

Git ??? ?? ?? ??: merge, rebase, cherry-pick

1. ?? ?? ??

명령어	설명
merge	현재 브랜치와 지정된 브랜치를 통합하여 새로운 커밋을 생성한다.
rebase	현재 브랜치의 커밋들을 지정된 브랜치의 커밋들 위에 다시 작성한다.
cherry-pick	지정된 브랜치의 특정 커밋을 현재 브랜치로 가져온다.

2. ????? ?? ??

`git merge <브랜치명>`

- 현재 브랜치 : 현재 브랜치의 커밋들을 지정된 브랜치의 커밋들 위에 다시 작성한다.
- 브랜치명 : merge할 브랜치의 이름

```
git checkout main          # 현재 브랜치로 체크아웃
git merge feature/login    # feature/login 브랜치와 merge
```

- 결과 : main ← feature/login 커밋

`git rebase <브랜치명>`

- 현재 브랜치 : 현재 브랜치의 커밋들을 지정된 브랜치의 커밋들 위에 다시 작성한다.
- 브랜치명 : rebase할 브랜치의 이름
- SHA : rebase할 커밋의 SHA (선택 사항)

```
git checkout feature/login
git rebase main
```

- `git : feature/login` `main` `git` `git`

`git cherry-pick <commit>`

- `git` : `git` `git` `git` `git`
- `git` `git` `git` `git` `git`

```
git checkout main
git cherry-pick 7a1f3b2
```

3. ?? ?? ??

`merge`

```
A---B---C---M      (main)
  \      /
   D---E      (feature)
```

`rebase`

```
A---B---C---D'---E' (feature rebased onto main)
```

`cherry-pick`

```
main:      A---B---C---E'
feature:      D---E
```

4. ??? ??

<code>git</code> <code>git</code> <code>git</code> <code>git</code> <code>git</code>	<code>merge</code>
<code>git</code> <code>git</code> <code>git</code> <code>git</code>	<code>rebase</code>

5. ?????

- merge: 1 2 3 4 5 6 7 8
- rebase: 1 2 3 4 5 6 7 8 (1 2 3 4 5 6 7 8)
- cherry-pick: 1 2 3 4 5 6 7 8

6. ?? ????: revert, reset, stash

📄	📁	📂 📅 📌
revert	1 2 3 4 5 6 7 8 1 2 3 4	1 2 3 4 5 6 7 8
reset	1 2 3 /456 /7 8 9 1 2 3	1 2 3 4 5 6 7 8 9
stash	1 2 3 4 5	1 2 3 4 5 6 7 8

```
# revert: 📄 📁
git revert <📄 📁>

# reset: 📄 📁📂
git reset --soft HEAD~1 # 📄 📁
git reset --hard HEAD~1 # 📄 📁📂 📅

# stash: 📄 📁 📂
git stash
git stash pop
```