Git을 사용한 버전 관리

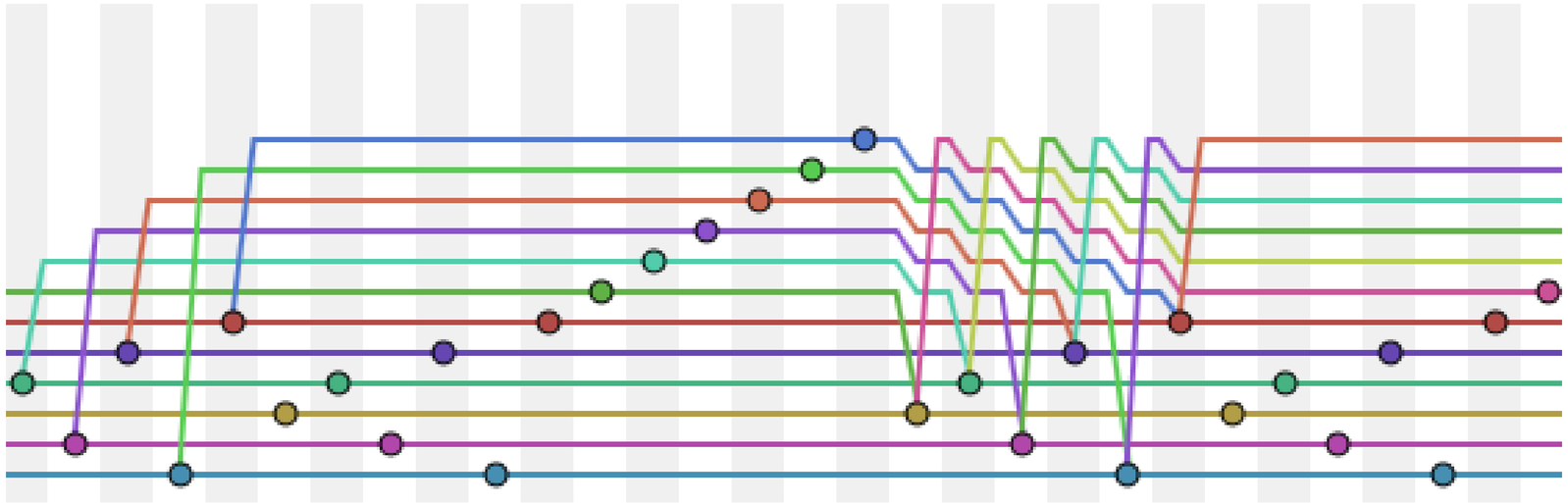
Git rebase

/* elice */

rebase

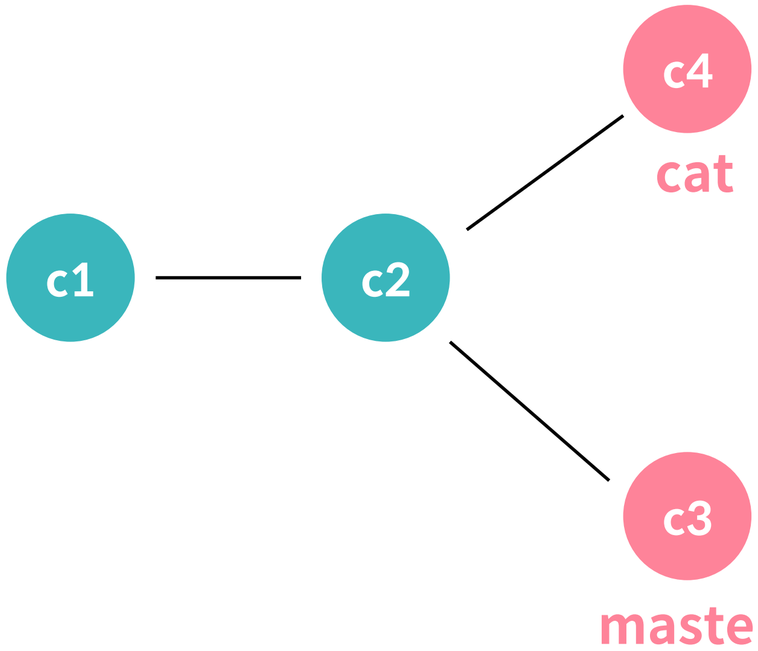
rebase

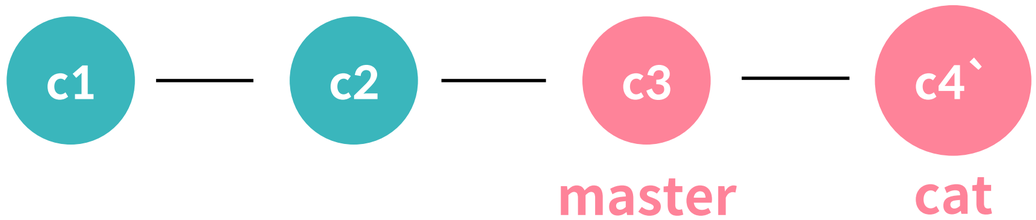
rebase는 브랜치가 너무 많아져서
history 정리가 필요한 상황에 사용합니다.



rebase

그래프를 선형으로 만듭니다.

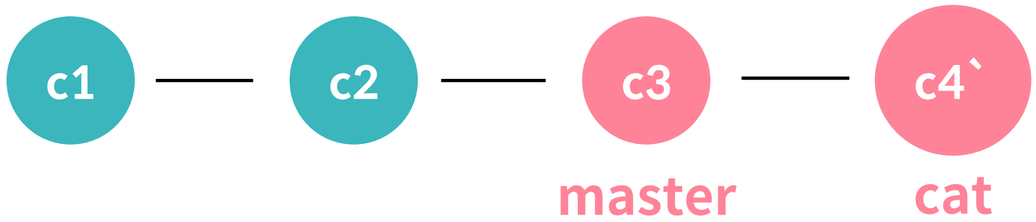




rebase

rebase는 공통된 부모까지의 commit을 가져와

원하는 브랜치 옆에 이어붙입니다.




```
git checkout cat
```

```
git rebase master
```

rebase conflict 해결하기

rebase conflict 해결

merge와 마찬가지로 rebase 과정 중에서도
충돌이 발생할 수 있습니다.

```
$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: 2
Using index info to reconstruct a base tree...
M       client_file
Falling back to patching base and 3-way merge...
Auto-merging client_file
CONFLICT (content): Merge conflict in client_file
error: Failed to merge in the changes.
```

rebase conflict 해결

status 로 파일의 상태를 확인하면 다음과 같습니다.

```
$ git status
rebase in progress; onto 1522887
You are currently rebasing branch 'cat' on '1522887'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --abort" to check out the original branch)
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   client_file
no changes added to commit (use "git add" and/or "git commit -a")
```

rebase conflict 해결

client_file에서 발생한 충돌을 해결하고
`git add` 명령을 수행해서 client_file을 staging 해줍니다.

```
$ git add client_file
$ git status
rebase in progress; onto 1522887
You are currently rebasing branch 'cat' on '1522887'.
  (all conflicts fixed: run "git rebase --continue")

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   client_file
```

rebase conflict 해결

staging 되었다면 이를 `git rebase --continue` 를 이용해서 rebase를 마무리 해줍니다.

```
$ git rebase --continue
Applying: 2
$ git log --all --graph --oneline
* 83e40a4 (HEAD -> cat) cat commit
|
* 1522887 (master) master commit
|
* e44bc57 init repo
```

rebase conflict 해결

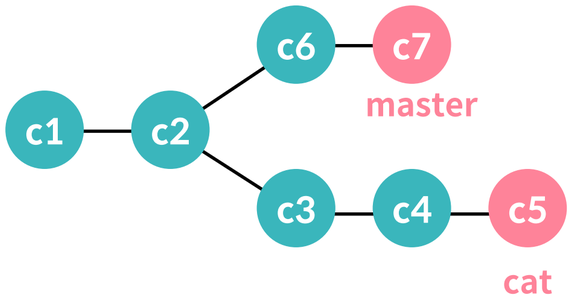
후에 master를 cat으로 merge 뿐만 아니라 rebase로도
fast-forward 시켜 줄 수 있습니다.

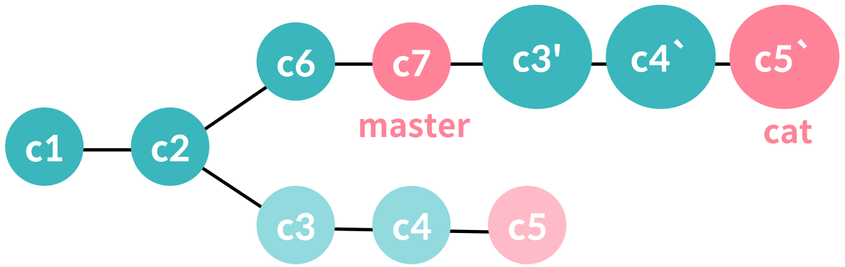
```
$ git checkout master
$ git rebase cat
First, rewinding head to replay your work on top of it...
Fast-forwarded master to cat.
$ git log --online
* 83e40a4 (HEAD -> master, cat) cat commit
* 1522887 master commit
* e44bc57 init repo
```

결론

```
git rebase master
```

위의 명령어를 요약하자면 **현재 위치해 있는 브랜치의 commit 부터**
master의 공통된 부모 전까지의 commit 을
master 옆에 이어 붙인다는 의미입니다.






/*elice*/
contact@elice.io