


[10/23] Git CL - branch, merge, log,

- Git branch & merge¹
 - main / dev / branch 1,2,3,4
 - main 과 dev 를 통해서만 병합 (최종 점검)
 - dev 와 branch 1,2,3,4 를 통해서만 병합 (관리 용이)
- git log
 - commit 뒤 **암호(SHA-256)** 을 통해 추후 파일 되돌리기(restore) 가능

```
C:\git-work\1023work>git log
commit 2f4a78c2b85934b3a494370af17e2453b7f72298 (HEAD -> main, origin/main)
Author: student02 <gkmh0925@gmail.com>
Date:   Mon Oct 23 10:37:51 2023 +0900

    10-23-02
```

- main protect
 - 관리자/레파지토리 생성자는 **protect** 설정 해도 올라감
 - **branch** 생성시 **main** 의 **push** 를 방지
 - **push failed** : Everything up-to-date 안내가 나와 변경된 것이 없다고 뜸(=권한없음)

 **Your main branch isn't protected**
Protect this branch from force pushing or deletion, or require status checks before merging.
[Learn more](#) Protect this branch ×

'protect this branch'

Protect matching branches

☒ **Require a pull request before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☒ **Require approvals**
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.

☒ **Lock branch**
Branch is read-only. Users cannot push to the branch.

'권장 체크박스 설정'

Branch protection rules

Add rule

main

Currently applies to 1 branch

Edit

Delete

setting > branches 에서 변경가능

¹ [Pro git](#) (korean) 65P 참조

- Git bash
 - 현재의 branch 위치를 한눈에 볼 수 있음 (dev) (main)

```

MINGW64:/c/git-work/1023work

student02@student02 MINGW64 ~
$ cd ..

student02@student02 MINGW64 /c/Users
$ cd c:\git-work\1023work
bash: cd: c:git-work1023work: No such file or directory

student02@student02 MINGW64 /c/Users
$ cd c:/git-work/1023work

student02@student02 MINGW64 /c/git-work/1023work (dev)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

student02@student02 MINGW64 /c/git-work/1023work (main)
$

```

- git branch -a
 - 각 브랜치마다 config 설정이 안되어 있을 때 발생
 - 로컬과 깃헙의 브랜치 연결이 필요
 - git push --set-upstream origin

```

PS C:\git-work\1023work> git branch -a
* dev
  main
remotes/origin/main

```

- branch 생성
 - git checkout -b dev
 - git branch dev
 - 동일한 명령어
 - 현재 디렉토리를 기반으로 복제하여 새로운 디렉토리(dev)를 생성
- Push
 - git push -u origin kmh
 - git push --set-upstream origin kmh
 - 동일한 명령어
- 공동 작업시
 - repository download (Pull) 를 우선 실행하여 작업 동기화
 - git pull origin edit
 - add / commit / push 순서로 작성
 - git push -u origin edit

ProGit 정독 필요