







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



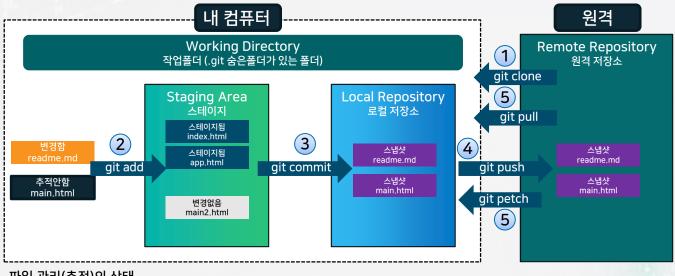
- 1. 깃허브에서 원격 저장소를 생성할 수 있다.
- 2. 원격 저장소와 연동해 원격 저장소의 수정을 pull할 수 있다.
- 3. 원격 저장소와 연동해 원격 저장소의 수정을 fetch한 후 병합할 수 있다.
- 4. 원격 저장소와 연동해 지역 저장소의 수정을 push할 수 있다.





clone과 push, pull, fetch 정리

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



파일 관리(추적)의 상태

untracked Tracked

추적안함 변경없음 변경함 스테이지됨 untracked unmodified Modified index.html





. DMI



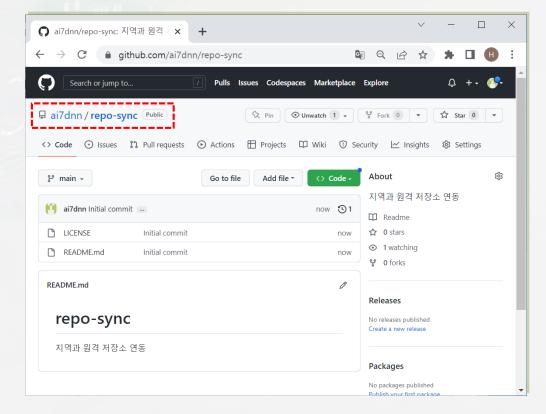
√ 깃허브원격저장소수정후지역에서 pull

💮 실습 사례

- ▼ 1. 원격 저장소에서 repo-sync 생성
- ★ 2. 지역 저장소에서 원격 저장소 repo-sync 복제
- ▼ 3. 원격 저장소에서 파일 hello.md 생성
- ★ 4. 지역 저장소에서 원격 저장소 repo-sync pull
- ▼ 5. 원격 저장소에서 파일 hello.md 수정
- ★ 6. 지역 저장소에서 원격 저장소 repo-sync fetch
- ▼ 7. 지역 저장소에서 원격 저장소 origin/main 병합



🏚 1. 원격 저장소에서 repo-sync 생성







BMUA 동양미리 인당지능소프

☑ 깃허브원격저장소수정후지역에서 pull □

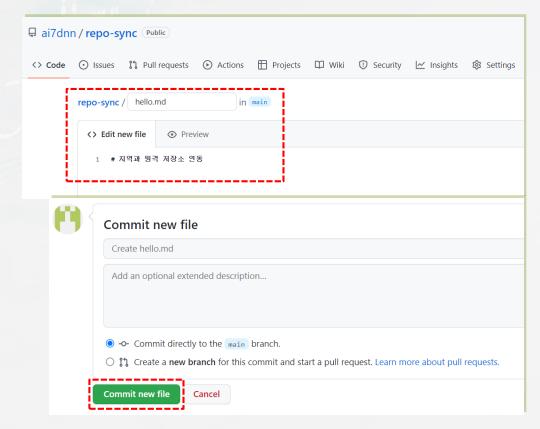
🌞 2. 지역 저장소에서 원격 저장소 repo-sync 복제

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ git clone https://github.com/ai7dnn/repo-sync.git
Cloning into 'repo-sync'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100\% (4/4), 12.51 KiB | 3.13 MiB/s, done.
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cd repo-sync
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git branch --all
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
```

DMU소 동양미래대학교 인공지능소프트웨어학교

🖊 깃허브원격저장소수정후지역에서 pull 🤈

🐞 3. 원격 저장소에서 파일 hello.md 생성



\$ git branch

* main



깃허브원격저장소수정후지역에서 pull

4. 지역 저장소에서 원격 저장소 repo-sync pull

```
#O|M@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 726 bytes | 48.00 KiB/s, done.
From https://github.com/ai7dnn/repo-sync
    f8a93b6..b2dbb15 main -> origin/main
Updating f8a93b6..b2dbb15

Fast-forward
hello.md | 1 +
1 file changed, 1 insertion(+)
create mode 100644 hello.md
```

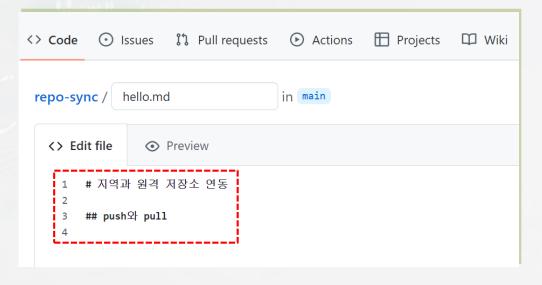
```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git diff main origin/main
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ ls
LICENSE README.md hello.md
```

- \$ git branch -a
 * main
 remotes/origin/HEAD ->
 origin/main
 remotes/origin/main
- \$ git branch -r
 origin/HEAD -> origin/main
 origin/main

- . DM
 - **조양미래대학교** 인공지능소프트웨어학과

1 기허브원격저장소수정후지역에서 pull 기





git fetch 전에 다음 검사한 후 진행

\$ git log --oneline --graph --all



깃허브원격저장소수정후지역에서 pull

🧰 6. 지역 저장소에서 원격 저장소 repo-sync fetch

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ ait fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 691 bytes | 38.00 KiB/s, done.
From https://github.com/ai7dnn/repo-sync
   b2dbb15..21c361e main
                               -> origin/main!
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 (HEAD -> main) Create hello.md
* f8a93b6 Initial commit
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git diff main origin/main
diff --git a/hello.md b/hello.md
index 9def62a..45d287b 100644
--- a/hello.md
+++ b/hello.md
00 -1 +1.3 00
 # 지역과 원격 저장소 연동
+## push와 pull
```

```
fetch는 원격저장소의
               최신 커밋 정보로 수정
  파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (mai
 $ git log -/one/line --graph --all
 * b2dbb15 (HEAD -> main, origin/main, origin/HEAD) Create
 hello.md
  * f8a93b6/Initial commit
 파이쌕@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (mai
 $ git fetch
  remote: Enumerating objects: 5, done.
 b2dbb15..21c361e main
                                  -> origin/main
 파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (mai
$ git log -=oneline =-graph -=all

* 21c36le (origin/main, origin/HEAD) Update hello.md

* b2dbb15 (HEAD -> main) Create hello.md

* f8a93b6 Initial commit
```

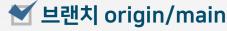




깃허브원격저장소수정후지역에서 pull

현재 상황 파악





- 원격 저장소와 다른 상태
- fetch로 원격 저장소와 동일

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 (HEAD -> main) Create hello.md
* f8a93b6 Initial commit
                                             파일 hello.md의 차이 보기:
$ cat hello.md
                                             각각의 브랜치에서 직접 보기
# 지역과 원격 저장소 연동
$ git switch origin/main # 옵션 -d 필요
fatal: a branch is expected, got remote branch 'origin/main'
hint: If you want to detach HEAD at the commit. try again with the --detach option.
$ git checkout origin/main
Note: switching to 'origin/main'.
You are in 'detached HEAD' state. You can look around, make experimental
HEAD is now at 21c361e Update hello.md
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync ((21c361e...))
$ cat hello.md
                                             파일 hello.md의 차이 보기:
# 지역과 원격 저장소 연동
                                             각각의 브랜치에서 직접 보기
## push와 pull
```





깃허브원격저장소수정후지역에서 pull

🍅 분리된 HEAD로 이동의 2가지 방법





```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
 * 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 (HEAD -> main) Create hello.md
* f8a93b6 Tnitial commit
<u>파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)</u>
$ git switch -d origin/main
HEAD is now at 21c361e Update hello.md
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync ((21c361e...))
$ git log --oneline --graph --all
* 21c361e (HEAD, origin/main, origin/HEAD) Update hello.md
* b2dbb15 (main) Create hello.md
* f8a93b6 Tnitial commit
파이쌕@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync ((21c361e...))
$ git switch -
Previous HEAD position was 21c361e Update hello.md
Switched to branch 'main'
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
   (use "git pull" to update your local branch)
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 (HEAD -> main) Create hello.md
* f8a93b6 Initial commit
```



기허브원격저장소수정후지역에서 pull

🦚 7. 지역 저장소에서 원격 저장소 origin/main 병합

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 (HEAD -> main) Create hello.md
* f8a93b6 Initial commit
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git merge origin/main
Updating b2dbb15..21c361e
Fast-forward
 hello.md | 2 ++
1 file changed, 2 insertions(+)
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ git log --oneline --graph --all
* 21c361e (HEAD -> main, origin/main, origin/HEAD) Update hello.md
* b2dbb15 Create hello.md
* f8a93b6 Initial commit
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/repo-sync (main)
$ cat hello.md
# 지역과 원격 저장소 연동
## push와 pull
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart_git]/repo-sync (main)
$ git diff main origin/main
```







🤍 지역에서 원격 저장소로 push 🦼

🌦 실습 사례

- ▼ 1. 지역 저장소에서 client.md 파일 생성 후 커밋
- ★ 2. 지역 저장소에서 원격저장소로 push 보내기
- ☑ 3. 깃허브 원격 저장소에서 확인



지역에서 원격 저장소로 push

🌞 1. 지역 저장소에서 client.md 파일 생성 후 커밋

```
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/ai7dnn/repo-sync.git
  Push URL: https://github.com/ai7dnn/repo-sync.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
$ echo local repository > client.md
$ cat client.md
local repository
$ git add client.md
$ git commit -m 'add client.md'
[main a4d1d60] add client.md
 1 file changed, 1 insertion(+)
 create mode 100644 client.md
$ git log --oneline --graph --all
* a4d1d60 (HEAD -> main) add client.md
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 Create hello.md
* f8a93b6 Initial commit
```

```
$ git remote show origin
* remote origin
Fetch URL: https://github.com/ai7dnn/repo-sync.git
Push URL: https://github.com/ai7dnn/repo-sync.git
HEAD branch: main
Remote branch:
   main tracked
Local branch configured for 'git pull':
   main merges with remote main
Local ref configured for 'git push':
   main pushes to main (fast-forwardable)
```

\$ git remote show origin





지역에서원격저장소로push

21c361e..a4d1d60 main -> main

🐞 2. 지역 저장소에서 원격저장소로 push 보내기

```
* remote origin
  Fetch URL: https://github.com/ai7dnn/repo-sync.git
  Push URL: https://github.com/ai7dnn/repo-sync.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (fast-forwardable)
$ git log --oneline --graph --all
* a4d1d60 (HEAD -> main) add client.md
* 21c361e (origin/main, origin/HEAD) Update hello.md
* b2dbb15 Create hello.md
* f8a93b6 Initial commit
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
writing objects: 100% (3/3), 347 bytes | 173.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ai7dnn/repo-sync.git
```

```
$ git log --oneline --graph --all
* a4d1d60 (HEAD -> main, origin/main, origin/HEAD) add client.md
* 21c361e Update hello.md
* b2dbb15 Create hello.md
* f8a93b6 Initial commit
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/ai7dnn/repo-sync.git
  Push URL: https://github.com/ai7dnn/repo-sync.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
```

현재 원격저장소에 로그인 되어 있는지 확인하며, push 이전에 반드시 원격저장소의 최신 내용이 지역저장소에 반영이 되어 있어야 함





지역에서원격저장소로push

Push에서 로그인 화면

- **™** Browser/Device
 - 이미 로그인되어 있다면 바로 수행
 - 로그인 안된 상태라면 로그인 화면에서 로그인



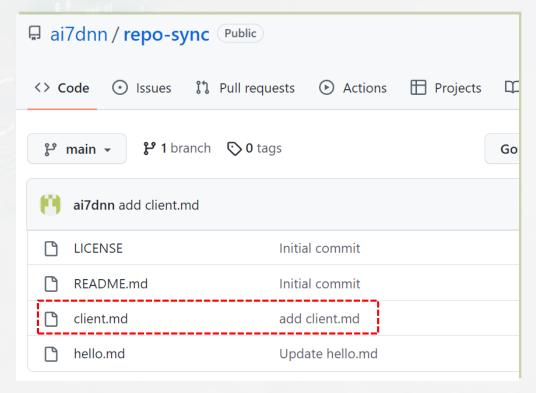


• PAT 입력



지역에서 원격 저장소로 push











지역에서 원격 저장소로 push

🤷 4. 지역 저장소에서 새로운 브랜치 생성 후 파일 추가 커밋

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(main)
$ git switch -c hotfix
Switched to a new branch 'hotfix'
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git ba
* hotfix
 main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ echo hotfix file > hotfix.md
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git add .
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git commit -m 'create branch hotfix and file'
[hotfix e8b4a23] create branch hotfix and file
 1 file changed, 1 insertion(+)
 create mode 100644 hotfix.md
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git lag1
* e8b4a23 (HEAD -> hotfix) create branch hotfix and
file
* 765b83c (origin/main, origin/HEAD, main) add
client.md
* 333a264 Update hello.md
* 11bb0a6 Create hello.md
* 9205272 Initial commit
```





지역에서원격저장소로push

🌞 5. 지역저장소의 새로운 브랜치를 원격저장소에 push: 오류 확인

▼ \$ git push -h로 명령 도움말 검사

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git ba
* hotfix
 main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/ai7dnn/repo-sync2.git
  Push URL: https://github.com/ai7dnn/repo-sync2.git
 HEAD branch: main
  Remote branch:
   main tracked
  Local branch configured for 'git pull':
```

```
main merges with remote main
Local ref configured for 'git push':
   main pushes to main (up to date)
```

```
(hotfix)
$ git push
fatal: The current branch hotfix has no upstream branch.
To push the current branch and set the remote as
upstream, use
```

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2

```
git push --set-upstream origin hotfix
```

To have this happen automatically for branches without a tracking upstream, see 'push.autoSetupRemote' in 'git help config'.



지역에서 원격 저장소로 push

🌞 6. 지역저장소의 새로운 브랜치를 원격저장소에 push -u [원격별칭] [새브랜치이름]

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git push -u origin hotfix
Enumerating objects: 4, done.
Counting objects: 100\% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 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.
remote:
remote: Create a pull request for 'hotfix' on GitHub
by visiting:
             https://github.com/ai7dnn/repo-
remote:
sync2/pull/new/hotfix
remote:
To https://github.com/ai7dnn/repo-sync2.git
```

```
* [new branch] hotfix -> hotfix
branch 'hotfix' set up to track 'origin/hotfix'.

PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2
(hotfix)
$ git ba
* hotfix
main
remotes/origin/HEAD -> origin/main
remotes/origin/hotfix
remotes/origin/main
```





2 지역에서원격저장소로push

🦚 7. 지역저장소의 새로운 브랜치를 원격저장소에 push -u 로 성공하기 위한 설정

★ \$ git config –global push.autoSetupRemote true

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2 (hotfix)
$ git switch -c develop
Switched to a new branch 'develop'
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2 (develop)
$ echo develop file > develop.md
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2 (develop)
$ git add .
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2 (develop)
$ git commit -m 'develop'
[develop 6f698aa] develop
 1 file changed, 1 insertion(+)
 create mode 100644 develop.md
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-sync2 (develop)
$ git config --global push.autoSetupRemote true
```



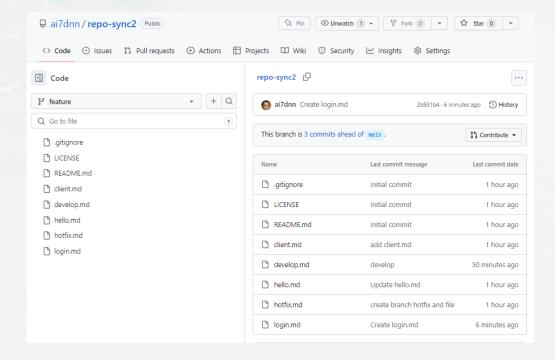


2 지역에서 원격 저장소로 push

🐞 8. 지역저장소의 새로운 브랜치를 원격저장소에 push -u 로 성공

```
sync2/pull/new/develop
PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-remote:
sync2 (develop)
                                              To https://github.com/ai7dnn/repo-sync2.git
$ git push
                                               * [new branch] develop -> develop
Enumerating objects: 4, done.
                                              branch 'develop' set up to track
Counting objects: 100\% (4/4), done.
                                               'origin/develop'.
Delta compression using up to 8 threads
Compressing objects: 100\% (2/2), done.
                                              PC@DESKTOP-482NOAB MINGW64 /c/[2023 git]/repo-
writing objects: 100\% (3/3), 267 bytes
                                              sync2 (develop)
267.00 KiB/s, done.
                                              $ git branch --all
Total 3 (delta 1), reused 0 (delta 0), pack-
                                              * develop
reused 0
                                                hotfix
remote: Resolving deltas: 100\% (1/1),
                                                main
completed with 1 local object.
                                                 remotes/origin/HEAD -> origin/main
                                                 remotes/origin/develop
remote:
                                                remotes/origin/hotfix
remote: Create a pull request for 'develop' on
GitHub by visiting:
                                                 remotes/origin/main
remote:
            https://github.com/ai7dnn/repo- $
```

- 지역에서 원격 저장소로 push
- 🐡 9. 원격저장소에서 새로운 브랜치 생성
 - ☑ 브랜치 이름 feature, 새로운 파일 login.md 생성 커밋









지역에서 원격 저장소로 push

🐞 10. 지역저장소에서 원격의 새로운 브랜치 가져오기

☑ 원격 브랜치 origin/feature를 지역 브랜치 feature 생성 후 연결 후 연결, 데이터 가져오기

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023
git]/repo-sync2 (develop)
$ git ba
* develop
 hotfix
 main
 remotes/origin/HEAD -> origin/main
 remotes/origin/develop
 remotes/origin/feature
  remotes/origin/hotfix
  remotes/origin/main
PC@DESKTOP-482NOAB MINGW64 /c/[2023
git]/repo-sync2 (develop)
$ git checkout -t origin/feature
Switched to a new branch 'feature'
branch 'feature' set up to track
'origin/feature'.
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[2023
git]/repo-sync2 (feature)
$ git ba
 develop
* feature
 hotfix
 main
 remotes/origin/HEAD -> origin/main
 remotes/origin/develop
  remotes/origin/feature
  remotes/origin/hotfix
  remotes/origin/main
PC@DESKTOP-482NOAB MINGW64 /c/[2023
git]/repo-sync2 (feature)
$ cat login.md
## login
```





Summary

- >> 원격 저장소 복제 연동
 - \$ git clone https://github.com/ai7dnn/repo-sync.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 원격 저장소 보내기
 - \$ git push origin main
 - \$ git push