

# git reflog

- 기본 기능
- 특정 Reflog 보기
- 시간 이력

Git은 reference log를 이용해 branch의 최신 버전을 유지한다. Reflog는 reference logs 를 뜻하며 reference 또는 ref는 commit 를 가르킨다.

## 기본 기능

Git 이력 정보를 얻을 수 있다.

```
git reflog [show HEAD]
```

결과:

```
eff544f HEAD@{0}: commit: migrate existing content
bf871fd HEAD@{1}: commit: Add Git Reflog outline
9a4491f HEAD@{2}: checkout: moving from master to git_reflog
9a4491f HEAD@{3}: checkout: moving from Git_Config to master
39b159a HEAD@{4}: commit: expand on git context
9b3aa71 HEAD@{5}: commit: more color clarification
f34388b HEAD@{6}: commit: expand on color support
9962aed HEAD@{7}: commit: a git editor -> the Git editor
```

## 특정 Reflog 보기

전체 Reflog 보기

```
git reflog show --all
```

특정 Reflog 보기

```
git reflog show <Reflog name>
```

otherbranch 브랜치 지정 결과:

```
git reflog show otherbranch
9a4491f otherbranch@{0}: commit: seperate articles into branch PRs
35aeee4a otherbranch@{1}: commit (initial): initial commit add git-init and
setting-up-a-repo docs
```

## 시간 이력

Reflog는 timestamp를 가지고 있으므로 시간을 활용한 다양한 기능을 이용할 수 있다.

master 브랜치에 대해 하루 이전의 이력 표시 예:

```
git diff master@{0} master@{1.day.ago}
```

시간 표현 방법)

- 1.minute.ago
- 1.hour.ago
- 1.day.ago
- yesterday
- 1.week.ago
- 1.month.ago
- 1.year.ago
- 2011-05-17.09:00:00

## Git 가이드

### 1. Git 시작하기

- Git 저장소
  - git init
  - git clone
  - git config
- 변경 저장하기
  - git add
  - git commit
  - git diff
  - git stash
  - .gitignore
- 저장소 점검하기
  - git status
  - git log
  - git tag
  - git blame
- 변경 취소하기
  - git 실행 취소
  - git clean
  - git revert
  - git reset
  - git rm
- Rewriting history
  - git commit -- amend
  - git rebase
  - git reflog

### 2. Git 협업하기

- 동기화하기
  - git remote
  - git fetch
  - git pull
  - git push
- 브랜치 사용하기
  - git branch
  - git checkout
  - git merge
  - 병합 충돌 해결하기 (Merge conflicts)
  - 병합 전략 (Merge strategies)
- Pull request 만들기

