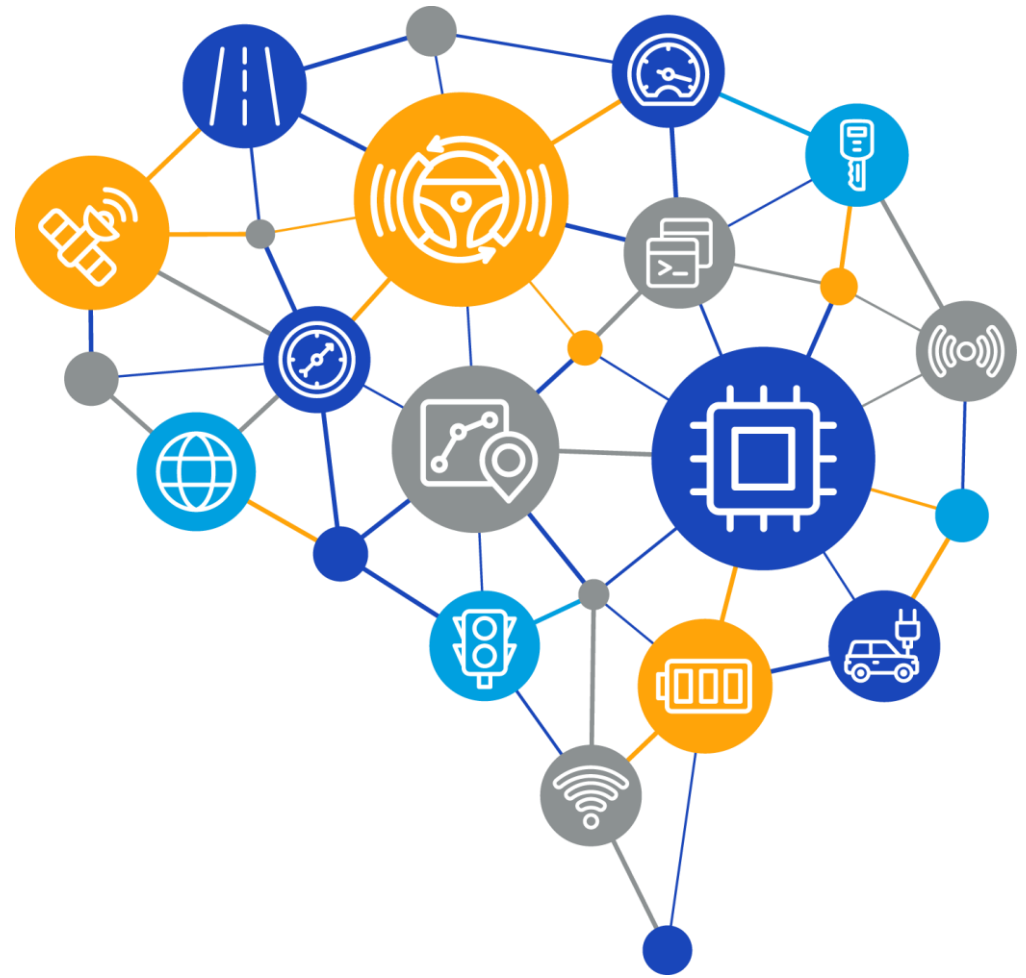


지역과 원격저장소 연동 기초 clone pull

Version Control and
Git & Github basics

2022.05

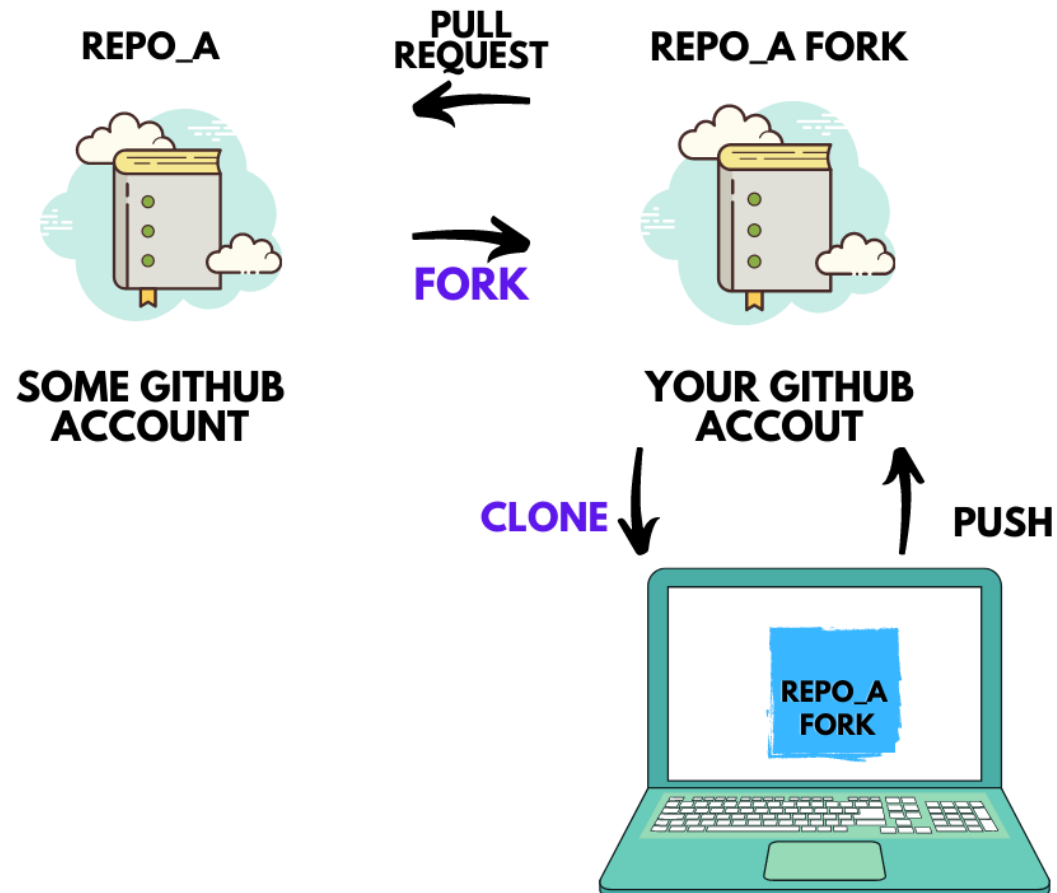
강환수 교수



AI Experts
Who Lead
The Future

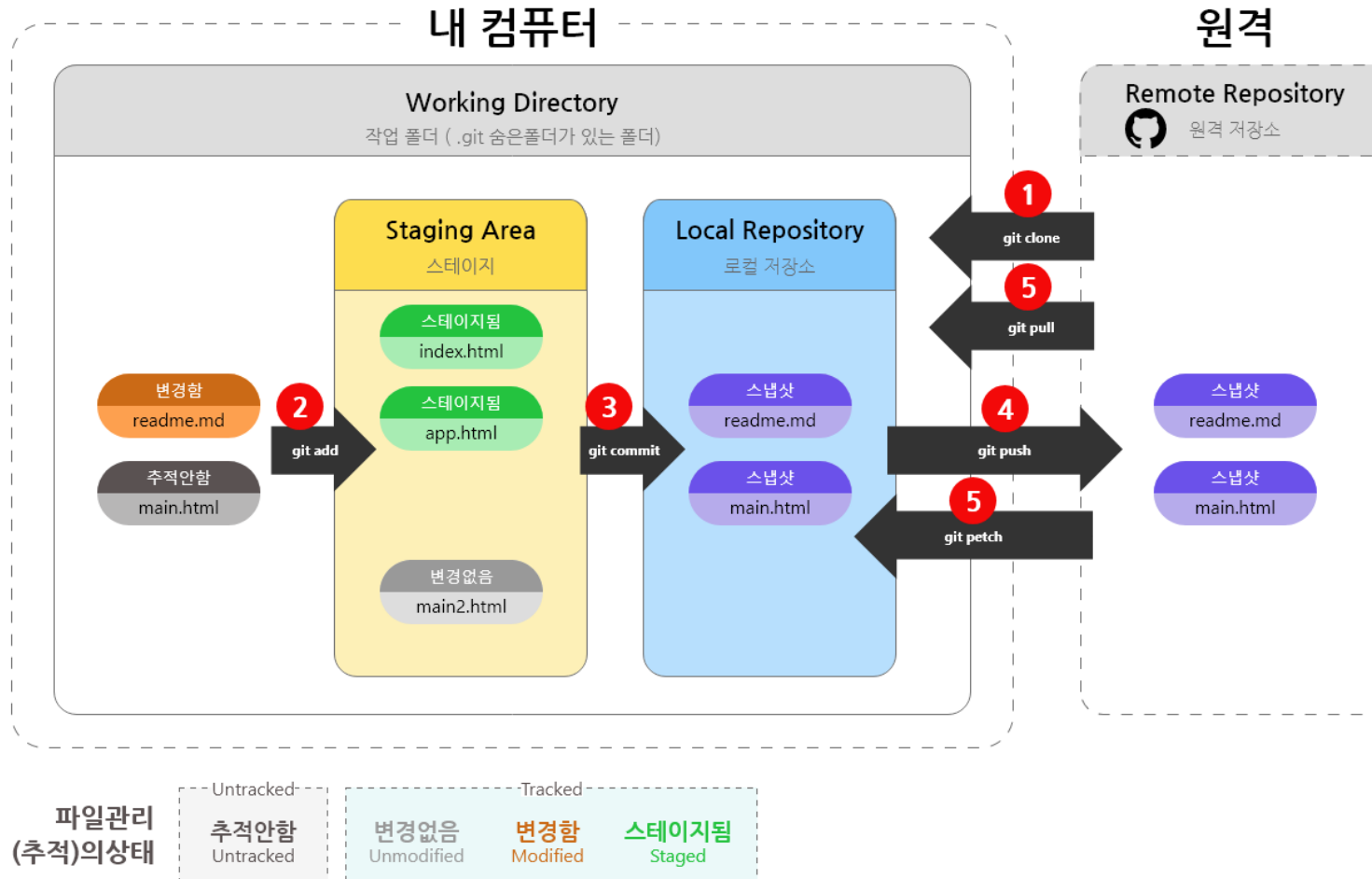
01

지역에서 깃허브 저장소 클론(clone)



 CHEDY HAMMAMI

- 원격 저장소를 지역 저장소에 복제
 - 공개된 저장소는 소유와 관계 없이 가능



- 1. 깃허브 원격저장소 생성
 - git-clone
 - https 주소 복사
- 2. PC 깃에서 원격저장소 복제(git clone)
 - \$ git clone [복사된-주소]
 - 원격저장소와 동일한 이름으로 복제
 - \$ git clone [복사된-주소] [새로운-폴더명]
 - 하부 폴더 [새로운-폴더명] 이름으로 복제
- 3. 원격 저장소 별칭 이름 점검
 - \$ git remote
 - 기본 이름 origin 점검
 - \$ git remote -v

- 이름 git-clone
- Add a README file 체크


Create a new repository


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

Owner * / Repository name *

Great repository names git-clone is available. orable. Need inspiration? How about [miniature-sniffle](#)?

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.](#)

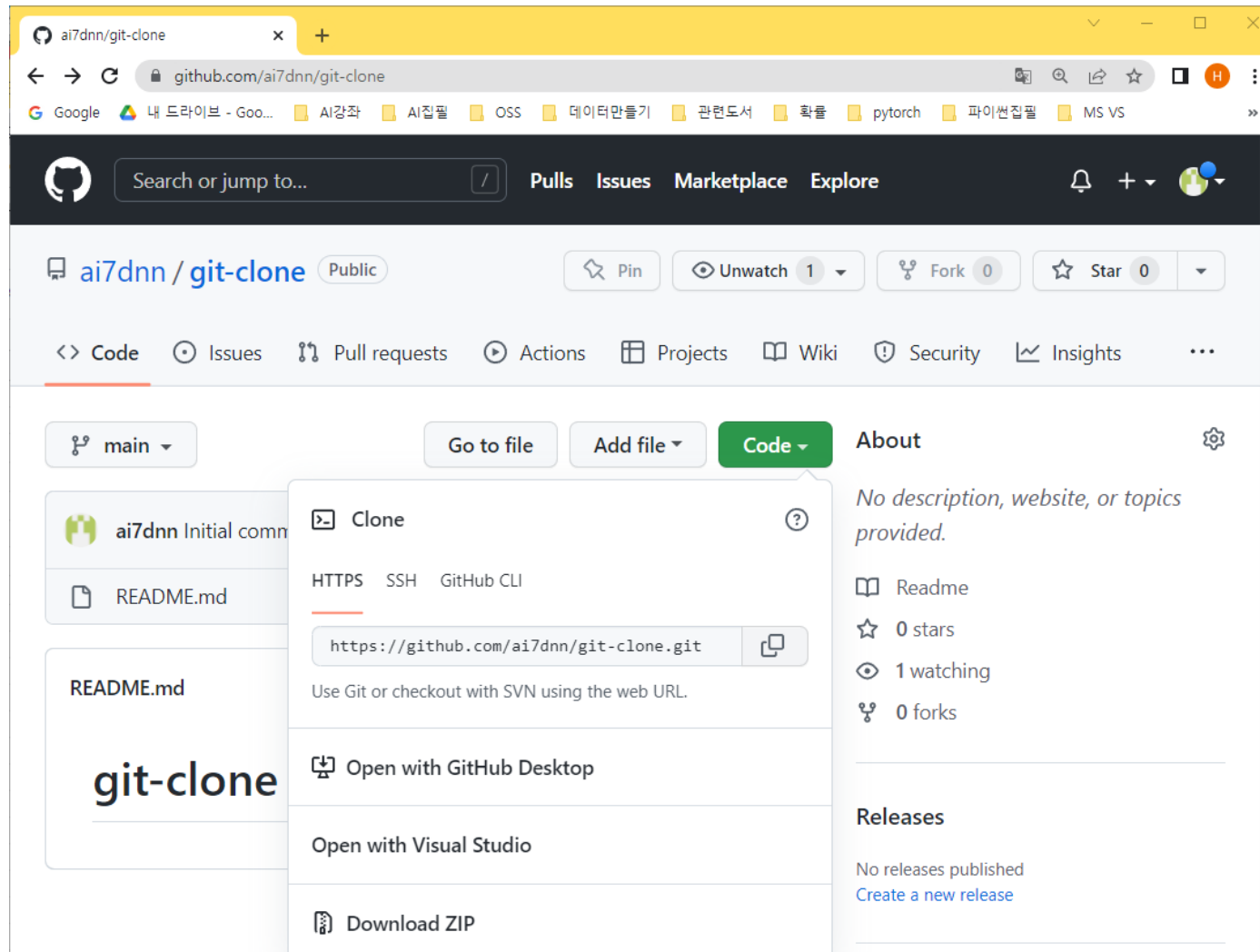
Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

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

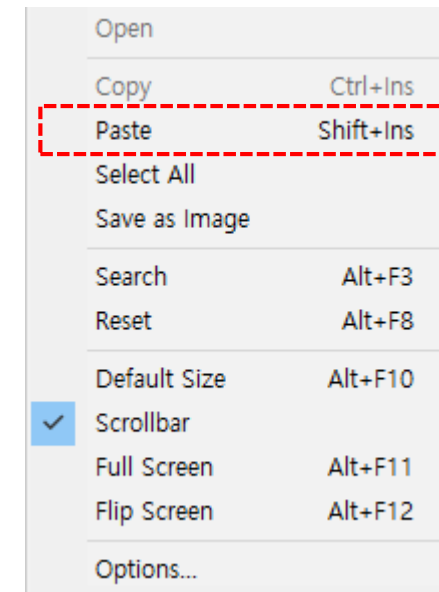
This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository



- 명령어
 - \$ `gi clone` 저장소주소 [저장소폴더명]
- 자신의 지역 저장소에 복사 명령
 - \$ `git clone https://github.com/ai7dnn/git-clone.git`
 - 오른쪽 마우스 팝업 메뉴로 붙여 넣기
- 복사된 폴더 확인



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
```

```
$ pwd  
/c/[git tutorial]
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
```

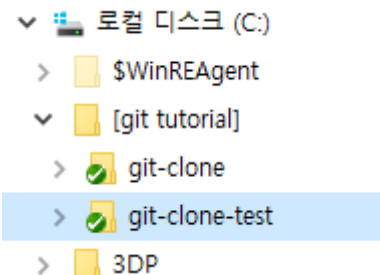
```
$ git clone https://github.com/ai7dnn/git-clone.git  
Cloning into 'git-clone'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
```

```
$
```


- `git clone https://github.com/ai7dnn/git-clone.git` 하부폴더명

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ git clone https://github.com/ai7dnn/git-clone.git git-clone-test
Cloning into 'git-clone-test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```



- **\$ git remote**
 - 원격 저장서 이름 목록만
- **\$ git remote -v**
 - 원격 저장서 주소와 이름 목록

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ pwd
/c/[git tutorial]
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]
$ cd git-clone
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
$ pwd
/c/[git tutorial]/git-clone
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
$ git remote -v
origin https://github.com/ai7dnn/git-clone.git (fetch)
origin https://github.com/ai7dnn/git-clone.git (push)
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-clone (main)
$
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git
tutorial]/git-clone (main)
$ git remote
origin
```

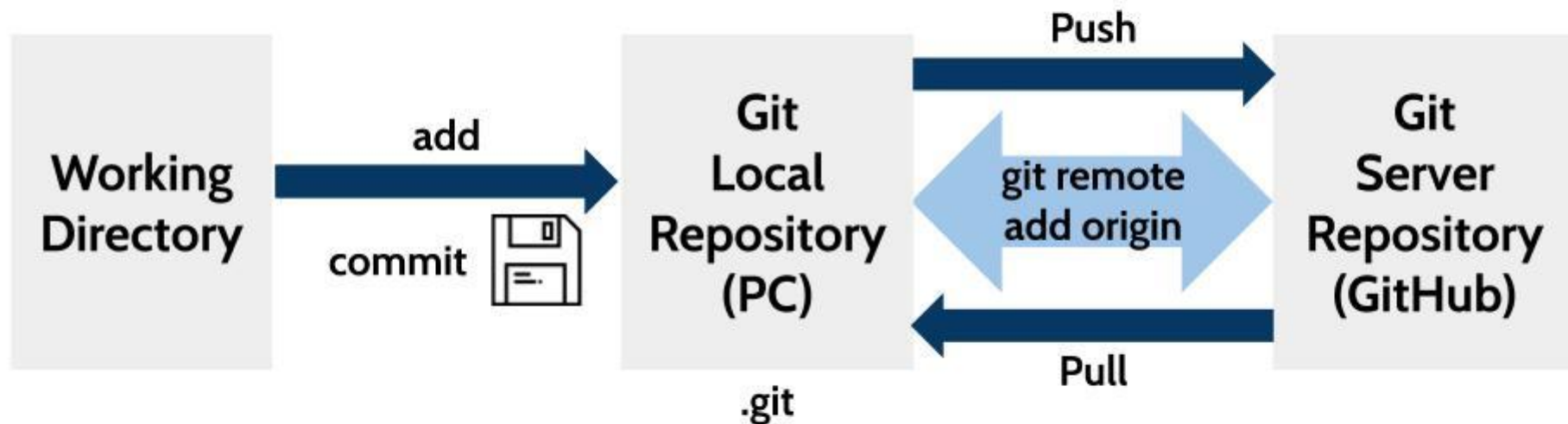
AI Experts
Who Lead
The Future

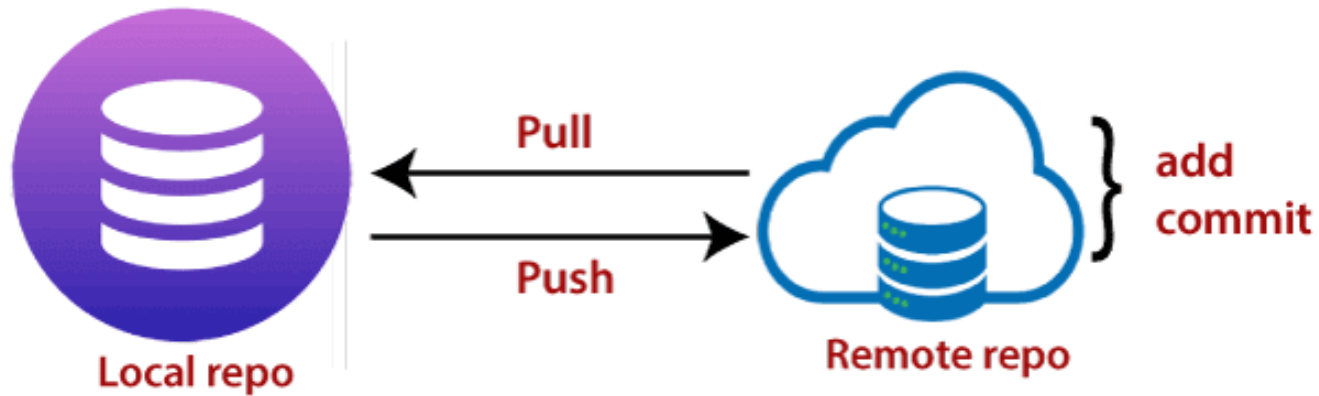
02

깃허브 저장소 수정 후 풀(pull)



git structure





- 깃허브 저장소 git-clone에서 작업
- 1. README.md 파일 수정
 - Markdown 학습
 - [Markdownguide.org](https://www.markdownguide.org) 참조
 - 이력 확인
- 2. PC에서 지역저장소와 원격저장소가 다른 것을 점검
 - \$ cat README.md
 - README.md 파일 수정 미반영
- 3. PC 지역저장소에서 원격저장소 끌어(가져)오기 pull
 - \$ git pull
 - \$ cat README.md

- 파일 README.md 수정 후 커밋

git-clone

파일 수정 1회 후 커밋, 헤드라인 마크다운

- # The largest heading
- ## The second largest heading
- ##### The smallest heading

Commit changes

Update README.md

파일 수정 1회 후 커밋

☒ 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

- 문서 형식화 언어

- 마크다운은 일반 텍스트 문서에 형식 요소를 추가하는 데 사용할 수 있는 경량 마크업 언어
- 현재 세계에서 가장 인기 있는 마크업 언어 중 하나

- 2004년 존 그루버가 개발

- <https://www.markdownguide.org/>

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	bold text

Link

[Markdown Guide](<https://www.markdownguide.org>)

Image

![alt text](<https://www.markdownguide.org/assets/images/tux.png>)


Extended Syntax

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

Table

Syntax Description
----- -----
Header Title
Paragraph Text


- 2 minutes ago | 2 클릭


 main ▾


Go to file





Add file ▾









Code ▾

 ai7dnn Update README.md ... 2 minutes ago ⌚ 2

 README.md Update README.md 2 minutes ago


 ai7dnn / git-clone Public


 Pin  Unwatch 1  Fork 0  Star 0


 Code  Issues  Pull requests  Actions  Projects  Wiki  Security  Insights ...

Update README.md

파일 수정 1회 후 커밋

 main


 ai7dnn committed 3 minutes ago Verified 1 parent a8273a1 commit b25494fd9618e446b273815a2a0ce3606462ea82

 Showing 1 changed file with 6 additions and 1 deletion. Split Unified

7

■■■■■

README.md



...	@@ -1 +1,6 @@	
1	- # git-clone	
1	+ # git-clone	
2	+	
3	+ ## 파일 수정 1회 후 커밋, 헤드라인 마크다운	
4	+ - # The largest heading	
5	+ - ## The second largest heading	
6	+ - ##### The smallest heading	

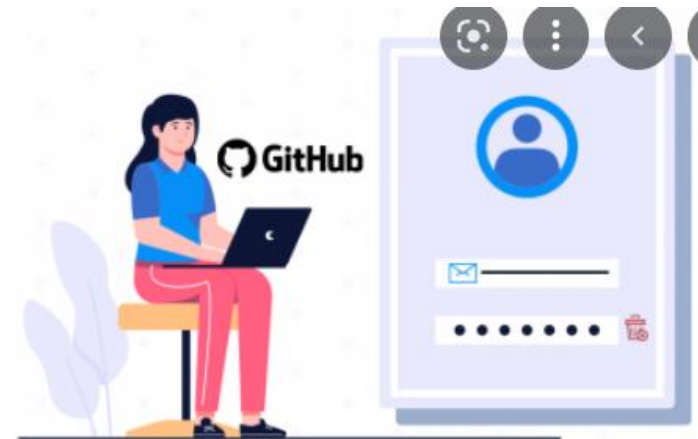
- `git pull <원격 저장소 명> <branch 명>`
- `$ git pull`
 - 현재 origin에서 기본 브랜치로 가져 오기
- 결과 확인
 - `$ cat readme.md`

AI Experts
Who Lead
The Future

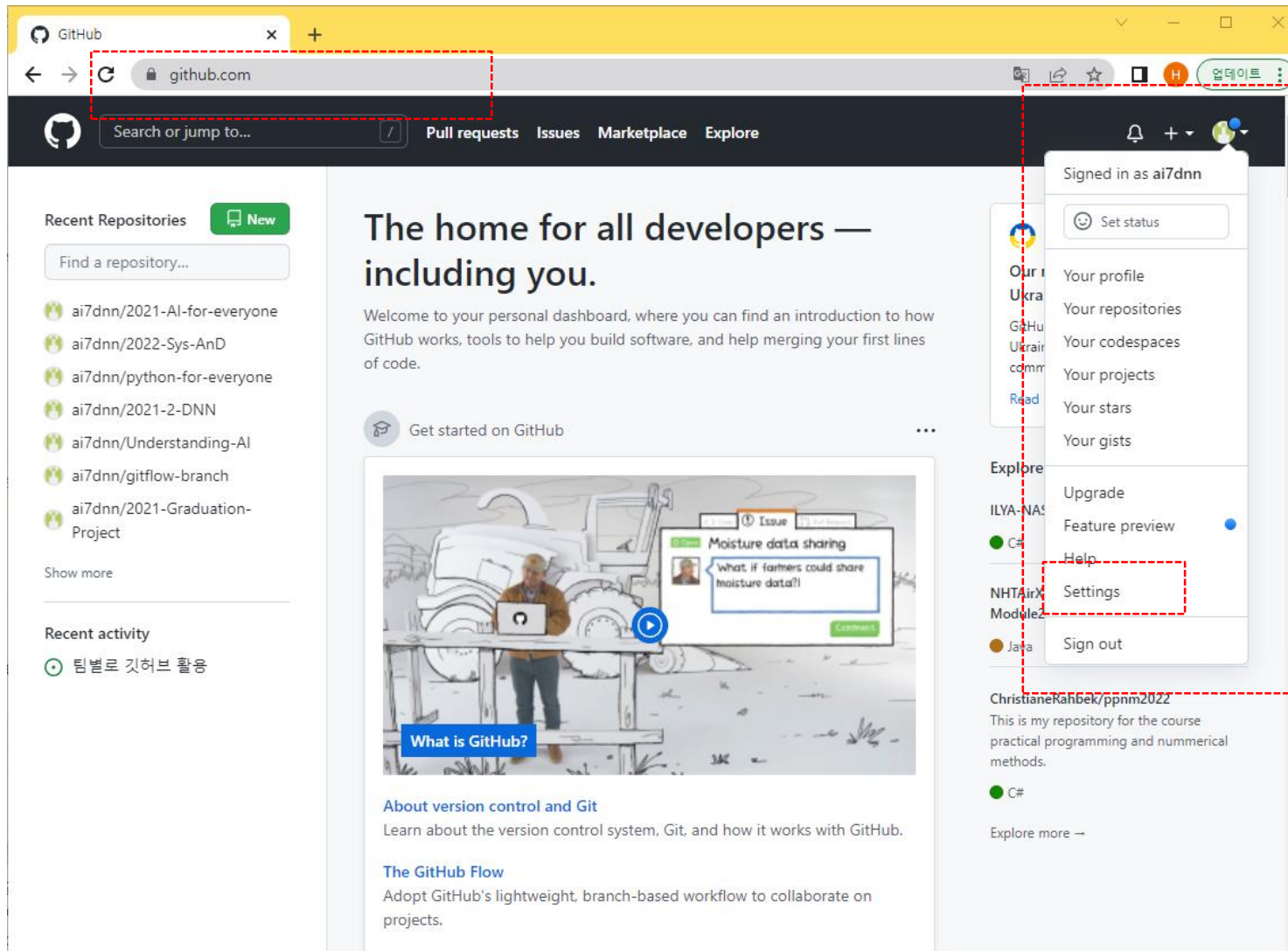
03

깃허브 PAT 생성

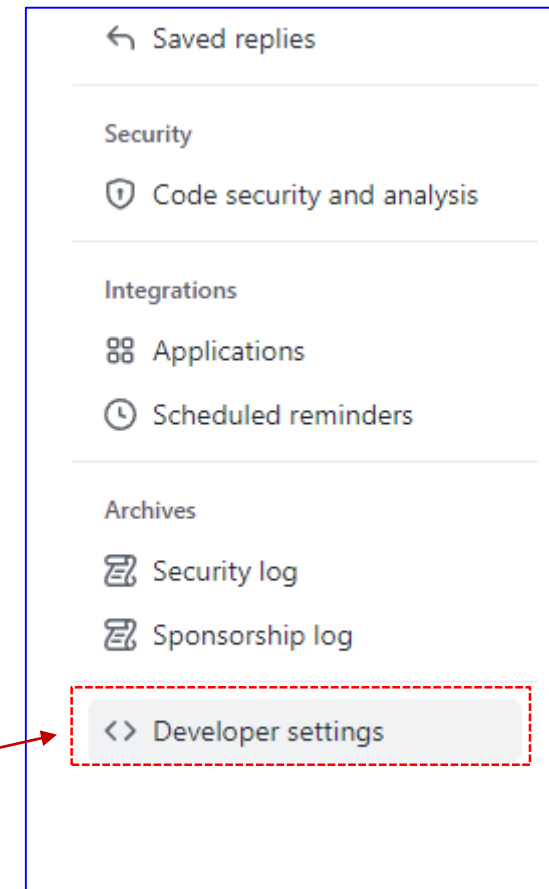
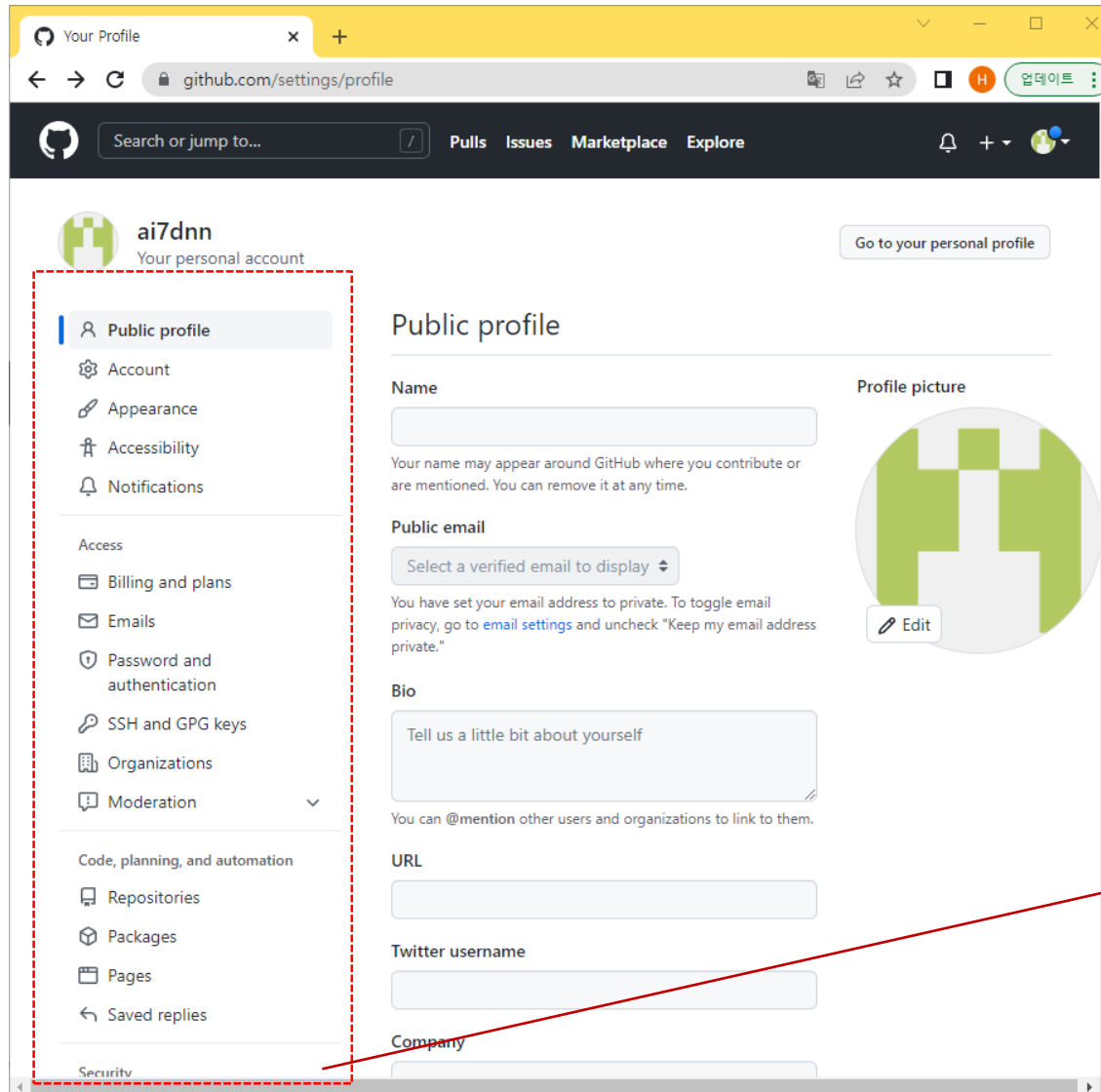
- **개인 프로젝트를 진행하던 중 아래와 같은 에러 메시지가 발생**
 - Support for password authentication was removed on August 13, 2021. Please use a personal access token instead. ... The requested URL returned error: 403
 - 이유는 2021년 8월 13일 이후부터는 기존의 비밀번호 대신 토큰을 사용해서 접근하도록 변경되었기 때문입니다. 따라서 github에서 Personal access token을 발급 받아 설정해주어야 합니다.
- **비밀번호 인증 대신 사용하는 강화된 인증 방법**
- **깃허브 계정마다 토큰 생성 필요**



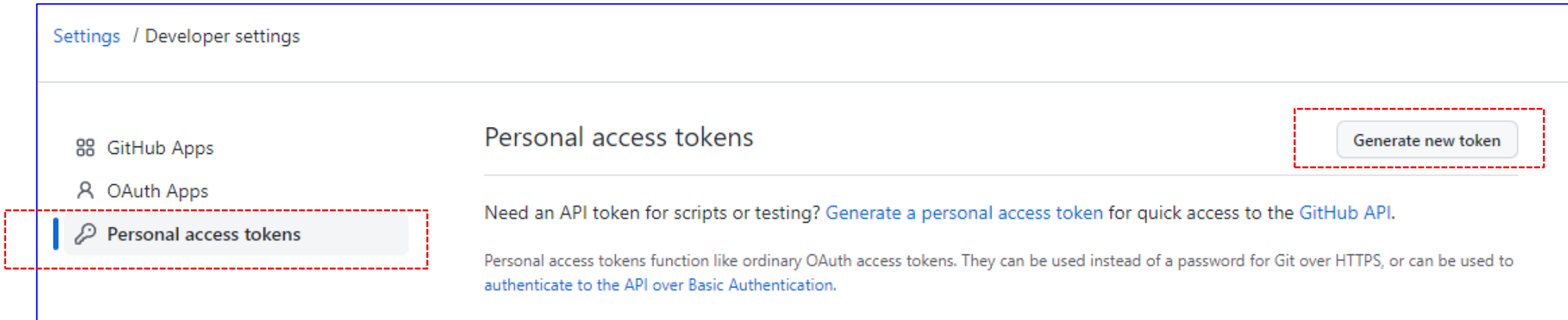
- 프로필 메뉴
 - 로그인 후 GitHub.com에서 우측 메뉴 Settings



- 왼쪽 메뉴 가장 하단: Developers settings



- Generate new token



The screenshot shows the GitHub 'Developer settings' page. On the left sidebar, 'Personal access tokens' is highlighted with a red dashed box. The main content area is titled 'Personal access tokens' and features a 'Generate new token' button in the top right corner, also highlighted with a red dashed box. Below the title, there is explanatory text about API tokens and a link to 'Generate a personal access token'.

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

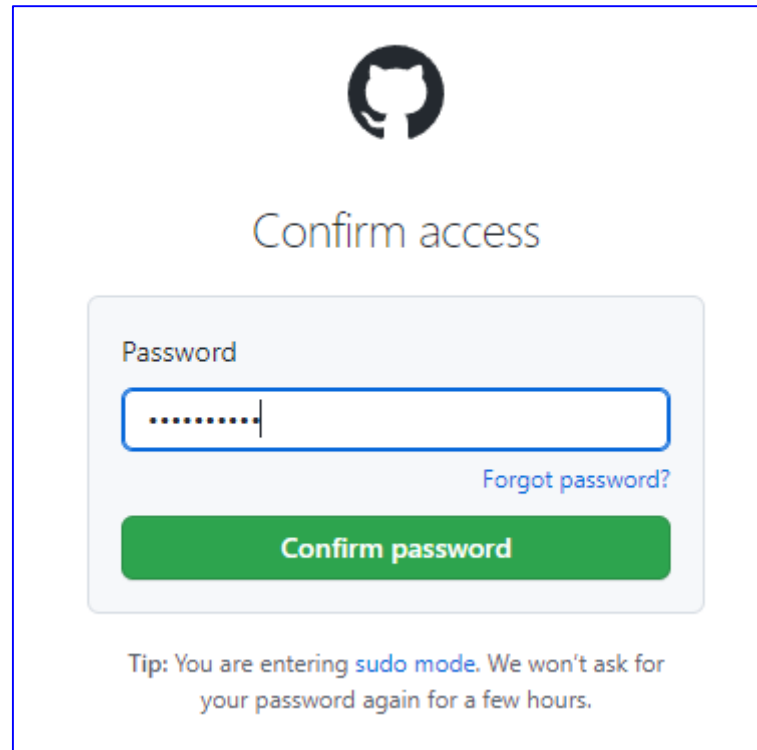
Personal access tokens

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 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](#).

- 암호 인증



The screenshot shows the GitHub 'Confirm access' screen. At the top is the GitHub logo. Below it, the text 'Confirm access' is centered. A password input field is shown with a blue border and a blue cursor. To the right of the input field is a link 'Forgot password?'. Below the input field is a green button labeled 'Confirm password'. At the bottom, a tip message states: 'Tip: You are entering sudo mode. We won't ask for your password again for a few hours.'

Confirm access

Password

.....

[Forgot password?](#)

Confirm password

Tip: You are entering [sudo mode](#). We won't ask for your password again for a few hours.

값 입력

- Generate token
- 범위 선택
 - 모두 선택
 - Publish 기능을 위해
 - 상황에 따라
 - Repo, Workflow, gist 정도 선택

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

token of repo workflow gist

What's this token for?

Expiration *

90 days The token will expire on Fri, Jul 22 2022

Select scopes

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

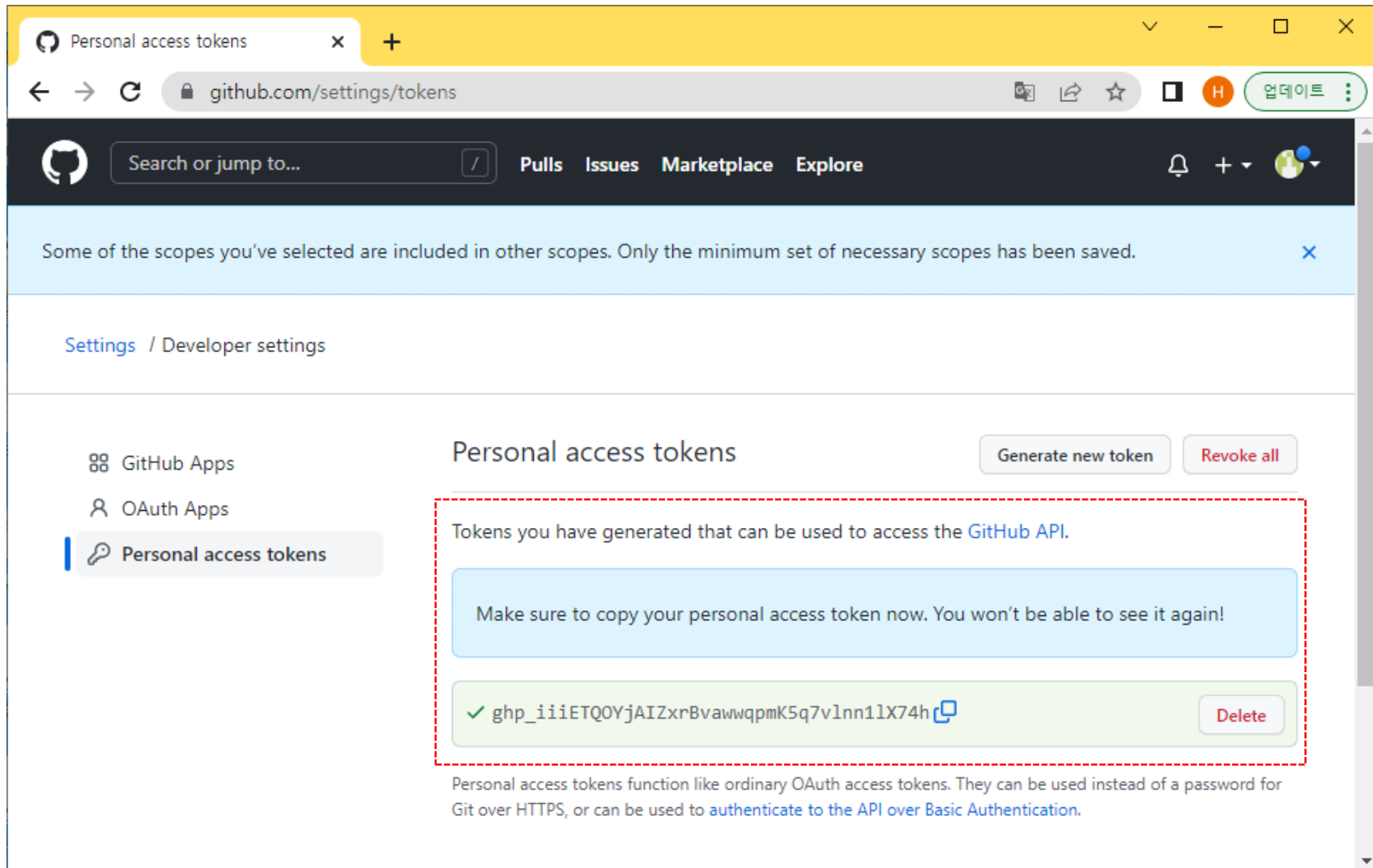
- ☒ repo Full control of private repositories
- ☐ repo:status Access commit status
- ☒ repo:deployment Access deployment status
- ☒ public_repo Access public repositories
- ☒ repo:invite Access repository invitations
- ☒ security_events Read and write security events

- ☒ workflow Update GitHub Action workflows
- ☐ write:packages Upload packages to GitHub Package Registry
 - ☐ read:packages Download packages from GitHub Package Registry
- ☐ delete:packages Delete packages from GitHub Package Registry
- ☐ admin:org Full control of orgs and teams, read and write org projects
 - ☐ write:org Read and write org and team membership, read and write org projects
 - ☐ read:org Read org and team membership, read org projects
- ☐ admin:public_key Full control of user public keys
 - ☐ write:public_key Write user public keys
 - ☐ read:public_key Read user public keys
- ☐ admin:repo_hook Full control of repository hooks
 - ☐ write:repo_hook Write repository hooks
 - ☐ read:repo_hook Read repository hooks
- ☐ admin:org_hook Full control of organization hooks
- ☒ gist Create gists
- ☐ notifications Access notifications
- ☐ user Update ALL user data
 - ☐ read:user Read ALL user profile data
 - ☐ user:email Access user email addresses (read-only)
 - ☐ user:follow Follow and unfollow users
- ☐ delete_repo Delete repositories
- ☐ write:discussion Read and write team discussions
 - ☐ read:discussion Read team discussions
- ☐ admin:enterprise Full control of enterprises
 - ☐ manage_runners:enterprise Manage enterprise runners and runner-groups
 - ☐ manage_billing:enterprise Read and write enterprise billing data
 - ☐ read:enterprise Read enterprise profile data
- ☐ admin:pgp_key Full control of public user GPG keys (Developer Preview)
 - ☐ write:pgp_key Write public user GPG keys
 - ☐ read:pgp_key Read public user GPG keys

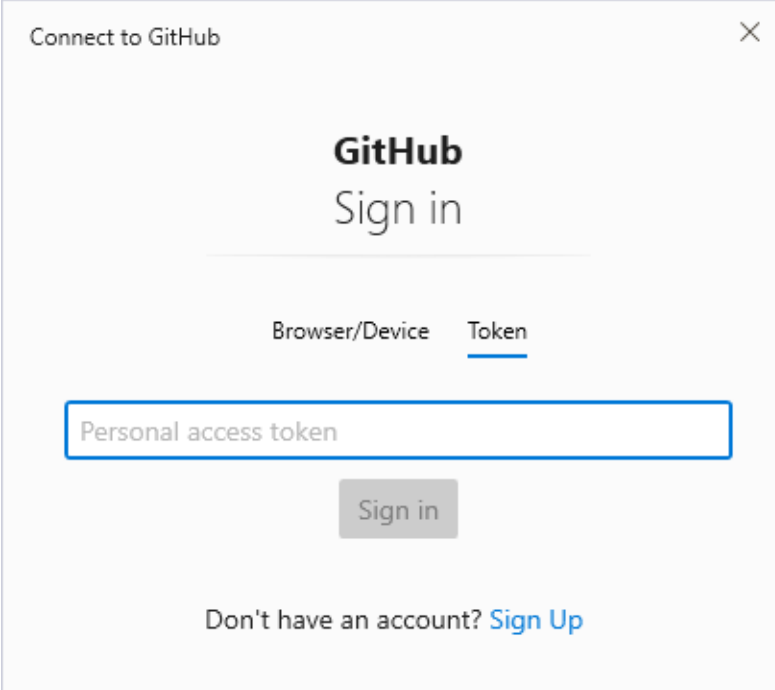
Generate token

Cancel

- 잘 보관 필요
 - 후에 토큰이 표시되지 않음



- 푸시하는 저장소의 권한이 필요하므로 다음 화면의 인증이 필요



The image shows a 'Connect to GitHub' dialog box. At the top, it says 'Connect to GitHub' with a close button (X). Below that is the GitHub logo and 'Sign in'. There are two tabs: 'Browser/Device' and 'Token', with 'Token' being the active tab. Below the tabs is a text input field labeled 'Personal access token'. Below the input field is a 'Sign in' button. At the bottom, it says 'Don't have an account? [Sign Up](#)'.

- 만일 현재 로그인되어 있다면 바로 Browser/Device 클릭
- Token 누르고 PAT 정확히 복사 후 sign in 클릭