

git reset

1. Git Reset ???

git reset Git 的 命令 可以 (HEAD) 的 状态 重置 , 但是 不会 丢失 任何 数据 , 且 可以 指定 重置 到 哪个 版本 .

“ 重置 到 哪个 版本 / 哪个 分支 ... 重置 到 哪个 版本 ! ” 的 意思 .

2. reset ?? ?? ??

命令	HEAD 指针	Staging Area 状态	工作区 状态
--soft	指向 O	保留 所有	保留 所有
--mixed	指向 O	保留 所有	保留 所有
--hard	指向 O	保留 所有	丢弃 所有

3. ??? ??

初始 状态:

A---B---C---D (HEAD)

↑

当前 分支

```
git reset --soft B
```

结果:

A---B (HEAD)

↑

현재 커밋 C, D로 리셋된 상태

```
git reset --hard B
```

HEAD:

A---B (HEAD)

↑

C, D 커밋은 삭제됨 (이전 브랜치에 있음)

4. ??? ??

? --soft

```
git reset --soft <커밋>
```

- 커밋된 파일은 유지
- 커밋된 파일은 유지, 커밋된 파일은 삭제

예시 :

```
“add 파일1 파일2 파일3 , 파일4 파일5 파일6 파일7
```

? --mixed (?? ??)

```
git reset --mixed <커밋>
```

- 파일 & 디렉토리 모두 리셋
- 파일은 리셋, 디렉토리는 유지

예시 :

```
“add 파일1 파일2 파일3 파일4 , git add 파일5 파일6 파일7
```

? --hard

```
git reset --hard <commit>
```

- 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

11. 12. :

“ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. ”

△ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. !

5. ?? ??

```
# 1. 2. 3. 4.
echo "a" > a.txt && git add . && git commit -m "A"
echo "b" > b.txt && git add . && git commit -m "B"
echo "c" > c.txt && git add . && git commit -m "C"

git log --oneline
```

```
# 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12.
git reset --soft HEAD~1 # 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12.
git reset --mixed HEAD~1 # 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12.
git reset --hard HEAD~1 # 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12.
```

6. git reset vs git revert ??

| 1. | 2. | 3. | 4. | 5. |
|--------|----|------|----|-------------------------------------|
| reset | 1. | 2. X | 3. | 4. 5. 6. 7. 8. 9. 10. , 11. 12. |
| revert | 1. | 2. O | 3. | 4. 5. 6. 7. 8. 9. 10. , 11. 12. 13. |

? ??

| ?? | ?? | ?? ?? |
|---------|--------------------------|---------------|
| --soft | ??? ?? , staging? ??? | ??? ?? ??? ?? |
| --mixed | ?? + staging ?? , ??? ?? | ??? |
| --hard | ?? + staging + ??? ?? ?? | ??? ? ?? *?? |

Revision #2
Created 20 May 2025 08:44:27 by Dain
Updated 28 May 2025 07:58:30 by Dain