

2024-2 기말고사 (오픈소스 소프트웨어)

누구나 특별한 제한 없이 그 코드를 보고 사용할 수 있는 오픈소스 라이선스를 만족하는 SW :
오픈소스 소프트웨어

리처드 스톨먼이 설립한 제단: 자유 소프트웨어 제단

Copyright의 반댓말: Copyleft

오픈소스 지원 관리 서버 3개 : github, gitlab, bitbucket

대부분의 웹을 지원하는 서비스 스택 모델: LMAP

빅데이터, 인공지능 분야 OSS 프로그램: Python, Scikit-learn, Tensorflow, Pytorch

대표적 오픈소스 라이선스 종류: GNU General Public, MIT, Apache, BSD, MySQL, SUSE, Ubuntu

소스코드 반환 의무가 있는 대표적인 라이선스: GPL, AGPL, LGPL, MPL, EPL

FSS에서 만든 라이선스: GPL

위 정답의 완화된 버전: LGPL

깃의 임시 저장소: Stash

깃 4영역: 작업 디렉토리, 스테이징 영역, 깃 저장소, Stash

Stash는 (스택) 구조이다

LIFO: Last In - First Out

git stash 하면: Working Dir과 Staging Area가 Stash, Git repo가 Working dir, Staged로

Stash 에서 꺼내기: git stash apply

그냥 apply는 (작업 디렉토리) 내용만 복구한다

둘다 복원하려면: --index

스테이징 영역을 stash에 저장하지 않는 명령어: -k, --keep-index

미추적 파일도 함께 stash에 저장: -u, --include-untracked

User가 선택하여 stash에 저장: -p, --patch

Stash 목록 보기: git stash list

가장 최신으로 저장한 항목의 번호는: 0

Stash 항목이 생성되었을 때의 커밋 자료와 최신 stash 항목간 차이: git stash show

최근 또는 지정된 임시 저장소 내용을 반영하고 삭제: git stash pop stash@{n}

최근 또는 지정된 임시저장소 내용을 반영, stash유지, wd만 반영: git stash apply stash@{n}

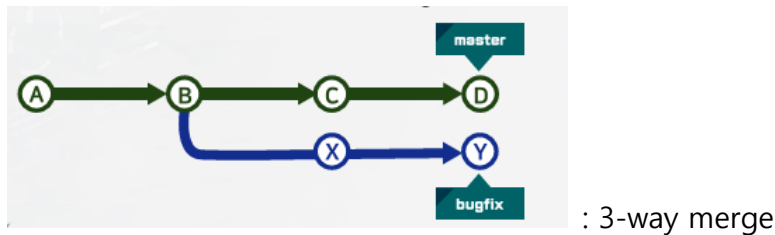
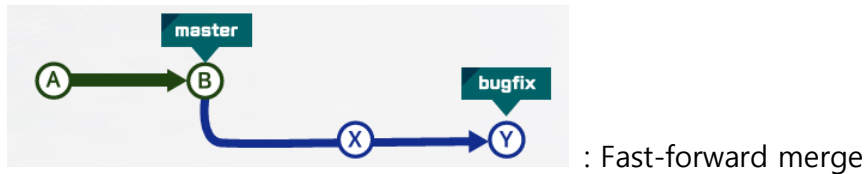
Wd와 staged 에도 반영 git stash apply --index stash@{n}

가장 최근 임시저장 내용 삭제: git stash drop

모든 stash 저장 내용 삭제: git stash clear

사용자와 상호작용하여 삭제 항목 선택하는 명령어: git clean -i

병합의 종류: Fast-forward merge, 3-way merge



3-way merge는 (병합 커밋) 을 생성한다.

Fast Forward 병합이 가능한 상태에서 3way 하는 방법: git merge --no-ff

커밋 이력과 병합보다는 브랜치 이력을 남기지 않는 merge: git merge --squash

3-way와 squash는 commit이 필요하다 (O, X)

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gffm (main)
$ git merge topic
Updating 3d00e27..8dd39e7
g | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 g

$ git lgag1
* 8dd39e7 (HEAD -> ) B
* eb0bfd8 A
* 3d00e27 1
```

1. Fast-forward 2. main, topic

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/branch-lab (main) 1. MERGING
```

병합 취소: git merge --abort

Rebase 는 branch에서

최근 커밋 수정: git commit --amend

이전 특정 커밋으로 완전히 돌아가버리는 방법: reset

깃 3저장소 내용을 모두 변경: --hard

깃 저장소만 변경: --soft

깃 저장소와 스테이징만 변경: --mixed

Reset 후 바로 이전 상태로 되돌아가기: `git reset --hard ORIG_HEAD`

커밋 취소: `revert`

Revert는 특정 커밋을 취소하는 새로운 커밋을 생성한다 (O, X)