
Open source Software
오픈소스 소프트웨어
09. [실습] 커밋과 로그 이력, 과거 여행

학습 개요

1. 깃 설정과 저장소 생성
2. 커밋 과정
3. 이전 버전으로 이동 후 다시 최신 버전으로 이동

학습 목표

1. 깃 초기 설정하고 저장소를 생성할 수 있다.
2. 파일을 만들어 버전관리를 위한 과정을 수행할 수 있다.
3. 버전관리 로그 이력을 확인할 수 있다.
4. 이전 버전으로 이동하고 다시 최신 버전으로 이동할 수 있다.

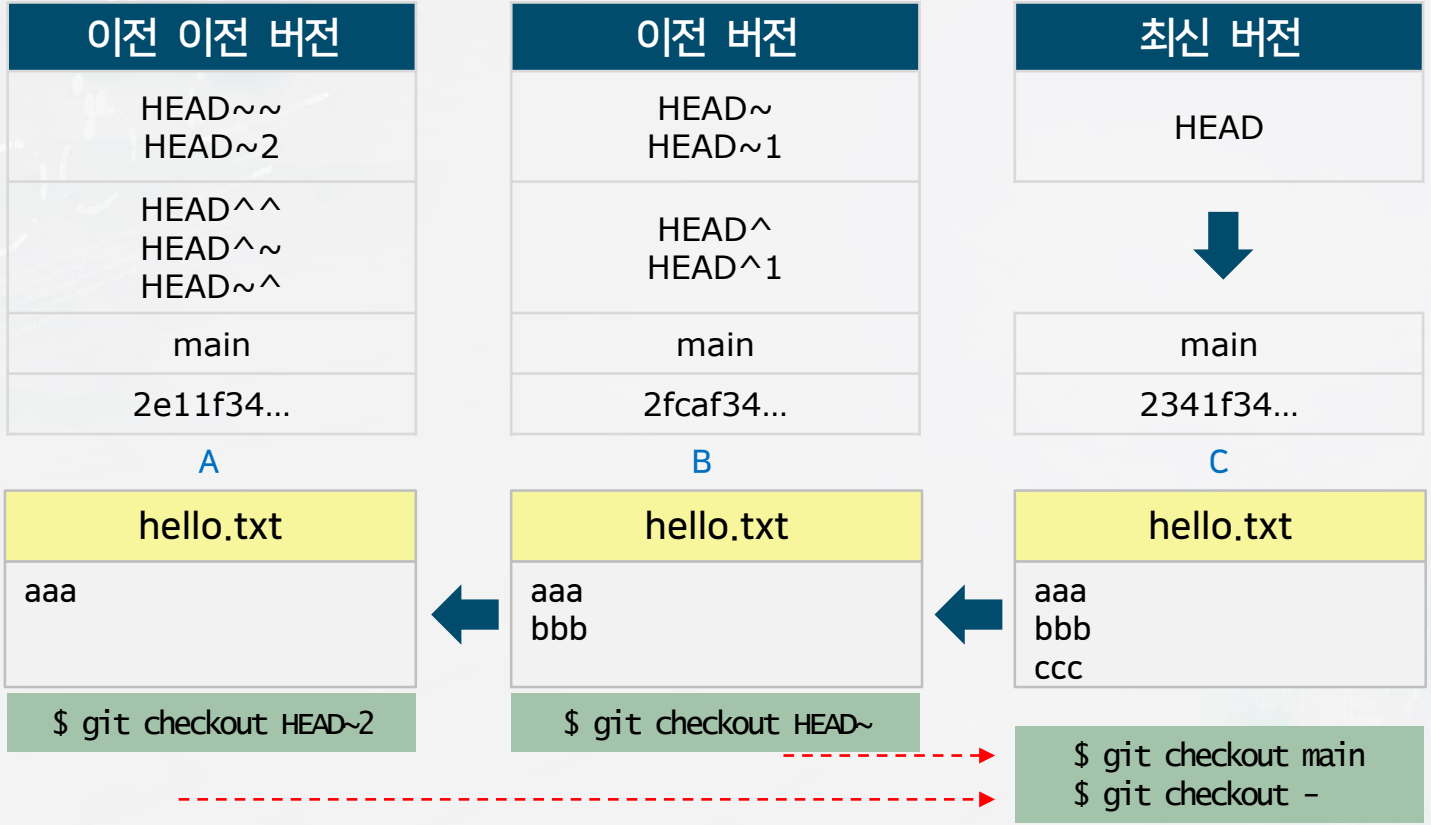
LESSON 01

커밋과 로그 이력

깃 설정과 깃 저장소 basic 생성

	명령	설명
①	$ git config --global user.name hskang	사용자 이름 설정
	$ git config --global user.email hskang@gmail.com	전자메일 설정 설정
	$ git config --global core.autocrlf true $ git config --global core.safecrlf false	자동 줄바꿈 설정 안전 줄바꿈 설정
	$ git config --global core.editor 'code --wait' $ git config --global init.defaultBranch main	기본 편집기 설정 기본 브랜치이름 설정
②	$ git init basic	저장소 생성
	$ cd basic $ ls -al	폴더 이동 파일 확인

3번의 커밋 후 과거와 다시 현재로의 여행



1 커밋과로그이력

1회의 커밋

		작업디렉토리	스테이징 영역	깃 저장소
① 파일 수정	$ echo aaa > hello.txt	aaa		
	$ cat hello.txt $ git status [--long]			
	$ git status -s	?? hello.txt		
② 파일 추가	$ git add hello.txt	aaa	aaa	
	$ git status			
	$ git status [--short | -s]	A hello.txt		
③ 첫 번째 커밋	$ git commit -m A	aaa	aaa	aaa
	$ git status			

	$ git log
	$ git log --oneline
	$ git log [--patch | -p]			

총 2회의 커밋 이력과 수정

		작업디렉토리	스테이징 영역	깃 저장소
④ 파일 수정	$ echo bbb >> hello.txt	aaa bbb		
	$ cat hello.txt $ git status [--long]			
	$ git status -s	M hello.txt		
⑤ 두 번째 커밋	$ git commit -am B	aaa bbb	aaa bbb	aaa bbb
	$ git status			

	$ git log
	$ git log --online
	$ git log [--patch | -p]			

⑥ 다시 파일 수정	$ echo ccc >> hello.txt	aaa bbb ccc	aaa bbb	aaa bbb

	$ cat hello.txt
	$ git status
	$ git status -s
		M hello.txt		

1 커밋과로그이력

총 3회의 커밋 이력과 수정, 추가, 수정

		작업디렉토리	스테이징 영역	깃 저장소
	$ git commit -am C	aaa bbb ccc	aaa bbb ccc	aaa bbb ccc

⑦
세 번째
커밋
	$ git status
	$ git log
	$ git log --online
	$ git log [--patch | -p]
	$ git log --reverse --oneline
	$ git log -2
	$ git log -2 --oneline
	$ git log --graph --oneline			

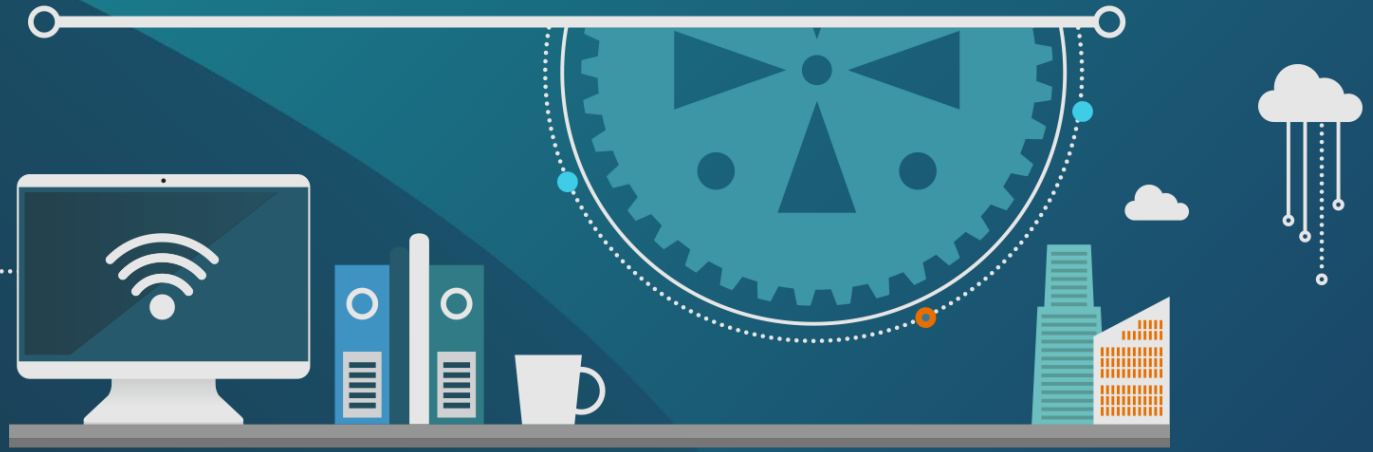
	$ git show
	$ git show --oneline
	$ git show HEAD
	$ git show HEAD~
	$ git show HEAD~2			

총 3회의 커밋 이력과 수정, 추가, 수정

		작업디렉토리	스테이징 영역	깃 저장소
⑧ 수정	$ echo ddd >> hello.txt	aaa bbb ccc ddd	aaa bbb ccc	aaa bbb ccc
	$ git status $ git status -s	M hello.txt		
⑨ 추가	$ git add hello.txt	aaa bbb ccc ddd	aaa bbb ccc ddd	aaa bbb ccc
	$ git status $ git status -s	M hello.txt		
⑩ 수정	$ echo eee >> hello.txt	aaa bbb ccc ddd eee	aaa bbb ccc ddd	aaa bbb ccc
	$ git status $ git status -s	MM hello.txt		

LESSON 02

과거 이전 버전으로 이동



2 과거 이전 버전으로 이동

현재 상태

3번의 커밋이 수행된 상태와 커밋 로그 이력

작업 폴더 (hello.txt)	스테이지 영역 (hello.txt)	깃 저장소 (hello.txt)
aaa bbb ccc ddd eee	aaa bbb ccc ddd	aaa bbb ccc

C	HEAD
aaa bbb ccc	07af7f6
B	HEAD~ HEAD^ HEAD~1 HEAD^1
aaa bbb	3a2d414
A	HEAD~~ HEAD^^ HEAD~^ HEAD~2
aaa	dd3f28a

2 과거 이전 버전으로 이동

명령 checkout

$ git checkout [이전커밋]

- 상태가 clean해야 함
  - 3 영역이 동일한 상태

현재 상태를 clean한 상태로 하는 방법

- 임시 저장인 git stash
  - 작업 디렉토리와 스테이징 영역의 저장하고 3 영역이 동일한 상태가 되도록 후에 다시 자세히 학습 예정

	작업디렉토리	스테이징 영역	깃 저장소
현재 상황	aaa bbb ccc ddd eee	aaa bbb ccc ddd	aaa bbb ccc
$ git checkout HEAD~	오류 발생		
$ git stash	aaa bbb ccc	aaa bbb ccc	aaa bbb ccc
$ cat hello.txt			

2 과거 이전 버전으로 이동

과거 버전 이동 준비

- 작업 영역은 깨끗(Clean)하게
- 스테이징 영역은 커밋과 같게

		작업디렉토리	스테이징 영역	깃 저장소
	현재 상황	aaa bbb ccc ddd eee	aaa bbb ccc ddd	aaa bbb ccc
준비	$ git checkout HEAD~	오류 발생		

	$ git stash
	위는 다음 2개의 명령으로도 같은 기능 수행
	$ git restore --staged hello.txt
	$ git restore hello.txt
		aaa
		bbb
		ccc
			aaa
			bbb
			ccc
				aaa
				bbb
				ccc

	$ cat hello.txt			

2 과거 이전 버전으로 이동

바로 이전 HEAD~으로 이동

바로 이전으로 이동 후 내용 살피고 다시 원복

		작업디렉토리	스테이징 영역	깃 저장소
	$ git checkout HEAD~	aaa bbb	aaa bbb	aaa bbb
	$ git status			
	$ cat hello.txt			

⑪	$ git log
	$ git log --oneline
	$ git log --graph
	$ git log --all --oneline			

	$ git checkout main	aaa bbb ccc	aaa bbb ccc	aaa bbb ccc
	$ cat hello.txt			

A

hello.txt

aaa

$ git checkout HEAD~2

B

hello.txt

aaa
bbb

$ git checkout HEAD~

C

hello.txt

aaa
bbb
ccc

$ git checkout main
$ git checkout -

2 과거 이전 버전으로 이동

명령 checkout 정리

설명	git checkout	git switch
이전 커밋으로 이동	$ git checkout [이전커밋]	$ git switch -d [이전커밋]
다른 브랜치로 이동	$ git checkout [branch]	$ git switch [branch]
새로운 브랜치를 생성하고 이동	$ git checkout -b [newBranch]	$ git switch -c [newBranch]

Summary

저장소 생성

- $ git init basic

깃 저장소에 저장

- $ git add, commit

모두 커밋 이력 보기

- $ git log
  - --oneline --graph --all -n

특정한 커밋 이력 보기

- $ git show [HEAD]
  - --oneline

Summary

바로 이전 버전으로 가기

- $ git checkout HEAD~

다시 최신 버전으로 돌아오기

- $ git checkout main

다시 checkout 이전으로 돌아오기

- $ git checkout -