

1. Git 과 Branch

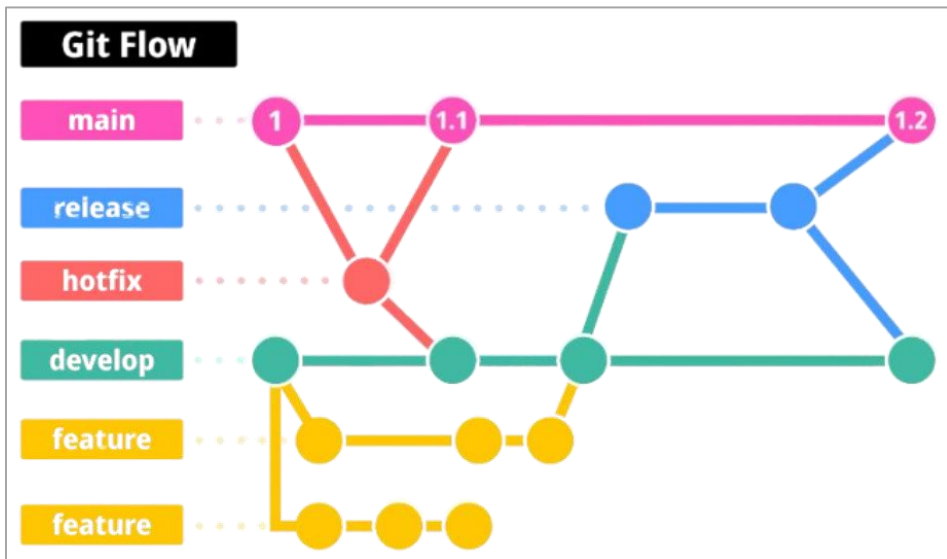


- Branch 개념
- Branch 만들기 및 이동하기
- Branch에서 커밋하기
- Branch 병합하기
- Branch 삭제하기
- 특정 commit 병합

Git의 Branch란?(1)

❖ Branch 개념

- Git에서 독립적인 작업 공간을 제공하여, 메인 코드에 영향을 주지 않고 새로운 기능 개발이나 버그 수정을 병렬로 진행할 수 있는 기능
- 작업 완료 후 메인 브랜치에 병합하여 통합 가능



출처: <https://leetrue.hashnode.dev/branch-merge-strategy>



Git의 Branch란?(2)

◆ Branch 핵심 장점 5가지

1. 안정성 유지

- 각 작업을 독립적으로 수행해, 메인 코드에 영향을 주지 않고 안전하게 개발 가능

2. 동시 작업 지원

- 여러 사람이 동시에 다른 기능이나 수정 작업을 병렬로 진행할 수 있음

3. 협업 효율성

- 팀원 간 코드 충돌을 최소화하며, 코드 리뷰 및 테스트 절차를 쉽게 관리할 수 있음

4. 문제 해결 용이성

- 버그 수정이나 기능 개발 중 문제가 발생해도, 기존 상태로 쉽게 복구 가능

5. 변경 이력 관리

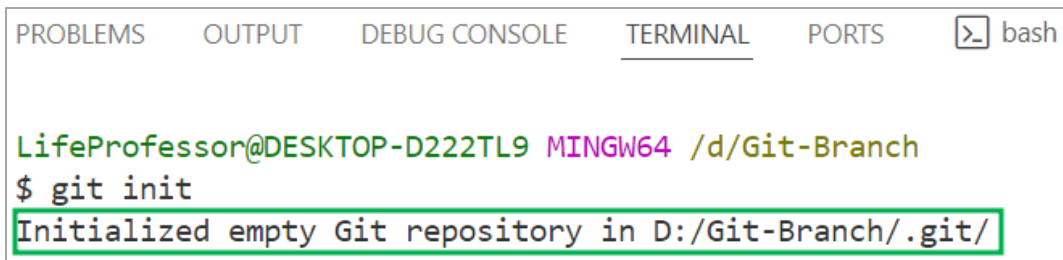
- 각 브랜치가 독립적으로 변경 사항을 기록해, 추적과 관리를 체계적으로 할 수 있음



Branch 생성(1)

◆ git 저장소 만들고 초기화

1. 드라이브 선정 → Git 저장소로 사용할 폴더 생성
2. 폴더 열기 → 파일 → PowerShell 열기
3. 명령어 입력 (code .) → VS Code 실행
4. VS Code 터미널 실행 (Ctrl+Shift+`) → Git Bash 열기
5. 명령어 입력 (git init)
6. .git 폴더 생성 확인



The screenshot shows a VS Code interface with the 'TERMINAL' tab selected. The terminal prompt is 'LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch'. The command '\$ git init' has been entered, and the output 'Initialized empty Git repository in D:/Git-Branch/.git/' is displayed and highlighted with a green box.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    [X] bash

LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch
$ git init
Initialized empty Git repository in D:/Git-Branch/.git/
```

Branch 생성(2)

◆ 파일 생성 및 커밋

1. feature.txt 파일 생성
2. 1번 라인에 "feature A" 추가 및 저장
3. git add feature.txt 입력
4. git commit -m "add feature A" 입력



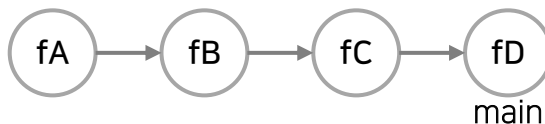
```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git add feature.txt
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git commit -m "add feature A"
[main (root-commit) 9d1cb7a] add feature A
1 file changed, 1 insertion(+)
create mode 100644 feature.txt
```

Branch 생성(3)

◆ 파일 생성 및 커밋

5. 2~4번 라인에 “feature B/C/D” 추가 및
저장하면서 커밋



ex) git commit -am “add feature B”

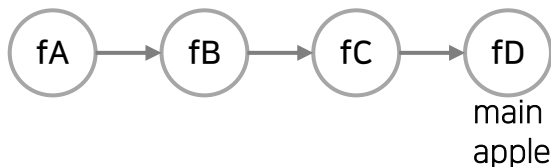
6. git log --oneline 입력

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git log --oneline
5c8cac9 (HEAD -> main) add feature D
3b72653 add feature C
7d26dfa add feature B
9d1cb7a add feature A
```

Branch 생성(4)

❖ branch 생성 및 확인: git branch

1. git branch 입력
2. git branch apple 입력
3. git log 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git branch
* main

LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git branch apple

LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git log
commit 5c8cac9aab95fca8f6f0f60dc8f19119abb3cd93 (HEAD -> main, apple)
Author: LimSeongMin <lifeprof@naver.com>
Date: Mon Oct 7 17:41:21 2024 +0900

    add feature D
```


Branch 생성(5)

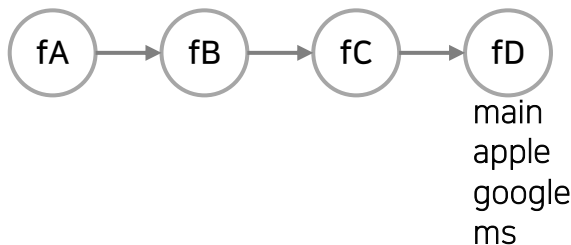
❖ branch 생성 및 확인: git branch

4. git branch google 입력

5. git branch ms 입력

6. git branch 입력

7. git log --oneline 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git branch google
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git branch ms
```

```
LifeProfessor@DESKTOP-D222TL9
$ git branch
apple
google
* main   현재 "main" branch에 있다
ms
```

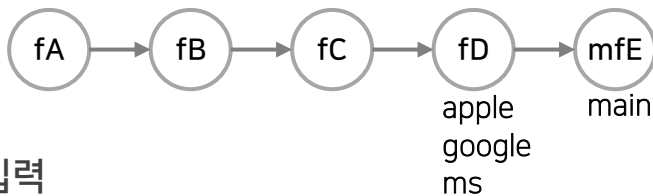
```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git log --oneline
5c8cac9 (HEAD -> main, ms, google, apple) add feature D
3b72653 add feature C
7d26dfa add feature B
9d1cb7a add feature A
```



Branch 생성(6)

main branch에 commit 추가

1. feature.txt 파일 열기
2. 5번 라인에 "main feature E" 추가 및 저장
3. git commit -am "add main feature E" 입력
4. git log --oneline 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
```

```
$ git commit -am "add main feature E"
```

```
[main 4659874] add main feature E
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
```

```
$ git log --oneline
```

```
4659874 (HEAD -> main) add main feature E
```

main branch의 최신 커밋

```
5c8cac9 (ms, google, apple) add feature D
```

ms, google, apple branch의 최신 커밋

```
3b72653 add feature C
```

```
7d26dfa add feature B
```

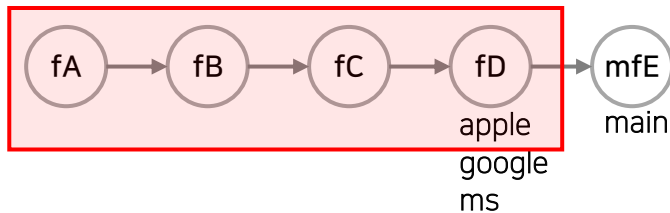
```
9d1cb7a add feature A
```



Branch 전환 후 커밋(1)

git switch

1. git switch apple 입력
2. git log --oneline 입력
3. feature.txt 파일 내용 확인



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git switch apple
Switched to branch 'apple'

LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
$ git log --oneline
5c8cac9 (HEAD -> apple, ms, google) add feature D
3b72653 add feature C
7d26dfa add feature B
9d1cb7a add feature A
```

Branch 전환 후 커밋(2)

❖ git switch

4. feature.txt의 5번 라인에 "apple feature E" 추가 및 저장
5. apple.txt 파일 생성
6. apple.txt 파일에 "apple feature E" 추가 및 저장

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ ls -a
```

```
./ ../ .git/ apple.txt feature.txt
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ cat feature.txt
```

```
feature A
```

```
feature B
```

```
feature C      feature.txt 파일 내용
```

```
feature D
```

```
apple feature E
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ cat apple.txt
```

```
apple feature E  apple.txt 파일 내용
```



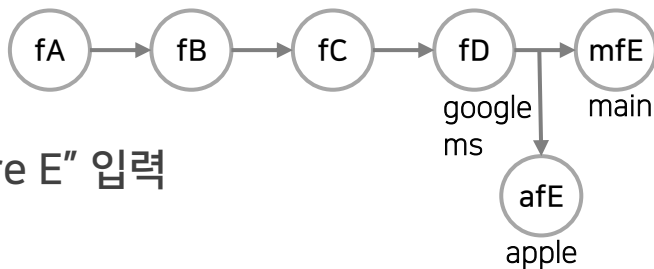
Branch 전환 후 커밋(3)

git switch

7. git add . 입력

8. git commit -m "add apple feature E" 입력

9. git log --oneline 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ git add .
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ git commit -m "add apple feature E"
```

```
[apple ab1204d] add apple feature E  
2 files changed, 3 insertions(+), 1 deletion(-)  
create mode 100644 apple.txt
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
```

```
$ git log --oneline
```

```
ab1204d (HEAD -> apple) add apple feature E
```

```
5c8cac9 (ms, google) add feature D
```

```
3b72653 add feature C
```

```
7d26dfa add feature B
```

```
9d1cb7a add feature A
```

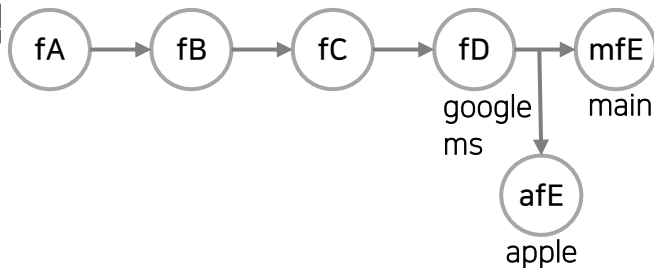


Branch와 Commit의 관계(1)

❖ git log --oneline --branches

1. git log --oneline --branches 입력

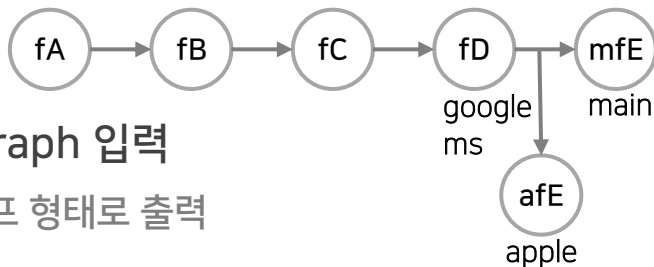
* branches 옵션: 브랜치별 최신 커밋 출력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
$ git log --oneline --branches
ab1204d (HEAD -> apple) add apple feature E
4659874 (main) add main feature E
5c8cac9 (ms, google) add feature D
3b72653 add feature C
7d26dfa add feature B
9d1cb7a add feature A
```

Branch와 Commit의 관계(2)

❖ git log --oneline --branches --graph



2. git log --oneline --branches --graph 입력

* graph 옵션: 브랜치와 커밋의 관계를 그래프 형태로 출력

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
$ git log --oneline --branches --graph
* ab1204d (HEAD -> apple) add apple feature E
| * 4659874 (main) add main feature E
|/
* 5c8cac9 (ms, google) add feature D
* 3b72653 add feature C
* 7d26dfa add feature B
* 9d1cb7a add feature A
```

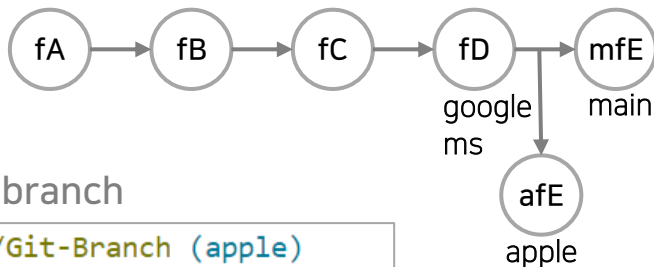
두 개의 Branch 비교

git log branch명..branch명

1. git log main..apple 입력

2. git log apple..main 입력

* A..B : A는 기준 branch, B는 비교 대상 branch



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
$ git log main..apple
commit ab1204d738c03e9c67461ca75efc6940f21109a7 (HEAD -> apple)
Author: LimSeongMin <lifeprof@naver.com>
Date: Mon Oct 7 20:33:52 2024 +0900
```

add apple feature E

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (apple)
$ git log apple..main
commit 46598741c9e7dc8cb81fa1951626b85ec37e7d30 (main)
Author: LimSeongMin <lifeprof@naver.com>
Date: Mon Oct 7 18:08:18 2024 +0900
```

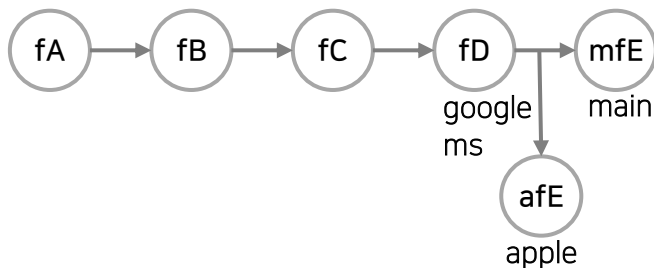
add main feature E



Branch 병합(1)

❖ git merge branch명

1. git switch main 입력
2. feature.txt 파일 내용 변화 확인
3. git merge apple 입력
4. Accept Both Changes 클릭
5. Merge해서 conflict 해결 후 commit



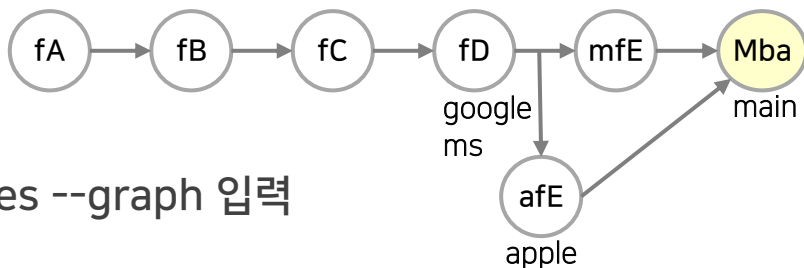
```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main)
$ git merge apple
Auto-merging feature.txt
CONFLICT (content): Merge conflict in feature.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Branch 병합(2)

❖ git merge branch명

6. apple.txt 추가 및 내용 확인

7. git log --oneline --branches --graph 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (main|MERGING)
$ git log --oneline --branches --graph
* 5702392 (HEAD -> main) Merge branch 'apple'
| \
| * ab1204d (apple) add apple feature E
| * | 4659874 add main feature E
| /
* 5c8cac9 (ms, google) add feature D
* 3b72653 add feature C
* 7d26dfa add feature B
* 9d1cb7a add feature A
```

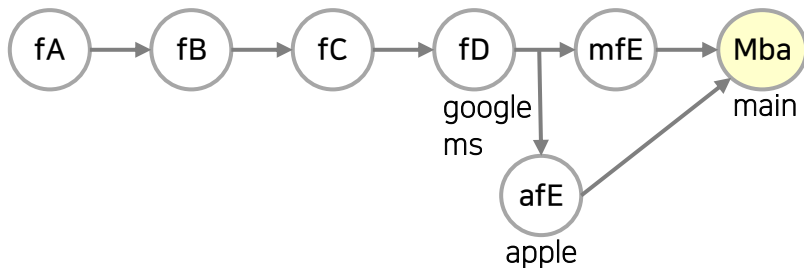


Branch 병합(4)

❖ git merge branch명

8. git switch google

9. git merge main



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
```

```
$ git merge main
```

```
Updating 5c8cac9..5702392
```

```
Fast-forward
```

```
apple.txt | 1 +
```

```
feature.txt | 4 +++-
```

```
2 files changed, 4 insertions(+), 1 deletion(-)
```

```
create mode 100644 apple.txt
```

Branch 병합(5)

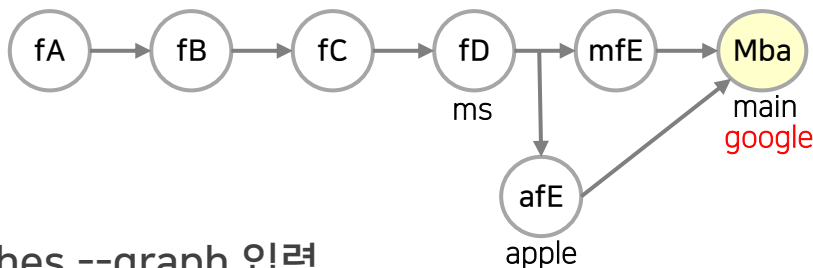
❖ git merge branch명

8. git switch google

9. git merge main

10. git log --oneline --branches --graph 입력

11. apple.txt 파일 추가 및 내용 확인



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
$ git log --oneline --branches --graph
* 5702392 (HEAD -> google, main) Merge branch 'apple'
| \
| * ab1204d (apple) add apple feature E
| * | 4659874 add main feature E
| /
* 5c8cac9 (ms) add feature D
* 3b72653 add feature C
* 7d26dfa add feature B
* 9d1cb7a add feature A
```



Branch 삭제(1)

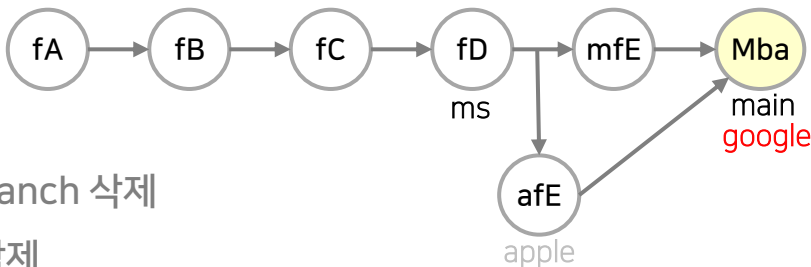
❖ git branch -d branch명

1. git branch 입력

2. git branch -d apple 입력

* -d 옵션: delete의 약어, 병합된 branch 삭제

* -D 옵션: 병합 안된 branch 강제 삭제



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
```

```
$ git branch
```

```
apple
```

```
* google
```

```
main
```

```
ms
```

```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
```

```
$ git branch -d apple
```

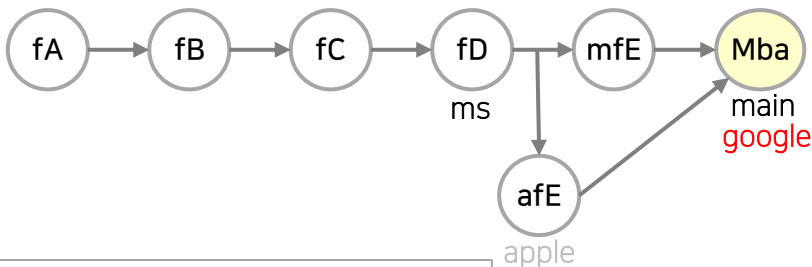
```
Deleted branch apple (was ab1204d).
```



Branch 삭제(2)

❖ git branch -d branch명

3. git log --oneline --branches --graph 입력



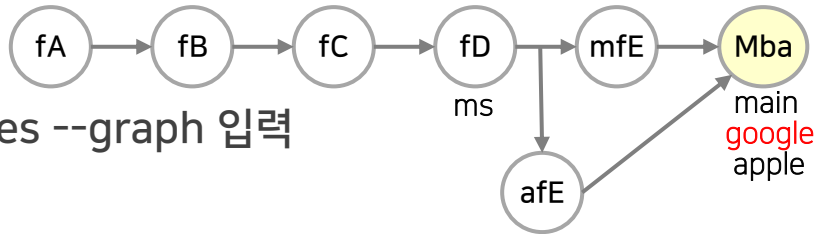
```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
$ git log --oneline --branches --graph
* 5702392 (HEAD -> google, main) Merge branch 'apple'
| \
| * ab1204d add apple feature E
| * | 4659874 add main feature E
|/
* 5c8cac9 (ms) add feature D
* 3b72653 add feature C
* 7d26dfa add feature B
* 9d1cb7a add feature A
```

Branch 복원

❖ git branch branch명

1. git branch apple 입력

2. git log --oneline --branches --graph 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
$ git branch apple

LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/Git-Branch (google)
$ git log --oneline --branches --graph
* 5702392 (HEAD -> google, main, apple) Merge branch 'apple'
| \
| * ab1204d add apple feature E
| * | 4659874 add main feature E
|/
* 5c8cac9 (ms) add feature D
* 3b72653 add feature C
* 7d26dfa add feature B
* 9d1cb7a add feature A
```



[Branch 실습]

◆ branch 생성, 병합, 삭제, 복원

※ GitFlow 상태를 만드세요

1. 프로젝트 폴더 생성 및 Git 초기화

- 1) my-branch-project의 이름으로 폴더 생성
- 2) Git 저장소 초기화해 프로젝트를 버전관리 되도록 준비

2. 파일, branch 생성 및 커밋

- 1) main.txt, google.txt, apple.txt를 GitFlow에 맞게 생성 및 커밋
- 2) google, apple branch를 GitFlow에 맞게 생성

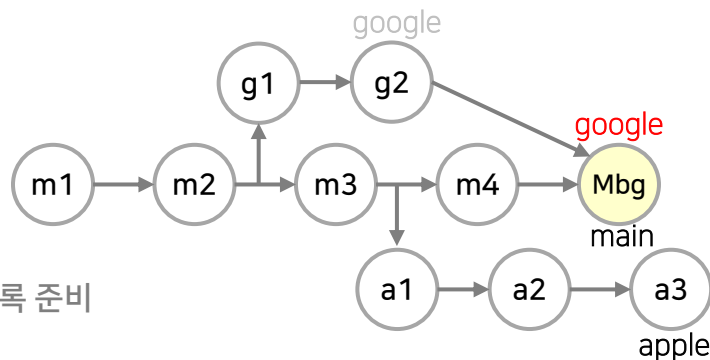
3. branch 병합

- 1) google의 g2와 main의 m4 branch 병합

4. branch 삭제 및 복원

- 1) google branch 삭제 후 복원

5. 최종 그래프 확인



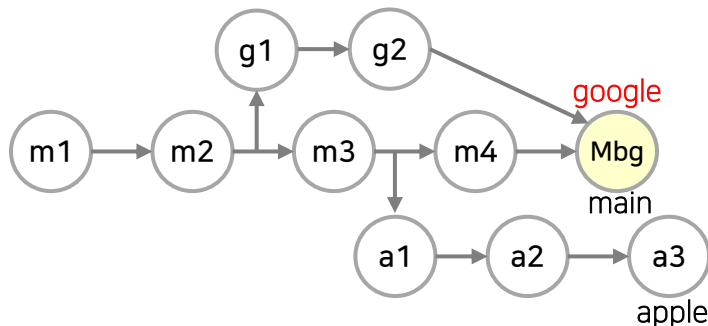
```
LifeProfessor@DESKTOP-D222TL9 MINGUI
$ git log --online --branches --graph
* 1eba0ab (HEAD -> main, google)
|
| * 19ba219 m4
| * 831bc83 g2
| * 863569e g1
| * 80e49b9 (apple) a3
| * 55957dd a2
| * 93a00e8 a1
| /
| * e1cd056 m3
| /
* 1f245e2 m2
* de045e8 m1
```



특정 commit 병합(1)

❖ git cherry-pick hash값

1. git log --oneline 입력
2. a2의 hash값 복사
3. git cheery-pick 'a2 hash값' 입력
4. conflict 문제 해결 후 커밋



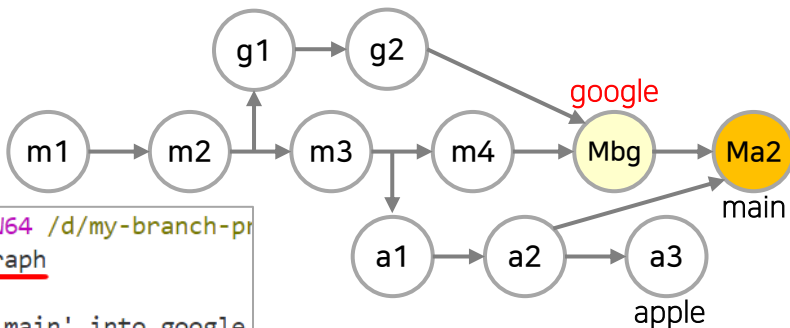
```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/my-branch-project (main)
$ git cherry-pick 55957dd
CONFLICT (modify/delete): apple.txt deleted in HEAD and modified in 55957dd
(a2).  Version 55957dd (a2) of apple.txt left in tree.
error: could not apply 55957dd... a2
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: "git add/rm <paths>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
hint: Disable this message with "git config advice.mergeConflict false"
```



특정 commit 병합(2)

❖ git cherry-pick hash값

5. git log --oneline --branches --graph 입력



```
LifeProfessor@DESKTOP-D222TL9 MINGW64 /d/my-branch-pr
$ git log --oneline --branches --graph
* 017ee6d (HEAD -> main) a2
* 1eba0ab (google) Merge branch 'main' into google
| \
| * 19ba219 m4
| * 831bc83 g2
| * 863569e g1
| | * 80e49b9 (apple) a3
| | * 55957dd a2
| | * 93a00e8 a1
| | /
| * e1cd056 m3
| /
* 1f245e2 m2
* de045e8 m1
```





수고했습니다

