12주차 OSS 필기 정리

#25

base : master와 bugfix의 공통 조상

1. $ git rebase master : 베이스가 master 자리로 이동
2. $ git merge bugfix : master 브랜치에서 bugfix 브랜치를 fast-forward 병합

rebase에서의 충돌 발생 후 해결 절차

1. 파일 수정
2. 파일 추가 - $ git add <수정파일>
3. Rebase 계속 수행, 마지막 메시지 수정 – $ git rebase –continue

[ 3-way, fast-forward와 rebase 비교 ] 단점 / 장점

3-way merge

* 여러 분기가 생긴 변경 내용의 이력이 그대로 남아서 복잡

Rebase

* 히스토리가 선형으로 단순하고 깨끗
* 원래의 커밋 이력 변경 ( 이력 남겨야하면 사용x )
* 두 갈래의 브랜치에서 기존의 베이스를 마지막 커밋으로 수정하는 병합

Fast-forward merge

* 조상에 위치한 브랜치에서 선행 브랜치의 마지막으로 이동하는 병합

[ 일반적인 rebase 방법 ]

1. $ git checkout topic
2. $ git rebase main
3. $ git checkout main
4. $ git merge topic

[ 다른 rebase 방법 ]

1. $ git rebase main topic
2. $ git checkout main
3. $ git merge topic

#26

커밋 메시지 수정

* $ git commit --amend : 설정된 편집기로 최근 커밋 메시지 수정, 새로운 커밋 ID(커밋 시간, 내용 등등)로 수정
* $ git commit --amend -m “an updated commit message” : 최근 커밋 메시지를 직접 입력해 수정, 새로운 커밋 ID로 수정
* $ git commit --amend --no-edit : 메시지 수정 없이 다시 커밋 수정

Rebase의 --interactive

* $ git rebase -I HEAD ~3 : 기본 편집기로 HEAD~2부터 HEAD까지의 행을 편집 가능