

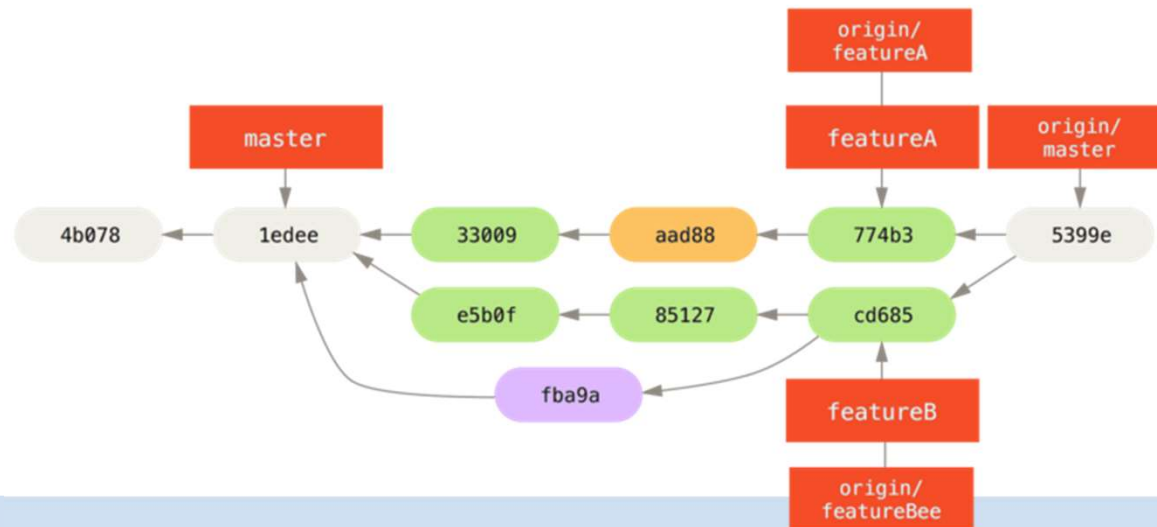
03

Branch

Git

브랜치란

- 하나의 코드에 개별 작업을 브랜치라 한다.
- 독립적으로 진행되는 개별 작업 단위를 의미
- 각 브랜치에서 다른 브랜치에 상호 영향 없이 코드를 추가, 수정, 삭제
- 작업한 브랜치를 다른 브랜치에 병합(merge)
- 새로운 저장소를 만들면 자동으로 main(master)브랜치가 만들어진다.

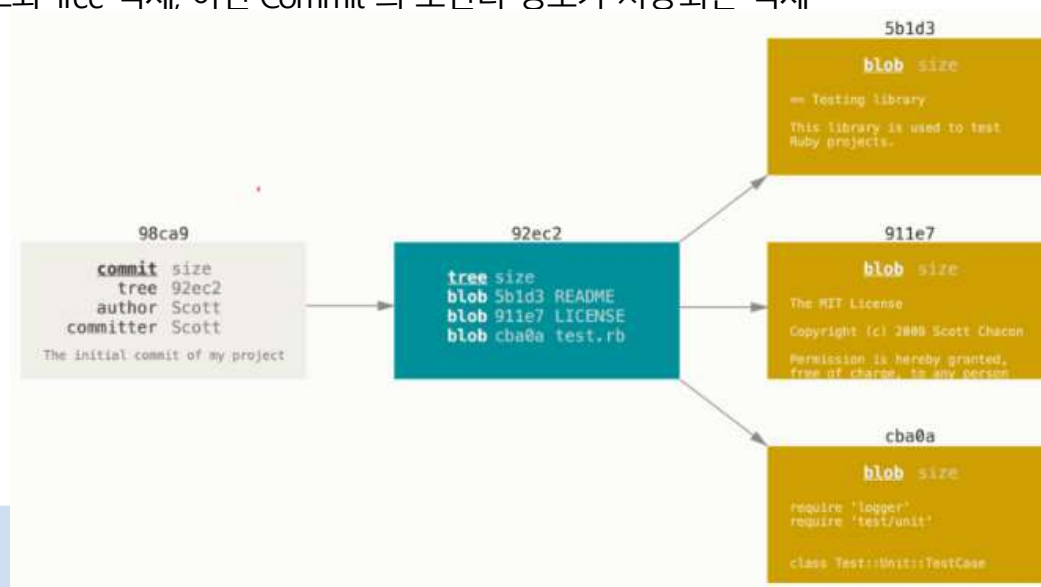


브랜치란

- 새로운 브랜치를 만드는 것은 코드 복사 개념이다.

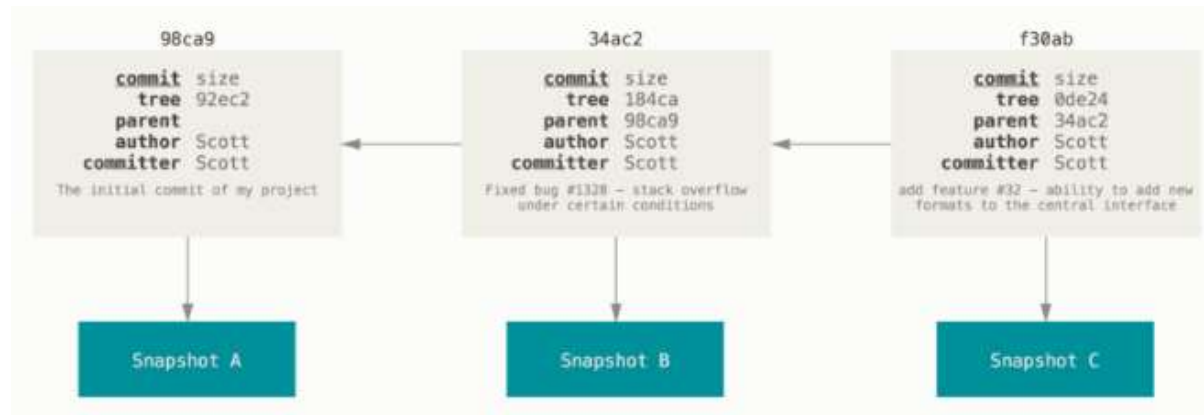
- Commit 객체

- Blob 객체 : 각 파일에 대한 객체
- Tree 객체 : 각 Blob 정보가 들어간 파일과 디렉토리의 구조를 표현하는 객체
- Commit 객체 : meta 정보와 Tree 객체, 이전 Commit 의 포인터 정보가 저장되는 객체

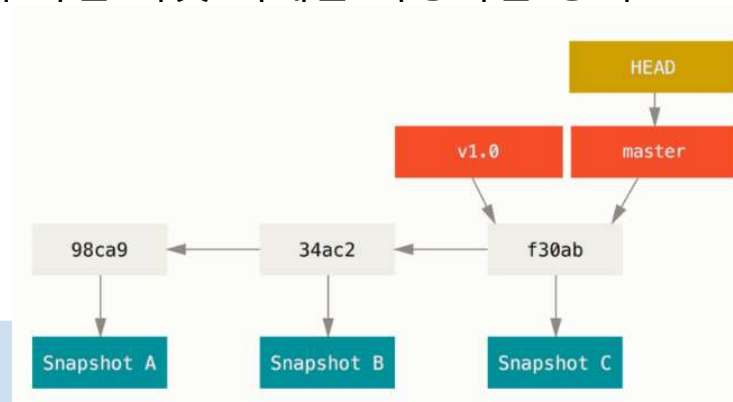


브랜치란

- 새로운 커밋이 발생하면 커밋 객체에 이전 커밋에 대한 정보가 들어간다.

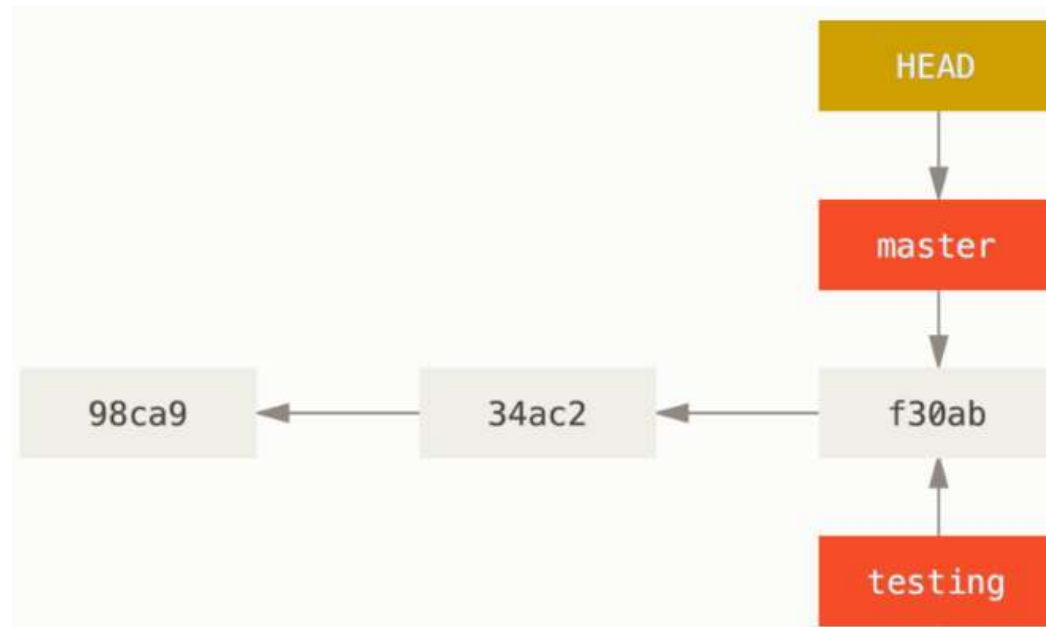


- HEAD 는 현재 작업하는 브랜치의 작업 커밋 객체를 지칭하는 용어



브랜치란

- 새로운 브랜치를 만드는 것은 HEAD 가 지칭하는 커밋 객체의 복제 개념



브랜치란

- **git branch**
 - 브랜치 목록 출력
- **git branch <name>**
 - 새로운 브랜치 만들기
- **git switch <name>**
 - 브랜치 이동
- **git switch -c <name>**
 - 브랜치를 만들고 이동
- **git branch -d <name>**
 - 브랜치 삭제

diff

■ 차이점을 확인하기 위한 명령어

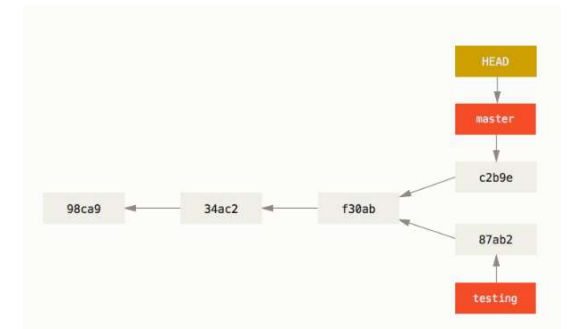
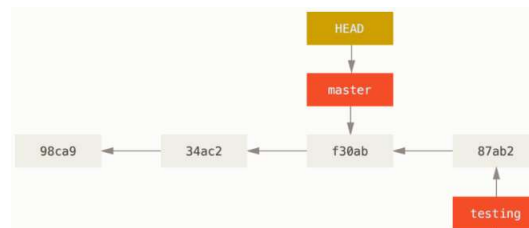
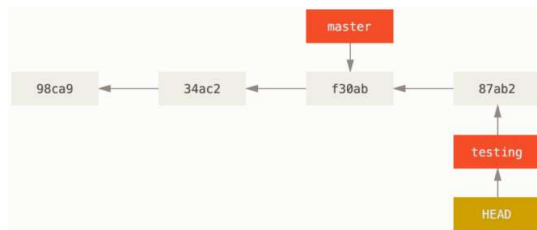
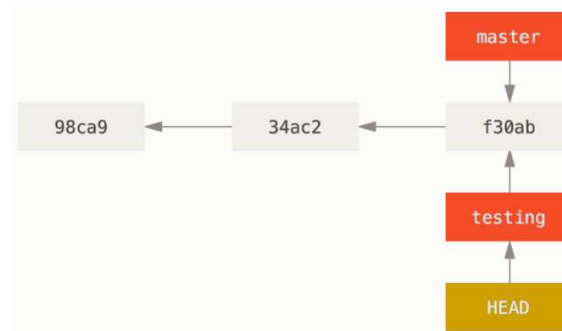
- Working Dir <-> Index
- Working Dir, Index <-> Head
- Index <-> Head
- 브랜치 <-> 브랜치
- Commit <-> Commit

diff

- **git diff HEAD**
 - Working Dir, Index <-> HEAD
- **git diff --staged**
 - Index <-> HEAD
- **git diff [option] <file>**
 - 특정 파일을 명시해 비교 가능
- **git diff <branch> <branch>**
 - 두 브랜치간의 비교
- **git diff <hash> <hash>**
 - 두 커밋간의 비교

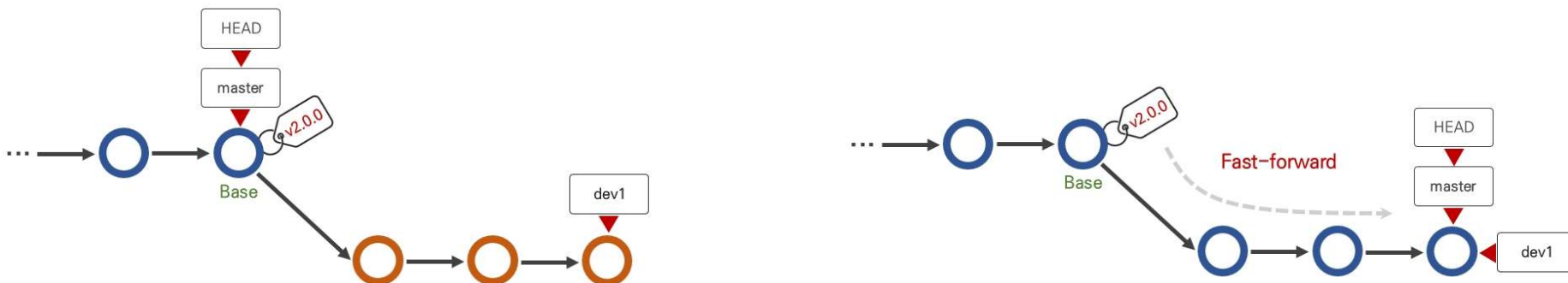
merge

- 브랜치를 만들고 커밋을 하면 두 브랜치가 지칭하는 HEAD 는 상이해진다.



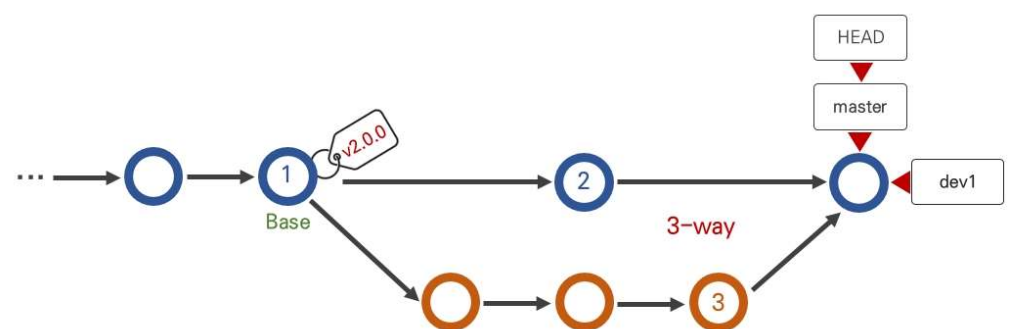
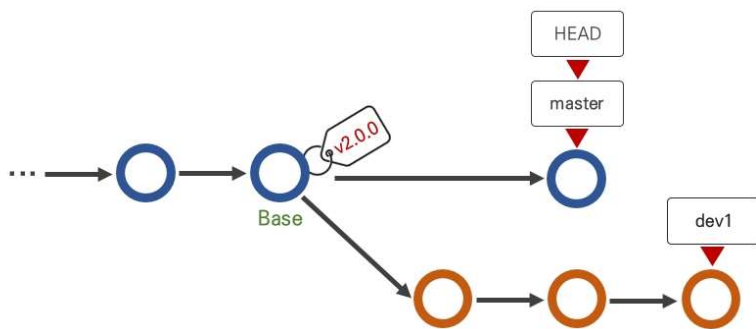
merge

- merge, 브랜치 병합
- fast-forward 병합과 3-way 병합
- 병합에 의해 새로운 commit 이 만들어질 수도 있고 아닐 수도 있다.
- fast-forward merge
- 두 브랜치가 동일 선상에 있을 때 병합이 발생하면 새로운 commit 이 만들어지지 않고 병합된다.



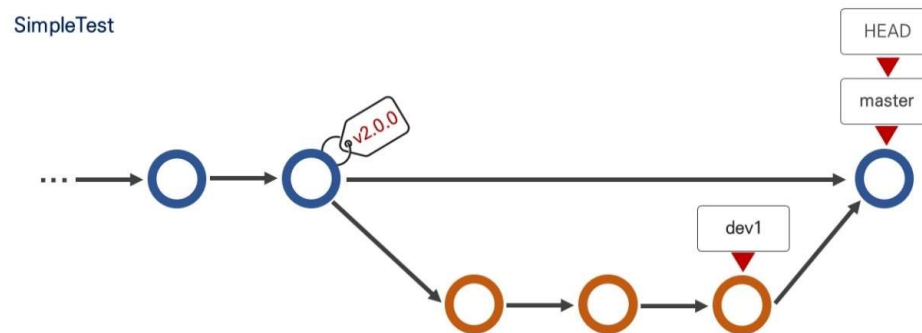
merge

- 3-way merge
- 두 브랜치가 base 에서 각각 commit 이 진행된 경우
- 병합을 하면 새로운 commit 이 만들어 진다.



merge

- **git merge <branch>**
- **git merge --ff <branch>**
 - --ff 는 merge 의 기본 값
 - fast-forward 관계인 경우 새로운 commit 을 발생하지 않고 병합
 - fast-forward 가 아닌 경우 새로운 commit 을 발생하고 병합
- **git merge --no-ff <branch>**
- 두 브랜치가 fast-forward 관계라고 하더라도 새로운 merge commit 을 발생시켜서 병합



merge

- 병합 시 충돌(Conflict)
- 충돌 문제를 해결하지 않으면 commit 할 수 없다.

```
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test6 (main)
$ git merge twoBranch
Auto-merging a.txt
CONFLICT (add/add): Merge conflict in a.txt
Auto-merging main.txt
CONFLICT (content): Merge conflict in main.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/ $ git status
On branch main
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both added:      a.txt
        both modified:   main.txt

no changes added to commit (use "git add" and/or "git commit -a")
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test6 (main|MERGING)
```

rebase

- 브랜치를 만들때의 commit 정보를 base 라고 한다.
- rebase 는 브랜치의 base 를 다시 지정하기 위한 명령어이다.
- `git rebase <branch>`



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare