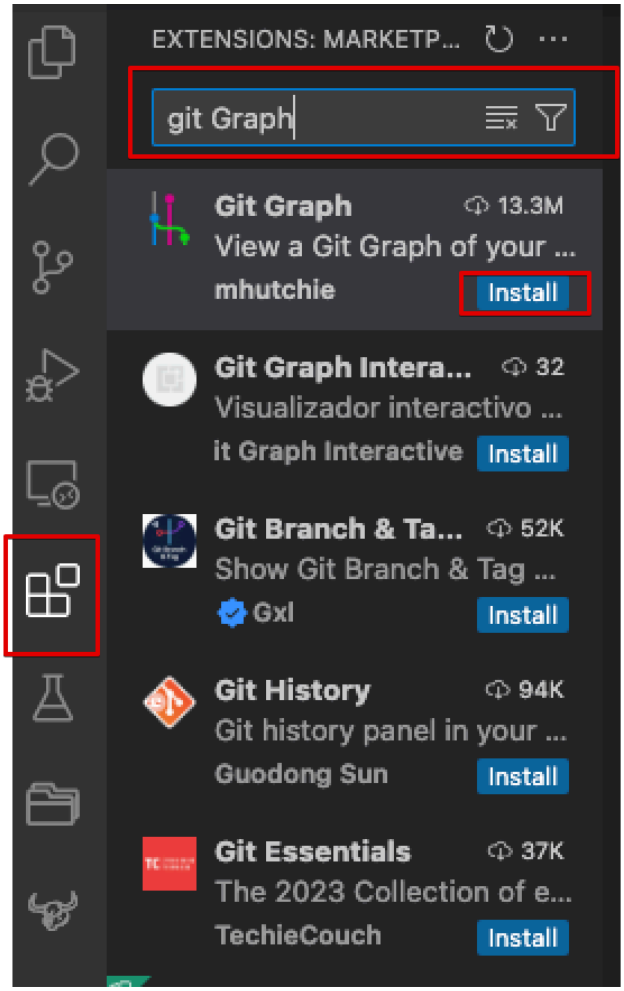
14. VScode 화면의 상단에 PAT 토큰을 입력란이 뜨면 PAT 붙혀넣기후 엔터쳐서 마무리 한다

- 정상적으로 push가 완료된다



5. Git Graph 플러그인 설치하기

- VScode 화면 > 왼쪽메뉴 > Extensions > git graph 검색 > 아래 그림처럼 install 클릭 하여 설치 한다


EXTENSIONS: MARKETP...
git Graph
Git Graph
13.3M
View a Git Graph of your ...
mhutchie
Install
Git Graph Intera...
32
Visualizador interactivo ...
it Graph Interactive
Install
Git Branch & Ta...
52K
Show Git Branch & Tag ...
Gxl
Install
Git History
94K
Git history panel in your ...
Guodong Sun
Install
Git Essentials
37K
The 2023 Collection of e...
TechieCouch
Install