







- 1. Personal Access Token
- 2. Push와 Pull
- 3. Fetch



- 1. 깃허브 계정에서 PAT를 생성해 저장할 수 있다.
- 2. 원격 저장소와 연동해 원격 저장소의 수정을 pull할 수 있다.
- 3. 원격 저장소와 연동해 원격 저장소의 수정을 fetch한 후 병합할 수 있다.
- 4. 원격 저장소와 연동해 지역 저장소의 수정을 push할 수 있다.









## 기가인접근토큰(Personal Access Token)생성

# 🦚 개인 접근 토큰 인증 개요

- ☑ 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



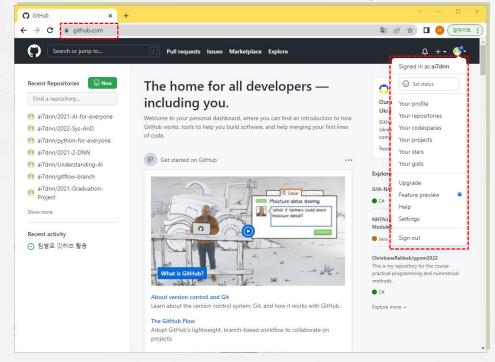


## 기인 접근 토큰(Personal Access Token)생성

## Personal Access Tokens

## ₩ 프로필 메뉴

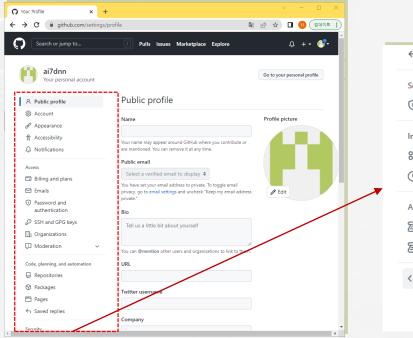
● 로그인 후 GitHub.com에서 우측 메뉴 Settings

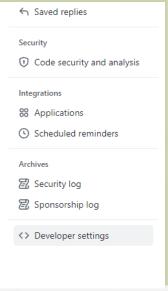






- 기가인접근토큰(Personal Access Token)생성
- 🐡 프로파일 settings 메뉴
  - ✓ 왼쪽 메뉴 가장 하단: Developers settings









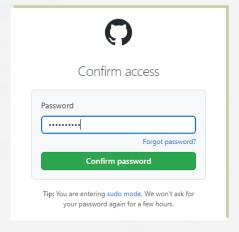
# 개인접근토큰(Personal Access Token)생성

# Personal access tokens



Settings / Developer settings		
88 GitHub Apps	Personal access tokens	Generate new token
8 OAuth Apps	Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.	
Personal access tokens		
<u> </u>	Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git authenticate to the API over Basic Authentication.	over HTTPS, or can be used to

# 營 암호 인증







# 개인접근토큰(Personal Access Token)생성





- ₩ 범위 선택
  - 상황에 따라
    - Repo, Workflow, gist 정도 선택

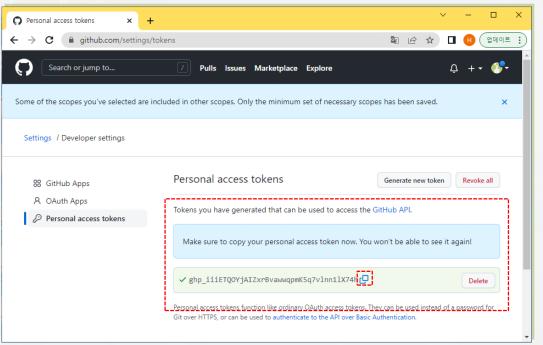
New personal acce	ss token	
	like ordinary OAuth access tokens. They can be uso sticate to the API over Basic Authentication.	ed instead of a password for Git over
Note		
token of repo workflow gist		
What's this token for?		
Expiration *		
90 days \$\Pincete \text{The token w}	ill expire on Fri, Jul 22 2022	
Select scopes		
Scopes define the access for per	sonal 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 or
☐ 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
<b>✓</b> gist	Create gists
notifications	Access notifications
user	Update ALL user data
read:user	Read ALL user profile data
usenemail	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:gpg_key	Full control of public user GPG keys (Developer Preview)
☐ write:gpg_key	Write public user GPG keys
☐ read:gpg_key	Read public user GPG keys





- 기가인접근토큰(Personal Access Token)생성
- 🤲 생성된 개인 접근 토큰
  - ❤ 잘 보관 필요
    - 후에 토큰이 표시되지 않음









### 깃허브원격저장소수정후pull

# 🦚 깃 깃허브 push pull

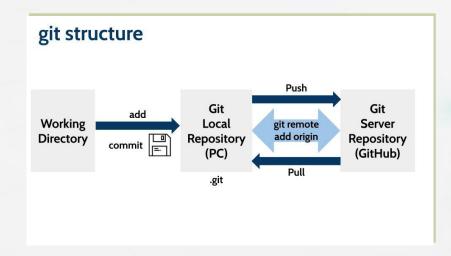


○ 밀기 끌기





○ 올리기 내리기

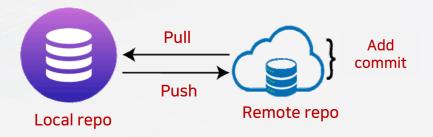




### 기허브원격저장소수정후pull /



- ☑ 깃허브 원격 저장소에서 수정 후 커밋
  - 원격 저장소의 수정을 지역 저장소에 반영
    - 지역 저장소에서 pull





### 기허브원격저장소수정후 pull

# 🏶 윈도 push 중 오류



# git push

remote: Permission to ...

fatal: unable to access 'https://github.com/....git/':

The requested URL returned error: 403

# ☑ 인증 오류 발생 가능

\$ git push

remote: Support for password authentication was removed on August 13, 2021. remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.

fatal: Authentication failed for 'https://github.com/ai7dnn/git-clone.git/'



### 깃허브원격저장소수정후pull

# 🏶 윈도 push 중 오류

# ★ 참조 오류 발생 가능

# git push

remote: Permission to ...

fatal: unable to access 'https://github.com/....git/':

The requested URL returned error: 403

## ★ Windows 환경에서 깃 사용자 계정과 암호 설정 및 변경

- 1. 제어판 사용자 계정 자격 증명 관리자
- 2. Windows 자격 증명
- 3. 일반 자격 증명
- 4. git 관련 자격 증명 편집







### 기허브원격저장소수정후 pull

# ₩ 인증 오류 해결 방법

## ✓ PAT(Personal Access Token)로 연결

- \$ git push -u https://{token}@github.com/{username}/{repo\_name}.git
- \$ git push -u <a href="https://ghp\_GeU8yVKNmKdlpZ787LHDb6HARqF8h@github.com/ai7dnn/git-clone.git">https://ghp\_GeU8yVKNmKdlpZ787LHDb6HARqF8h@github.com/ai7dnn/git-clone.git</a>

```
파이썬@DESKTOP-8TN3J1L MINGW64 ~/git-clone (main)
$ git push
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/
about-remote-repositories#cloning-with-https-urls for information on currently
 recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/ai7dnn/git-clone.git/'
파이쌕@DESKTOP-8TN3J1L MINGW64 ~/qit-clone (main)
$ git push -u https://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@github.com/ai7dnn/git-clone.git
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ai7dnn/git-clone.git
  1470320..358adce main -> main
branch 'main' set up to track
'https://ghp_TafJF7ghxKWXAh1M8FvlLg3r1Doi832m2gJz@github.com/ai7dnn/git-clone.git/main'.
```





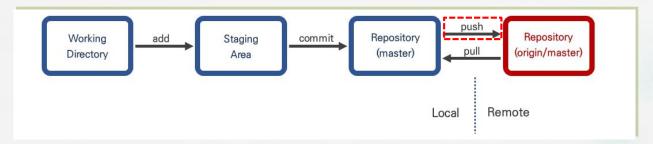




### 지역에서 원격 저장소로 push

# 🦚 지역 저장소에서 push

- ✓ 쓰기 권한이 있어야 push가 가능
  - 자신의 저장소나 다른 사람의 저장소라면 협업자(collaborator)로 등록
- **M** Push
  - 로컬 저장소에서 남겨놓은 코드 변경 이력들이 원격 저장소로 전송
- ₩ 명령
  - \$ git push <저장소별칭명> <브랜치명>
    - \$ git push origin topic







### 지역에서 원격 저장소로 push

## 🥶 명령 push에서 인자 생략하기

- ☑ 옵션 -u 사용
  - 최초에 한 번만 저장소명과 브랜치명을 입력하고 그 이후에는 모든 인자를 생략 가능
    - \$ git push origin topic
  - 성공 이후 간단히 다음으로 가능
    - \$ git push
- - 설정 변수 push.default 를 current로 설정
    - \$ git config --global push.default current
  - 어느 브랜치에서 작업을 하든 git push만 날리면 원격 저장소에 동일한 브랜치로 코드 변경이 업로드
    - \$ git push



### 지역에서 원격 저장소로 push

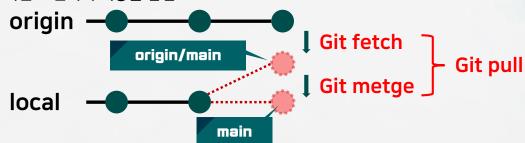




- fetch 명령과 병합하는 merge 명령이 순차적으로 진행
- o fetch 명령
  - 원격 저장소의 정보를 로컬(remote/origin) 저장소로 가져오는 명령
- o merge 명령
  - 변경된 정보를 로컬 저장소의 내용과 병합

# ★ fetch 명령으로 작업한 내용을 확인한 후

- merge 여부를 결정하는 과정
  - 병합(merge)
    - ➡ 다른 브랜치의 내용을 합침







### 지역에서 원격 저장소로 push

# # fetch

- ☑ 원격 저장소에서 로컬 저장소로 소스를 가져와 병합을 미수행하는 명령
  - 단지 소스를 가져올 뿐 병합을 하지 않음
- ₩ 명령어
  - \$ git fetch <remote>
    - \$ git fetch origin
- ▼ fetch후 이후 origin/main, origin/HEAD가 더 최신의 위치에 있음
  - 지역인 HEAD의 main이 이후에 있음
    - \$ git log --all --oneline --graph

```
$ git fetch origin
From https://github.com/ai7dnn/git-clone
   1470320..389ab6c main -> origin/main
$ git lgag1
* 389ab6c (origin/main, origin/HEAD) Update test.md
* 358adce (HEAD -> main) 1
* 1470320 add test.md
* d510717 add file
* b04bff5 Add hello.py
* f49fff7 Update README.md
* 6be0ab8 Initial commit
```





### 지역에서 원격 저장소로 push

# 🐞 fetch 후 병합

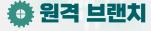
# ☑ 깃허브의 내용을 반영하려면 fetch 후 merge 해야함

\$ git merge origin/main

```
파이썬@DESKTOP-8TN3J1L MINGW64 ~/git-clone (main)
$ git merge origin/main
Updating 358adce..389ab6c
Fast-forward
 test.md | 3 +++
 1 file changed, 3 insertions(+)
파이썬@DESKTOP-8TN3J1L MINGW64 ~/git-clone (main)
$ git lgag1
* 389ab6c (HEAD -> main, origin/main, origin/HEAD) Update test.md
* 358adce 1
* 1470320 add test.md
* d510717 add file
* b04bff5 Add hello.py
* f49fff7 Update README.md
* 6be0ab8 Initial commit
```



### 지역에서 원격 저장소로 push



# ☑ 원격을 포함한 모든 브랜치 파악

- 지역의 브랜치와 HEAD
  - main HEAD
- 원격의 브랜치와 HEAD
  - origin/main origin/HEAD

remotes/origin/main

```
파이썬@DESKTOP-8TN3J1L MINGW64 ~/git-clone (main)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/ai7dnn/git-clone.git
Push URL: https://github.com/ai7dnn/git-clone.git
HEAD branch: main
Remote branch:
    main tracked
Local ref configured for 'git push':
    main pushes to main (up to date)

파이썬@DESKTOP-8TN3J1L MINGW64 ~/git-clone (main)
$ git branch --all
* main
    remotes/origin/HEAD -> origin/main
```





# Summary

- >> 원격 저장소 복제 연동
  - \$ git clone https://github.com/atom/atom.git
- >> 원격 저장소 수정 사항 pull로 지역 저장소로 가져오기
  - \$ git pull origin main
  - \$ git pull
- >> 원격 저장소 수정 사항 fetch로 지역 저장소로 가져와 병합하기
  - \$ git fetch origin main
  - \$ git fetch
  - ◆ \$ git merge origin/main
- >> 지역 저장소 수정 사항 push로 원격저장소 보내기
  - \$ git push origin main
  - \$ git push