







- 1. 브랜치 필요성
- 2. 브랜치 생성과 이동
- 3. 브랜치 목록 확인과 삭제



- 1. 브랜치를 이해하고 활용할 수 있다.
- 2. 브랜치 목록을 파악하고 필요하면 생성할 수 있다.
- 3. 생성된 브랜치로 이동해 커밋할 수 있다.
- 4. 필요 없는 브랜치를 삭제할 수 있다.









/ 브랜치개요

🍅 [실습] 기본 브랜치

- ☑ 기본 브랜치 이름 설정
 - 기본 설정으로 수정
 - \$ git config --global init.defaultBranch main
- ✓ 저장소 생성 gbrch
- ☑ 이미 생성된 저장소의 브랜치 이름 수정
 - \$ git branch -M newBname

```
$ git init gbrch
Initialized empty Git repository in C:/[smart git]/gbrch/.git/
$ cd gbrch

파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git branch -M master

파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (master)
$ git branch -M main
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$
```





브랜치개요





- o main 브랜치는 생성된 커밋을 가리킴
- 커밋이 계속 발생
 - main 브랜치는 자동으로 가장 마지막 커밋을 가리킴

```
파이쌘@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ touch README
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git add README
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ ait commit -m A
[main (root-commit) 7672f9e] A
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git log --oneline
7672f9e (HEAD -> main) A
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
```

HEAD



main



7672f92



Α

README

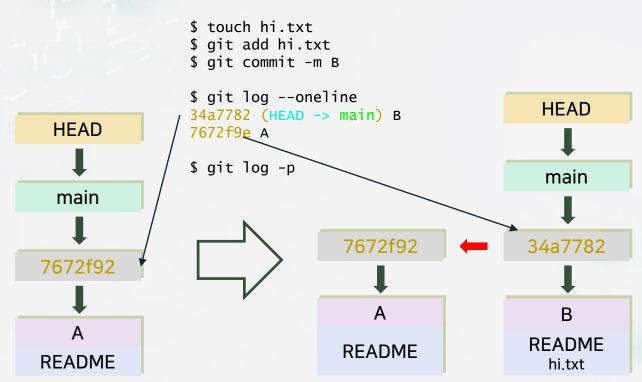




브랜치개요

🐡 두 번째 커밋 이후의 로그 이력

₩ 브랜치 이름 main



/ 브랜치개요

🦚 새로운 브랜치 생성



- 단순히 생성하고 이동은 안함
 - \$ git branch bname
- 생성하고 새 브랜치로 이동도 수행
 - \$ git switch -c bname
 - \$ git checkout -b bname

브랜치 topic을 생성하고 HEAD는 이동하지 않음

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/gbrch (main)

\$ git branch topic

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/gbrch (main)

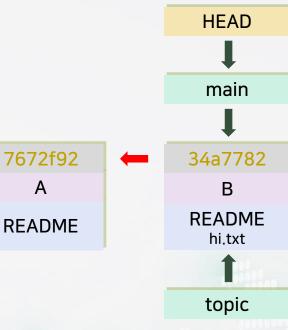
\$ git branch

* main

topic











1 브랜치개요



☑ 명령 git branch로 현재 저장소의 브랜치 목록을 확인

| 주요 명령 | |
|------------------|-----------------------------|
| \$ git branch | 커밋이 발생한 브랜치 목록 보이기 |
| \$ git branch -v | 브랜치마다 마지막 커밋 ID와 메시지도 함께 표시 |

브랜치 main과 topic은 동일한 커밋을 가리키고

HEAD는 main 브랜치를 가리킴을 의미

✓ 결과 표시 * main

■ *은 현재 작업하고 있는 브랜치를 표시

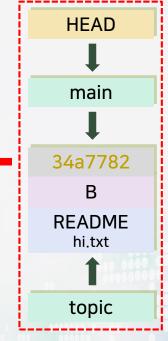
\$ git branch * main topic

\$ git branch -v
* main 34a7782 B
topic 34a7782 B

파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main) \$ git log --oneline --all 34a7782 (HEAD -> main, topic) B 7672f9e A 7672f92

Α

README



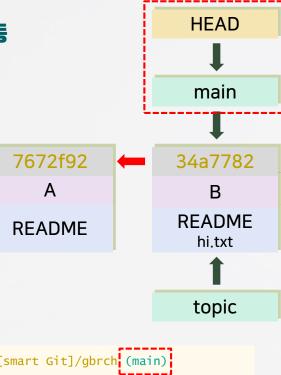
DMUA SSUP URANGAL

1 브랜치개요

🐡 생성된 새로운 브랜치로 이동

☑ 이동

- 지정한 브랜치로 이동
 - \$ git switch [bname]
 - \$ git checkout [bname]
- 이전 브랜치로 이동
 - \$ git switch -
 - \$ git checkout -



main 7672f92 34a7782 Α В **README** README hi.txt topic **HEAD**

PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/gbrch (main)
\$ git switch topic
Switched to branch 'topic'
PC@DESKTOP-482NOAB MINGW64 /c/[smart Git]/gbrch (topic)





브랜치개요

🐞 분리된(detached) HEAD

★ HEAD가 현재 브랜치(마지막 커밋)가 아닌 그 이전 커밋을 가리키는 상태

- \$ git checkout HEAD~
 - 현재 브랜치에서 마지막 커밋 이전 커밋으로 이동
- \$ git switch -d HEAD~
 - 이전 커밋으로 이동하려면 옵션 -d 또는 --detach를 사용





브랜치개요



★ HEAD가 현재 브랜치(마지막 커밋)가 아닌 그 이전 커밋을 가리키는 상태











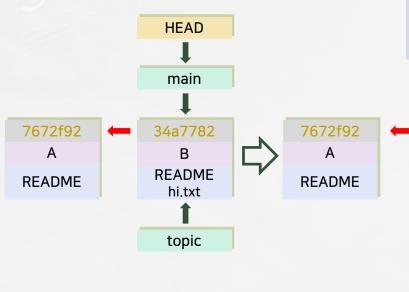
2 브랜치관리

🍅 새 브랜치에서 커밋



₩ 브랜치 topic으로 이동

₩ 브랜치 topic에서 커밋



파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main) \$ git switch topic Switched to branch 'topic' 파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (topic) \$ echo 111 > to.txt \$ git add to.txt \$ git commit -m TOP1 \$ git log --oneline --all c6c76a4 (HEAD -> topic) TOP1 34a7782 (main) B **\$** 1s 7672f9e A README hi.txt to.txt main main 34a7782 7672f92 34a7782 c6c76a4 Α В

README

README

hi,txt

topic

HEAD

B
README
hi.txt

topic

HEAD 1:



브랜치관리

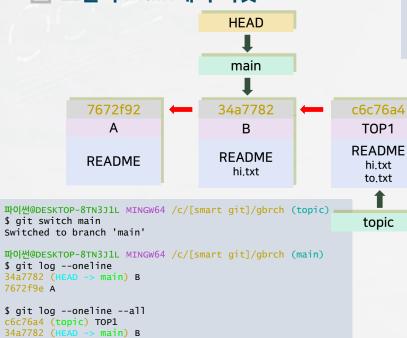
7672f9e A

7672f9e A

👛 기본 브랜치로 이동 후 다시 커밋







\$ git commit -m C \$ git log --oneline --all f25934d (HEAD -> main) C c6c76a4 (topic) TOP1 34a7782 B 7672f9e A \$ git log --oneline --all --graph * f25934d (HEAD -> main) C | * c6c76a4 (topic) TOP1 * 34a7782 B * 7672f9e A 7672f92 34a7782 В Α

README

README

hi.txt

\$ echo main > ma.txt \$ git add ma.txt

> c6c76a4 TOP1 **README** hi.txt to.txt

HEAD

main

f2594d

README

hi.txt

ma.txt

topic

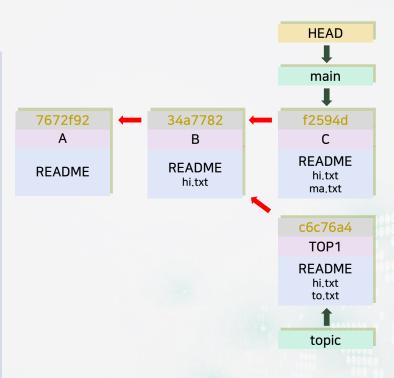






❤ 각각의 브랜치에서

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git log --oneline
f25934d (HEAD -> main) C
34a7782 B
7672f9e A
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git log --oneline --graph --all
* f25934d (HEAD -> main) C
| * c6c76a4 (topic) TOP1
* 34a7782 B
* 7672f9e A
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git switch topic
Switched to branch 'topic'
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (topic)
$ git log --oneline
c6c76a4 (HEAD -> topic) TOP1
34a7782 B
7672f9e A
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (topic)
$ git log --oneline --graph --all
* f25934d (main) C
| * c6c76a4 (HEAD -> topic) TOP1
* 34a7782 B
* 7672f9e A
```





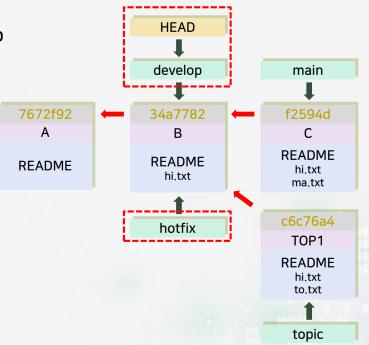


🍅 최신 커밋 이전에서 두 브랜치 생성

\$ git switch main

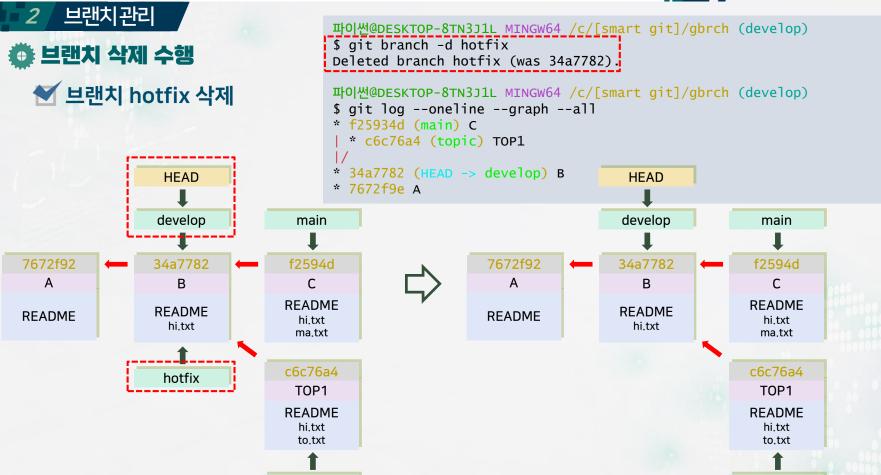
- \$ git switch -c hotfix
- \$ git checkout HEAD~
- \$ git checkout -b develop

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (topic)
$ git switch main
Switched to branch 'main'
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (main)
$ git checkout HEAD~
Note: switching to 'HEAD~'.
You are in 'detached HEAD' state. You can look around, make experimental
HEAD is now at 34a7782 B
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch ((34a7782...))
$ git switch -c hotfix
Switched to a new branch 'hotfix'
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (hotfix)
$ git checkout -b develop
Switched to a new branch 'develop'
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (develop)
$ git branch
* develop
 hotfix
  main
  topic
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gbrch (develop)
$ git log --oneline --graph --all
* f25934d (main) C
| * c6c76a4 (topic) TOP1
* 34a7782 (HEAD -> develop, hotfix) B
* 7672f9e A
```









topic

topic



2 브랜치관리

🍅 브랜치 관련 명령 정리

- ★ \$ git checkout bname
 - 전환, 이동
- **★** \$ git switch bname
 - 전환, 이동
- ★ \$ git checkout -
 - 이전 브랜치로 전환, 이동
- **S** \$ git switch -
 - 이전 브랜치로 전환, 이동
- **★** \$ git branch -h
 - 도움말 보기

| 설명 | git checkout | git switch |
|--------------------------|--|---|
| | 1 | 1 |
| 이전 커밋으로 이동 다른 브랜치로 이동 | \$ git checkout [이전커밋] \$ git checkout [branch] | \$ git switch -d [이전커밋] \$ git switch [branch] |
| | 1 | 1 |
| 새로운 브랜치를 생성하고 이동 | \$ git checkout -b [newBranch] | \$ git switch -c [newBranch] |





Summary

- >> \$ git branch
 - ◆ 저장소 목록 보기
- >> \$ git branch (new-branch)
 - ◆ 저장소 생성만
- >> \$ git checkout -b (new-branch)
 - ◆ 저장소 생성하고 이동
- >> \$ git switch -c (new-branch)
 - ◆ 저장소 생성하고 이동
- >> \$ git branch -d branch-name
 - ◆ 저장소 삭제
- >> \$ git branch -D branch-name
 - ◆ 저장소 삭제, 강제 삭제